FJB

JOSEFF BETANCOURT

AI PLATFORM ARCHITECT | JOSEFFB.COM

FERNAIN RESEARCH | RESEARCH NOTE

AI-Assisted Software Delivery Management

A Human-Legibility Test and Candidate Execution Profile

A research framework for projecting agentic execution into accountable human control.
Written for AI enthusiasts, engineering leaders, and delivery professionals.

Prepared 13 Aug. 2026
Version 1.1.0
Author and publisher: Joseff Betancourt
Research support: Fernain Research

A Human-Legibility Test and Candidate Execution Profile

Author: Joseff Betancourt **Version:** 1.1.0 **Publication date:** 2026-08-13 **Publisher:** Joseff Betancourt **Research support:** Fernain Research **Status:** Publication version **Document type:** Research note **Review method:** Bounded, non-systematic narrative review **Research approach:** Design-science proposal with adversarial premise evaluation **Empirical status:** No original experiment or measurement **Citation style:** MLA 9 **Persistent identifier:** Not assigned **Canonical URL:** Pending public release **License:** To be selected before public release

Authorship and AI-assistance disclosure: Research direction and methodology: Joseff Betancourt. Evidence discovery, analysis, adversarial review, and drafting: AI-assisted. Final evaluation, editorial judgment, and publication responsibility: Joseff Betancourt.

Abstract

AI-assisted and agentic software development can produce a branching digital execution graph: multiple sessions, retries, candidates, evidence artifacts, dependencies, and terminal events may arise from one human-owned outcome. Conventional project-management systems intentionally compress delivery into work items, assignees, estimates, and statuses. This research note tests whether those abstractions remain sufficient not only to store agentic work, but to make it accurately understandable and governable by the accountable human. It uses a bounded, non-systematic narrative review of current work-management platforms, adaptive delivery methods, standards, empirical software-development studies, and human-factors and computer-supported cooperative-work literature. The novelty premise is evaluated adversarially against three outcomes: an existing method is sufficient; an existing method needs a minimum execution profile; or a distinct methodology is warranted. The evidence reviewed does not justify replacing Kanban or Scrum with a new methodology. It supports testing a risk-proportional AI Delivery Execution Profile that projects machine execution into human-legible outcome, attempt, candidate, evidence, authority, clock, cost, and effect records. Proposed studies compare that profile with the strongest reasonably configured conventional control using comprehension accuracy, reconstruction time, intervention quality, recording burden, and accepted-delivery outcomes. A controlled visual comparison holds one synthetic stage-and-ROM dataset constant while rendering it as responsibility bands and as a Gantt view; the study tests representation-form fit rather than presuming one view is universally superior. No original experiment or measurement is reported.

Keywords: AI-assisted software development; agentic software engineering; project management; software delivery; human-AI collaboration; Kanban; Scrum; situation awareness; distributed cognition; evidence-based acceptance; AI governance

Suggested citation: Betancourt, Joseff. *AI-Assisted Software Delivery Management: A Human-Legibility Test and Candidate Execution Profile*. Version 1.1.0, 13 Aug. 2026. Research note.

Executive Summary

This paper tests a premise rather than assuming a conclusion:

> **Premise:** A new delivery-management methodology may be needed for AI-assisted and agentic software development.

The burden of proof rests with novelty. If configured Kanban, Scrum, or another established method can represent AI-assisted delivery without excessive ambiguity or administrative contortion, inventing a new methodology would be unnecessary. The research must try to disprove the premise before proposing a replacement **[OPINION | ADM-C042]**.

The evidence reviewed for this paper does not support the claim that conventional project-management tools are obsolete or incapable. Each reviewed platform links AI-assisted execution to an existing work item, session, branch, or pull-request flow. Linear can preserve a human assignee while delegating execution to an agent. Jira and GitHub expose agent sessions. Azure Boards can display a GitHub Copilot operation as in progress, ready for review, or failed **[FACT | ADM-C002]** (Atlassian, “View and Manage Agent Sessions”; GitHub, “Managing Agent Sessions”; Linear, “Assign and Delegate”; Microsoft, “Use GitHub Copilot”).

Established methods and current standards are stronger counterevidence than a simple tool survey. The May 2025 *Kanban Guide* already requires teams to define their units of value, started and finished boundaries, workflow states, work-in-progress controls, explicit flow policies, and probabilistic service level expectations. The *Scrum Guide* describes Scrum as purposefully incomplete and able to contain other processes and techniques **[FACT | ADM-C036]** (*Kanban Guide*; Schwaber and Sutherland). PMI’s June 2026 AI standard and July 2026 *Agile Practice Guide* explicitly address AI-enabled project work, human-in-the-loop practice, governance, tailoring, AI and generative AI, flow, outcomes, and DORA metrics **[FACT | ADM-C037]** (Project Management Institute, *Standard for Artificial Intelligence*; Project Management Institute, *Agile Practice Guide*). These publications substantially raise the novelty burden.

The narrower concern is semantic. Across the reviewed platforms, the ordinary work item remains the dominant planning object, while AI execution introduces additional objects and events:

- multiple execution attempts for one requested outcome;
- parallel agents whose aggregate runtime may exceed elapsed time;
- failed, rejected, interrupted, and superseded attempts;
- immutable candidate artifacts;
- evidence that applies to one exact candidate;
- human interventions and acceptance decisions;
- requested versus observed provider and model provenance;
- separate execution, validation, acceptance, and release states; and
- costs accumulated across successful and unsuccessful attempts.

These concepts can often be approximated through custom fields, comments, linked pull requests, agent-session records, APIs, or external analytics. However, the reviewed documentation does not establish a common, portable delivery record that gives all of them stable meaning **[INFERENCE | ADM-C003]**. The possible gap is therefore not a lack of fields. It is a lack of consistent delivery semantics and a reliable projection from machine execution into a human-operable view.

That distinction is central. A digital system may treat several concurrent tasks as invocations of the same underlying model. The human operator still needs them separated into inspectable contexts with bounded scope, state, evidence, unresolved obligations, and handoff points. Those contexts are not people and do not acquire accountability. They are external coordination objects that help the accountable human perceive, understand, and act on the execution topology **[INFERENCE | ADM-C046]**.

External studies also caution against reducing AI-assisted development to generation speed. A bounded GitHub Copilot experiment reported that treatment participants completed one JavaScript task 55.8 percent faster, and pooled field experiments across 4,867 developers reported 26.08 percent more completed tasks. By contrast, a randomized study of 16 experienced maintainers completing 246 tasks in familiar mature repositories reported that allowing early-2025 AI tools increased completion time by 19 percent **[REPORTED | ADM-C004]** (Peng et al.; Cui et al.; Becker et al.). These results are not contradictory proofs about AI in general. They measure different operators, tools, tasks, repositories, and outcome boundaries.

DORA's 2025 research describes AI as an amplifier of the surrounding organizational system, and the SPACE framework rejects a single activity measure as a sufficient definition of developer productivity **[REPORTED | ADM-C005]** (DeBellis et al.; Forsgren et al.). The practical implication is that a model completion, code diff, commit count, or pull request cannot by itself establish an accepted engineering outcome.

This paper reaches a **provisional, not experimentally validated, disposition**:

> **The replacement-methodology interpretation is not supported. The narrower extension premise remains plausible.** Established project-management methods should remain the governing layer, while AI-assisted delivery may benefit from an explicit execution profile for attempts, authority, isolation, evidence, acceptance, provenance, concurrent clocks, WIP, and cost. The current evidence provisionally selects Outcome 2, but a strong configured control may still reduce the result to Outcome 1 **[INFERENCE | ADM-C006]**.

The operational proposal is therefore a candidate **AI Delivery Execution Profile**, not a replacement project-management method:

1. Keep the conventional **work item** as the demand, planning, and portfolio object.
2. Link material AI-assisted work to an **AI Delivery Record (AIDR)** as the evidence-bearing execution object.
3. Define the managed unit as an **accepted delivery outcome**, not an agent run or completion message.
4. Preserve human authority separately from AI execution.
5. Represent delivery as a state vector rather than one overloaded status.
6. Record all material attempts and their lineage.
7. Bind validation evidence to an immutable candidate.

8. Keep human-active time, machine-active time, waiting, validation, and elapsed schedule separate.
9. Calculate cost per accepted outcome across accepted and nonaccepted attempts.
10. Control new execution against downstream verification and integration capacity rather than agent concurrency alone.

The profile and its AIDR are candidate interventions, not validated standards. A registered comparison must allow both to lose. If a reasonably configured conventional workflow performs equivalently or better after accounting for comprehension accuracy, reconstruction time, intervention quality, forecast error, quality, authority compliance, delivery time, and recording burden, the extension premise should be rejected for the tested setting **[HYPOTHESIS | ADM-C007]**.

1. Research Mandate

1.1 Premise

A new delivery-management methodology may be needed because AI changes the relationship among execution, effort, time, ownership, evidence, acceptance, and the human operator's ability to maintain an accurate view of the work **[HYPOTHESIS | ADM-C001]**.

“May be needed” is deliberate. It is not a claim that Agile has failed, that Kanban cannot represent agentic work, or that current platforms must be replaced.

The research separates two nested propositions:

Proposition	Test
`P-extension`	A portable AI execution profile materially improves a strong configured existing method
`P-replacement`	A distinct methodology materially improves the configured existing method plus the profile

Outcome 1 requires affirmative equivalence or non-inferiority evidence against `P-extension`; Outcome 2 requires support for `P-extension` without support for `P-replacement`; Outcome 3 requires support for both. An inconclusive study produces `UNKNOWN`, not a favorable conclusion by default.

1.2 Burden

The research must attempt to prove or disprove the premise. “New methodology” carries the higher burden. A new method is warranted only if reasonable configuration or bounded extension of an established method performs materially worse or becomes operationally incoherent.

The comparison must not use an artificially weak control. “Traditional project management” means the best preregistered conventional implementation available to the participating team, with equal training and reasonable configuration.

1.3 Required deliverable

Regardless of the result, the research must produce an operational model for AI-assisted software delivery. A paper that concludes only that existing methods “feel wrong,” or that they are “probably sufficient,” does not satisfy the mandate.

1.4 Three legitimate outcomes

Outcome	Research conclusion	Required operational result
1. Premise disproved	A reasonably configured traditional method is sufficient	Specify the method, work unit, estimation rules, agent representation, human gates, evidence, and metrics
2. Premise partially supported	Traditional project management remains the governing layer, but some abstractions require extension	Define the minimum extensions and show how they map into existing tools
3. Premise supported	Existing methods fail materially or require excessive contortion	Propose a new method with explicit primitives, lifecycle, roles, operating procedure, and validation evidence

The current paper provisionally selects Outcome 2 because established methods and tools already provide the governing primitives, while the residual AI-execution semantics are not yet consistently normalized **[INFERENCE | ADM-C006]**. This is close to Outcome 1 at the framework level. The disposition must change if a strong configured control performs equivalently, or if a genuinely different lifecycle proves necessary.

2. Problem

Conventional software-delivery systems intentionally compress complexity. A work item usually answers a small set of questions:

- What needs to be done?
- Who owns it?
- What is its priority?
- What state is it in?
- How large is it?
- What source changes, reviews, or releases relate to it?

That compression makes work searchable, assignable, and reportable. It is a feature, not a defect.

AI-assisted execution adds a second layer of activity. One human owner may delegate to one or several agents. A single request may produce multiple candidate implementations. One agent may finish while another remains active. A candidate may pass unit tests and fail an acceptance review. A later correction may invalidate earlier evidence. The implementation may be accepted but not released. A low-cost run may become expensive after retries and human correction.

At the runtime level, this activity forms a digital execution graph. Nodes may be tasks, sessions, attempts, candidates, checks, decisions, or effects. Edges may represent delegation, dependency, retry, correction, supersession, validation, or handoff. Several nodes may use the same underlying model and still require separate operational identities because the human must inspect, pause, resume, compare, or close them independently.

Project management therefore has a cognitive as well as a transactional function. It externalizes enough of the work for people to coordinate without reconstructing the entire runtime from logs. A system can be technically capable of storing every event and still fail if the accountable human cannot quickly determine:

- what is running, blocked, complete, or abandoned;
- which bounded objective and authority apply to each execution context;
- what changed and which evidence remains valid;
- which obligations have not transferred at handoff;
- where intervention is required; and
- what may safely be accepted or released.

The paper calls the required projection **human-legible execution topology**. The phrase “inspectable desks” is a useful metaphor: each desk is a bounded execution context with visible work and obligations, not a fictional person. Human accountability remains with the named authority **[OPINION | ADM-C047]**.

When all of those conditions are reduced to one assignee, estimate, and status, the record can become ambiguous:

- Does “assigned to AI” transfer accountability, or only execution?
- Does “finished” mean the agent stopped, a candidate exists, tests passed, the outcome was accepted, or the change was released?
- Does an estimate describe human effort, aggregate machine runtime, elapsed schedule, or cost?
- Is a rejected attempt part of the record, or is only the successful path retained?
- Was evidence produced against the exact artifact being accepted?
- Did the requested provider and model actually serve the run?
- Which person had authority to accept residual risk or trigger an external effect?

