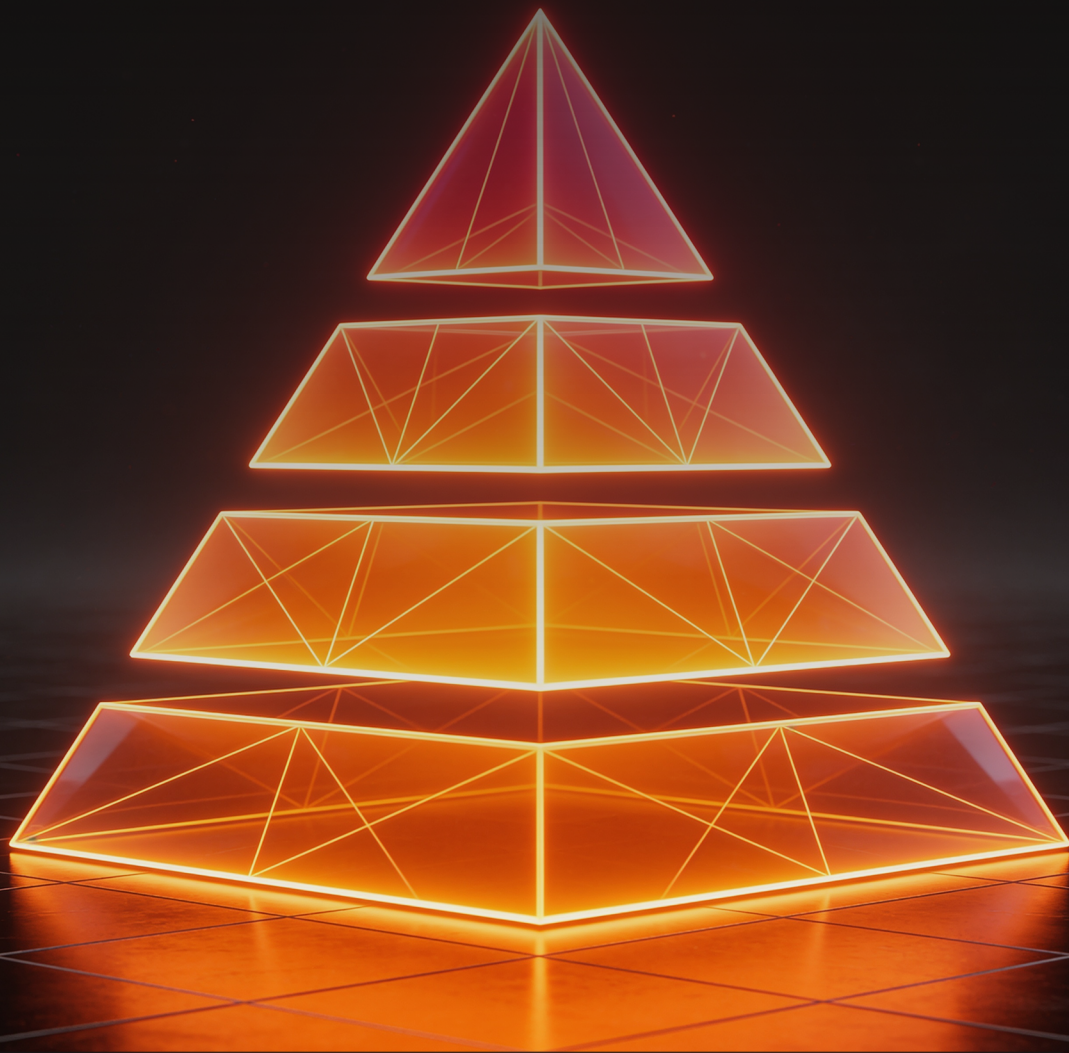


HoloV

A Holonic Delivery Model for the AI Era



AI capabilities are improving quickly. The bigger challenge is creating a structured model for how work moves between people and AI.

Whitepaper

Written by
Arun Kumar Sasidharan
SVP - Enterprise Architecture





Contents

1. The Problem Statement.....	3
2. Intellectual Lineage.....	4
3. Core Philosophy.....	6
4. The Model Defined.....	8
5. Operationalizing HoloV.....	10
6. Tooling Ecosystem.....	12
7. KPIs and the HoloV Harness.....	14
8. The Agentic Autonomy Progression.....	17
9. Why This Matters.....	19



The Problem Statement

The software industry is short of a delivery model for the AI era. The dominant pattern of 2024–2025 was adoption without architecture. Teams reached for AI coding assistants and discovered that individual productivity gains did not translate into organizational delivery improvement. Code was generated faster and accumulated technical debt faster. The bottleneck was never writing code. It was always the clarity of what should be built, and the governed path from that clarity to something running in production.

Three structural failures characterize organizations navigating this transition without a framework:

➤ Operating-Model Confusion

Existing delivery models assume a single team, cadence, and maturity path. They have no common model for an enterprise delivery system where human teams, AI agents, and hybrid holons can coordinate work consistently across the organization.

➤ Spec Dissolution

AI generation without a governing specification produces what the industry has begun to call automated legacy—internally consistent yet architecturally brittle code that no one fully understands or owns.

➤ Autonomy by Accident

Organizations are moving toward AI-led delivery implicitly, without a framework. The transition happens by default through tool adoption, often with no governance, no reversibility, or a clear signal for when it is safe to reduce human oversight.

