University of New Mexico

HERA 2.0: Human-Governed Agentic Software Development with External Cognitive Orchestration -- An Industrial Experience Report on AI-DLC with ChatGPT Web as Cognitive Continuity Layer

Manuel-Enrique Puebla-Martínez
Universidad del Trabajo del Uruguay, Montevideo, Uruguay; mpuebla12@gmail.com

Abstract--The adoption of LLM-based agents in industrial software engineering has surfaced a structural problem: local agents can conduct AI-DLC workflows correctly while simultaneously pursuing incomplete or misguided problem interpretations when context is fragmented or prompts are imprecise. This experience report presents HERA 2.0, a methodology that addresses this problem through architectural separation of responsibilities: local agents perform situated execution (repository observation, command execution, evidence generation under AI-DLC rules), while ChatGPT Web acts as an external cognitive continuity layer (knowledge integration, operative prompt generation, adversarial review). The synchronization protocol relies on curated Markdown living artifacts, a single executable instruction file (instruccion-agente.md), and a Run ID anti-old-instruction mechanism introduced in V7. Evidence classification in HERA uses four operational states -- confirmed, reasonable inference, unvalidated assumption, and discarded after disconfirmation -- that naturally corresponds to the (T, I, F) structure of neutrosophic evidence degrees, providing a formal epistemic grounding for the methodology's knowledge management layer. An anonymized industrial case on POS-switch integration with timeout inconsistency and transactional compensation illustrates differences between HERA and direct AI-DLC across nine comparison rows in one industrial case. The methodology is relevant to SDG 9 (Industry, Innovation and Infrastructure) through a potentially replicable, evidence-governed pattern for sustainable AI-assisted software engineering that does not require hyperscale AI infrastructure -- a relevant constraint for technology organizations in emerging economies.

Keywords--HERA 2.0; AI-DLC; agentic software development; ChatGPT; cognitive continuity; operative prompt chain; agent autonomy governance; living context; experience report; human-in-the-loop.

1 Introduction

The integration of large language model (LLM) agents into software engineering practice has progressed rapidly from IDE-level code suggestions [2] toward autonomous agents capable of inspecting repositories, generating multi-file changes, executing tests, and producing compliance artifacts [9]. Methodologies such as AI-DLC [1] structure this process through phases, Markdown rule files, human checkpoints, and formal audit trails, converting agent interactions into traceable, reversible software engineering processes. However, a structural problem persists in industrial deployments: local agents may correctly follow AI-DLC phases while operating on incomplete problem interpretations -- what this paper terms methodologically ordered but cognitively misguided autonomy. The risk is not the absence of methodology; it is the execution of a disciplined process in the wrong direction.

This experience report presents HERA 2.0, an industrial methodology that addresses this problem through explicit architectural separation: local agents execute AI-DLC workflows in the workspace (situated execution), while ChatGPT Web governs the strategic reasoning layer (cognitive continuity). The separation is not arbitrary -- it respects the functional design of each tool: code agents are optimized for situated repository operations, while ChatGPT Web's conversational architecture is particularly suited for sustained reasoning over incrementally discovered, heterogeneous evidence. HERA does not compete with AI-DLC; it complements it.

One feature of HERA's knowledge management that deserves attention is the mandatory classification of findings into three categories: confirmed by code, reasonable inference, or unvalidated assumption. This tripartition was present in the original HERA design as a practical discipline; what the neutrosophic evidence degree framework [12,13] adds is a formal language for it. In that framework, T captures confirmed support, I captures epistemic indeterminacy, and F captures evidenced disconfirmation -- a correspondence that is not superficial. Industrial software investigations routinely encounter states where evidence is partial, hypotheses compete, and some uncertainty genuinely cannot be resolved before the next execution cycle. The formal grounding opens a path toward quantitative tracking of knowledge state health across HERA cycles, which the original methodology could not provide.

The paper makes five contributions. The primary one is HERA 2.0 itself: an integration pattern between AI-DLC, local agents, and ChatGPT Web as external cognitive orchestrator, including a synchronization protocol built on curated Markdown living artifacts and a single executable instruction file (instruccion-agente.md). Secondary contributions include the concept of governed agent autonomy and its operational distinction from strategic direction; the V7 anti-old-instruction mechanism based on Run ID; and an anonymized industrial case that makes the comparison between direct AI-DLC and HERA 2.0 described across nine comparison rows. A final contribution is the formal epistemic grounding of HERA's C/I/U evidence classification via neutrosophic evidence degrees -- a connection that was implicit in the methodology and is here made explicit.