The central research problem is:

> Can a conventional project-management method, reasonably configured, retain and project enough information for an accountable human to understand, govern, reconstruct, and safely accept AI-assisted delivery without unacceptable ambiguity or burden; and if not, what is the smallest sufficient extension or replacement?

3. Purpose and Scope

This paper is intended for AI practitioners, engineering leaders, project managers, researchers, and technically informed stakeholders.

It addresses:

- software delivery involving AI assistance or bounded agent delegation;
- work-item semantics;
- the translation of digital execution graphs into human-legible control views;
- operator situation awareness, reconstruction burden, and intervention decisions;
- human and AI responsibility;
- execution-attempt lineage;
- evidence and acceptance;
- effort, time, and cost accounting;
- mapping into current work-management platforms; and
- a validation protocol for deciding whether the extension is useful.

It does not:

- claim that AI always improves or reduces productivity;
- define a universal model or provider ranking;
- treat configured runtime identity as observed fact;
- replace legal, regulatory, security, or organizational authority;
- prescribe one software platform;
- validate the proposed AIDR through original data; or
- claim that every AI-assisted task needs a full execution record.

4. Methodology

4.1 Study type

This is a research note. Its review method is a bounded, non-systematic narrative review. Its research approach is a design-science proposal with adversarial premise evaluation. It is not a systematic review, scoping review, meta-study, or meta-analysis. It does not pool effect sizes.

The note combines:

1. a bounded review of official platform documentation;
2. a bounded review of empirical and governance literature;
3. adversarial evaluation of the novelty premise;
4. a candidate delivery representation;
5. falsifiable hypotheses; and
6. a proposed validation study.

No original benchmark, controlled trial, or internally reproduced experiment was conducted for this paper **[FACT | ADM-C008]**.

4.2 Author contribution and design provenance

This paper began with author-directed design artifacts rather than with a request to summarize a literature. Joseff Betancourt directed the creation of an AI-Assisted Engineering Delivery Model and supplied a real-project Gantt as a comparison reference. He identified that separating work sequence, responsibility bands, engineering effort, external dependencies, and human acceptance felt more truthful for orchestrated AI delivery than beginning with a calendar-first view.

That response is not empirical validation. It is the motivating author observation from which the research problem was derived **[OPINION | ADM-C050]**. Neither supplied real artifact is reproduced in this paper. Figures 1A and 1B instead render the same synthetic stage-and-ROM dataset so that differences in underlying project content do not confound the proposed comparison. The paper turns the observation into testable design questions:

1. Which distinctions does the figure preserve that an ordinary schedule view may visually compress?
2. Are those distinctions necessary for accurate human understanding and control of agentic work?
3. Can a reasonably configured existing method express them with acceptable burden?
4. Does the candidate profile improve decisions and delivery, or merely add documentation?

The original contribution proposed here is therefore not the external facts summarized in the review. It is:

- the digital-to-human legibility problem;
- the human-legible execution-topology construct;
- the distinction between accountable people and inspectable execution contexts;
- the responsibility-band interpretation of AI-assisted delivery;
- the candidate AI Delivery Execution Profile and AIDR; and
- the falsifiable comparison that allows the design to lose.

This structure follows a design-science sequence of problem identification, solution objectives, artifact design, demonstration, evaluation, and communication **[REPORTED | ADM-C051]** (Peffers et al.). This paper completes the problem, objectives, and candidate-artifact stages. Demonstration and evaluation remain future work, so it is a design-science **proposal**, not a validated design-science result.

Research direction, problem framing, methodology, design judgment, and final publication responsibility remain Joseff Betancourt's. Evidence discovery, analysis, adversarial review, drafting, and document production are AI-assisted, as disclosed at the beginning and end of the paper.

4.3 Source-selection rules

Sources were selected in the following order:

1. official product documentation for current platform behavior;
2. publisher or institutional pages for research metadata;
3. original papers or standards for empirical and governance claims; and
4. the author’s prior published methodology for internal conceptual continuity.

Vendor documentation is used to establish what a product documents, not to prove independent effectiveness. A research paper is cited only for the setting and result it actually studied.

4.4 Citation-verification rule

Each internet source included in the Works Cited was opened from its first-party, institutional, publisher, or primary-paper location during manuscript preparation. The title, named authors or organization, publication information when available, URL, and cited proposition were checked as of 13 August 2026.

This is a source-integrity check. It is not an independent reproduction of the reported result. If a source cannot be verified, the claim must be removed or labeled `UNKNOWN`, not preserved because the citation appears plausible.

4.5 Evidence taxonomy

Each material claim uses one primary epistemic status:

Label	Meaning in this document
`FACT`	A directly verifiable property of a document, specification, artifact, or defined method
`MEASURED`	A raw or directly derived result recorded by this project under a documented procedure
`REPORTED`	A result reported by an external source and not reproduced here
`EXPERIMENT`	A conclusion reproduced internally under a registered protocol but not externally validated
`INFERENCE`	A conclusion derived from identified evidence and stated assumptions
`HYPOTHESIS`	A falsifiable proposition proposed for testing
`OPINION`	A normative design choice or recommendation
`UNKNOWN`	Evidence is absent, inaccessible, contradictory, or inconclusive

This paper contains no `MEASURED` or `EXPERIMENT` claims.

4.6 Claim format

Claims use `[STATUS | ADM-C###]`. MLA 9 parenthetical citations identify the source. Epistemic status and source citation answer different questions:

- the status states what kind of knowledge claim is being made;
- the citation states where supporting information can be inspected.

4.7 Adversarial rule

The strongest reasonable conventional control must be tested before a new method is accepted. Evidence that an existing tool can be customized is evidence against a strong replacement claim. Evidence that one product already implements a proposed distinction is a counterexample, not an inconvenience.

4.8 Operational definitions

These terms must be bound before data collection:

Term	Working definition
Material attempt	An execution whose cost, candidate, evidence, intervention, failure, or effect could change a named delivery decision
Material intervention	A human action that changes scope, authority, runtime, candidate direction, evidence interpretation, or acceptance
Material candidate change	A change capable of invalidating at least one required evidence item or acceptance conclusion
Reasonable configuration	A current, supported implementation selected by qualified users, given equal training and configured without intentionally withholding normal fields, policies, automation, or gates
Operational contortion	Repeated shadow records, manual reconciliation, semantic exceptions, or tool workarounds that breach a preregistered burden, accuracy, or timeliness threshold
Digital execution graph	The tasks, sessions, attempts, candidates, evidence, dependencies, decisions, effects, and relationships produced while pursuing an outcome
Human-legible execution topology	A bounded projection that lets the intended human identify scope, authority, active and terminal work, dependencies, evidence, unresolved obligations, and next decisions within preregistered accuracy, time, and workload limits
Inspectable execution context	A separately viewable and governable task or attempt boundary; a coordination object, not a person or transfer of accountability
Reconstruction burden	Human time and effort required to derive a correct current and historical delivery state from retained records
Representational truthfulness	The degree to which a view keeps materially different quantities and states distinct, exposes exclusions and uncertainty, and avoids implying precision not supported by the underlying record

5. Observations from Current Tools

5.1 Conventional work-item semantics remain useful

Jira's documented default model uses work-item fields such as status, priority, and resolution. Its statuses include in progress, in review, approved, done, resolved, and closed, and administrators can modify or create statuses **[FACT | ADM-C009]** (Atlassian, "What Are Work Item Statuses").

Linear assigns an issue to one person, uses estimates for issue complexity or size, and measures cycle time from issue start to completion **[FACT | ADM-C010]** (Linear, "Assign and Delegate"; Linear, "Estimates"; Linear, "Insights").

Azure Boards defines work items with fields, workflow states, estimates, acceptance criteria, and links to development artifacts. Its Agile workflow moves a story to resolved after implementation tasks and unit tests, then to closed when the product owner agrees that acceptance criteria and tests pass **[FACT | ADM-C011]** (Microsoft, "Agile Workflow").

GitHub issues, pull requests, projects, reviews, checks, and rulesets provide a combined planning and code-acceptance surface. Assigning an issue to GitHub Copilot starts a session; when the coding task finishes, Copilot opens a pull request for human review and iteration **[FACT | ADM-C012]** (GitHub, "Get Started with Copilot Agents").

These are not obsolete primitives. An AI delivery method should preserve them unless evidence shows they cause a material failure.

5.2 Existing platforms are configurable

Jira allows administrators to modify or create statuses. Azure Boards supports inherited-process custom fields that can feed queries, charts, analytics views, and Power BI **[FACT | ADM-C013]** (Atlassian, “What Are Work Item Statuses”; Microsoft, “Add and Manage Fields”).

Therefore, “the tools have no field for this” is not a sufficient argument for a new methodology. Fields can be added. The research question is whether teams can give them stable meaning, capture them reliably, and use them without excessive overhead.

5.3 Agent execution is becoming first-class

Current products already contradict the broad claim that project-management systems cannot represent agents:

- Linear permits a human teammate to remain the assignee and owner while an agent is the delegate **[FACT | ADM-C014]** (Linear, “Assign and Delegate”).
- Linear’s `AgentSession` exposes pending, active, error, awaiting-input, complete, and stale states **[FACT | ADM-C015]** (Linear, “Agent Interaction”).
- Jira exposes agent sessions as needing input, working, or finished; a finished session is ready for review **[FACT | ADM-C016]** (Atlassian, “View and Manage Agent Sessions”).
- Azure Boards can show a Copilot operation as in progress, ready for review, or error while linking the resulting branch and draft pull request **[FACT | ADM-C017]** (Microsoft, “Use GitHub Copilot”).
- GitHub exposes session logs, token usage, session duration, files read, changes, and validation activity **[FACT | ADM-C018]** (GitHub, “Managing Agent Sessions”; GitHub, “Get Started with Copilot Agents”).

These are material counterexamples. They weaken the case for an entirely new project-management system.

5.4 Human review boundaries also exist

The reviewed products do not uniformly equate agent completion with automatic merge. GitHub’s documented flow returns a pull request for review. Jira’s coding agent can create a draft pull request but does not merge it; the team reviews, approves, and merges. Azure’s Copilot integration similarly produces a draft pull request and directs review to GitHub **[FACT | ADM-C019]** (Atlassian, “Generate Code”; GitHub, “Get Started with Copilot Agents”; Microsoft, “Use GitHub Copilot”).

This is evidence against the claim that human gates are absent. The residual question is whether those gates preserve sufficient candidate-specific evidence and authority for the organization’s risk.

5.5 Residual representational gaps

Across the reviewed first-party documentation, no common work-item schema was identified that normalizes all of the following:

- typed retry, correction, fork, replay, and supersession lineage;

- queue, human-active, agent-active, tool-active, validation, wait, and elapsed clocks;
- requested route separately from observed provider and model;
- evidence validity bound to an immutable candidate;
- cost accumulated across accepted and nonaccepted attempts;
- acceptance authority independently from execution identity; and
- release or external effect separately from outcome acceptance.

This is a bounded negative finding: “not identified in the reviewed documentation,” not proof that no private API, extension, marketplace product, or organization-specific configuration can represent the concept.

The defensible inference is that the gap concerns **portable first-class semantics**, not absolute technical capability **[INFERENCE | ADM-C003]**.

6. Observations from Research

6.1 AI productivity effects are heterogeneous

Peng and colleagues randomized 95 professional programmers working on a bounded JavaScript HTTP-server task. The treatment group with GitHub Copilot completed the task 55.8 percent faster. The authors explicitly caution that their standardized task does not represent collaboration on large proprietary or open-source projects and does not resolve code-quality effects **[REPORTED | ADM-C004]** (Peng et al.).

Cui and colleagues report three randomized field experiments at Microsoft, Accenture, and an anonymous Fortune 100 company. The pooled analysis covered 4,867 developers and reported a 26.08 percent increase in completed tasks, with substantial noise across the individual experiments **[REPORTED | ADM-C004]** (Cui et al.).

Becker and colleagues randomized 246 real tasks completed by 16 experienced open-source developers in mature repositories they knew well. Allowing early-2025 AI tools increased completion time by 19 percent, even though participants expected or perceived a speedup **[REPORTED | ADM-C004]** (Becker et al.).

These studies do not supply a universal coefficient. They demonstrate why the task, operator, repository, tool, outcome boundary, and measurement protocol must remain attached to any result.

6.2 The surrounding system matters

DORA’s 2025 report characterizes AI as an amplifier of organizational strengths and weaknesses and argues that the greatest returns depend on the underlying organizational system **[REPORTED | ADM-C005]** (DeBellis et al.).

The inference for project management is that faster candidate production can move the constraint rather than remove it. Review, integration, security, deployment, stakeholder decisions, or acceptance may become the limiting stage **[INFERENCE | ADM-C020]**.