HoloV is a delivery model designed specifically for that transition. It does not replace what teams are already doing. It provides the structural language that makes the transition governable and repeatable.

Intellectual Lineage

HoloV does not arrive from nowhere. It is the synthesis of four converging lines of thinking, each of which solved something real and left something unresolved.

- **The V-Model (early 1980s)**

First appearing at Hughes Aircraft circa 1982 and formalized in Germany through the late 1980s, the V-Model gave delivery its most enduring insight: that building and verifying are not sequential phases but mirror strokes. The down-stroke converts intent into form. The up-stroke tests form against intent.

What the V-Model could not do was iterate. It assumed intent was fully understood before a line was written, and it coupled every delivery phase so tightly that no single stage could change without cascading through the whole.

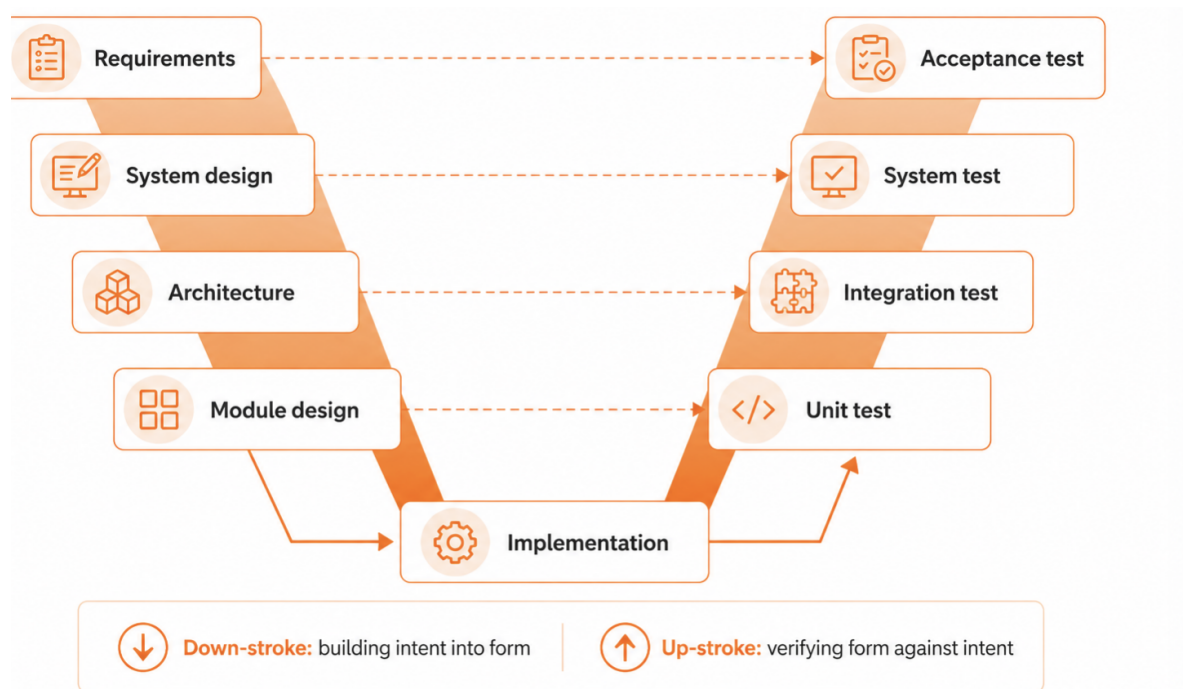


Figure 1 – The V-Model geometry. Every build phase has a mirror verification phase.

- **Agile (2001)**

Agile resolved the iteration problem and in doing so dissolved the verification rigor. Teams gained adaptability and lost the quality spine. Specifications became so fluid they ceased to govern. For a decade and a half this trade-off was acceptable – the bottleneck in delivery was human writing speed, and Agile optimized for that.



- The AI inflection (2023–2024)

AI changed the bottleneck. By 2023–24, developers could generate code faster than any prior methodology assumed possible. The constraint shifted from writing speed to intent clarity – the precision of specification and the rigor of verification against it.

- V-Bounce (Hymel, Crowdbotics, August 2024)

V-Bounce was the first formal attempt to adapt the V-Model for an AI-native team. It reintroduced down-stroke and up-stroke geometry into an AI-assisted workflow, compressing the implementation phase dramatically by placing AI as the primary code generator and repositioning humans as validators and verifiers.

It was a correct and important insight. But it remained a single-team model. It did not describe how enterprise delivery should coordinate work as AI takes on a greater role and delivery becomes progressively more autonomous.

- Spec-Driven Development (2025)

SDD arrived as a parallel answer from a different direction. Vibe coding had produced a specific failure mode: plausible code that drifted from intent, hallucinated dependencies, and collapsed under maintenance. The industry converged on a governed specification layer that served as a version-controlled, machine-readable source of truth for AI generation and output verification.

- HoloV (2026) – The Synthesis

HoloV takes the stroke language of the V-Model and V-Bounce, the specification discipline of SDD, and introduces a structural layer that may be relevant for enterprise adoption: the holonic delivery unit. The concept is drawn from Arthur Koestler's holon, something that is simultaneously whole in itself and part of a larger whole.

V-Bounce explains how a team's motion changes with AI. HoloV answers the next question: how should an enterprise delivery system be structured around Run, Change, Explore, and Modernize priorities?