2 Background and Related Work

2.1 AI-DLC and Agentic Software Engineering

AI-DLC [1] provides the operative infrastructure for agent-assisted development: Markdown rule files, state documents, audit trails, phased workflows, and human checkpoints. Its principal strength is converting agent interactions into locally traceable repository processes. Recent agentic frameworks such as SWE-agent have extended this direction, showing that LLM agents can address repository-level GitHub issues through systematic exploration [17]. However, these frameworks primarily address single-task resolution and do not address the governance of extended, multi-cycle industrial developments where context evolves iteratively.

2.2 Cognitive Continuity in Human-AI Teaming

Bommasani et al. [9] survey foundation model capabilities, noting that conversational LLMs and code agents exhibit complementary strengths: conversational models excel at sustained reasoning and knowledge integration, while code agents excel at situated repository operations. HERA 2.0 exploits this complementarity explicitly, assigning each tool the tasks aligned with its design. Amershi et al. [16] identify AI-human responsibility distribution as a primary determinant of ML-augmented engineering quality -- a finding that directly motivates HERA's separation of strategic direction from local execution.

2.3 Vibe Coding and Its Methodological Risks

Vibe coding -- the practice of exploring technical solutions through fluent conversational interaction with LLMs -- has accelerated discovery and prototyping in software development. Its principal risk is that conversation advances faster than understanding and evidence: when advancement criterion is seeming functionality, development is exposed to silent errors, technical debt, and incomplete validations. HERA 2.0 can be understood as a disciplined evolution of vibe coding: it preserves the conversational fluidity and exploratory speed while anchoring each advance in verifiable evidence, structured decisions, and governed prompt chains.

3 HERA 2.0 Architecture

3.1 Four-Layer Design

HERA 2.0 organizes responsibilities across four layers. Layer 1 (Living Artifacts) provides the shared memory substrate: Markdown documents encoding current state, decisions, risks, open questions, synchronization packages, executed prompts, agent responses, verdicts, and

postmortems -- a structured living context per development distinct from the formal AI-DLC audit trail. Layer 2 (Local Agents) performs situated execution: repository observation, command execution, code modification, test execution, and evidence artifact generation under AI-DLC rules. Layer 3 (ChatGPT Web) provides cognitive continuity: reading synchronization packages, integrating accumulated findings, classifying evidence by epistemic status, generating operative instructions, and performing adversarial review. Layer 4 (Human Control) maintains final authority: approving scope, curating living context, validating evidence, and assuming risk ownership. The living-artifact and multi-agent coordination layers can also benefit from causal maps that reconcile judgments expressed on heterogeneous linguistic scales; the extended hierarchical linguistic fuzzy-cognitive-map model provides such a mechanism [20].

3.2 Evidence Classification and Epistemic Grounding

A defining feature of HERA's knowledge management is the mandatory classification of findings into four operational states: (C) confirmed by code or tests, (R) reasonable inference pending verification, (U) unvalidated assumption, and (D) discarded after disconfirmation. This is not an identity with a neutrosophic triplet. Rather, the retained evidence can be summarized by degrees (T, I, F), where support for C contributes to T, unresolved R/U contributes to I, and explicit counterevidence recorded in D contributes to F [12,13]. A claim may weaken from C to R or U when its support is challenged, whereas contradictory evidence can move it to D. This gives the living context folder an auditable epistemic interpretation without forcing mutually exclusive workflow labels into three probability-like components. Directed graphs and fuzzy cognitive maps can complement this representation when relations among findings must be made explicit [21]. Template-controlled neutrosophic auditing of LLM verdicts is a relevant precedent for testing whether conclusions follow evidence rather than prompt wording [22].

Table 1. HERA 2.0 evidence classification with neutrosophic correspondence.

Category	HERA Label	Neutrosophic	Trigger	Action
Confirmed	C -- confirmed by code/test	$T > 0.7, I < 0.2, F < 0.1$	Code inspection or test passing	Include in operative instruction as constraint
Inference	I -- reasonable inference	$T \sim 0.5, I \sim 0.4, F < 0.2$	Logical deduction, no direct evidence	Flag explicitly; request verification
Unvalidated	U -- unvalidated assumption	$T < 0.4, I > 0.4$, F variable	Hypothesis without code support	Prohibit in operative scope; re-investigate
Discarded	D -- hypothesis discarded	$T < 0.2, I < 0.2, F > 0.7$	Contradicted by evidence	Archive in decisions.md; exclude from instruction