6.3 Generated output attenuates before accepted delivery

In an ICSE-SEIP industrial conference paper on Atlassian's HULA workflow, 663 Jira issues entered the online evaluation. The pipeline produced plans for 527, received plan approval for 433, generated code for 376, raised pull requests for 95, and merged 56. The study did not use a control group, so these transitions do not establish the causal effect of HULA. They do show material stage-by-stage filtering between requested work, generated artifacts, and merged changes in that deployment **[REPORTED | ADM-C041]** (Takerngsaksiri et al.).

A May 2026 NBER working paper used a matched event-study design with telemetry from more than 100,000 GitHub developers. For autonomous agents, the authors reported a cumulative effect of approximately 180 percent on commits, 50 percent on projects, and 30 percent on releases. This is observational working paper evidence, not a randomized estimate or a universal conversion rate **[REPORTED | ADM-C041]** (Demirer et al.).

The bounded inference is that generated output should be treated as intermediate inventory rather than accepted or released value **[INFERENCE | ADM-C044]**. The sources do not form one common conversion funnel, and neither proves that human review is always the binding constraint.

6.4 Activity is not the whole of productivity

The SPACE framework treats developer productivity as multidimensional across satisfaction and well-being, performance, activity, communication and collaboration, and efficiency and flow. It explicitly rejects a single activity measure as a complete account **[REPORTED | ADM-C005]** (Forsgren et al.).

Agent runs, tokens, commits, lines changed, and pull requests are therefore activity measures. They may be useful inputs, but none alone establishes quality, value, safety, or accepted delivery.

6.5 Governance requires explicit roles and evidence

The NIST AI Risk Management Framework organizes lifecycle risk work around Govern, Map, Measure, and Manage. It calls for documented human-AI roles, defined oversight, deployment-representative evaluation, documented limitations, and independent review where appropriate **[FACT | ADM-C021]** (Tabassi).

This framework does not validate the AIDR. It supports the underlying discipline that roles, evidence, limitations, and acceptance should be explicit rather than inferred from an AI system's confidence.

6.6 Delivery artifacts participate in human cognition and coordination

Human-computer interaction and collaborative-work research provides a stronger foundation for the author's "digital-to-human" concern than tool capability alone.

Hollan, Hutchins, and Kirsh describe distributed cognition as an approach that examines coordination across people, technologies, and material representations, rather than treating cognition as confined to one person **[REPORTED | ADM-C045]** (Hollan et al.). Under that lens, a project board, execution record, handoff, or evidence bundle is not merely administrative storage. It participates in how work is remembered, coordinated, and decided.

Dourish and Bellotti's shared-workspace study argues that awareness of individual and group activity is critical to collaboration and reports that awareness supplied through the shared workspace can support dynamic coordination **[REPORTED | ADM-C045]** (Dourish and Bellotti). Their study concerns human collaborative editing, not AI software agents, but it supports examining whether execution state should be visible in the work surface rather than recoverable only through separate logs.

Endsley's situation-awareness model distinguishes perception of relevant elements, comprehension of their meaning, and projection of their future status. It also identifies attention, working memory, system complexity, and automation as factors affecting situation awareness **[REPORTED | ADM-C045]** (Endsley, “Toward a Theory”). In a separate navigation-task experiment, Endsley and Kiris found that lower situation awareness under automation was associated with slower decisions after automation failure **[REPORTED | ADM-C045]** (Endsley and Kiris). Those results should not be transplanted as an effect size for software delivery. They establish a plausible human-factors risk to test.

Amershi and colleagues synthesized 18 human-AI interaction guidelines and evaluated them through multiple rounds that included 49 design practitioners examining 20 AI-infused products **[REPORTED | ADM-C045]** (Amershi et al.). The study supports treating status communication, correction, control, and failure handling as interface-design concerns, but it does not validate the specific AIDR or responsibility-band view.

The bounded inference is that **storage sufficiency is not the same as legibility sufficiency**. A delivery method should be tested on whether the accountable human can form an accurate operational picture and make correct decisions with acceptable effort, not only on whether every fact exists somewhere in the system **[INFERENCE | ADM-C046]**.

6.7 Decomposition and calibration remain relevant

The U.S. Government Accountability Office’s cost-estimating guide emphasizes a technical baseline, work breakdown structure, assumptions, data collection, sensitivity and risk analysis, documentation, and updates using actual costs **[FACT | ADM-C022]** (U.S. Government Accountability Office).

The prior *AI-Assisted Estimation Methodology* similarly separates baseline effort, operator-active effort, AI execution, schedule, and cost, while treating its planning coefficients as locally calibrated inputs rather than universal findings **[FACT | ADM-C023]** (Betancourt).

6.8 Existing methods and 2026 standards raise the novelty burden

The current *Kanban Guide* defines Kanban as a strategy for optimizing the flow of value through a process. Its Definition of Workflow already accommodates a locally chosen unit of value, multiple started and finished points, explicit states and flow policies, WIP control, probabilistic service level expectations, and context-specific measures. It also states that Kanban can augment other delivery techniques **[FACT | ADM-C036]** (*Kanban Guide*).

The *Scrum Guide* defines Scrum as a lightweight, purposefully incomplete framework. It supplies human accountabilities, goals, inspection and adaptation, an Increment, and a Definition of Done, while allowing other processes and techniques inside the framework **[FACT | ADM-C036]** (Schwaber and Sutherland). Under the proposed mapping, AI agents are delegated execution resources; human Scrum accountabilities remain with people **[INFERENCE | ADM-C043]**.

PMI’s June 2026 *Standard for Artificial Intelligence in Portfolio, Program and Project Management* addresses AI used in project work and AI-driven initiatives, including human-in-the-loop review, governance, risk, lifecycle, and tailoring across predictive, adaptive, and hybrid environments. PMI’s July 2026 *Agile Practice Guide*, second edition, advertises framework-neutral coverage of AI-enabled delivery, AI and generative AI, outcomes, flow metrics, DORA, and fit-for-purpose lifecycle selection **[FACT |**

ADM-C037] (Project Management Institute, *Standard for Artificial Intelligence*; Project Management Institute, *Agile Practice Guide*).

This review verified PMI's official public descriptions, not the full 275-page standard or 182-page guide. Detailed comparison against those full texts is required before any publication-level novelty claim **[FACT | ADM-C038]**.

Together, this prior art defeats a broad claim that adaptive delivery methods cannot express AI-assisted work. The remaining question is narrower: does an explicit AI execution profile improve operation or evidence enough to justify its added burden?

6.9 Direct evidence gap

No comparative study was identified in this bounded review that tests a reasonably configured conventional work-item workflow against a linked delivery record containing a state vector, full attempt lineage, candidate-bound evidence, multiple clocks, and explicit human acceptance **[UNKNOWN | ADM-C024]**.

That gap is the reason to test the proposal. It is not evidence that the proposal works.

7. Evaluation of the Premise

7.1 Strongest case against a new methodology

A strong conventional approach can already do much of the work:

1. Use Kanban, or Scrum with Kanban, for demand, priority, WIP, flow, and cadence.
2. Keep a human accountable owner.
3. Treat the agent as a delegate or implementation resource.
4. Use child issues, sessions, or pull requests for attempts.
5. Require automated checks and human review before "done."
6. Add custom fields for provider, model, evidence, and cost.
7. Use code-hosting and analytics systems for detailed telemetry.

Kanban's Definition of Workflow can encode accepted outcomes as work items, evidence gates as explicit policies or states, review queues as WIP, and local historical delivery as probabilistic service level expectations. Scrum can retain human accountabilities and a Definition of Done while incorporating those flow controls. Linear directly represents human ownership plus agent delegation. Jira, Azure Boards, and GitHub can link agent activity to work items and pull requests. All four can participate in configurable approval or review flows.

PMI's two 2026 publications further undermine a claim that the project profession has no AI-specific governance or delivery guidance. A fair control must incorporate this current prior art rather than compare the proposal with an obsolete or deliberately weak implementation.

If those conventions yield accurate decisions, reconstructable delivery history, acceptable forecasts, and low record burden, the premise is disproved. A new label would add ceremony without value.

7.2 Strongest case for extension

The conventional approach becomes less clear when:

- several attempts contribute to one candidate;
- machine-readable events exist but the human must reconstruct current state across work items, sessions, branches, logs, and review surfaces;
- several contexts use the same underlying model but still need separate scope, lifecycle, and handoff identities;
- failed attempts consume meaningful cost;
- parallel agents make aggregate runtime nonadditive with elapsed schedule;
- evidence becomes stale after candidate changes;
- agent completion is mistaken for outcome acceptance;
- one field must carry both accountable owner and active executor;
- parallel attempts collide because ownership, dependencies, or isolation are not explicit;
- execution starts exceed downstream verification or integration capacity;
- provider configuration is reported as runtime fact without attestation;
- acceptance and release require different authorities; or
- analysis needs the cost of all attempts per accepted outcome.

These conditions can be encoded in existing products, but the organization must define an explicit semantic contract. That contract is the methodological extension.

7.3 Threshold for replacement

A fully new method should be considered only if comparative evidence shows that:

- conventional methods cannot represent the required lifecycle without material decision error;
- their configured views cannot keep the execution topology human-legible within preregistered comprehension, reconstruction, and workload limits;
- configuration complexity or record burden becomes unacceptable;
- teams repeatedly confuse execution, evidence, acceptance, and release;
- tool-specific workarounds prevent portable reporting or auditability; and

• a replacement method materially improves the preregistered outcomes.

The present literature and product documentation do not meet that threshold **[INFERENCE | ADM-C006]**.

7.4 Provisional disposition

The most defensible current position is:

> AI-assisted and agentic software development does not yet justify replacing adaptive delivery methods. Test a minimal, risk-proportional AI Delivery Execution Profile layered on Kanban or Scrum with Kanban: human accountability, a human-legible projection of execution, bounded outcome slices, isolated execution, evidence-based acceptance, and WIP controlled against downstream capacity. Promote it to a distinct methodology only if evidence shows that these requirements create a different governing lifecycle rather than a better-defined workflow **[INFERENCE | ADM-C006]**.

8. Candidate AI Delivery Execution Profile

The profile is an explicit Definition of Workflow and execution-evidence contract layered on an existing project-management method. It does not replace the work-item system, portfolio method, product cadence, or accountable human roles.

8.1 The two-record model

The model separates planning from execution evidence:

Record	Primary purpose	Primary question
Work item	Demand, priority, owner, dependencies, portfolio state	What outcome is requested, by whom, and why?
AI Delivery Record	Attempts, candidates, evidence, clocks, cost, acceptance	How was the outcome attempted, evaluated, and accepted?

The work item remains the system of record for project management. The AIDR is linked only when AI involvement, risk, concurrency, cost, or evidentiary need justifies it.

8.2 Human-legible projection

The AIDR is a data contract, not a demand that humans read raw telemetry. Its human-facing projection should expose the minimum state needed to answer:

1. What accepted outcome is being pursued?
2. Which human is accountable?
3. Which bounded execution contexts exist, and what scope and authority does each have?

4. Which contexts are active, blocked, terminal, superseded, or awaiting input?
5. Which candidate is current, and what evidence applies to it?
6. What obligations remain before acceptance or an external effect?
7. Where do durable lifecycle evidence and a UI projection disagree?
8. What decision or intervention is required next?

Each task, session, or attempt can be shown as an **inspectable execution context**—the “separate desk” metaphor—without presenting it as a separate accountable person. Stable labels help the human locate and compare work; they do not establish agency, authority, or moral responsibility **[OPINION | ADM-C047]**.

The projection should use progressive disclosure. Outcome, accountability, state, current candidate, unresolved obligations, and next decision remain visible. Prompts, logs, token details, and complete provenance remain available on demand. Maximum telemetry can reduce rather than improve legibility if it obscures the decision state.

8.3 Unit of delivery

The primary unit is the **accepted delivery outcome** **[OPINION | ADM-C025]**.

An accepted outcome requires:

1. a defined objective and exclusions;
2. an immutable candidate;
3. required evidence gates passed or explicitly waived by authorized human authority;
4. an explicit acceptance decision; and
5. recorded residual risks and limitations.

An agent’s terminal response is evidence about an attempt. It is not outcome acceptance.

8.4 Delivery objects

Object	Definition
Work item	The planning and coordination object
Attempt	One bounded execution under a defined contract and base state
Candidate	An immutable artifact or artifact set proposed for evaluation
Evidence artifact	A recorded observation produced by a defined procedure against a candidate
Evidence gate	A rule that evaluates candidate-bound evidence
Acceptance decision	Authorized disposition of a candidate for the requested outcome

Object	Definition
Effect event	Preview, deployment, publication, migration, release, rollback, or other external effect
AI Delivery Record	The linked record connecting all of the above

8.5 Responsibility bands

Band	Responsibilities
Accountable Human Authority	Objective, scope, delegated authority, risk acceptance, final acceptance, release authorization, override
AI control or coordination	Bounded decomposition, routing, coordination, evidence assembly, escalation
AI execution	Investigation, implementation, tests, documentation, correction under a bounded contract
Independent validation	Deterministic checks and reviewers sufficiently independent for the risk
External dependencies	Access, vendor input, stakeholder decisions, change windows, infrastructure, compliance

Provider and model names are provenance fields, not roles or authority **[OPINION | ADM-C026]**.