It provides a way to coordinate human teams, AI agents, and hybrid holons through a single living specification. This allows organizations to move toward autonomous delivery in a deliberate and reversible way.

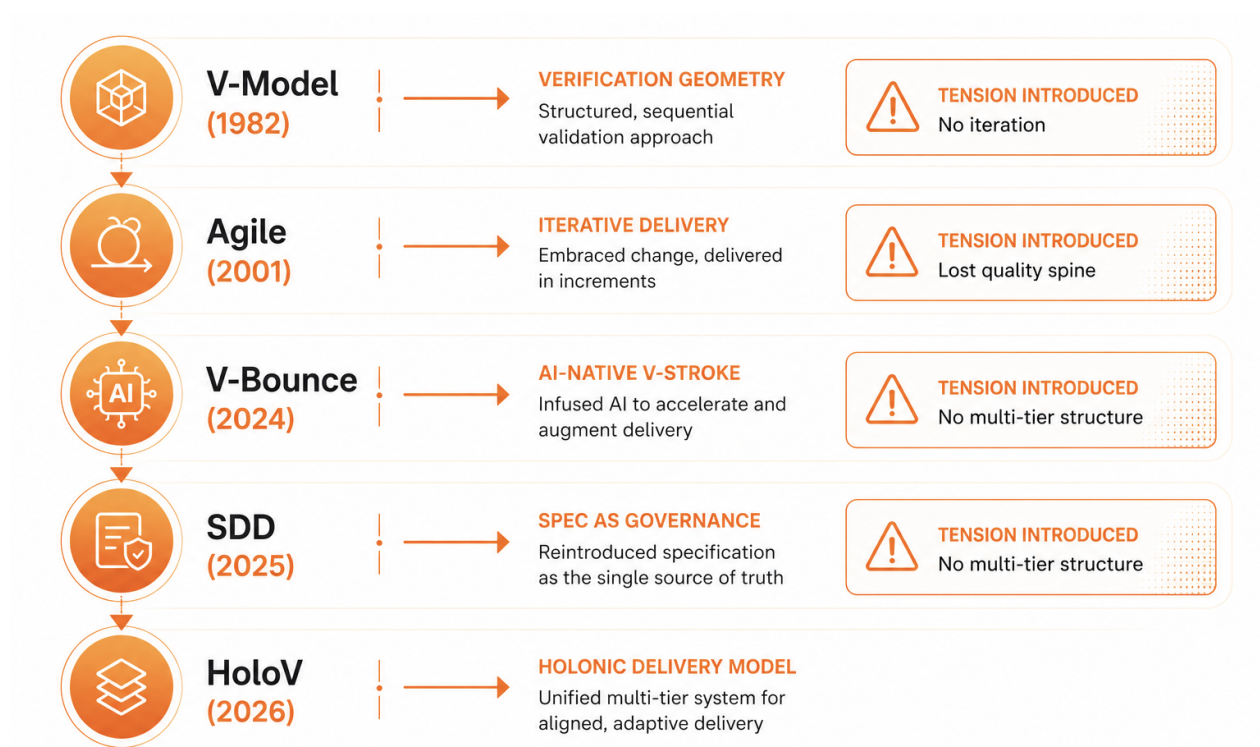


Figure 2 – Intellectual lineage. Each model resolved a prior tension and introduced a new one that the next model

Core Philosophy

HoloV rests on three philosophical commitments that distinguish it from every prior delivery framework.

- **Delivery Units are Holons, Not Steps**

A step is complete only when it feeds the next step. Arthur Koestler introduced the term "holon" in *The Ghost in the Machine* (1967) to describe something that is simultaneously a whole and a part. A holon is complete in itself while contributing to a larger whole. It operates independently and delivers measurable outcomes.

This distinction matters because it changes the relationship between delivery units from dependency to composition. A Run holon, Change holon, Explore holon, or Modernize holon does not wait for another unit to finish before it can



act. Each operates within its own bounded context and cognitive load profile, drawing from and contributing back to the live spec spine. The holons run against the spec, not against each other.

- **The Spec is the Only Interface**

In most delivery models, teams are coupled by artifacts, timelines, handoff meetings, shared tools, and implicit context. In HoloV, context stays inside the holon; the only thing that crosses a holon boundary is the spec. This makes each holon genuinely independent: you can change the tools, people, AI agents, and models inside any holon without affecting the others, as long as the spec contract is honored.

The spec is not a document that lives in one holon. The governed projection of context across holon boundaries provides the minimum expression of intent, constraints, acceptance criteria, and decisions needed for another holon or agent to act safely. It begins as a statement of intent and progressively hardens into machine-executable specification as each holon reads from it, acts on it, and writes back to it.

- **Autonomy is a Property of Spec and Context Maturity, not AI Capability**

The question most organizations ask is: when is AI good enough to work autonomously?

HoloV asks a different question: when is the spec mature enough, and the holon context bounded enough, for a human team, AI agent, or hybrid holon to reduce human verification without increasing delivery risk?

Autonomy is not granted by tool capability alone. It is earned by the combined maturity of the spec and the bounded context in which the holon operates. This reframes the AI transition as a question of governance and how work is designed.

The Model Defined

HoloV has four structural elements. They can be understood independently but only operate meaningfully together.

• The V-Stroke

The V-stroke is the unit of work motion inside any holon. The down-stroke translates spec into action – design, generation, change, experiment, remediation, or modernization. The up-stroke verifies the result back against the spec through tests, validation evidence, operational signals, or decision records.

