Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2026) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2026

Reconciling Two Time Logics: An Agile–CPMAI Integration Framework for Managing AI Projects

Jeonghyun Park[a], Byungjun Kang[a,b,c], Md. Sharfaraz Ibne Syeed[d], Mark Nyarkoh[a], Young Hoon Kwak[e], Changwoo Park[a,*]

[a]*Seoul National University, Seoul, Korea*
[b]*aSSIST University, Seoul, Korea*
[c]*NFN Co., Ltd., Seoul, Korea*
[d]*UK Ministry of Defence, London, United Kingdom*
[e]*George Washington University, Washington, D.C., United States*

Abstract

Artificial intelligence (AI) projects continue to exhibit high failure rates, partly because conventional project-management practice presupposes deterministic, feature-driven work, whereas AI work is data-centric, experimental, and probabilistic. The Cognitive Project Management for AI (CPMAI) methodology was devised for AI initiatives, yet most organizations already operate Agile, and the two are often treated as competing rather than complementary. Following a Design Science Research stance, we first formalize the conflict that arises when Agile is applied to AI work as a dual-timeline problem: Agile sprints are short, fixed, feature-oriented cycles, whereas CPMAI phases are variable, data-centric stages with distinct completion semantics. We then design an integration framework in which CPMAI phases form a strategic backbone and Agile sprints provide the execution rhythm, coupled by two devices we introduce—phase gates and a two-tier Definition of Done—that we distinguish from classical stage-gate models and from conventional Scrum DoD. We provide an analytical demonstration through a worked execution trace showing how the framework dissolves each facet of the dual-timeline problem, including a gate-driven loop-back. We then specify a staged evaluation plan whose first step is an explicitly bounded LLM-simulated Delphi used to pre-test the instrument, and we discuss boundary conditions under which the framework should not be applied. The contribution is a phase-level operating model that lets Agile organizations accommodate the data-centric and experimental nature of AI projects without abandoning their cadence.

© 2026 The Authors. Published by ELSEVIER B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)

* Corresponding author. *E-mail address:* cwpak@snu.ac.kr

1877-0509 © 2025 The Authors. Published by ELSEVIER B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the CENTERIS/Proj MAN/HCist Conferences

Peer-review under responsibility of the scientific committee of the CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2026
Keywords: AI project management; Agile; CPMAI; Design Science Research; dual-timeline problem; stage-gate; large language models

1. Introduction

The adoption of artificial intelligence (AI) and machine learning (ML) has accelerated across industries, yet a substantial share of AI initiatives fails to deliver the value originally intended; inadequate project management and weak alignment with business objectives are recurrently cited among the causes [1], [2]. A structural reason underlies the pattern. Conventional project-management and software-engineering methods assume that deliverables behave deterministically with respect to a specification, whereas AI solutions are data-centric rather than software-centric: their quality is governed by data, and learning outcomes cannot be guaranteed ex ante, rendering much of the work experimental and probabilistic [2].

To close this gap, the Project Management Institute introduced CPMAI (Cognitive Project Management for AI), a data-centric, iterative methodology of six phases [1]. In practice, however, most development organizations have already institutionalized Agile. Improving AI delivery therefore calls less for wholesale replacement than for a coherent reconciliation of the Agile cadence already embedded in organizations with the AI-specific requirements CPMAI encodes. Treating the two as rivals forces a false choice; the open question is how to operate them together.

We pursue that reconciliation through a Design Science Research (DSR) lens [3], [4] and answer three questions: **RQ1**: what is the nature of the conflict when Agile is applied directly to AI projects; **RQ2**: how can the two methods be combined into a coherent operating model that is distinct from prior gating and DoD practice; and **RQ3**: how can the resulting artifact be evaluated and bounded. Our contributions are: (i) a conceptual articulation of the conflict as a dual-timeline problem; (ii) an integration framework whose novel elements—gate decoupling and a two-tier Definition of Done—are explicitly differentiated from classical stage-gate models and from conventional layered DoD practice; (iii) an analytical demonstration via a worked execution trace; and (iv) a staged, threat-aware evaluation plan together with boundary conditions. Sections 2–8 develop these in turn.

2. Background and Related Work

Agile organizes work into short, fixed-length sprints that yield a potentially shippable increment and respond rapidly to change [5]. Its effectiveness rests on an implicit premise: that work can be expressed as estimable units with a definable notion of "done." CPMAI organizes AI work into six phases—business understanding, data understanding, data preparation, model development, model evaluation, and operationalization—with trust, bias mitigation, and governance embedded throughout [1], [2]. CPMAI is best read as an AI-specific extension of CRISP-DM [6] that also absorbs Agile and DataOps practices, making it data-first and highly iterative [7].