3.3 Synchronization Protocol and Run ID Mechanism

The V7 synchronization protocol establishes an auditable channel between ChatGPT Web and local agents. The ChatGPT-to-agent direction materializes through a single executable local file (instruccion-agente.md) containing the active instruction generated by ChatGPT, reviewed and approved by the human, and identifiable by a unique Run ID. The agent-to-ChatGPT direction relies on living context artifacts synchronized to an accessible source (Google Drive or repository). Before executing, the agent must verify that the instruccion-agente.md Run ID matches active-run-id.txt and does not appear as completed, canceled, or replaced in completed-runs.md. This mechanism eliminates the most common failure mode in manual ChatGPT-agent workflows: execution of old, replaced, or cross-development instructions due to informal copy-paste without identity controls.

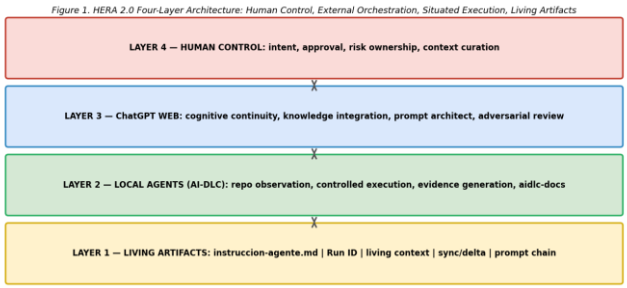

Figure 1. HERA 2.0 Four-Layer Architecture: responsibilities per layer and bidirectional information flow between cognitive and execution planes

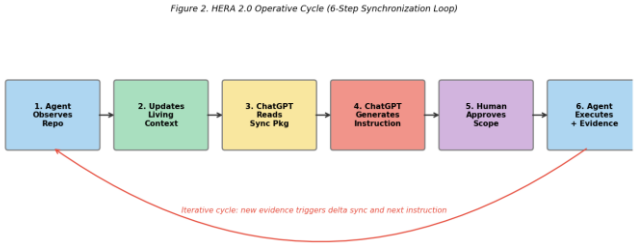

Figure 2. HERA 2.0 Six-Step Operative Cycle showing the agent-context-ChatGPT-instruction-approval-execution synchronization loop

4 Industrial Case: POS-Switch Timeout and Transactional Compensation

4.1 Technical Context

The anonymized case involves a legacy enterprise Java Point-of-Sale (POS) system integrated with a payment services switch via SOAP/HTTP. The analyzed operation is a credit authorization with quota discount. The critical failure mode occurs when the POS sends an authorization request to the switch, the switch processes and discounts the quota, but the POS receives a timeout or no-response exception. The POS records the operation as uncertain; the switch has applied an irreversible quota change. Without transactional compensation, the customer suffers an effective quota reduction without a completed POS transaction -- a regulatory and service quality violation.

4.2 Why This Case Requires HERA

The case cannot be addressed with a generic prompt ('fix the timeout'). Resolution requires: reconstructing the distributed transaction flow; distinguishing functional cancellation, technical reversal, transport timeout, annulment, and quota discount -- five concepts an agent may conflate without explicit operative constraints; identifying persistence points, idempotency requirements, and existing compensation builders/handlers; and producing tests covering late responses, null responses, and connection errors. In direct AI-DLC, an agent receiving an incomplete initial prompt could produce a formally correct AI-DLC flow tracing the wrong problem. HERA addresses this by making ChatGPT Web -- fed the full living context including the C/I/U evidence classification -- the architect of the operative instruction that constrains the agent's search space.

4.3 Operative Prompt Chain Applied

Table 2. Operative prompt chain for the POS-switch timeout case.

Moment	Operative Function	Primary Risk Controlled
Inception	Reconstruct POS -> Switch flow; classify authorization, reversal, annulment, timeout, quota discount	Prevents implementation before flow reconstruction

Moment	Operative Function	Primary Risk Controlled
Clarification	Do not assume quota reverses from POS; verify transaction layer registration and POS exception receipt	Prevents wrong layer responsibility assignment
Reorientation	Previous analysis conflated functional cancellation and technical reversal; re-classify and re-identify failure points	Prevents contaminated hypothesis propagating into plan
Restriction	Do not modify payment classes; inspect only authorizers, switch client, exception handling	Prevents premature or out-of-scope changes
Validation	Propose minimal tests for timeout before/after authorization, null response, late response, connection error	Prevents declaring success without edge case coverage
Adversarial Review	Search risk of double reversal, omitted reversal, message change to user, regression in unrelated flows	Identifies real failures before closure
Closure	Update living context, decisions, residual risks, evidence matrix, PR description	Converts case into reusable knowledge artifact

5 Comparison: Direct AI-DLC versus HERA 2.0

Table 3. Comparative analysis: direct AI-DLC versus HERA 2.0.