Every Run, Change, Explore, and Modernize holon runs V-strokes, but the amplitude of the V differs by work priority and by the cognitive load carried inside that bounded context.

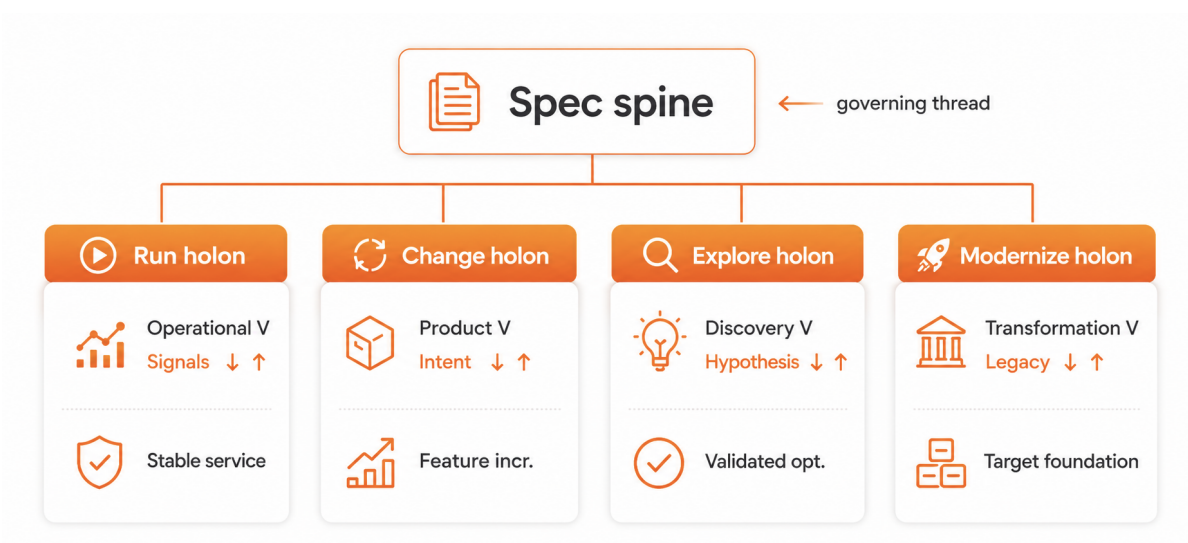


Figure 3 – The full HoloV model. Run, Change, Explore, and Modernize holons each run V-strokes against a shared spec spine, while preserving their own bounded context and cognitive load profile.

• The Holon

The holon is the bounded delivery unit. It has a defined context, receives input from the spec spine, follows a process through V-strokes, and returns its output to the spec. In HoloV, holons are organized primarily around work priorities rather than team seniority or AI maturity. Four holon types characterize an enterprise HoloV program:

Holon type	Work priority	Context and cognitive load	Primary output
 Run	Maintain and manage production systems that serve customers	Operational stability, reliability, incidents, service quality, and change risk	Stable service, verified fixes, updated runbook and production spec
 Change	Build new features and extend current product capability	Product intent, backlog interpretation, acceptance criteria, and release impact	Validated feature increment and refined product spec
 Explore	Explore, synthesise, and validate new capabilities	Ambiguity, discovery, option shaping, feasibility, and learning velocity	Validated option, experiment evidence, and emerging capability spec
 Modernize	Transform legacy constraints into future-ready product foundations	Legacy knowledge, architectural debt, migration risk, reverse engineering, and target-state design	Modernized foundation, transition evidence, and target-state spec

Figure 4 – The four work-priority holons. Each has a bounded context, a distinct cognitive load profile, and a defined output back into the spec spine.

• The Spec Spine

The spec spine is the horizontal governance thread that connects all holons without collapsing their bounded contexts. It persists across the entire program arc, beginning as a statement of intent and progressively hardening into machine-executable specification. The only artifact that exists outside holon boundaries is the spec spine. It provides the governed projection of context that allows Run, Change, Explore, and Modernize holons to act independently while keeping the delivery system coherent.

• The Autonomy Gradient

HoloV follows a clear progression, but it is not a one-way march from human work to AI work. Each Run, Change, Explore, or Modernize holon progresses toward greater autonomy only when its spec is mature enough and its bounded context is stable enough to reduce human verification without increasing delivery risk. The autonomy gradient is therefore not a timeline; it is a maturity signal driven by the combined readiness of the spec, the holon context, and the verification harness.



Operationalizing HoloV

- **Operating the Run Holon**

Operationalizing HoloV means assigning enterprise delivery work to bounded holons rather than arranging teams in a maturity sequence. The Run holon protects production stability. It converts service signals, incidents, and operational constraints into verified fixes, runbook updates, and production spec refinements.

Its V-strokes are operational: the down-stroke moves from signal to remediation, and the up-stroke verifies service stability, regression safety, and production readiness. The Run holon writes back only the governed production learning that other holons need, keeping operational context bounded while strengthening the spec spine.

- **Operating the Change Holon**

The Change holon extends current product capability. It receives product intent from the spec spine, interprets backlog priorities, shapes acceptance criteria, and turns them into validated feature increments. Its bounded context contains the product area, user flows, release constraints, dependencies, and quality expectations required to deliver change safely.

