Intent: AI SDLC Methodology

Intent ID: INT-AISDLC-001 **Date**: 2024-01-01 (originated), 2026-02-19 (v2.1 revision) **Priority**: Critical **Status**: v2.1 — Asset Graph Model complete, implementation in progress

---

The Problem

AI coding assistants are powerful but chaotic:

1. **No methodology** — ad-hoc usage, no traceability from intent to runtime
2. **Lost context** — no persistent memory of project decisions across sessions
3. **No quality enforcement** — TDD skipped, debt accumulates, no gates
4. **Enterprise gap** — can't prove AI-generated code meets specs (BCBS 239, SOC 2)
5. **No shared framework** — every team reinvents their own prompts

---

What We Want

An AI SDLC methodology grounded in the Constraint-Emergence Ontology, where:

- Software development is an **asset graph** — typed artifacts connected by admissible transitions
- A **universal iteration function** is the only operation — parameterised per edge
- **Feature vectors** (REQ keys) trace trajectories through the graph from intent to runtime
- **{Human, Agent, Tests}** evaluators compose per edge to define convergence
- **Context[]** (ADRs, data models, templates, policy) is the constraint surface preventing degeneracy
- The **full lifecycle** closes: CI/CD → telemetry → homeostasis → new intent → back into the graph

Four primitives. One operation. The rest is parameterisation.

Business Value

Developer Productivity: Structured AI assistance, persistent context, no tool-switching **Quality Assurance**: TDD/BDD as edge evaluators, feature vector traceability, constraint density prevents hallucination **Enterprise Enablement**: Audit trails via feature vector trajectories, deterministic evaluators for compliance **Cost Reduction**: Auto-generated documentation, standardised methodology, no vendor lock-in

Key Documents

Document	Purpose
AI_SDLC_ASSET_GRAPH_MODEL.md	Canonical methodology — the 4 primitives, ontology grounding
AISDLC_IMPLEMENTATION_REQUIREMENTS.md	32 implementation requirements derived from the model
Constraint-Emergence Ontology	Parent theory

Current Status

Asset Graph Model (v2.1): Complete — replaces v1.x 7-stage pipeline **Implementation Requirements (v3.1)**: Complete — 32 requirements, 9 critical **Tooling (Claude Code plugins)**: Partial — agents, workspace, commands exist from v1.x, need realignment to graph model **Examples**: ai_sdlc_examples — needs update for v2.1

History

- **v1.0** (2024): Key Principles + TDD workflow (ai_init)
- **v1.2** (2025-11): 7-stage pipeline model, 47 implementation requirements, ~4,400 lines across 4 methodology docs
- **v2.0** (2026-02): Asset Graph Model — 4 primitives, 1 operation, ontology-grounded. Prior version tagged `v2.0`.
- **v2.1** (2026-02): Revised framing — composite vectors, zoomable graph, scale-dependent assurance, 3-param iterate(). ~570-line canonical doc replaces 4,400 lines. 32 implementation requirements replace 47.