Dimension	Direct AI-DLC	HERA 2.0
Flow Conduction	Local agent conducts cycle; asks, plans, proposes	Agent executes AI-DLC; strategic direction governed by human + ChatGPT
Agent Autonomy	High local autonomy; sensitive to incomplete context	Governed autonomy: agent observes/executes; ChatGPT restricts and reorients
Initial Prompt	Written directly by human; variable quality	Generated or refined by ChatGPT with curated living context
Prompt Chain	Intermediate responses may remain informal conversation	Complete chain treated as versioned engineering artifact
Evidence Classification	aidlc-docs, audit, workflow state	C/I/U tripartition + living context + delta + prompt history
Cognitive Continuity	Local agent concentrates reasoning and execution	ChatGPT sustains global case logic; agent provides situated evidence
Human Control	Phase approval and local review	Context curation + external adversarial review + risk ownership
Operative Friction	Low; everything local to workspace	Higher; requires ChatGPT sync and Drive/repository artifacts
Primary Advantage	Integrated local execution and traceability	Autonomy governance, strategic prompt quality, knowledge integration

The comparison makes explicit that HERA's advantage is not universal. For brief, well-specified, single-system tasks, direct AI-DLC offers lower friction with equivalent quality. HERA's differential value emerges in complex industrial contexts: legacy distributed systems, long investigations with iteratively discovered evidence, multiple interdependent repositories, high regression risk, or regulatory documentation requirements. The cost is explicit: greater operative friction from synchronization, Drive maintenance, and ChatGPT context curation.

6 Threats to Validity and SDG Alignment

6.1 Threats to Validity

Internal validity: improvements may result from developer experience or case-specific knowledge rather than the methodology. Mitigation: register prompts with Run IDs, define and prospectively register a rubric to compare direct and HERA-assisted prompts on matched tasks. External validity: the legacy POS-switch context may not generalize to greenfield projects or microservices architectures. Applicability conditions: HERA's differential value is highest when problems combine distributed integration, legacy code, long investigation cycles, and high regression risk. Construct validity: 'cognitive continuity' and 'governed autonomy' require observable proxies for quantitative evaluation. Researcher bias: the first author is the methodology's developer; mitigation through explicit threats, traceable artifacts, and external adversarial review built into the methodology itself.

6.2 SDG 9 Alignment

HERA 2.0 advances SDG 9 Target 9.b (support domestic technology development in developing countries) through two mechanisms. First, the methodology is designed for industrial organizations without hyperscale AI infrastructure: it requires only ChatGPT Web (accessible) and any AI-DLC-capable local agent (open-source). Second, HERA formalizes the governance practices that enable sustainable AI-assisted software engineering -- preventing the technical debt accumulation and silent error introduction that commonly follow ungoverned vibe coding adoption in resource-constrained development teams. The Open Group Architecture Framework (TOGAF) analogy is apt: HERA provides the architectural governance layer for agentic software development that emerging economy technology organizations need to adopt AI tools sustainably.

7 Conclusions and Future Work

HERA 2.0 presents a formally motivated integration pattern illustrated in one industrial case for AI-assisted software development that addresses the central weakness of direct AI-DLC deployment: cognitively misguided agent autonomy arising from incomplete context or imprecise prompts. The architectural separation between ChatGPT Web (cognitive continuity, knowledge integration, prompt governance) and local agents (situated execution, evidence generation) respects the functional design of each tool and reduces the risk of ordered but misguided development cycles. The V7 Run ID mechanism operationalizes instruction custody, converting informal copy-paste workflows into auditable, identity-controlled execution units. The (T, I, F) correspondence of HERA's evidence classification provides a provisional epistemic interpretation for the living context knowledge management layer.

Future work includes: (1) prospective rubric-based evaluation comparing matched direct and HERA-assisted prompts across diverse industrial tasks; (2) quantitative tracking of neutrosophic evidence state transitions (C -> I -> U) as a development health metric; (3) extension to multi-agent architectures where multiple local agents operate under HERA governance concurrently; and (4) empirical replication across domains beyond legacy POS integration.

Cross-domain evidence governance

The evidence-governance principle is not limited to software development. A field pose-estimation protocol distinguishes repeatable acquisition from unvalidated interpretation [18], while a neutrosophic stress test of TOPSIS carries stakeholder dissent as indeterminacy instead of collapsing it into one coefficient [19]. These applications motivate cross-domain tests of HERA's confirmed, inferred, and unvalidated evidence states.