Its V-strokes are product-oriented. The down-stroke turns intent into design, generation, and implementation. The up-stroke verifies the feature against acceptance criteria, test evidence, and release impact. The Change holon contributes refined product specification, feature evidence, and decision records back to the spec spine.

- **Operating the Explore Holon**

The Explore holon carries ambiguity deliberately. It investigates emerging capabilities, synthesizes options, validates hypotheses, and converts uncertainty into governed choices. Its bounded context is discovery-heavy, spanning market signals, user needs, technology possibilities, feasibility constraints, and continuous learning.

• Operating the Modernize holon

The Modernize holon transforms legacy constraints into future-ready product foundations. It operates where cognitive load is highest, including reverse engineering, architectural debt, migration risk, hidden business rules, data dependencies, and target-state design. Its purpose is not merely to replace old systems, but to convert legacy knowledge into governed, executable specification.

Its V-strokes are transformational: the down-stroke translates legacy insight into target architecture, migration actions, and modernized implementation; the up-stroke verifies behavioral equivalence, transition safety, architectural fitness, and business continuity. The Modernize holon writes back target-state specifications, transition evidence, and modernization decisions into the spec spine.

• The Spec Transfer Ritual

The spec transfer ritual is the only inter-holon ceremony HoloV prescribes. It is not a team handoff and it is not a gate review. It is the governed transfer of the minimum sufficient spec contract required for another holon, human team, AI agent, or hybrid holon to act safely without inheriting the full context of the originating holon.

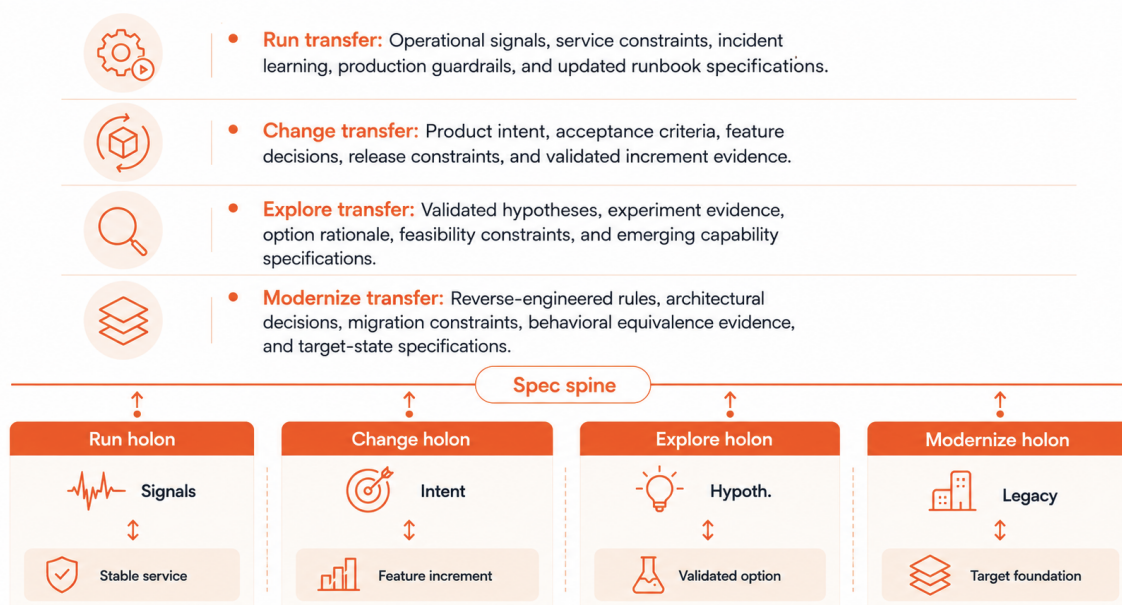


Figure 5 – Operationalizing HoloV. Run, Change, Explore, and Modernize holons operate in parallel against the spec spine, transferring only governed spec contracts across bounded contexts.



Tooling Ecosystem

HoloV is tool-agnostic at the philosophy level but agentic at the operating level. Tool selection is not merely a vendor choice; it defines what work can be delegated to agents inside each holon, what must remain human-governed, and what evidence is required before human verification can be reduced. Each Run, Change, Explore, and Modernize holon therefore has a distinct agentic stance shaped by its work priority, bounded context, cognitive load, spec maturity, and verification readiness.

- **Agentic Stance by Holon and Layer**

Layer	Holon applicability	Agentic role	Tool examples	Governance stance
 Spec authoring	All holons	Agents structure, refine, decompose, and validate specs before execution.	AWS Kiro, GitHub Spec Kit, Claude Code	Human-approved, version-controlled
 Production operations	Run	Agents detect signals, propose fixes, update runbooks, and execute low-risk remediation.	Observability, incident, runbook, CI/CD agents	Guard railed, risk-tiered
 Product change	Change	Agents translate intent into design, code, tests, and release evidence.	Cursor, Claude Code, Copilot, test agents	Spec-gated, test-verified
 Discovery and synthesis	Explore	Agents scan signals, synthesize patterns, generate prototypes, and compare options.	Research agents, LLM workbench, prototyping tools	Human-governed, evidence-backed
 Modernization	Modernize	Agents reverse-engineer code, extract rules, propose target architectures, and generate migration specs.	Lifter, code analysis, migration agents	Evidence-led, equivalence-verified
 Verification harness	All holons	Agents execute tests, scan risks, detect drift, and trigger escalation.	CI/CD, security, quality, test agents	Automated up-stroke with escalation
 Delivery telemetry	All holons	Agents monitor flow, detect bottlenecks, and recommend progression or step-back.	LinearB, Faros AI, Jellyfish	Observable, reversible

Figure 6 – HoloV tooling ecosystem by holon, agentic role, and governance stance.