8.6 State vector

One status should not carry four different decisions. The AIDR uses a state vector:

```
state:
work_state: authorized
evidence_state: partial
acceptance_state: undecided
effect_state: none
```

Recommended values:

Axis	Values
`work_state`	`proposed`, `authorized`, `executing`, `paused`, `candidate_ready`, `closed`, `cancelled`
`evidence_state`	`absent`, `partial`, `complete`, `failed`, `invalidated`
`acceptance_state`	`undecided`, `conditional`, `accepted`, `rejected`, `withdrawn`
`effect_state`	`none`, `preview`, `released`, `rolled_back`

Example:

```
work_state: candidate_ready
evidence_state: failed
acceptance_state: undecided
effect_state: none
```

This state is coherent: an attempt produced a candidate, but the evidence failed, no authorized acceptance occurred, and no external effect was taken.

8.7 Attempt lineage

Each material attempt records:

- stable attempt ID;
- parent attempt IDs;
- relationship such as root, retry, correction, fork, replay, or merge;
- bounded contract;
- base repository, document, environment, or data state;
- requested, routed-upstream, and provider-reported runtime evidence, when available;
- start, end, terminal state, and terminal reason;
- direct lifecycle-evidence reference and any conflicting parent or UI projection;
- human interventions;
- candidate and evidence outputs;
- cost and clock events; and
- limitations.

Failed, rejected, interrupted, and superseded attempts remain part of the record. Success-only retention creates survivor bias and understates rework **[INFERENCE | ADM-C027]**.

8.8 Runtime and lifecycle evidence

Runtime identity has three distinct evidentiary layers:

Field	Meaning
Requested	What the human or control plane asked to use
Routed upstream	What the routing or transport layer sent to the provider

Field	Meaning
Provider reported	What the provider response claims served the request

The provider-reported value is an observed provider claim. It is not cryptographic proof of the provider's physical implementation. If the routing or provider layer does not supply a correlated value, that field remains `Unverified`; the method must not silently copy the requested value into an observed field **[OPINION | ADM-C048]**.

An evidence adapter may improve provenance without becoming a universal execution dependency. Whether an `Unverified` runtime blocks acceptance is a preregistered authority and risk rule. For ordinary work it may be a recorded limitation; for a provider-restricted or regulated task it may fail a required gate.

Lifecycle evidence follows the same principle. A durable terminal event from the exact child task, session, or attempt is stronger evidence of that context's state than a parent-page badge or cached projection. Conflicts must be recorded and reconciled rather than resolved by guessing **[OPINION | ADM-C049]**.

8.9 Evidence gates

A minimum gate set is:

Gate	Question
Objective and authority	Is the outcome defined, and are actions authorized?
Candidate integrity	Is the exact artifact and base state identified?
Technical validation	Did required build, test, static, and integration checks run?
Risk validation	Did applicable security, privacy, compliance, or operational checks run?
Independent review	Did an appropriate reviewer evaluate the candidate and evidence?
Outcome acceptance	Did authorized human authority accept the intended outcome and residual risk?
Effect authorization	Was any release or external effect separately authorized?

Evidence is admissible only for the candidate and environment it actually evaluated. A material candidate change invalidates affected evidence.

8.10 Multiple clocks

At minimum, preserve:

Clock	Definition
Operator-active	Human time actively scoping, directing, reviewing, correcting, deciding, or accepting
Agent-compute	Sum of agent-run durations; may exceed elapsed time under concurrency
Validation-active	Human and deterministic validation effort, recorded without double counting where required

Clock	Definition
External-wait	Waiting for access, people, vendors, infrastructure, or change windows
Elapsed	Calendar time from authorization to acceptance or another named boundary

```
T_elapsed = accepted_at - authorized_at

T_agent_compute = Σ run_duration(j)
```

Do not derive elapsed time by adding operator, agent, validation, and wait clocks. Their intervals may overlap **[INFERENCE | ADM-C028]**.

8.11 Total cost

Two reporting modes are valid. They must not be mixed.

Currency-normalized mode

```
C_total_currency =
C_provider
+ C_local_compute
+ C_operator
+ C_validation
+ C_external
```

Every term must use the same currency and preregistered allocation rules. Operator and validation time require declared labor rates.

Disaggregated mode

```
C_vector = (
provider_currency,
local_compute_currency,
operator_minutes,
validation_minutes,
external_currency
)
```

Time-valued components remain separate and are not added to monetary components. Each human interval receives one primary allocation category so operator and validation time are not double counted.

```
CostPerAcceptedOutcome =
Σ all in-scope costs across accepted, rejected, interrupted,
failed, and superseded attempts
/ count(accepted outcomes)
```

```
RetryBurden =
cost of nonaccepted attempts
/ total delivery cost
```

`CostPerAcceptedOutcome` and `RetryBurden` are scalar only in currency-normalized mode. In disaggregated mode, report each component separately. In either mode, state inclusions, exclusions, rates, currency, allocation rules, and the observation boundary.

9. AI Delivery Record Schema

The schema is an interchange proposal, not a required storage product.

```
schema_version: "0.1"

record:
id: "AIDR-<stable-id>"
work_item:
system: "<jira|linear|azure-boards|github|other>"
id: "<work-item-id>"
url: "<link>"

outcome:
objective: "<accepted outcome>"
included: []
excluded: []
acceptance_criteria: []

authority:
accountable_human: "<identity-or-role>"
scope_authority: "<identity-or-role>"
risk_authority: "<identity-or-role>"
acceptance_authority: "<identity-or-role>"
release_authority: "<identity-or-role>"
allowed_effects: []
prohibited_effects: []

classification:
task_class: "<code|documentation|data|infrastructure|mixed>"
risk: "<low|moderate|high|regulated>"
data_classification: "<public|internal|confidential|regulated|unknown>"

execution_contract:
claimed_scope: []
dependency_refs: []
reservation_ref: "<reservation-or-lock-ref-or-null>"
workspace_isolation_ref: "<worktree-sandbox-or-environment-ref-or-null>"
data_boundary: []
retry_budget:
attempts: null
cost: null
elapsed: null
validation_plan_ref: "<plan-or-checklist-ref>"
```

```
state:
work_state: "authorized"
evidence_state: "absent"
acceptance_state: "undecided"
effect_state: "none"
transitions:
- axis: "<work|evidence|acceptance|effect>"
from: null
to: "authorized"
occurred_at: "<ISO-8601>"
actor_ref: "<identity-system-or-rule>"
reason: "<reason>"

timeline:
queued_at: null
authorized_at: "<ISO-8601>"
candidate_ready_at: null
evidence_ready_at: null
review_started_at: null
review_ended_at: null
accepted_at: null
effect_authorized_at: null
released_at: null

forecast:
made_at: "<ISO-8601-or-null>"
target_boundary: "<acceptance|integration|release|stable-production|null>"
value: null
unit: "<hours|days|null>"
confidence: null
```

```
versions:
repository_base: "<commit-or-digest>"
agent_system: "<version-or-digest-or-null>"
routing_policy: "<version-or-digest-or-null>"
router: "<version-or-digest-or-null>"
shim: "<version-or-digest-or-null>"
provider_adapter: "<version-or-digest-or-null>"
harness: "<version-or-digest-or-null>"
prompts: "<digest-or-null>"
tools: []
environment: "<digest-or-null>"

operators:
- participant_ref: "<pseudonymous-or-approved-identity>"
identity_class: "<primary_author|team_member|independent_reviewer|other>"
prior_exposure_to_task: false
prior_exposure_to_repository: true
intervention_count: 0
```

```
attempts:
- attempt_id: "ATT-001"
parent_attempt_ids: []
lineage_relation: "root"
contract_ref: "<digest-or-link>"
base_ref: "<immutable-reference>"
started_at: "<ISO-8601>"
ended_at: "<ISO-8601-or-null>"
status: "<queued|running|returned|failed|interrupted|timed_out|superseded>"
runtime:
requested:
provider: null
model: null
reasoning: null
observed:
provider: null
model: null
reasoning: null
evidence_ref: null
interventions: []
candidate_refs: []
evidence_refs: []
clock_interval_refs: []
cost_entry_refs: []
limitations: []
```

```
candidates:
- candidate_id: "CAN-001"
produced_by: ["ATT-001"]
immutable_ref: "<commit-version-or-digest>"
supersedes: null
status: "<ready|under_review|accepted|rejected|superseded>"
ready_at: "<ISO-8601-or-null>"

required_gates:
- gate_id: "GATE-001"
name: "<gate-name>"
required_for: "<acceptance|effect>"
independent_review_required: false
waiver_authority: "<identity-or-role-or-null>"
```

```
evidence:
- evidence_id: "EV-001"
gate_id: "GATE-001"
candidate_id: "CAN-001"
procedure_ref: "<command-protocol-or-checklist>"
environment_ref: "<identity-or-digest>"
result: "<pass|fail|inconclusive>"
artifact_ref: "<log-report-review-or-receipt>"
admissibility: "<admissible|inadmissible|invalidated|pending>"
observed_at: "<ISO-8601>"
reviewer_ref: "<identity-or-system>"
independent_review:
required: false
reviewer_ref: null
decided_at: null
waiver:
waived: false
authorized_by: null
reason: null
decided_at: null
limitations: []
```

```
clocks:
unit: "minutes"
overlap_policy: "<allow-cross-category-overlap-but-not-duplicate-same-actor>"
intervals:
- category: "<operator|agent_compute|validation|external_wait>"
actor_ref: "<identity-or-system>"
started_at: "<ISO-8601>"
ended_at: "<ISO-8601>"
allocation_rule: "<single-primary-category>"

costs:
reporting_mode: "<currency_normalized|disaggregated_vector>"
currency: "<ISO-4217-or-null>"
labor_rates_ref: "<preregistered-rate-table-or-null>"
entries:
- category: "<provider|local_compute|operator|validation|external>"
attempt_id: "<ATT-001-or-null>"
amount: null
currency: null
minutes: null
allocation_rule: "<rule>"
```

```
acceptance:
candidate_id: null
status: "undecided"
decided_by: null
decided_at: null
evidence_refs: []
residual_risks: []

effect:
status: "none"
target: null
authorized_by: null
authorized_at: null
executed_at: null
rollback_ref: null
```

Unknown fields remain `null` or `unknown`. A requested provider, model, or reasoning level does not prove observed runtime identity.

Minimum invariants:

- the accepted candidate ID must match a recorded immutable candidate;
- `evidence_state: complete` requires a pass or authorized waiver for every required gate;
- evidence invalidated by a material candidate or environment change cannot support acceptance;
- a released effect requires recorded release authority and authorization, unless a separately authorized emergency exception is recorded; and
- summary state must be derivable from the transition, evidence, acceptance, and effect records rather than manually contradicting them.

10. Operating Procedure

Lifecycle at a glance: define the outcome → preserve the work item → select record depth → bind authority → isolate and register attempts → identify the candidate → run evidence gates → accept or reject → authorize any external effect → reconcile clocks and cost → close at the defined boundary.

Step 1: Define the accepted outcome

State the result a human authority can accept. Record exclusions and the boundary between implementation acceptance and release.

Step 2: Retain the conventional work item

Use the existing tool for demand, owner, priority, dependencies, milestone, and portfolio reporting.

Step 3: Decide record depth

Level	Appropriate use
D0 — Conventional only	Trivial, reversible, low-risk assistance where ordinary history is sufficient
D1 — Minimal AIDR	One bounded AI attempt; record owner, candidate, evidence, acceptance
D2 — Full AIDR	Multiple attempts, material cost, concurrency, production impact, or audit need
D3 — Controlled AIDR	High-consequence, security-sensitive, regulated, destructive, or externally consequential work

These levels are proposed, not validated. D3 inherits every D2 field and adds the applicable organization-specific security, privacy, legal, compliance, change-control, separation-of-duties, and emergency-recovery controls; this paper does not claim one universal D3 control set.

Step 4: Define authority before execution

Record who owns scope, risk, acceptance, and external effects. Record actions the AI may and may not take.

Step 5: Reserve scope and isolate execution

For concurrent or consequential work, record claimed files or surfaces, dependencies, a reservation reference, and an isolated workspace or environment. A new attempt must not silently overlap another active attempt.

Step 6: Register attempts

Give each material attempt an ID, contract, base, lineage, requested runtime, retry budget, and permitted effect boundary.

Step 7: Produce an immutable candidate

Use a commit, patch plus base, document version, build digest, configuration bundle, or equivalent stable identity.

Step 8: Run candidate-bound gates

Record procedures, environments, results, skipped surfaces, artifacts, and limitations. Invalidate evidence when its candidate or material environment changes.

Step 9: Make an explicit acceptance decision

The accountable human or delegated human authority accepts, conditionally accepts, rejects, or withdraws the candidate.