Two adjacent bodies of work bear directly on our contribution. First, stage-gate models, long established in new-product development [8], structure innovation as stages separated by go/kill decision gates. Agile–Stage-Gate hybrids subsequently embedded time-boxed sprints within those stages, with reported performance gains in manufacturing firms [9], [10]. Second, Scrum's Definition of Done (DoD) provides completion criteria for increments, and practitioners sometimes layer a per-team and per-release DoD. Prior integration attempts also exist: one proposal departs from a shared business need and bifurcates into two concurrent iterative loops—Agile development and an Agile-enabled data method—yielding a synthesized methodology [7]; practitioner accounts likewise observe that AI work strains Agile ceremonies and propose running sprints within phase structures [11].

Our framework is deliberately positioned against these approaches; Table 1 summarizes the contrast across five dimensions. Unlike [7], [11], which remain largely conceptual or practitioner-oriented, we specify a phase-level operating model and subject it to analytical demonstration. The need for such an operating model is recognized rather than merely assumed: survey and review work reports that data-science teams lack an agreed framework reconciling iterative delivery with data-centric lifecycles [12], [13], and design-oriented Scrum adaptations for data science [14]

stop short of a phase-level cadence–gate coupling. To our knowledge, a systematic, peer-reviewed operating model at this granularity is not yet established; this paper addresses that gap.

Table 1. Positioning of the proposed framework against adjacent cadence–gate approaches. P1 (phase primacy) is adapted from [11]; the differentiating elements are P2, P3, and the two-tier Definition of Done (Section 4.3).

Dimension	Agile–Stage-Gate [9,10]	Scrabbl [7]	Mielke [11]	This framework
Gate vs. sprint clock	Synced to stage boundary	Two parallel loops	Sprints inside phases	Decoupled, asynchronous (P2)
Gate criterion	Business go / kill	Not specified	Qualitative phase checklist	Pre-registered data/model predicate
Sprint increment	Working product	Feature	Feature	Data artifact in data phases (P3)
Definition of Done	Single tier + gate	Not specified	Single tier	Two nested tiers (Section 4.3)
Gate authority & loop-back	External management gatekeepers	Not specified	Not specified	Team proposes; PO + steward ratify; ratified loop-back (Section 4.2)

The term dual-timeline also appears in practitioner discourse [11]; our contribution is not the label but its formalization—as a structural conflict between three Agile premises and three properties of AI work (Section 3) and the design response that follows (Section 4).

3. The Dual-Timeline Problem

AI projects exhibit three properties that destabilize Agile's premises. First, non-determinism: model behavior is probabilistic and outcomes cannot be guaranteed, so much of the work is genuinely experimental. Second, the dominance of data preparation: cleaning, labeling, and structuring data is typically the most time-consuming and decisive activity [15], displacing feature construction as the principal scheduling variable. Third, post-deployment continuity: models degrade under data drift and require ongoing retraining and monitoring [1], blurring any fixed notion of completion.

These properties undermine three Agile premises: estimability collapses because experiment outcomes are uncertain; the increment definition collapses because data work does not reduce to a shippable feature; and the Definition of Done collapses because model quality depends on data and varies after release. We characterize the core of this conflict as a dual-timeline problem—Agile and CPMAI operate on different time logics [11]. Agile timelines are short, fixed, feature-oriented sprints; CPMAI timelines are multi-week, data-centric phases [1] (Table 2).

Table 2. Contrasting time logics of Agile sprints and CPMAI phases.

Dimension	Agile (sprint)	CPMAI (phase)
Time unit	1–4 weeks, fixed	Multiple weeks, variable until goal met
Focus	Feature delivery	Data/model objective
Increment	Shippable feature increment	Phase-level data/model artifact
Completion	Feature works	Data-quality / model-performance threshold met
Uncertainty	Estimate, then iterate	Experiment, then decide whether to proceed

Naively juxtaposing the two timelines produces a recurring mismatch: a sprint may close while the phase objective remains unmet, or a phase may remain open while a sprint is forced to terminate. As the mismatch accumulates, teams abandon either the Agile rhythm or the phase-level rigor of CPMAI—an outcome equivalent to methodological failure. The framework in Section 4 is designed to dissolve this specific mismatch.