- **The Spec Toolchain**

The spec spine is the most critical infrastructure in a HoloV program because it is the contract through which human teams, AI agents, and hybrid holons act safely.

The spec toolchain must support version control, machine readability, governed spec contracts, and cross-holon access without collapsing bounded contexts. It must also make agentic execution observable, reversible, and auditable.



The following properties are required of any spec toolchain adopted:

- **Version-controlled:** Every human or agent-initiated spec change is traceable, attributed, reviewable, and reversible.
- **Structured:** Expressed in a format that agents can decompose, reason over, and execute against without relying on implicit context.
- **Machine-readable:** Consumable directly by generation, verification, remediation, discovery, and modernization agents without manual transformation.
- **Living:** Updated by every holon as it runs, with agent contributions reviewed, accepted, or rejected through governed workflow.
- **Contracted:** Exposes only the minimum sufficient spec contract required for another holon or agent to act safely, while keeping local context bounded.
- **Auditable:** Records agent actions, human approvals, verification evidence, escalations, and step-back decisions as part of the spec history.

• Tool Transition Across the Gradient

A critical design principle of the HoloV tooling ecosystem is that agentic delegation should trail spec and context maturity, not lead it. A Run, Change, Explore, or Modernize holon does not adopt autonomous or agentic tooling simply to accelerate.

It adopts higher-agency tooling only when the spec is mature enough, the bounded context is stable enough, and the verification harness is strong enough to make reduced human verification safe.

The transition is also reversible at the holon level. If a production incident, scope change, discovery ambiguity, legacy constraint, or verification failure introduces uncertainty that an agent cannot resolve safely, the holon steps back from autonomous or agentic execution to assistant-led or human-led operation.

The spec spine absorbs the new learning, the bounded context is restored, and the holon re-qualifies before increasing agentic agency again.



KPIs and the HoloV Harness

HoloV introduces a measurement harness that operates across four layers.

- Within-holon health
- AI effectiveness and autonomy
- Cross-holon progression
- Harness governance

Together, these metrics show whether each Run, Change, Explore, or Modernize holon is executing its V-strokes well, whether agentic work is effective, and whether the program is becoming more coherent, governable, measurable, and safely autonomous across the spec spine.

- Within-Holon KPIs

KPI	Measures	Target signal	Holon	Tool source
 Spec coverage ratio	% of features with complete spec before code gen	> 90% before down-stroke	Change	GitHub Spec Kit, Kiro
 Spec drift rate	Divergence between spec and generated code at review	< 5% per iteration	Change	Spec diff tools, Claude Code
 Architecture decision log completeness	% of decisions documented with rationale	100% at spec transfer	All holons	ADR tools, Confluence
 Refactor fidelity	% of refactored code that passes original acceptance criteria	> 95%	Modernize	CI pipeline, SonarQube
 Test coverage against spec	% of acceptance criteria covered by automated tests	> 85%	All holons	Copilot, Diffblue, Jest
 Spec ambiguity surfaced	Ambiguities raised back to spine per iteration	Tracked, trend down	Explore	Jira, spec toolchain

Figure 7 – Within-holon KPIs. Each Run, Change, Explore, and Modernize holon has health metrics tied to its V-strokes, bounded context, and agentic operating stance.



- **AI Effectiveness and Autonomy KPIs**

AI effectiveness KPIs measure whether agentic work is producing useful, safe, and high-quality outputs inside each holon. Autonomy KPIs measure whether the system can safely reduce human verification without increasing delivery risk.

Together, they prevent HoloV from treating agentic adoption as a productivity claim alone. Agents must improve flow while keeping the spec reliable and the delivery process reversible.

KPI	Measures	Target signal	Scope	Tool source
 Agent acceptance rate	% of agent outputs accepted with no or minor human edits	Trend up, > 80% for mature holons	Per holon	Agent logs, review tools
 Agent rework rate	% of agent outputs requiring significant correction or redesign	Trend down, < 10% for mature holons	Per holon	Pull requests, review tools
 Agent defect escape rate	Defects traced to agent-generated or agent-modified artifacts after verification	Near zero; any escape triggers review	Run, Change, Modernize	Defect tracker, CI/CD, incident logs
 Prompt/spec clarification rate	How often agents require additional human clarification before safe execution	Trend down as spec matures	All holons	Spec toolchain, agent prompts
 Evidence completeness rate	% of agent actions accompanied by test, rationale, trace, or decision evidence	> 95% before autonomy increase	All holons	Verification harness, ADR logs
 Agent productivity leverage	Cycle-time or throughput improvement attributable to agentic execution without quality degradation	Improves with stable defect and rework rates	Per holon	Delivery telemetry, review analytics

Figure 8 – AI effectiveness and autonomy KPIs. These measure whether agentic execution is useful, safe, evidence-backed, and worthy of increased autonomy.

