Aurora Workflow Orchestration – Method Specification v1.2

Shawn C. Wright, Aurora Research Initiative / Waveframe Labs Division

October 2025

Aurora Workflow Orchestration (AWO) — Method Specification v1.2 (Final)

Author:
Shawn C. Wright

Affiliation:
Aurora Research Initiative, Waveframe Labs (Independent Researcher)

Version:
1.2 · **Date:** 2025-10-19
Status: Final (Canonical Specification)
Supersedes: AWO_Method_Spec_v1.1 (2025-10-07)

DOI:
10.5281/zenodo.17345552

Purpose

Define a **repeatable, falsifiable, and auditable** method for AI-assisted research and analysis so that a third party can independently reproduce both the process and the evidence supporting its outcomes.

Scope

AWO governs how to structure AI-assisted research into falsifiable claims, audited iterations, and immutable, citable releases.

AWO defines the **methodological layer** that enforces reproducibility and auditability.
CRI-CORE provides one possible **implementation layer** that automates AWO-compliant runs and manifests, but AWO remains valid without it.

Normative Requirements (must / should)

1. **Falsifiability** — Every claim MUST have a defined procedure that could falsify it.

2. **Full Logging** — Every reasoning step and output MUST be logged with timestamps and (when provided) schema validation.

3. **Independent Audit** — Logic, data, and peer audits MUST be performed by separate agents or processes.

4. **Rejection Loop** — Failed audits MUST trigger revision or withdrawal, not defense.

5. **Portability** — Artifacts SHOULD be domain-agnostic and reusable across contexts.

6. **Version Locking** — All claims and outputs MUST reference immutable version identifiers (tags, hashes, DOIs).

Roles

- **Orchestrator (Human)** — Frames questions, defines falsifiability criteria, resolves conflicts, and approves releases.

- **Main Model (Continuity)** — Maintains project context, synthesizes results, and integrates audit feedback.

- **Auxiliary Auditors (Independent)** —
  - *Logic Auditor* — Ensures internal consistency and valid reasoning.

  - *Data Validator* — Tests claims empirically.

  - *Peer Critic* — Performs adversarial review to surface conceptual errors.

- **System Auditor (Optional)** — Verifies runtime integrity and provenance (e.g., CRI-CORE execution logs).

Core Artifacts (per repository)

- **Falsifiability Manifest** (`/docs/FALSIFIABILITY_MANIFEST.md`) — claim IDs, tests, datasets (if any), thresholds, and status.
- **Workflow Logs** (`/logs/*.md`) — dated entries with actions, insights, next steps, skills.
- **Decision Records (ADRs)** (`/decisions/*.md`) — context, decision, consequences, and evidence links.
- **Evidence Registry — use actual folders present in this repo**:
  - `/figures/` — images/plots referenced by manifests and ADRs.
  - `/models/` — optional saved model artifacts (if produced).
  - `/scripts/` — helper scripts used during an iteration (referenced from ADRs).
  - `/workflows/` — executable workflow specifications used during runs.
  - `/schemas/` — validation and reproducibility schemas.
- **Run Manifests** (`/runs/run_*/run_manifest.json`) — canonical record of runtime state and results for AWO-compliant executions.
- **Attestation & Sums** (`/runs/run_*/ATTESTATION.txt`, `/runs/run_*/SHA256SUMS.txt`, plus `.sig`/`.cert`) — cryptographic proof and checksums for each run.
- **Release Artifacts** — `CHANGELOG.md`, `CITATION.cff`, `.zenodo.json`, Git tag, and Zenodo DOIs (concept + version).

  Note: This specification intentionally does **not** reference `/notebooks` or `/data`, because those folders are **not present** in this repository. If future projects need them, add explicitly and update ADRs/manifests accordingly; the AWO method itself does not require them.

Lifecycle (one iteration)

0. **Setup** — Define claims, initialize Falsifiability Manifest, prepare `/templates`, assign auditors via Model Roster.

1. **Draft (Main Model)** — Produce reasoning and outputs tagged with claim IDs.

2. **Audit (Independent)** — Logic, data, and peer auditors record pass/fail results and notes.

3. **Synthesis (Main Model)** — Reconcile audit outputs; revise claims or methods.

4. **Decision** — Record outcome in ADR (accepted / revised / withdrawn) with evidence links.

5. **Evidence Capture** — Save figures and scripts, capture any produced models, and update Manifest status.

6. **Release Gate** — Validate reproducibility (schema if present, e.g., `/schemas/run.schema.json`); verify attestation and checksums (`ATTESTATION.txt`, `SHA256SUMS.txt`, signatures); confirm cross-references; tag release and archive on Zenodo.

Logging Schema (minimum fields)

- **Log Entry:** date, action, lesson, next step, skills.

- **Audit Record:** claim ID, auditor, check type (logic | data | peer), criteria, result, evidence links.

All logs SHOULD conform to JSON schemas under `/schemas/` when available. Schema version SHOULD match the repository release tag (e.g., v1.2).

Rejection Handling

Any failed audit → revise draft or withdraw claim.
Partial failures → enter **conditional revision** state until all criteria pass.
Update Manifest and ADR accordingly. No appeals without new evidence.

Portability Guidelines

- Keep templates generic; avoid domain jargon in checklists.

- Parameterize datasets and metrics in the Manifest **only if they exist** in the repo.
- Use the Model Roster to swap models or auditors without changing the process.
- Derived projects (e.g., CRI-CORE or domain forks) MUST preserve field-level schema compatibility.

Conformance Checklist

- ☐ Manifest exists with at least one falsifiable claim and test.
- ☐ Logs present for each iteration (draft → audit → synthesis → decision).
- ☐ At least one ADR captures a non-trivial trade-off or decision.
- ☐ Release artifacts present; latest tag archived with Concept + Version DOIs.
- ☐ Attestation artifacts present and verifiable (`ATTESTATION.txt`, `SHA256SUMS.txt`, signatures).
- ☐ (If schemas present) Schema validation passed for logs and manifests.

Example Reference

Waveframe v4.0 — Canonical case study demonstrating AWO artifacts and citable release under Aurora Research Initiative.

File and Folder Conventions

- `/templates/*.md|yaml` — Reusable templates for manifest, audit record, ADR, release checklist.
- `/schemas/*.json` — Validation and reproducibility schemas.
- `/decisions/` — Governance layer.
- `/logs/` — Execution layer.

- `/docs/` — Whitepapers, manifests, and specifications.
- `/runs/` — Runtime results and manifests (`run_manifest.json`) plus attestation/sums.
- `/figures/`, `/models/`, `/scripts/`, `/workflows/` — Evidence and execution assets used by this repository.

Maintained by Waveframe Labs
`swright@waveframelabs.org`
https://waveframelabs.org

Status: Finalized under Aurora Research Initiative · October 2025
Future changes appear only as *Implementation Notes*, not method revisions.