

AI SDLC — Project Genesis: Intent

Intent ID: INT-AISDLC-001 **Date:** 2024-01-01 (originated), 2026-02-21 (v2.6 revision)
Priority: Critical **Status:** v2.8 — Asset Graph Model complete, multi-agent coordination, two-command UX layer, gradient unification, processing phases, sensory systems, implementation Phase 1a

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
<u>AI SDLC ASSET GRAPH MODEL.md</u>	Canonical methodology — the 4 primitives, ontology grounding
<u>AISDLC IMPLEMENTATION REQUIREMENTS.md</u>	64 implementation requirements derived from the model
<u>Constraint-Emergence Ontology</u>	Parent theory

Current Status

Asset Graph Model (v2.8): Complete — multi-agent coordination, two-command UX, gradient unification, processing phases, sensory systems, protocol hooks

Implementation Requirements (v3.12): Complete — 64 requirements, 10 critical

Tooling (e.g. Claude Code plugins): Phase 1a — 4 agents, 13 commands, 4 hooks, configurable graph **Examples:** [ai_sdlc_examples](#) — CDME dogfood (5/5 edges converged)

Version

v3.0 — Asset Graph Model (v2.8), 64 implementation requirements. See [AI SDLC ASSET GRAPH MODEL.md](#) for the canonical specification.