• Cross-Holon Progression Metrics

KPI	Measures	Target signal	Scope	Tool source
Spec maturity index	Composite score: coverage + structure + machine-readability	> 80 to progress	Spine	Spec toolchain audit
Spec transfer success rate	% of spec transfers accepted first time without rework or context leakage	> 90%	Between holons	Spec transfer log
Autonomy readiness score	% of spec and context layers meeting safe agentic execution criteria	> 85% to transition	Per holon	Spec completeness audit
Delivery cycle time	End-to-end time from spec to production per feature	Trend down across gradient	Program	LinearB, Jellyfish
Human intervention rate	% of autonomous cycles requiring human input	< 8%, trend to < 3%	Per holon	Faros AI, observability
Spec regression rate	% of spec layers requiring rework after autonomy transition	< 5%	Program	Spec version control

Figure 9 – Cross-holon progression metrics. These govern spec transfer quality, holon-level autonomy readiness, and reversible progression toward safer agentic execution.

• The HoloV Harness

The HoloV harness is the observability and governance infrastructure that feeds these metrics. It is not a single tool – it is a composed pipeline drawing from the tooling ecosystem defined in Section 6, the spec spine, the verification harness, delivery telemetry, and agent-action logs across all holons.

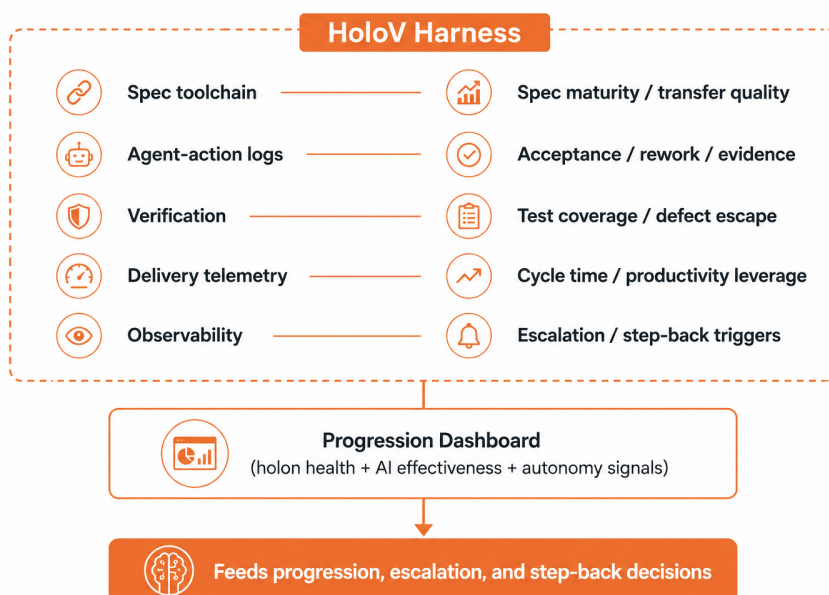


Figure 10 – The HoloV harness. A composed observability pipeline that surfaces within-holon health, AI effectiveness, autonomy, and cross-holon progression signals.



Three governance principles apply to the harness:

- **No Manual KPI**
Every metric in the harness is derived from tool telemetry, spec history, verification evidence, and agent-action logs rather than self-reported status.
- **Spec and Context Quality Gates Before Progression**
No holon increases agentic agency without meeting thresholds for spec maturity, bounded-context stability, and autonomy readiness.
- **Reversibility is Instrumented**
The harness monitors agentic execution continuously. Escalation rates, spec regression, context instability, or verification failure trigger step-back to assistant-led or human-led operation until the holon re-qualifies.

The Agentic Autonomy Progression

Organizations need to know when each Run, Change, Explore, or Modernize holon is ready to increase agentic agency without increasing delivery risk. HoloV answers this through a structured approach. Autonomy progresses only when the spec is mature, the bounded context is stable, the verification harness is strong, and evidence shows AI can safely reduce human oversight.

- **Progression Criteria**

A holon is ready to progress toward higher agentic autonomy when five conditions hold simultaneously:

- **Spec Maturity:** The spec layer is complete, structured, version-controlled, machine-readable, and unambiguous enough for agents to execute without resolving hidden intent.
- **Bounded-Context Stability:** The holon's domain, constraints, interfaces, tools, data, and decision rights are stable enough that agents can act without inheriting unresolved external context.
- **Verification Completeness:** The up-stroke can be expressed through automated tests, operational signals, evidence checks, or decision records that verify agent output against the spec.



- **AI Effectiveness Evidence:** Agent acceptance, rework, defect escape, clarification, and evidence-completeness metrics show that agentic execution is useful, safe, and improving flow without degrading quality.
- **Escalation and Step-Back Protocol:** A governed protocol exists to route ambiguity, verification failure, context instability, or risk escalation back to human-led or assistant-led operation.

• Holon-Specific Progression

Autonomy progresses independently inside each holon. A Run holon may safely delegate low-risk remediation while an Explore holon remains human-governed because ambiguity is still high. A Change holon may progress to agentic design, implementation, and test generation once acceptance criteria and release evidence are stable. A Modernize holon may remain evidence-led for longer because legacy context, equivalence risk, and migration constraints carry higher cognitive load.

These conditions put the decision in the hands of the delivery system, not the AI vendor. Teams evaluate whether a holon's spec, context, verification evidence, and agent performance are mature enough to safely reduce human

• The Progression is Reversible

The progression is reversible at the holon level. If a production incident, scope change, discovery ambiguity, legacy constraint, verification failure, or agent effectiveness drop introduces uncertainty that the agent cannot resolve safely, the holon steps back from autonomous or agentic execution to assistant-led or human-led operation.