Step 10: Authorize external effects separately

Release, publish, migrate, deploy, or otherwise affect an external system only under the applicable authority.

Step 11: Reconcile clocks and cost

Include nonaccepted attempts, operator intervention, validation, waiting, and external fees. Do not report only the final successful run.

Step 12: Close the work item at the defined boundary

The organization may define “done” as accepted implementation or released outcome. Whichever boundary it chooses must be explicit and consistent.

11. Mapping into Existing Tools

Platform	Conventional record	AI execution support	Minimum extension
Jira	Work item, assignee, status, estimate, worklog, development links	Agent sessions and coding-agent draft PR	AIDR link, state vector, attempt lineage, candidate/evidence IDs
Linear	Issue, human assignee, estimate, status, insights	Separate agent delegate and `AgentSession` lifecycle	AIDR link, candidate evidence, all-attempt cost, separate effect state
Azure Boards	Work item, state, effort, acceptance criteria, development links	Copilot operation status and linked draft GitHub PR	AIDR custom link or child record, attempt lineage, multi-clock and acceptance record
GitHub	Issue/Project, assignee, PR, reviews, checks, rulesets	Agent sessions, logs, token use, draft PR	Repository AIDR or attestation linked to issue/PR, authority and cost fields

Sources: first-party platform documentation cited in Sections 5.1–5.4.

The recommended implementation is not a second project-management platform. Start with a linked structured record or small set of typed fields. Automate capture only after the semantics are stable.

12. Metrics

12.1 Cost per accepted outcome

Defined in Section 8.9. Report task class, risk, provider/model provenance status, and observation window.

12.2 First-candidate acceptance rate

```
FCAR =
outcomes accepted on first candidate
/ outcomes producing at least one candidate
```

12.3 Retry and rework burden

Report nonaccepted attempt count, cost, operator time, and elapsed impact.

12.4 Evidence completeness

```
EvidenceCompleteness =
admissible required evidence items
/ required evidence items
```

Completeness does not imply success. A complete evidence bundle may establish that a candidate failed.

12.5 False-completion rate

```
FalseCompletionRate =
work items declared done before required evidence and acceptance
/ work items declared done
```

12.6 Reconstruction time

Measure the time an independent reviewer needs to identify:

- objective and exclusions;
- accountable authority;
- attempts and lineage;
- accepted candidate;

- evidence and limitations;
- total cost; and
- release status.

12.7 Operator intervention rate

```
InterventionRate =
material human interventions
/ agent attempts
```

12.8 Recording burden

```
RecordBurden =
human-active time maintaining delivery records
/ accepted outcomes
```

The execution profile fails economically if its decision value does not justify this burden.

12.9 Verification-constrained WIP

Track execution, verification, human review, and integration as distinct queues:

```
VerificationWIP =
count(candidates awaiting or undergoing required verification)

ReviewQueueAge =
(review_started_at if present, otherwise observation_time)
- evidence_ready_at
```

The candidate policy is to admit new agent execution only when downstream verification and integration remain within locally defined WIP controls and service level expectations. The stronger proposition—that human verification capacity rather than agent concurrency should govern overall AI work admission—is a hypothesis, not an established fact **[HYPOTHESIS | ADM-C039]**.

Rates with a zero denominator are reported as `not applicable`, not zero.

13. Figures 1A and 1B — Controlled Representation Comparison

The comparison holds the work packages and ROM effort ranges constant. Only the representation changes.

Stage	Synthetic work package	ROM active engineering effort
Stage 0	Planning and acceptance criteria	1.0–1.5 hours
Stage 1	Data and permission boundary	4.0–6.0 hours
Stage 2	Core implementation	10.0–15.0 hours
Stage 3	Logging and failure handling	5.0–7.0 hours
Stage 4	Integration hardening	6.0–9.0 hours
Stage 5	Automated QA and regression	5.0–8.0 hours
Stage 5A	Human acceptance and validation	8.0–12.0 hours
Stage 6	Documentation and handoff	4.0–6.0 hours
Total	**Active engineering effort**	**43.0–64.5 hours**

Shared comparison inputs:

- the traditional comparison baseline is 24–36 person-days;
- the Gantt projection is 8–12 business days;
- one accountable human retains acceptance authority;
- up to three agents may execute concurrently; and
- human review occurs in daily review windows.

The traditional baseline is a comparison input, not a measured conversion from the active-engineering-hour range. The 8–12-day projection is also not derivable from ROM effort alone; it depends on the stated concurrency, dependency, calendar, and review-window assumptions.

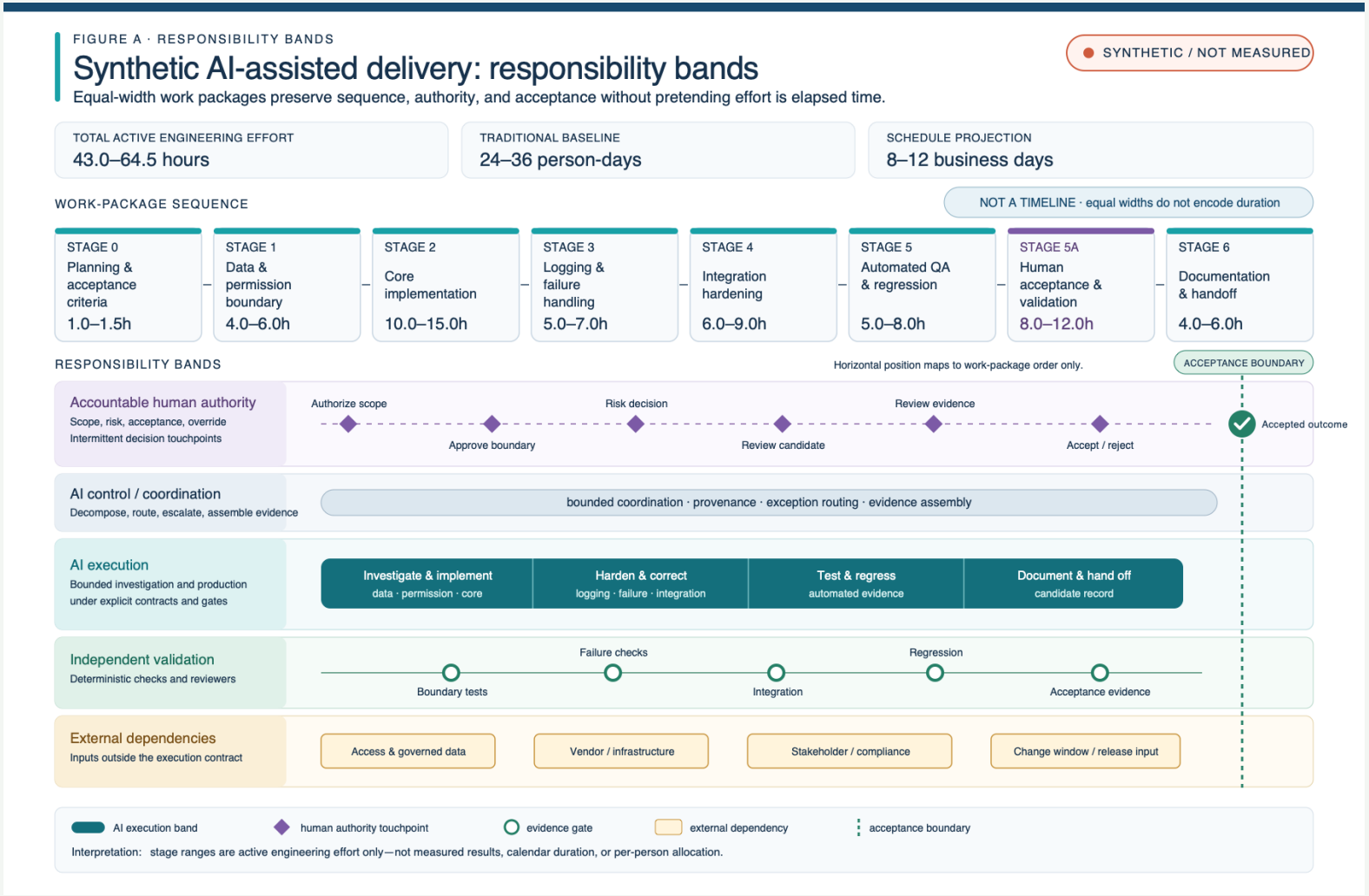

Figure 1A. Synthetic responsibility-band view. This view emphasizes ROM allocation, accountable authority, AI execution, independent validation, and external-dependency boundaries across the controlled stage dataset.

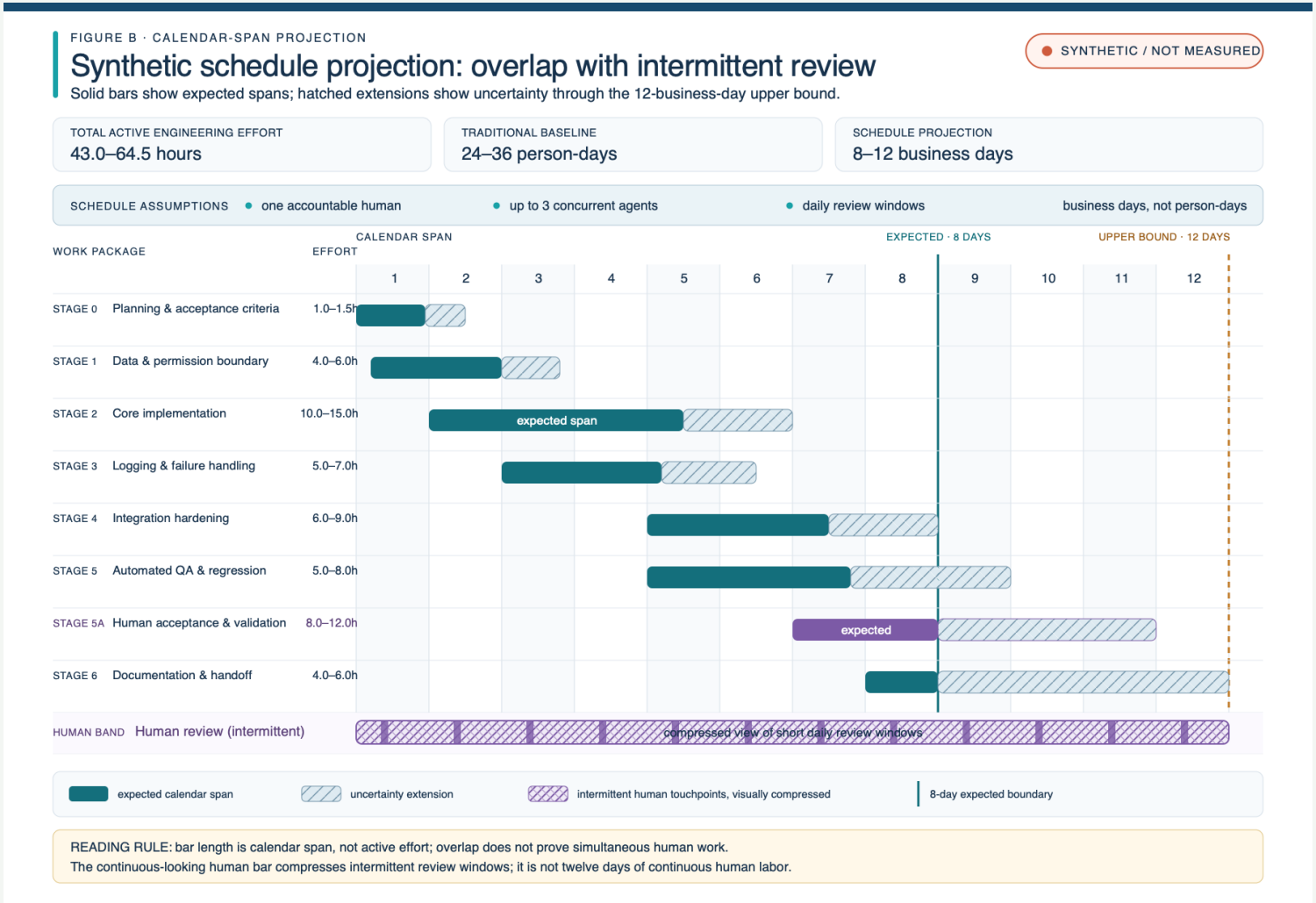

Figure 1B. Synthetic Gantt view. This view applies the stated schedule assumptions to the same stages and ROM ranges to emphasize sequence, concurrency, review windows, and projected elapsed schedule.

Interpretation rule: Neither representation is presumed universally superior. The responsibility-band view is better matched to questions about ROM allocation and authority. The Gantt view becomes useful for schedule interpretation only after dependency, concurrency, calendar, and review assumptions have been supplied. Whether either form improves interpretation is a hypothesis to test, not a result **[HYPOTHESIS | ADM-C034]**.

Both figures and every numeric input in this section are synthetic, illustrative, and not measured or validated by this paper **[FACT | ADM-C029]**. The author-supplied real Gantt is deliberately not reproduced; it served only as a design reference and is excluded from the controlled comparison.