Acknowledgments

The author thanks the practitioners who contributed to the industrial case, and the supporting institution for research support.

References

[1] M. E. Puebla Martinez, "AI-DLC: A Methodology for AI-Assisted Software Development with Human Control and Verifiable Evidence," preprint, 2024.

[2] GitHub, "GitHub Copilot," GitHub Inc., 2024. Available: https://github.com/features/copilot

[3] OpenAI, "ChatGPT Technical Report," OpenAI, 2023.

[4] S. R. Chidamber and C. F. Kemerer, "A metrics suite for object-oriented design," IEEE Trans. Softw. Eng., vol. 20, no. 6, pp. 476-493, 1994.

[5] M. Fowler, Refactoring: Improving the Design of Existing Code, 2nd ed. Addison-Wesley, 2018.

[6] K. Beck, Test-Driven Development: By Example. Addison-Wesley, 2002.

[7] A. Basili and B. T. Perricone, "Software errors and complexity: An empirical investigation," Commun. ACM, vol. 27, no. 1, pp. 42-52, 1984.

[8] M. Brundage et al., "Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims," arXiv:2004.07213, 2020.

[9] R. Bommasani et al., "On the Opportunities and Risks of Foundation Models," arXiv:2108.07258, 2021.

[10] X. Hou et al., "Large Language Models for Software Engineering: A Systematic Literature Review," ACM Trans. Softw. Eng. Methodol., vol. 33, no. 6, Art. 148, 2024. https://doi.org/10.1145/3695988

[11] M. Chen et al., "Evaluating Large Language Models Trained on Code," arXiv:2107.03374, 2021. [HumanEval benchmark]

[12] F. Smarandache, "A Unifying Field in Logics: Neutrosophic Logic," American Research Press, 1999.

[13] M. Leyva-Vazquez and F. Smarandache, "Neutrosophic paraconsistent logic: Evidence degrees, ontological indeterminacy, and scientific evidence synthesis," NCML, vol. 43, pp. 211-221, 2026.

[14] C. Wohlin et al., Experimentation in Software Engineering. Springer, 2012.

[15] R. Yin, Case Study Research and Applications: Design and Methods, 6th ed. Sage, 2017.

[16] S. Amershi et al., "Software Engineering for Machine Learning," in Proc. ICSE-SEIP 2019.

[17] J. Yang et al., "SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering," arXiv:2405.15793, 2024.

[18] D. M. Ramirez Guerra, C. E. De la Torre Choque, J. G. Herrera Chamorro, and J. K. Flores Zabaleta, "Field-Deployable Pose Estimation for Youth Athlete Screening: A Reproducible Protocol and Its Sensitivity Limits in a Genetically Controlled Case Series," accepted manuscript, CITIS 2026, paper 240, 2026.

[19] L. G. Lopezdominguez Rivas, S. D. Lopezdominguez Rivas, D. M. Lopezdominguez Rivas, and F. Parrales-Bravo, "How Much of a Multicriteria Ranking Is the Data and How Much Is the Arithmetic? A Neutrosophic Stress Test on Stakeholder Perception Data," accepted manuscript, CITIS 2026, paper 241, 2026.

[20] M. Leyva-Vázquez, E. Santos-Baquerizo, M. Peña-González, L. Cevallos-Torres, and A. Guijarro-Rodríguez, "The Extended Hierarchical Linguistic Model in Fuzzy Cognitive Maps," in Technologies and Innovation, Communications in Computer and Information Science, vol. 658, Springer, Cham, 2016, pp. 39–50, doi: 10.1007/978-3-319-48024-4_4.

[21] M. Leyva-Vázquez, K. Pérez-Teruel, A. Febles-Estrada, and J. Gulín-González, "Causal knowledge representation techniques: A case study in medical informatics," Revista Cubana de Información en Ciencias de la Salud, vol. 24, no. 1, pp. 73–83, 2013. https://www.medigraphic.com/pdfs/acimed/aci-2013/aci131f.pdf

[22] M. Leyva-Vázquez, A. Matheu Pérez, and F. Smarandache, "Los veredictos internos siguen a la evidencia: una lectura neutrosófica con control de plantilla de los estados epistémicos en grandes modelos de lenguaje mediante la lente jacobiana," Neutrosophic Computing and Machine Learning, vol. 44, pp. 465–475, 2026, doi: 10.5281/zenodo.21421596.