4. The Proposed Agile–CPMAI Integration Framework

4.1. Design principles

The framework treats the CPMAI phases as a strategic backbone and Agile sprints as the execution rhythm operating within them, adding explicit devices to absorb the time-axis mismatch. Five principles, each answering a Section 3 conflict, govern the artifact. P2 and P3, together with the two-tier DoD of Section 4.3, constitute the elements we claim as distinct from prior gating and DoD practice (Section 2).

- **P1 — Phase primacy**: CPMAI phases supply strategic objectives and structure; sprints execute within and across them [11].
- **P2 — Gate decoupling**: phase completion is judged by a separate phase-exit gate, asynchronous to sprint boundaries, resolving the timeline mismatch (the conflict at the heart of Section 3).
- **P3 — Redefined increment**: in data-centric phases the sprint increment is a data artifact (e.g., a cleaned dataset and quality report) rather than a feature.
- **P4 — Explicit uncertainty**: experiments whose success is uncertain are managed as time-boxed spikes rather than committed stories, so a failed experiment is a recorded learning, not a sprint failure.
- **P5 — Embedded trust**: ethics, bias, and compliance checks enter the sprint Definition of Done, mirroring CPMAI's per-phase governance [1].

4.2. Phase-sprint mapping and gate governance

Each phase maps to a sprint mode, a redefined increment, and a phase-exit gate (Table 3). Because CPMAI is iterative, downstream phases may loop back to earlier ones when problems emerge. Figure 1 depicts the overall structure.

Table 3. Mapping of CPMAI phases to sprint mode, redefined increment, and phase-exit gate. Spike modes in phases 2 and 4 instantiate P4.

CPMAI phase	Sprint mode	Sprint increment	Phase-exit gate
Business understanding	Discovery	Problem statement, KPIs	Goals and success criteria agreed
Data understanding	Spike (exploratory)	Data inventory and gap report	Data feasibility confirmed
Data preparation	Data-work	Cleaned dataset, quality metrics	Training-ready data threshold met
Model development	Spike (experimental)	Candidate model, experiment log	Baseline performance reached
Model evaluation	Verification	Evaluation and bias report	Business/ethical criteria passed
Operationalization	Deployment	Deployed model, monitoring	Operational stability and governance

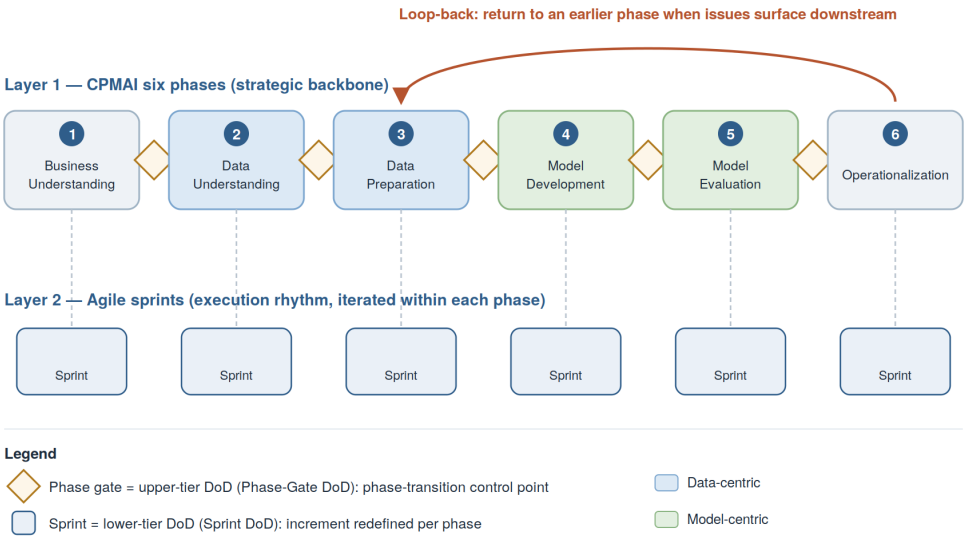


Fig. 1. The Agile–CPMAI integration framework: CPMAI phases as a strategic backbone, Agile sprints as the execution rhythm, coupled by phase gates and a two-tier Definition of Done.