The spec spine absorbs the new learning, the bounded context is restored, and the holon re-qualifies before increasing agentic agency again. Reversibility is a feature, not a failure mode.

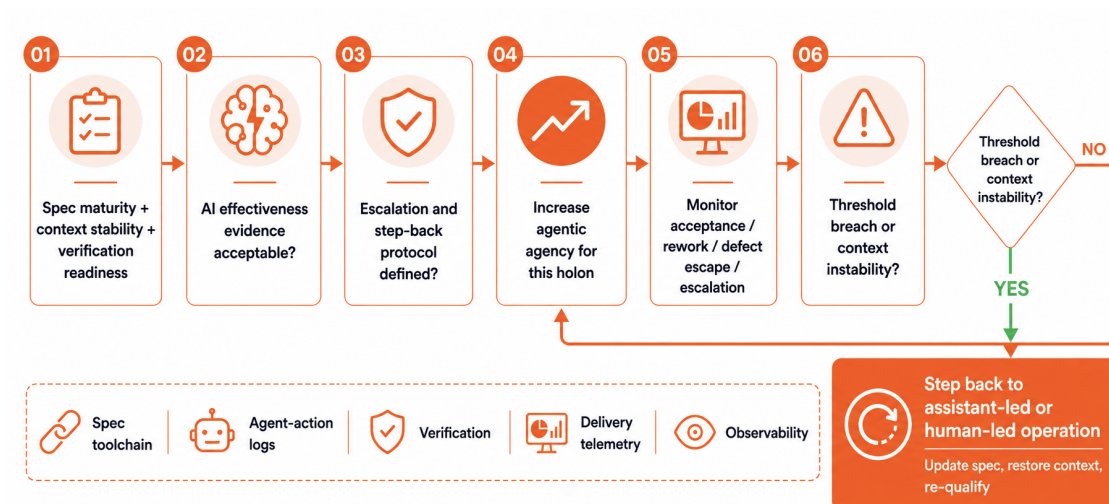


Figure 11 – Agentic autonomy progression decision tree. Entry is governed by spec maturity, context stability, verification readiness, and AI effectiveness evidence; exit is triggered by harness signals and reversible step-back.

Why This Matters

Every delivery framework is a claim about the hardest problem in software delivery. Waterfall claimed it was requirements volatility. Agile claimed it was plan rigidity. SAE claimed it was organizational alignment. V-Bounce claimed it was the separation of building from verifying.

HoloV makes a different claim. The next stage of software delivery depends on more than AI capability. Organizations lack a structural model for managing the transition from human-led to agentic delivery without forcing a one-way shift to agent-led execution.

• The Structural Gaps HoloV Closes

Problem	Prior frameworks	HoloV
Different work priorities, bounded contexts, and cognitive loads	No structural vocabulary beyond team type or maturity level	Run, Change, Explore, and Modernize holons. Each whole, bounded, and composable
AI generation without spec governance	Spec drift, automated legacy, and unclear ownership	Spec spine, the governed projection of context and the only interface across holons
No governed path to agentic autonomy	Ad hoc tool adoption and vendor-led maturity claims	Agentic autonomy progression, governed by spec maturity, context stability, verification readiness, and AI effectiveness evidence
Human teams, AI agents, and hybrid holons working without a shared operating model	Tools adopted locally with inconsistent roles, controls, and evidence	Agentic operating model, clear delegation, governance stance, evidence requirements, and human/agent boundaries by holon
Autonomy without reversibility	One-way transition from human-led work to agent-led execution	Harness-triggered step-back, reversible movement between agentic, assistant-led, and human-led operation

Figure 12 – HoloV problem/resolution map.



• The Broader Claim

HoloV provides a structured path through the AI transition by strengthening human delivery with AI.

- Human leaders govern intent, risk, context boundaries, and spec quality rather than merely supervising task execution.
- Human teams and AI agents collaborate inside Run, Change, Explore, and Modernize holons, each with a bounded context, cognitive load profile, and governed output into the spec spine.
- Agents take on structured execution only when the spec, context, verification evidence, and AI performance show it is safe. The verification harness can reduce agent autonomy when conditions change.

The spec becomes the durable asset, while holons provide the operating model and the V brings quality discipline across the system. Autonomy is continuously managed through the clarity of human intent rather than treated as a final destination.

HoloV governs the transition to agentic software delivery across different work priorities, bounded contexts, and levels of agentic autonomy.



About Indium

We are an AI services company specializing in Agentic AI, Data & Analytics, Application Engineering, and Quality Engineering.

By combining autonomous AI systems, modern engineering, and data intelligence, we build innovative AI solutions that drive measurable business outcomes and long-term value.

With 5,000+ associates worldwide, Indium partners with Fortune 500 companies, Global 2000 enterprises, and leading technology firms in Financial Services, Healthcare, Manufacturing, Retail, and Technology. Our teams operate in North America, India, the UK, Singapore, Australia, and Japan, helping businesses stay ahead in an AI-first world.

USA

Cupertino | Princeton
Toll-free: +1-888-207-5969

INDIA

Chennai | Bengaluru | Mumbai
Hyderabad | Pune
Toll-free: 1800-123-1191

UK

London
Ph: +44 1420 300014

SINGAPORE

Singapore
Ph: +65 6812 7888

www.indium.tech



For Sales Inquiries
sales@indium.tech



For General Inquiries
info@indium.tech

