Aurora Workflow Orchestration – Method Specification v1.2.1

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

October 2025

Aurora Workflow Orchestration (AWO)

Method Specification — v1.2.1 (Scaffold)

Maintainer: Waveframe Labs

Preface

This document defines the **normative specification** for Aurora Workflow Orchestration (AWO).
It replaces descriptive or philosophical language with enforceable procedural logic.
All future automation layers (e.g., CRI-CORE) must validate conformance against these requirements.

Interpretation of Compliance Language
- **MUST** — absolute requirement for AWO-compliant repositories.
- **SHOULD** — strong recommendation; deviations must be justified in documentation.
- **MAY** — optional behavior permitted for flexibility.

---

1. Introduction

Purpose: Establish a formal, reproducible framework for AI–human research collaboration.
AWO defines the structural and procedural rules necessary for falsifiable, auditable, and citable research.

Scope: This specification governs the structure, artifacts, and operational requirements of AWO projects.

It does not define runtime enforcement (covered by CRI-CORE) but specifies all conditions that runtime tools must verify.

TODO: Insert concise background paragraph linking to AWO Whitepaper rationale.

---

2. Definitions

Define key entities and concepts used throughout the AWO standard.

Core Terms: - **Run:** A discrete, traceable research execution instance.
- **Provenance:** The recorded lineage of all data, logic, and decisions that produced a result.
- **Artifact:** Any persistent output (report, manifest, ADR, checksum, dataset).
- **Attestation:** Human or automated confirmation that artifacts are complete, correct, and verified.
- **ADR:** Architecture Decision Record documenting the reasoning behind changes.
- **Manifest:** A falsifiability declaration defining disproof conditions before execution.

TODO: Refine definitions list and cross-link to CRI-CORE schema references later.

---

3. Roles and Responsibilities

AWO distinguishes between procedural roles to ensure accountability and non-circular validation.

Primary Roles: - **Researcher:** Executes runs and maintains artifacts.
- **Maintainer:** Oversees repository integrity and version control.
- **Reviewer:** Performs verification and attestation of completed runs.

TODO: Add explicit permissions/responsibilities (who can sign approvals, tag releases, modify manifests).

---

4. Repository Requirements

Every AWO project MUST follow a consistent repository layout to ensure verifiability and interoperability.

Required Directories:

```
/docs/       → manifests, specs, reports
/decisions/  → ADRs (0001-NNNN)
```

```
/logs/        → timestamped workflow notes
/runs/        → attested run artifacts
/figures/     → diagrams, lifecycle visuals
```

TODO: Add detailed artifact rules and cross-link schema expectations.

---

5. Lifecycle and Run Phases

Each research cycle proceeds through four canonical phases:

1. **Fan-out (Planning)** — Define hypotheses, manifests, ADRs.
2. **Consensus (Execution)** — Perform runs and collect data.
3. **Attestation (Verification)** — Approve or reject based on falsifiability criteria.
4. **Archival (Publication)** — Freeze results, compute checksums, tag releases.

TODO: Create table describing inputs/outputs for each phase.

---

6. Artifacts and File Rules

Every run MUST produce a verifiable set of artifacts:

File	Description	Required
`workflow_frozen.json`	Captures executed parameters and inputs.	Yes
`report.md`	Describes outcomes, metrics, and observations.	Yes
`approval.json`	Signed validation record by human reviewer.	Yes
`SHA256SUMS.txt`	Hash registry for all outputs.	Yes
`manifest.json` or `manifest.md`	Defines falsifiability boundaries.	Yes

TODO: Add versioning, format validation (JSON schema references), and CRICORE integration hooks.

---

7. Compliance Language

This section defines the mandatory, recommended, and optional behaviors for implementers.

Level	Definition	Enforcement
MUST	Required for compliance.	Hard validation
SHOULD	Recommended unless documented exception.	Warning
MAY	Optional feature.	No enforcement

TODO: Map existing AWO clauses to each compliance level.

---

8. Governance and Attestation

Each run requires human or automated attestation of validity and completeness.

Core Requirements: - Runs MUST include `approval.json` with reviewer signature and timestamp.
- Attestation MAY include checksum verification and peer confirmation.
- Failed attestations MUST be logged under `/logs/attestation_failures/`.

TODO: Specify acceptable digital signature methods and verification workflows.

---

9. Release and Versioning

AWO-compliant repositories MUST version all outputs and preserve immutability.

Release Requirements: - Each release corresponds to a reproducible state of the repository.
- Tags MUST follow semantic versioning (e.g., `v1.2.1`).
- Releases MUST attach PDF artifacts, SHA256SUMS, and ADR references.
- Released runs MUST NOT be altered post-publication.

TODO: Add instructions for checksum regeneration and Zenodo linkage.

---

10. Licensing and Attribution

AWO uses dual licensing to separate executable and textual components.

- **Code:** Licensed under Apache 2.0.

- **Documentation:** Licensed under CC BY 4.0.
- Attribution MUST include author, ORCID, and concept DOI in derivative works.

TODO: Add structured attribution metadata schema reference.

---

11. Falsifiability Manifests

Each experiment MUST include a falsifiability manifest before execution.

Manifest Contents: - Hypothesis statement
- Predicted outcomes
- Disproof criteria
- Experimental plan
- Acceptance thresholds
- Known risks

TODO: Formalize manifest schema for CRI-CORE parsing.

---

12. Conformance Checklist

Each repository MUST pass the following before claiming AWO compliance:

☐ Standard directory structure present.

☐ At least one signed run in `/runs/`.

☐ ADRs and falsifiability manifests linked.

☐ SHA256SUMS.txt present at root.

☐ PDF artifacts built successfully.

☐ CHANGELOG includes version reference.

☐ README links to Whitepaper, Method Spec, Adoption Guide.

TODO: Add automated compliance script references (future CRI module).

---

13. Appendix C — Rationale Summary (Reserved)

TODO: When the Method Spec text is finalized, reintroduce Appendix C summarizing why each rule exists in concise bullet form.

(Placeholder retained for structural continuity.)

End of Specification — Aurora Workflow Orchestration (AWO) v1.2.1 Scaffold