14. Worked Example

ILLUSTRATIVE — NOT MEASURED

Work item:

> Add a CSV export to an internal administration report while preserving the report's permission filter and spreadsheet-injection protections.

Conventional record

```
id: EX-17
owner: engineering_lead
status: in_progress
estimate: 5
priority: medium
```

This record supports planning but does not show which candidate passed which checks.

AI-assisted attempts

Attempt	Result	Candidate	Disposition
ATT-001	Export generated	CAN-001	Rejected: permission filter missing
ATT-002	Permission corrected	CAN-002	Rejected: formula-prefix handling missing
ATT-003	Both defects corrected	CAN-003	Accepted after evidence gates

At the end of ATT-001:

```
work_state: candidate_ready
evidence_state: failed
acceptance_state: undecided
effect_state: none
```

At the end of ATT-003:

```
work_state: closed
evidence_state: complete
acceptance_state: accepted
effect_state: none
```

The implementation is accepted but not deployed. Release belongs to a separate change window and authority.

Illustrative clocks and cost

Quantity	Value
Operator-active time	70 minutes
Aggregate agent-compute time	54 minutes
Validation-active time	25 minutes
External waiting	24 hours
Authorization-to-acceptance elapsed time	29 hours
Provider charges	$3.80

These values do not add into elapsed time because some intervals overlap. The example retains all three attempts when computing cost and rework.

15. Assumptions

ID	Assumption	Consequence if false
A1	The organization already has a work-item system worth preserving	A separate integrated system may be simpler
A2	Material work can be expressed as an accepted outcome	Acceptance remains subjective or fragmented
A3	A human authority can own scope, risk, acceptance, and release	The responsibility model is inapplicable
A4	Candidates can be identified immutably	Evidence cannot be bound reliably
A5	Attempts can be captured without prohibited data exposure	The record requires redaction or a narrower schema
A6	Required evidence can be defined proportionally to risk	Gates become ceremonial or inconsistent
A7	Teams will preserve nonaccepted attempts	Cost and rework analysis remains biased
A8	Event timestamps are sufficiently reliable	Clock analysis becomes approximate
A9	AIDR maintenance can be partly automated	Manual burden may outweigh value
A10	Tool configuration and training can be equalized in comparison	The premise test becomes biased
A11	Verification and integration capacity can be observed well enough to set WIP controls	Admission policy becomes subjective or unstable

16. Falsifiable Hypotheses

H0a — Configured-method sufficiency

After reasonable configuration and equal training, the best preregistered existing Kanban or Scrum-with-Kanban workflow is equivalent or non-inferior to the AI Delivery Execution Profile under the endpoint hierarchy in Section 17.1 while imposing equal or lower recording burden **[HYPOTHESIS | ADM-C007]**.

Materiality and non-inferiority margins must be bound before data collection.

```
phase_1:
acceptance_accuracy_equivalence_margin: preregistered
correct_reconstruction_time_equivalence_margin: preregistered
recording_burden_noninferiority_margin: preregistered
phase_2:
accepted_outcome_lead_time_equivalence_margin: preregistered
accepted_outcome_throughput_equivalence_margin: preregistered
quality_noninferiority_margin: preregistered
stability_noninferiority_margin: preregistered
authority_compliance_noninferiority_margin: preregistered
```

Outcome 1 may be selected only when an adequately powered equivalence or non-inferiority analysis places the prespecified confidence intervals wholly within every required margin and Arm A's recording burden is no greater than Arm B's. Failure to demonstrate Arm B's superiority alone yields `UNKNOWN`; it does not disprove the premise.

H0b — Replacement non-superiority

A genuinely different methodology does not materially outperform the best configured existing method plus the minimum execution profile after accounting for the same preregistered endpoint hierarchy **[HYPOTHESIS | ADM-C040]**.

Failure to demonstrate Arm C's superiority leaves Outcome 3 unsupported; it does not prove that replacement is universally irrelevant.

H1 — Representation value

The AIDR improves reviewer accuracy in distinguishing execution completion, candidate readiness, evidence completion, outcome acceptance, and release **[HYPOTHESIS | ADM-C030]**.

H2 — Evidence-gate value

Candidate-bound evidence gates reduce false-completion declarations without violating the preregistered delivery-time and recording-burden margins **[HYPOTHESIS | ADM-C031]**.

H3 — Lineage value

Complete attempt lineage reduces independent reconstruction time and improves the accuracy of retry-cost estimates relative to success-only records **[HYPOTHESIS | ADM-C032]**.

H4 — Multiple-clock value

Separating operator-active, agent-compute, validation, waiting, and elapsed clocks reduces forecast error for recurring comparable work relative to one blended duration **[HYPOTHESIS | ADM-C033]**.

H5 — Authority clarity

For the same underlying delivery record, explicit responsibility bands improve accuracy and speed when identifying accountable human authority and separating that authority from AI execution, without exceeding the preregistered reading-burden margin **[HYPOTHESIS | ADM-C034]**.

H6 — Overhead boundary

For small, low-risk, single-attempt work, AIDR maintenance cost offsets or exceeds its decision value **[HYPOTHESIS | ADM-C035]**.

H6 is important because a method that benefits only complex work should not be imposed universally.

H7 — Verification-constrained WIP

When agent candidate-production capacity exceeds downstream review and integration capacity, WIP controls based on verified-acceptance capacity reduce queue age, rework, and total cost per accepted outcome relative to controls based only on active agent count **[HYPOTHESIS | ADM-C039]**.

H8 — Representation-form fit

When the underlying stages and ROM inputs are identical, representation performance interacts with the question being answered: a responsibility-band view improves ROM-allocation and authority interpretation, while a Gantt view improves sequence, concurrency, and projected-schedule interpretation only when its dependency, concurrency, calendar, and review-window assumptions are explicit. Neither form is expected to dominate every interpretation endpoint **[HYPOTHESIS | ADM-C034]**.

17. Validation Plan

17.1 Endpoint hierarchy

Phase	Co-primary endpoints	Safety / non-inferiority endpoints	Secondary endpoints
Phase 1 — representation	Correct acceptance classification; time to a correct reconstruction; representation-by-question-family interpretation accuracy in the paired visual sub-study	Recording and reading burden	Authority, candidate, evidence, cost, schedule, and release reconstruction accuracy; schedule-assumption recognition; confidence calibration
Phase 2 — delivery	Accepted-intent-to-stable-production lead time; accepted-outcome throughput per observation period	Quality, delivery stability, authority compliance	Human review hours, queue age, first-pass acceptance, rework, forecast error, evidence completeness, integration conflicts, recording burden, cost vector or currency-normalized cost per accepted outcome

The operational definition, observation boundary, statistical estimand, materiality or equivalence margin, missing-data policy, and multiplicity treatment for every endpoint must be preregistered. Phase 1 can establish representation and reconstruction value. It cannot establish that a workflow improves delivery.

17.2 Phase 1 — Representation feasibility

Select completed engineering slices with enough retained evidence to establish a reference record. Encode each slice in three forms:

1. ordinary conventional work-item record, used as an exploratory baseline;
2. best reasonably configured conventional record; and
3. conventional work item plus AIDR.

Do not intentionally weaken the configured control. Before reviewer scoring, an independent adjudication panel must construct the reference answers and report agreement and unresolved ambiguity.

The Figure 1 visual-form sub-study is nested within Phase 1 but does not change the record content. It compares two projections of the same controlled synthetic data, not a sparse conventional record against a richer AIDR.

17.3 Reviewer study

Use randomized, counterbalanced presentation. Reviewers should not see more than one representation of the same slice unless the study design controls carryover.

Ask reviewers to:

• decide whether the outcome is acceptable;

- identify the accountable authority;
- identify the accepted candidate;
- determine which evidence is admissible;
- reconstruct attempt lineage;
- estimate total cost and elapsed time; and
- identify whether release occurred.

Analyze the endpoints under the Phase 1 hierarchy in Section 17.1.

17.3.1 Paired, counterbalanced visual-form sub-study

Figures 1A and 1B form one paired stimulus: stage names, stage order, ROM ranges, total active engineering effort, traditional baseline, accountable human, agent-concurrency limit, review cadence, and projected schedule are held constant. Each image is presented with the same plain-text shared-input card from Section 13, so neither view gains an unstated datum needed for scoring. The views may change emphasis, but only the visual form changes.

Reviewers are assigned to counterbalanced `AB` or `BA` sequences, where `A` is the responsibility-band view and `B` is the Gantt view. Question order is randomized, no correctness feedback is shown between views, and the analysis includes representation, question family, order, and period effects. Reviewer and case are modeled as repeated or random effects as preregistered. If the study adds isomorphic cases, each case must still use an identical data payload under both forms.

Interpretation endpoints are:

Question family	Scored endpoint
ROM allocation	Correct stage effort range, total active engineering range, and allocation boundary
Authority	Correct accountable human, AI-execution boundary, validation boundary, and acceptance authority
Sequence and concurrency	Correct predecessor, permissible overlap, and agent-concurrency interpretation
Schedule	Correct 8–12-business-day projection and distinction between projected elapsed schedule, active engineering hours, and the 24–36-person-day comparison baseline
Assumption recognition	Correct identification of dependency, concurrency, calendar, and daily-review assumptions required for the schedule projection
Human performance	Time to a correct answer, confidence calibration, and perceived workload

The preregistered primary test is the `representation × question-family` interaction, not an aggregate preference score. H8 is supported only if Figure 1A improves the allocation/authority family and Figure 1B improves the sequence/schedule family within prespecified accuracy, time, and burden margins. A Gantt response that expresses schedule confidence without recognizing its schedule assumptions is scored as an interpretation error. Stated preference is exploratory. One form winning an omnibus score does not establish universal superiority.

17.4 Pilot

A pilot of approximately 12 to 20 work items may refine the instrument, definitions, and data-capture process. It must not be presented as confirmatory evidence unless its sample size and analysis were powered and preregistered for that purpose.

17.5 Confirmatory sample

Determine sample size through power analysis using pilot estimates. Bind:

- eligible task classes;
- exclusions;
- primary endpoint;
- materiality threshold;
- non-inferiority margins;
- missing-data policy;
- multiple-comparison treatment;
- stopping rule; and
- analysis method.

17.6 Phase 2 — Prospective delivery pilot

After representation feasibility, compare workflows prospectively through a cluster-randomized, stepped-wedge, or other preregistered design appropriate to the available teams:

Arm	Treatment
A	Best configured existing Kanban or Scrum-with-Kanban workflow using AI tools
B	The same governing method plus the minimum AI Delivery Execution Profile
C	A genuinely different candidate methodology, only if a coherent replacement has been specified

Arm A must receive reasonable configuration, current guidance, and equal training. Arm C must not be introduced merely to create a third label; it is eligible only when its lifecycle contains material primitives that Arm B cannot represent without documented operational contortion.

Before assignment, preregister:

- the exact fields, states, policies, gates, automation, and training allowed in Arms A and B;

- the D1, D2, or risk-based profile assignment rule used in Arm B;
- treatment-fidelity and cross-arm contamination checks;
- the portable profile semantics that distinguish B from ordinary local configuration; and
- a rule that treats a site already implementing all required B semantics as B-equivalent rather than stripping effective practice to manufacture Arm A.

Arm C and H0b remain deferred until a complete replacement lifecycle, implementation, and fidelity test exist.

Record:

```
versions:
repository_base: <commit>
agent_system: <version-or-digest>
routing_policy: <version-or-digest>
router: <version-or-digest>
shim: <version-or-digest>
provider_adapter: <version-or-digest>
harness: <version-or-digest>
prompts: <digest>
tools: [...]
environment: <digest>

operators:
- participant_ref: <pseudonymous-or-approved-identity>
identity_class: <class>
prior_exposure_to_task: <boolean>
prior_exposure_to_repository: <boolean>
intervention_count: <integer>
```

Stratify or model task class, risk, repository familiarity, operator experience, concurrency, review capacity, and tool version.

17.7 Parallelism and review-capacity test

Within comparable task and risk classes, randomize one active agent attempt versus multiple parallel attempts while holding downstream review capacity approximately fixed. Measure:

- accepted-outcome cycle time;
- execution, verification, and review queue age;
- integration conflicts;
- abandoned or superseded work;
- first-pass acceptance;
- rework and escaped defects;

- human review time; and
- total cost per accepted outcome.

This test distinguishes useful concurrency from upstream inventory growth. It also directly tests H7.

17.8 Decision rules

Result	Disposition
Adequately powered equivalence/non-inferiority analysis places every required Arm A versus B confidence interval within its prespecified margin, with no greater Arm A burden	Outcome 1: reject `P-extension` for the tested setting; `P-replacement` is not needed there
Arm B materially improves outcomes over A while C adds no meaningful benefit	Outcome 2: adopt the minimum profile for eligible work
C reproducibly outperforms B across settings without degrading quality, stability, cost, or human load, and the advantage depends on otherwise unrepresentable primitives	Outcome 3: justify development of a new methodology
Superiority is not shown, but equivalence/non-inferiority is also not established; or evidence is underpowered, contradictory, contaminated, or confounded	`UNKNOWN`: do not select an outcome; run the next registered experiment

Phase 1 alone cannot select Outcome 1, 2, or 3. Those dispositions require the Phase 2 delivery evidence appropriate to the proposition being tested.