Crucially, a gate is not self-evaluating. We specify its governance: each phase-exit gate is a predicate over explicit, pre-registered acceptance metrics (e.g., a maximum missing-value rate and a label-quality score for the phase-3 gate), proposed by the technical team during phase entry, ratified by the product owner together with a data steward, and recorded in the backlog as the phase's acceptance criteria. The same governance authorizes loop-back: when a downstream gate fails for reasons traceable to an upstream artifact, the product owner and data steward jointly re-open the implicated earlier phase rather than forcing closure. This makes both gate passage and regression auditable decisions rather than ad-hoc judgments. The trust checks of P5 are governed by the same apparatus rather than left as DoD line items: each is bound to a named control objective, an accountable reviewer, and a recorded evidence artifact, so that ethics, bias, and compliance criteria are ratified through the gate rather than self-attested—an elaboration prompted by the pre-test in Section 6, where a checklist-only reading of P5 was judged to risk "compliance theater."

4.3. Two-tier Definition of Done

The mismatch in completion semantics (Section 3) is addressed by a two-tier Definition of Done. The lower tier (sprint DoD) is defined per phase—for example, a data-quality threshold in phase 3 or a performance gain over baseline in phase 4. The upper tier (phase-gate DoD) specifies the transition condition for the phase as a whole; a phase advances only when the cumulative output of its sprints satisfies the gate. We do not claim the two tiers render completion wholly unambiguous—doing so presupposes fully pre-agreed quantitative predicates—but rather that they reduce ambiguity by separating sprint-level closure from phase-level transition and binding each to an explicit, pre-registered predicate; the strength of the reduction is contingent on the specificity of those predicates (a point reinforced by the pre-test in Section 6). The construct differs from a conventional layered DoD because the lower tier redefines what the increment is, not merely the granularity at which completion is asserted. Standard ceremonies are retained but adapted: the backlog admits experiment hypotheses and data tasks as first-class items; sprint planning is calibrated to the current phase [11]; reviews demonstrate data and model artifacts rather than working features; retrospectives explicitly record learning from failed experiments; and data engineers, ML engineers, and a data steward join the team so that governance enters the execution unit.

5. Analytical Demonstration

In DSR terms this section provides an ex-ante, analytical demonstration: rather than asserting effectiveness, we trace a representative project through the framework and show, facet by facet, that the dual-timeline problem is dissolved. We use a customer-churn-prediction project, a common AI use case; Fig. 2 shows the execution trace.

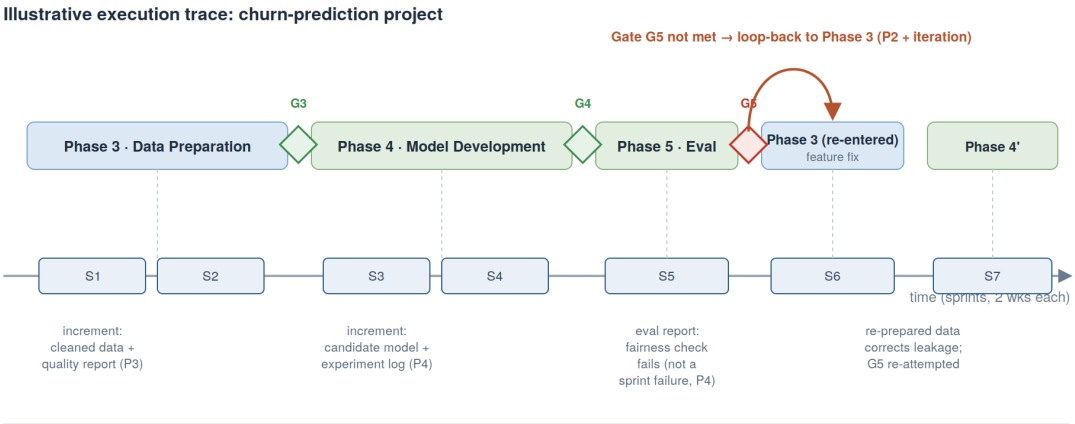


Fig. 2. Illustrative execution trace. Sprints (S1–S7) proceed on a fixed two-week clock while phase gates (G3–G5) are evaluated asynchronously; the failure of gate G5 triggers a governed loop-back to phase 3.

The trace dissolves each facet of Section 3 as follows. Increment collapse (P3): during phase 3 the increments delivered at S1–S2 are a cleaned dataset and a data-quality report—reviewable artifacts—so sprint reviews remain meaningful even though no feature ships. Estimability and uncertainty (P4): phase-4 sprints S3–S4 are experimental spikes; when an experiment underperforms, the sprint still closes with a recorded learning, decoupling experimental risk from sprint success. Timeline mismatch (P2): at S5 the model passes accuracy but fails a fairness check, so gate G5 is not met; rather than forcing phase closure on the sprint boundary, the phase remains open and a governed loop-back re-enters phase 3 (S6) to correct data leakage, after which phase 4 is repeated (S7) and G5 is re-attempted. Completion clarity (two-tier DoD): S6 satisfies its sprint DoD (leakage corrected) independently of whether the phase-5 gate ultimately passes, so "done" is well-defined at each tier against its own pre-registered predicate throughout.

The demonstration is analytical, not empirical: it establishes internal consistency and shows that the artifact behaves as designed against the problem it targets. It does not establish field effectiveness, which we address next.

6. Instrument Pre-Test and Refinement (Stage One)

We adopt a staged evaluation plan. Evaluation criteria are derived from the Section 3 problems and Section 4 principles to preserve traceability (Table 4); each is operationalized as a five-point Likert item with accompanying open commentary.

Table 4. Traceability matrix linking problems, design principles, and evaluation criteria.

Criterion	Problem (Section 3)	Design (Section 4)	Operationalization
Schedule coherence	Timeline mismatch	Gate decoupling (P2)	Reduced conflict between phase goals and sprint cadence
Increment visibility	Increment collapse	Redefined increment (P3)	Reviewable increments produced even in data phases
Completion clarity	DoD collapse	Two-tier DoD	Ambiguity in completion judgment reduced (not eliminated)
Uncertainty handling	Non-determinism	Spikes (P4)	Failure-prone experiments handled controllably
Trust/governance	Post-deployment continuity	Embedded trust (P5)	Ethics/bias/compliance checks governed within execution

Stage one is an explicitly bounded LLM-simulated Delphi, reported here as a pre-test of the instrument and not as evidence about the framework. Recent work reports that LLM agents can simulate interdisciplinary expert panels in iterative Delphi-style elicitation and act as adjuncts to human consensus, with such elicitation positioned as a complement under resource or time constraints rather than a substitute [16], [17]. The rationale for information value despite its limits is that, because the criteria (Table 4) are framed as predicates over the artifact, persona-conditioned readers can expose ambiguous wording, missing criteria, and obvious design objections—signals about instrument quality that do not depend on the authenticity of any "consensus."

Procedure. A single model (claude-opus-4-8, temperature 1.0) instantiated eight pre-registered expert personas; each independently rated the six statements on a five-point scale in Round 1, received anonymized panel medians, IQRs, and critiques, and re-rated in Round 2. Consensus was pre-registered as median ≥ 4 and IQR ≤ 1; the run date and full specification are given in Appendix A and the public replication package.

Results. Under the pre-registered rule, two criteria reached consensus at Round 2: schedule coherence (C1; median 4, IQR 0) and increment visibility (C2; median 4, IQR 0.25). Three criteria attracted a tight cluster of low ratings—completion clarity (C3; median 2, IQR 0), trust/governance (C5; median 2, IQR 1, mean falling 2.0→1.63), and practical adoptability (C6; median 2, IQR 0)—and uncertainty handling (C4) lost consensus between rounds, its median declining from 4 to 3 (Table 5). Notably, feedback drove convergence downward (C4, C5) rather than upward.

Table 5. Per-criterion ratings and consensus outcomes across Round 1 and Round 2 of the LLM-simulated Delphi.

Criterion	R1 median (IQR)	R2 median (IQR)	R2 consensus	Dominant critique theme
C1 Schedule coherence	4 (0)	4 (0)	Yes	Accepted; risk of phases drifting without escalation rules
C2 Increment visibility	4 (0.25)	4 (0.25)	Yes	Accepted; needs minimum data-quality-report schema
C3 Completion clarity	2 (0)	2 (0)	No	"Unambiguous" overclaims; predicates underspecified
C4 Uncertainty handling	4 (0)	3 (0)	No	Spikes not novel; spike-output standard missing
C5 Trust/governance	2 (0)	2 (1)	No	DoD checkbox risks "compliance theater"
C6 Practical adoptability	2 (0)	2 (0)	No	Cumulative overhead; adoption playbook absent

Interpretation as instrument signal. The pattern is diagnostic rather than evaluative. The unanimous low rating on C3, with a consistent critique that "unambiguous" overclaims, motivated the claim calibration already applied in Section 4.3 and a rewording of the C3 item (Table 4). The C5 and C6 results indicate that those items conflate the framework's intrinsic design with the availability of implementation scaffolding (tooling, ownership, escalation, an adoption playbook); the C5 critique specifically motivated the governance elaboration in Section 4.2, by which trust checks are ratified through the gate rather than self-attested. The criteria tied to the framework's structural core (C1, C2) were clear and stable. These are precisely the refinements a stage-one pre-test is empowered to inform, and they were incorporated without altering the five design principles.