18. Limitations

18.1 No original empirical result

This paper proposes a method and an evaluation. It does not establish that the AIDR improves cost, quality, speed, safety, or governance **[FACT | ADM-C008]**.

18.2 Documentation snapshot

Platform findings are based on first-party documentation reviewed on 13 August

2026. Product features, plans, APIs, and terminology may change.

18.3 Full PMI texts were not inspected

The official PMI product pages establish scope, publication dates, length, and advertised topics for the June 2026 AI standard and July 2026 agile guide. This review did not have the full texts. It therefore cannot establish whether specific AIDR fields or execution-profile rules are novel relative to those publications **[FACT | ADM-C038]**.

18.4 Negative findings are bounded

Failure to identify a native concept in reviewed documentation is not proof that it does not exist in another product surface, private API, extension, or custom configuration.

18.5 Fast product evolution

Linear's delegate model, Jira agent sessions, Azure's Copilot integration, and GitHub agent telemetry demonstrate that existing platforms are adapting. Evidence favoring a new method may weaken as platforms normalize these semantics.

18.6 Administrative burden

Detailed attempt, clock, and evidence records can become bureaucracy. Missing or low-quality manual data can create an illusion of auditability.

18.7 Measurement changes behavior

Teams may reduce recorded attempts, delay state transitions, or optimize metrics. Qualitative review and data-integrity checks remain necessary.

18.8 Acceptance is not correctness

Human acceptance establishes accountability, not infallibility. Review quality, independence, time, competence, and context affect the decision.

18.9 Runtime observability

Requested provider, model, reasoning, or tool settings may not match actual execution. Without authoritative evidence, observed identity remains `UNKNOWN` or `Unverified`.

18.10 Privacy and security

Prompts, logs, source, customer data, credentials, and tool output may contain protected information. AIDRs should use approved references, redaction, and digests rather than copying sensitive content indiscriminately.

18.11 External validity

Results from one team, repository, toolchain, provider mix, or risk class may not transfer. Conclusions must name the tested setting.

18.12 Synthetic representation comparison

Figures 1A and 1B are controlled synthetic stimuli, not observations of a delivery. They establish a fair comparison input, not a comparative advantage. One constructed dataset cannot validate H8, and the Gantt bars cannot validate the 8–12-business-day projection because the bars are themselves derived from the stated schedule assumptions **[FACT | ADM-C029]**.

19. Future Work

1. Obtain and compare the full 2026 PMI AI standard and *Agile Practice Guide* before making a publication-level novelty claim.
2. Complete outside review of the premise and operational schema.
3. Inventory current-tool capabilities against the proposed field dictionary.
4. Define D0 through D3 record profiles precisely.
5. Create a machine-validatable AIDR JSON Schema.
6. Preregister the representation-feasibility pilot.
7. Run the paired, counterbalanced Figure 1A/1B form-fit sub-study and report results by question family.
8. Run the pilot on real completed engineering slices.
9. Measure record burden explicitly.
10. Remove fields that do not improve a named decision.
11. Test automatic capture without allowing telemetry to overwrite authoritative human decisions.
12. Run the prospective comparison only after the record and outcome definitions stabilize.

20. Adoption Checklist

- [ ] Preserve the existing work-item system.
- [ ] Define “done” as a named delivery boundary.
- [ ] Keep a human accountable owner.
- [ ] Distinguish AI delegate or executor from owner.
- [ ] Define the accepted outcome and exclusions.
- [ ] Select D0, D1, D2, or D3 record depth.
- [ ] Reserve overlapping scope and isolate concurrent execution where needed.
- [ ] Record all material attempts, not only the successful one.
- [ ] Identify the immutable candidate.
- [ ] Bind evidence to candidate and environment.
- [ ] Keep execution, evidence, acceptance, and effect states separate.
- [ ] Separate operator-active, agent-compute, validation, wait, and elapsed clocks.
- [ ] Include nonaccepted attempts in cost.
- [ ] Record requested and observed runtime separately.
- [ ] Set separate WIP controls for execution, verification, review, and integration.
- [ ] Match the visual form to the question being answered.
- [ ] Do not interpret a Gantt as schedule evidence without explicit dependency, concurrency, calendar, and review assumptions.
- [ ] Measure recording burden.
- [ ] Preregister materiality and non-inferiority thresholds.
- [ ] Stop adding methodology if configured conventional practice is sufficient.

Appendix A — Claim Register

Each register row defines one claim ID. `Status` must match every in-text use. `Source / maturity` states how far the evidence has progressed; it is not a second epistemic status.

PDF layout note: Render this six-column register on landscape pages with the header repeated. Wrap cell text rather than shrinking the table below the document's normal note size.

Claim ID	Summary	Status	Claim / source type	Source / maturity	Validation path
ADM-C001	A new methodology may be needed because AI changes delivery relationships, but novelty carries the higher burden	HYPOTHESIS	Author research premise	Proposed; untested	Comparative study against a strong configured conventional control
ADM-C002	Current tools support ownership, workflow, review, customization, and growing agent features	FACT	Cross-platform documentation synthesis	First-party documentation snapshot, reviewed 13 Aug. 2026	Repeat documentation audit and verify representative tenants
ADM-C003	The residual gap concerns portable semantics rather than absolute tool capability	INFERENCE	Cross-platform synthesis	Bounded documentation review; portability not tested	Cross-platform schema and tenant study
ADM-C004	Published AI-development productivity effects are heterogeneous across studied settings	REPORTED	External empirical-study synthesis	Three cited studies; not pooled or reproduced	Review designs and reproduce on registered local task classes
ADM-C005	Organizational systems and multidimensional productivity affect how AI-development outcomes should be interpreted	REPORTED	External report and framework synthesis	DORA and SPACE; not reproduced	Prospective multidimensional measurement
ADM-C006	Current evidence does not support replacement and provisionally supports testing a minimum execution profile	INFERENCE	Adversarial premise synthesis	Provisional narrative-review disposition	Registered Arms A/B comparison, with Arm C only when eligible
ADM-C007	Configured-method sufficiency is the primary null	HYPOTHESIS	Preregistered comparison proposition	Proposed; untested	Adequately powered equivalence or non-inferiority study
ADM-C008	This paper contains no original experiment or measurement	FACT	Document property	Direct manuscript inspection	Reinspect empirical disclosures before publication
ADM-C009	Jira uses configurable work-item workflow fields and statuses	FACT	Product-documentation claim	Atlassian documentation snapshot	Documentation refresh and tenant verification
ADM-C010	Linear uses single-person ownership, issue estimates, and cycle-time reporting	FACT	Product-documentation claim	Linear documentation snapshot	Documentation refresh and tenant verification
ADM-C011	Azure Boards separates resolved implementation from product-owner closure	FACT	Product-documentation claim	Microsoft documentation snapshot	Documentation refresh and tenant verification
ADM-C012	GitHub Copilot issue delegation produces a pull request for human review	FACT	Product-documentation claim	GitHub documentation snapshot	Documentation refresh and live workflow verification
ADM-C013	Existing platforms support substantial customization	FACT	Cross-platform documentation claim	Documented capability; configured tenants not tested	Platform configuration and representative-tenant study
ADM-C014	Linear preserves a human assignee while delegating execution to an agent	FACT	Product-documentation claim	Linear documentation snapshot	Documentation refresh and live workflow verification
ADM-C015	Linear exposes a six-state agent-session lifecycle	FACT	Product-documentation claim	Linear developer-documentation snapshot	API and UI contract verification
ADM-C016	Jira exposes needs-input, working, and finished agent-session states	FACT	Product-documentation claim	Atlassian documentation snapshot	API and UI contract verification

Claim ID	Summary	Status	Claim / source type	Source / maturity	Validation path
ADM-C017	Azure exposes Copilot in-progress, ready-for-review, and error states	FACT	Product-documentation claim	Microsoft documentation snapshot	Integration-state verification
ADM-C018	GitHub exposes agent-session progress, tokens, duration, changes, and logs	FACT	Product-documentation claim	GitHub documentation snapshot	Documentation refresh and session inspection
ADM-C019	Reviewed agent flows preserve a human review boundary before merge	FACT	Cross-platform documentation claim	First-party workflow documentation	Live workflow and policy verification
ADM-C020	Faster generation can move rather than remove a delivery-system constraint	INFERENCE	Flow-system inference	Evidence-informed; binding constraint not measured here	Prospective queue and flow study
ADM-C021	NIST calls for explicit human-AI roles, oversight, evaluation, limitations, and evidence	FACT	Governance-document claim	NIST AI RMF inspected	Trace proposed controls to current NIST guidance
ADM-C022	GAO cost guidance emphasizes decomposition, assumptions, risk, documentation, and actuals	FACT	Government-guide claim	GAO guide inspected	Independent guide-to-method trace
ADM-C023	The prior estimation method separates effort, execution, schedule, and cost	FACT	Prior-publication claim	Published methodology inspected	Publication and worked-example inspection
ADM-C024	No direct comparative AIDR study was identified in this bounded review	UNKNOWN	Bounded absence claim	Non-systematic search; nonexistence not established	Expanded systematic or scoping review
ADM-C025	Accepted outcome should be the primary delivery unit	OPINION	Normative construct choice	Candidate design rule; unvalidated	Construct-validity and outcome study
ADM-C026	Provider and model identity are provenance, not authority	OPINION	Normative governance rule	Candidate design rule; unvalidated	Governance, legal, and operational review
ADM-C027	Success-only retention understates rework and creates survivor bias	INFERENCE	Measurement-design inference	Conceptual; not quantified here	Complete-lineage versus success-only analysis
ADM-C028	Overlapping operator, agent, validation, wait, and elapsed clocks cannot be summed to derive elapsed time	INFERENCE	Interval-arithmetic inference	Definition-level; timestamp reliability untested	Timestamp, allocation, and interval validation
ADM-C029	Figures 1A and 1B and their shared inputs are synthetic and not measured; the supplied real Gantt is not reproduced	FACT	Document and artifact property	Controlled publication figures; no empirical result	Inspect table and figure provenance, then run the registered study
ADM-C030	State vectors improve completion-state accuracy	HYPOTHESIS	Representation hypothesis	Proposed; untested	Reviewer study with adjudicated reference answers
ADM-C031	Candidate-bound gates reduce false completion	HYPOTHESIS	Process-control hypothesis	Proposed; untested	Prospective comparative study
ADM-C032	Attempt lineage improves reconstruction and retry-cost accuracy	HYPOTHESIS	Traceability hypothesis	Proposed; untested	Timed reconstruction study
ADM-C033	Multiple clocks improve recurring-work forecasts	HYPOTHESIS	Forecasting hypothesis	Proposed; untested	Prospective forecast study
ADM-C034	Representation performance depends on question type: responsibility bands fit ROM-allocation and authority questions, while Gantt fits schedule questions only with explicit assumptions; neither is universally superior	HYPOTHESIS	Representation-form-fit hypothesis	Paired synthetic stimuli; untested	Paired, counterbalanced form-by-question study
ADM-C035	Full delivery records are net-negative for some small, low-risk work	HYPOTHESIS	Burden-boundary hypothesis	Proposed; untested	Burden and decision-value analysis by risk class
ADM-C036	Kanban and Scrum already provide adaptable workflow, flow, accountability, and quality primitives	FACT	Current method-guide synthesis	2025 Kanban Guide and 2020 Scrum Guide inspected	Guide refresh and implementation mapping
ADM-C037	PMI's 2026 AI standard and agile guide publicly address AI-enabled project work and delivery	FACT	Official publication-page claim	Public PMI descriptions inspected; full texts unavailable	Inspect complete publications and trace relevant requirements