Interpretation as methodological evidence. Two observations bound the reading. First, the elicitation did not exhibit sycophantic inflation: medians moved downward under feedback, contrary to the concern that LLM panels converge upward to agree. Second, and more consequentially, within-round variance was near zero (IQR 0 for four of six criteria at Round 2; the eight personas collapsed onto only three distinct rating vectors), which should be read not as agreement among independent experts but as the internal consistency of one model prompted eight ways. This is the independence violation made directly observable: panelists drawn from a single model share parameters and are

correlated, so the "consensus" is substantially an artifact of single-model conditioning—a population-level effect documented independently in LLM agent populations, where collective bias emerges even when individual agents exhibit none [18]—and the downward shift in Round 2 reflects a closed loop in which the model reacts to the uniformly caveated tone of its own Round-1 critiques rather than independent reconsideration. Accordingly, high agreement is not treated as evidence of validity [16], and persona representativeness—frequently underspecified in persona research [19]—is documented in Appendix A and the public replication package together with model identity, version, temperature, and run date for reproducibility; because the run used temperature 1.0 on a dated checkpoint, the values are reported as a single illustrative run rather than stable point estimates.

Temperature 1.0 was deliberate rather than an oversight: at near-zero temperature, the within-round collapse could be attributed to greedy decoding itself, whereas under standard sampling diversity the collapse onto three rating vectors is the more conservative demonstration of single-model conditioning. Since provider-side nondeterminism precludes bit-exact replication even at temperature 0, reproducibility is secured at the protocol level—pre-registered personas, prompts, consensus rule, and complete raw outputs are published in the replication package.

Stage two, reserved as future work, is a human-expert Delphi across AI project managers, data scientists/ML engineers, and Agile coaches, using content-validity indices [20] for consensus and incorporating the revised instrument; stage three is multi-case field application for ex-post evaluation.

7. Discussion: Boundary Conditions and Threats

The framework is not universally appropriate, and stating where it should not be used guards against over-generalization. Its overhead—pre-registered gates, gate governance, a data steward role, and two-tier DoD—is justified only when phase structure and governance carry real value. We identify three boundary conditions. First, small exploratory PoCs: for a short proof-of-concept whose goal is to learn whether a problem is tractable at all, imposing six gated phases adds ceremony without payoff, and lightweight Agile or a single CRISP-DM pass is preferable. Second, single-practitioner projects: gate governance presupposes a separation between a technical proposer and a product-owner/data-steward ratifier; with one data scientist the separation is fictional and the gates become self-approval. Third, continuous-delivery ML with mature MLOps: where retraining is automated and monitored on an established pipeline, the phase backbone is better treated as a one-time setup than as a recurring cadence, and an MLOps-centric model may dominate.

The framework should also be positioned against the ML-specific process and operations literature. Quality-assured refinements of the data-mining lifecycle, notably CRISP-ML(Q) [21], add per-phase quality-assurance and risk gates but leave reconciliation with a fixed Agile cadence out of scope; our coupling device (P2) is precisely what those phase gates do not supply. MLOps reference architectures [22] close the deployment and continuous-training gap through an automated operational layer, but they operate after the upstream phase-versus-cadence conflict is resolved—which is why mature MLOps is a stated boundary condition, not a competitor to the phase backbone. Finally, among data-centric Agile variants, Data Driven Scrum [23] resolves the mismatch by abandoning the fixed iteration rather than decoupling a gate from it—the opposite design choice to ours, and a useful contrast: we keep the fixed sprint heartbeat and absorb the mismatch at the phase boundary.

Threats to the present contribution follow from its stage. The empirical input so far is a single-model simulated pre-test and an analytical demonstration rather than field evidence, so causal claims about delivery outcomes (schedule adherence, model quality) are deferred to stages two and three. The demonstration uses a single illustrative trace and does not exhaust the space of project shapes. Finally, gate predicates are domain-specific; the framework specifies how to govern them but not which thresholds to choose, which remains a matter of project context.

A limitation of the evidence base for RQ1 should be stated explicitly. CPMAI is a recent methodology whose canonical description exists primarily in PMI's certification and practitioner materials [1], [2], [15]; no peer-reviewed academic treatment exists. PMI has since published *The Standard for Artificial Intelligence in Portfolio, Program, and Project Management* [24], an ANSI-consensus standard that briefly acknowledges the tension between fixed sprint cadences and AI's longer experimental timelines (PMI's Section 4.1) but addresses it only as general tailoring guidance, without a phase-gate mechanism, predicate criteria, or a two-tier DoD. The coupling problem this paper formalizes therefore remains unaddressed even in PMI's most current standard-level treatment.

Adoption overhead deserves separate emphasis, since it was the most consistent reservation in the stage-one pre-test (C6). The framework deliberately concentrates ceremony—pre-registered predicates, two-tier DoD, gate ratification, and spike protocols—and we do not claim this is costless. Outside the boundary conditions above, we expect the overhead to be justified only where governance and auditability are first-order requirements (e.g., regulated domains), and we regard a phased adoption path and supporting tooling as a prerequisite for practice rather than an afterthought. Specifying that adoption path is itself future work; the contribution here is the operating model and its governance, not a turnkey rollout.

8. Conclusion

This paper formalized the conflict between Agile and AI work as a dual-timeline problem (RQ1) and proposed an Agile–CPMAI integration framework that couples CPMAI phases and Agile sprints through phase gates and a two-tier Definition of Done, elements differentiated from classical stage-gate models, Agile–Stage-Gate hybrids, and conventional layered DoD (RQ2). The framework was examined through an analytical demonstration via a worked execution trace, including a governed gate-driven loop-back, and through a bounded, single-model LLM-simulated Delphi serving as a stage-one instrument pre-test (RQ3). The pre-test left the five design principles intact while motivating disciplined refinements—claim calibration for the two-tier DoD and an extension of gate governance to trust checks—and, through its near-zero within-round variance, furnished a concrete illustration of why single-model elicitation cannot substitute for independent expert judgment. The principal contributions are the conceptual articulation of the dual-timeline problem and a phase-level operating model, complete with gate governance, that enables Agile organizations to accommodate the data-centric, experimental nature of AI projects without discarding their cadence. The work remains limited by the absence of naturalistic, longitudinal evidence; future work will conduct a human-expert Delphi with the revised instrument, followed by multi-case field studies, to assess the framework empirically.

Acknowledgements

This research was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (RS-2024-00436124, Human Resource Development Program for Industrial Innovation (Global)).

Appendix A. Transparency materials for the simulated Delphi

Model: claude-opus-4-8. Run date: 2026-05-29 (UTC). Temperature: 1.0. Panel size: 8 personas. Rounds: 2. Consensus rule (pre-registered): median ≥ 4 and IQR ≤ 1. Single model by design; persona independence is not satisfied. Full persona specifications, verbatim prompt templates, the runner implementation, and complete per-persona raw ratings (Round 1 and Round 2, including all critiques) are provided in the study's public replication package at https://github.com/camp-epm/reconciling-two-time-logics, together with a README describing the file structure and reproduction steps.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used Anthropic's Claude in order to improve language and readability and to check the consistency and formatting of references, citations, and tables. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] Project Management Institute. “PMI Certified Professional in Managing AI (PMI-CPMAI).” Project Management Institute. Available at: https://www.pmi.org/certifications/ai-project-management-cpmai (accessed July 2026).

[2] Walch, Kathleen, and Ron Schmelzer. (2025) “6 Phases to Run a Successful AI Project.” PMI Blog. Available at: https://www.pmi.org/blog/6-stages-to-run-a-successful-ai-project (accessed July 2026).

[3] Peffers, Ken, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. (2007) “A design science research methodology for information systems research.” *Journal of Management Information Systems* **24** (**3**): 45–77. DOI: 10.2753/MIS0742-1222240302.

[4] Hevner, Alan R., Salvatore T. March, Jinsoo Park, and Sudha Ram. (2004) “Design science in information systems research.” *MIS Quarterly* **28** (**1**): 75–105. DOI: 10.2307/25148625.

[5] Beck, Kent, et al. (2001) “Manifesto for Agile Software Development.” Available at: https://agilemanifesto.org (accessed July 2026).

[6] Chapman, Pete, et al. (2000) *CRISP-DM 1.0: Step-by-step Data Mining Guide.* SPSS Inc. / CRISP-DM Consortium.