Claim ID	Summary	Status	Claim / source type	Source / maturity	Validation path
ADM-C038	This review did not inspect the complete 2026 PMI publications	FACT	Review-scope property	Direct review record	Obtain and inspect the complete texts
ADM-C039	Verification-capacity-based WIP improves delivery when candidate production exceeds downstream capacity	HYPOTHESIS	Queue-control hypothesis	Proposed; untested	Controlled parallelism and review-capacity experiment
ADM-C040	A replacement methodology adds no material benefit over an existing method plus the minimum profile	HYPOTHESIS	Replacement null	Deferred until a coherent Arm C exists	Registered Arms B/C comparison
ADM-C041	In two reported settings, upstream code-production effects attenuated before merged or released outcomes	REPORTED	External empirical-study synthesis	ICSE-SEIP paper and NBER working paper; not reproduced	Review both studies and reproduce with local stage-conversion data
ADM-C042	Research must try to disprove the novelty premise before proposing a replacement	OPINION	Author methodological rule	Adopted for this paper; not an empirical result	Independent protocol review and preregistration audit
ADM-C043	Under the proposed Scrum mapping, AI agents are delegated execution resources while human accountabilities remain with people	INFERENCE	Method-to-design mapping	Scrum Guide plus conceptual mapping; untested in teams	Practitioner review and comparative implementation study
ADM-C044	Generated output should be treated as intermediate inventory rather than accepted or released value	INFERENCE	Stage-conversion inference	Bounded synthesis of ADM-C041; not a universal causal result	Prospective stage-conversion and acceptance study
ADM-C045	Distributed cognition, workspace awareness, situation awareness, and human-AI interaction research motivate testing visible delivery state and control	REPORTED	External HCI, CSCW, and human-factors synthesis	Adjacent-domain literature; no software-delivery effect reproduced	Primary-source review and software-delivery reviewer experiment
ADM-C046	Storage sufficiency is not legibility sufficiency; inspectable execution contexts can help humans understand execution topology	INFERENCE	Cross-domain synthesis	Tool review plus ADM-C045; conceptual and untested	Timed comprehension and intervention study
ADM-C047	Inspectable execution contexts are coordination objects, not persons; accountability remains with named human authority	OPINION	Normative ontology and governance rule	Candidate design rule; unvalidated	Governance, usability, and incident review
ADM-C048	Requested, routed-upstream, and provider-reported runtime evidence must remain distinct; an absent observed value remains unverified	OPINION	Normative evidence rule	Candidate provenance rule; unvalidated	Adapter trace, correlation, and assurance review
ADM-C049	Exact durable lifecycle evidence should outrank a conflicting UI projection, and conflicts require recorded reconciliation	OPINION	Normative evidence-hierarchy rule	Candidate lifecycle rule; unvalidated	Fault-injection and reconciliation study
ADM-C050	The author’s response to the original responsibility-band model and real Gantt motivated the research problem but does not validate it	OPINION	Author design observation	Motivation only; supplied real artifacts not reproduced	Paired controlled representation study
ADM-C051	Design-science research proceeds through problem, objectives, artifact design, demonstration, evaluation, and communication	REPORTED	External methodology report	Peffers et al.; this paper reaches proposal stages only	Complete demonstration and evaluation, then audit stage coverage

Appendix B — Compact Field Dictionary

Field	Required at D1	Required at D2	Meaning
Work-item link	Yes	Yes	Existing planning record
Accepted outcome	Yes	Yes	Human-acceptable result
Accountable human	Yes	Yes	Authority owner
Claimed scope and isolation	If concurrent	Yes	Ownership, reservation, dependencies, workspace
State vector	Yes	Yes	Work, evidence, acceptance, effect
Attempt ID	Yes	Yes	Bounded execution identity
Attempt lineage	No	Yes	Retry, correction, fork, replay, merge
Base identity	Yes	Yes	Repository/document/environment base
Requested runtime	No	Yes	Requested provider/model/tool configuration
Observed runtime evidence	No	Yes	Attested or recorded actual identity
Candidate identity	Yes	Yes	Immutable artifact reference
Evidence bundle	Yes	Yes	Candidate-bound results and limitations
Acceptance decision	Yes	Yes	Authorized disposition
Multiple clocks	No	Yes	Operator, agent, validation, wait, elapsed
Queue and WIP state	No	Yes	Execution, verification, review, integration inventory
All-attempt cost	No	Yes	Total attributable cost
Effect record	If applicable	If applicable	Release or external action

Appendix C — Search and Citation Verification Record

This paper uses a bounded, non-systematic review. It is not a systematic review or meta-analysis, and it does not report formal screening counts.

Item	Record
Review date	13 Aug. 2026
Product-documentation surfaces	Atlassian Support, GitHub Docs, Linear Docs and developer documentation, Microsoft Learn
Method and standards surfaces	Kanban Guides, Scrum Guides, PMI, NIST, U.S. GAO
Research surfaces	Publisher DOI pages, proceedings pages, arXiv primary records, DORA, ACM HCI and CSCW literature, human-factors literature, design-science methodology

Item	Record
Query families	AI-assisted software delivery methodology; agentic software project management; Kanban AI agents and WIP; Scrum AI-assisted development; Jira, Linear, Azure Boards, and GitHub agent sessions; AI coding productivity controlled studies; human review and AI delivery evidence
Inclusion rule	First-party product documentation, official method or standards publication, institutional report, publisher record, or primary research paper directly supporting a proposition in scope
Exclusion rule	Unsourced marketing summary, secondary paraphrase where a primary source was available, inaccessible claim without verifiable metadata, or source unrelated to software delivery management
Absence-claim treatment	A source not found in this review is `UNKNOWN`, not proof of nonexistence

For every item retained in the Works Cited, the cited title, author or institution, publication metadata when available, destination, and relevant proposition were checked against the primary or official page. Mutable documentation carries an access date. A successful citation check establishes source identity and claim fit; it does not independently validate a vendor’s effectiveness claim or a study beyond its reported design.

Works Cited

Amershi, Saleema, et al. “Guidelines for Human-AI Interaction.” *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ACM, 2019, pp. 1–13, https://doi.org/10.1145/3290605.3300233.

Atlassian. “Generate Code from a Work Item in Jira.” *Atlassian Support*, https://support.atlassian.com/jira-software-cloud/docs/generate-code-from-a-work-item-in-jira/. Accessed 13 Aug. 2026.

Atlassian. “View and Manage Agent Sessions in Jira.” *Atlassian Support*, https://support.atlassian.com/jira-software-cloud/docs/view-and-manage-agent-sessions-in-jira/. Accessed 13 Aug. 2026.

Atlassian. “What Are Work Item Statuses, Priorities, and Resolutions?” *Atlassian Support*, https://support.atlassian.com/jira-cloud-administration/docs/what-are-issue-statuses-priorities-and-resolutions/. Accessed 13 Aug. 2026.

Becker, Joel, et al. “Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity.” *arXiv*, 2025, https://doi.org/10.48550/arXiv.2507.09089. Accessed 13 Aug. 2026.

Betancourt, Joseff. *AI-Assisted Estimation Methodology*. Version 1.1.1, Fernain Research, 3 Aug. 2026, https://joseffb.com/assets/ai-assisted-estimation-methodology-v1.1.1.pdf.

Cui, Kevin Zheyuan, et al. “The Effects of Generative AI on High-Skilled Work: Evidence from Three Field Experiments with Software Developers.” *Management Science*, 27 Feb. 2026, https://doi.org/10.1287/mnsc.2025.00535.

DeBellis, Derek, et al. *State of AI-Assisted Software Development 2025*. Version 2025.2, Google Cloud, 2025, https://dora.dev/research/2025/dora-report/. Accessed 13 Aug. 2026.

Demirer, Mert, et al. “Writing Code vs. Shipping Code: Productivity Effects Across Generations of AI Coding Tools.” *NBER Working Paper Series*, no. 35275, National Bureau of Economic Research, May 2026, https://doi.org/10.3386/w35275.

Dourish, Paul, and Victoria Bellotti. “Awareness and Coordination in Shared Workspaces.” *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*, ACM, 1992, pp. 107–14, https://doi.org/10.1145/143457.143468.

Endsley, Mica R. “Toward a Theory of Situation Awareness in Dynamic Systems.” *Human Factors*, vol. 37, no. 1, 1995, pp. 32–64, https://doi.org/10.1518/001872095779049543.

Endsley, Mica R., and Esin O. Kiris. “The Out-of-the-Loop Performance Problem and Level of Control in Automation.” *Human Factors*, vol. 37, no. 2, 1995, pp. 381–94, https://doi.org/10.1518/001872095779064555.

Forsgren, Nicole, et al. “The SPACE of Developer Productivity: There’s More to It than You Think.” *ACM Queue*, vol. 19, no. 1, 2021, pp. 20–48, https://doi.org/10.1145/3454122.3454124.

GitHub. “Get Started with Copilot Agents on GitHub.” *GitHub Docs*, https://docs.github.com/en/copilot/how-tos/copilot-on-github/use-copilot-agents/overview. Accessed 13 Aug. 2026.

GitHub. “Managing Agent Sessions.” *GitHub Docs*, https://docs.github.com/en/copilot/how-tos/copilot-on-github/use-copilot-agents/manage-and-track-agents. Accessed 13 Aug. 2026.

Hollan, James, et al. “Distributed Cognition.” *ACM Transactions on Computer-Human Interaction*, vol. 7, no. 2, 2000, pp. 174–96, https://doi.org/10.1145/353485.353487.

The Kanban Guide. May 2025, *Kanban Guides*, https://kanbanguides.org/the-kanban-guide/2025.5/. Accessed 13 Aug. 2026.

Linear. “Agent Interaction.” *Linear Developers*, https://linear.app/developers/agent-interaction. Accessed 13 Aug. 2026.

Linear. “Assign and Delegate Issues.” *Linear Docs*, https://linear.app/docs/assigning-issues. Accessed 13 Aug. 2026.

Linear. “Estimates.” *Linear Docs*, https://linear.app/docs/estimates. Accessed 13 Aug. 2026.

Linear. “Insights.” *Linear Docs*, https://linear.app/docs/insights. Accessed 13 Aug. 2026.

Microsoft. “Add and Manage Fields for an Inherited Process.” *Microsoft Learn*, https://learn.microsoft.com/en-us/azure/devops/organizations/settings/work/customize-process-field?view=azure-devops. Accessed 13 Aug. 2026.

Microsoft. “Agile Workflow in Azure Boards.” *Microsoft Learn*, https://learn.microsoft.com/en-us/azure/devops/boards/work-items/guidance/agile-process-workflow?view=azure-devops. Accessed 13 Aug. 2026.

Microsoft. “Use GitHub Copilot with Azure Boards.” *Microsoft Learn*, https://learn.microsoft.com/en-us/azure/devops/boards/github/work-item-integration-github-copilot?view=azure-devops. Accessed 13 Aug. 2026.

Peng, Sida, et al. “The Impact of AI on Developer Productivity: Evidence from GitHub Copilot.” *arXiv*, 13 Feb. 2023, https://doi.org/10.48550/arXiv.2302.06590.

Peffers, Ken, et al. “A Design Science Research Methodology for Information Systems Research.” *Journal of Management Information Systems*, vol. 24, no. 3, 2007, pp. 45–77, https://doi.org/10.2753/MIS0742-1222240302.

Project Management Institute. *Agile Practice Guide*. 2nd ed., July 2026, https://www.pmi.org/standards/agile. Accessed 13 Aug. 2026.

Project Management Institute. *The Standard for Artificial Intelligence in Portfolio, Program and Project Management*. June 2026, https://www.pmi.org/standards/artificial-intelligence. Accessed 13 Aug. 2026.

Schwaber, Ken, and Jeff Sutherland. *The Scrum Guide*. Nov. 2020, https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf.

Tabassi, Elham. *Artificial Intelligence Risk Management Framework (AI RMF 1.0).* NIST AI 100-1, National Institute of Standards and Technology, 26 Jan. 2023, https://doi.org/10.6028/NIST.AI.100-1.

Takerngsaksiri, Wannita, et al. “Human-In-the-Loop Software Development Agents.” *2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, IEEE, 2025, pp. 342–52, https://doi.org/10.1109/ICSE-SEIP66354.2025.00036.

U.S. Government Accountability Office. *Cost Estimating and Assessment Guide: Best Practices for Developing and Managing Program Costs*. GAO-20-195G, 12 Mar. 2020, https://www.gao.gov/products/gao-20-195g.

Research Interpretation

The reviewed evidence supports six bounded conclusions:

1. Existing work-item tools remain useful and configurable.
2. Agent delegation and session telemetry are already entering those tools.
3. Kanban and Scrum are sufficiently adaptable to express most proposed lifecycle and flow concepts.
4. Current PMI guidance already addresses AI-enabled project work and agile delivery at a governance level.
5. Published AI-development effects vary materially by setting.
6. Delivery outcomes require more than activity or generation measures, while explicit roles, evidence, limitations, and acceptance are consistent with established governance practice.

The evidence does **not** establish that:

- conventional project management is universally insufficient;
- an AI execution profile improves delivery over a well-configured Definition of Workflow;
- the AIDR improves any outcome;
- a new branded methodology is warranted;
- every AI-assisted task needs a detailed record; or
- the proposed schema is complete.

The paper’s contribution is a falsifiable premise, an adversarial decision rule, and an operational candidate that remains useful whether the premise is disproved, partially supported, or supported.

Publication and Attribution

Publication field	Information
Version	1.1.0
Status	Publication version
Prepared	2026-08-13
Author and publisher	Joseff Betancourt
Website	joseffb.com
Research support	Fernain Research
Study type	Structured methodology note
Original measurement	None
Original experiment	None
Citation style	MLA 9
Evidence discovery and drafting	AI-assisted
Final evaluation and publication responsibility	Joseff Betancourt

Research direction and methodology: Joseff Betancourt. Evidence discovery, analysis, adversarial review, and drafting: AI-assisted. Final evaluation, editorial judgment, and publication responsibility: Joseff Betancourt.