[7] Mohanty, Asamanya. (2020) “Agile, CRISP-DM and CPMAI methodologies in AI and ML projects.” Available at: https://www.scrabbl.com/agile-crisp-dm-and-cpmai-methodologies-in-ai-and-ml-projects (accessed July 2026).

[8] Cooper, Robert G. (1990) “Stage-gate systems: a new tool for managing new products.” *Business Horizons* **33** (**3**): 44–54. DOI: 10.1016/0007-6813(90)90040-I.

[9] Sommer, Anita Friis, Christian Hedegaard, Iskra Dukovska-Popovska, and Kenn Steger-Jensen. (2015) “Improved product development performance through Agile/Stage-Gate hybrids: The next-generation Stage-Gate process?” *Research-Technology Management* **58** (**1**): 34–45. DOI: 10.5437/08956308X5801236.

[10] Cooper, Robert G., and Anita Friis Sommer. (2016) “The Agile–Stage-Gate hybrid model: A promising new approach and a new research opportunity.” *Journal of Product Innovation Management* **33** (**5**): 513–526. DOI: 10.1111/jpim.12314.

[11] Mielke, Chris. (2025) “AI-Powered PM: CPMAI–Agile integration and team management playbook.” The AI-Powered Project Manager. Available at: https://theaipoweredprojectmanager.substack.com/p/ai-powered-pm-cpmai-agile-integration (accessed July 2026).

[12] Saltz, Jeffrey S., and Nicholas Hotz. (2020) “Identifying the Most Common Frameworks Data Science Teams Use to Structure and Coordinate Their Projects.” *2020 IEEE International Conference on Big Data (Big Data)*: 2038–2042. DOI: 10.1109/BigData50022.2020.9377813.

[13] Saltz, Jeffrey S., and Iva Krasteva. (2022) “Current Approaches for Executing Big Data Science Projects—A Systematic Literature Review.” *PeerJ Computer Science* **8**: e862. DOI: 10.7717/peerj-cs.862.

[14] Baijens, Jeroen, Remko Helms, and Deniz Iren. (2020) “Applying Scrum in Data Science Projects.” *2020 IEEE 22nd Conference on Business Informatics (CBI)*: 30–38. DOI: 10.1109/CBI49978.2020.00011.

[15] IMT-PM. (2025) “Mastering AI Project Success with CPMAI: A Roadmap to Innovation.” Available at: https://www.imt-pm.com/mastering-ai-project-success-with-cpmai-a-roadmap-to-innovation/ (accessed July 2026).

[16] Park, Young Suk, Dongjae Jeon, Songchang Shi, Eric G. Sheu, Ali Tavakkoli, Abdelrahman Nimeri, and Ahram Han. (2025) “How does AI compare to the experts in a Delphi setting: simulating medical consensus with large language models.” *International Journal of Surgery* **112** (**2**): 2374–2385. DOI: 10.1097/JS9.0000000000003631.

[17] Lorenz, Tobias, and Mario Fritz. (2026) “Scalable Delphi: large language models for structured risk estimation.” arXiv:2602.08889.

[18] Ashery, Ariel Flint, Luca Maria Aiello, and Andrea Baronchelli. (2025) “Emergent social conventions and collective bias in LLM populations.” *Science Advances* **11** (**20**): eadu9368. DOI: 10.1126/sciadv.adu9368.

[19] Batzner, Jan, et al. (2025) “Whose personae? Synthetic persona experiments in LLM research and pathways to transparency.” *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES).* arXiv:2512.00461.

[20] Lawshe, Charles H. (1975) “A quantitative approach to content validity.” *Personnel Psychology* **28** (**4**): 563–575. DOI: 10.1111/j.1744-6570.1975.tb01393.x.

[21] Studer, Stefan, Thanh Binh Bui, Christian Drescher, Alexander Hanuschkin, Ludwig Winkler, Steven Peters, and Klaus-Robert Müller. (2021) “Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology.” *Machine Learning and Knowledge Extraction* **3** (**2**): 392–413. DOI: 10.3390/make3020020.

[22] Kreuzberger, Dominik, Niklas Kühl, and Sebastian Hirschl. (2023) “Machine Learning Operations (MLOps): Overview, Definition, and Architecture.” *IEEE Access* **11**: 31866–31879. DOI: 10.1109/ACCESS.2023.3262138.

[23] Saltz, Jeffrey S., Alex Sutherland, and Nicholas Hotz. (2022) “Achieving Lean Data Science Agility Via Data Driven Scrum.” *Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS-55)*: 7287–7296. DOI: 10.24251/HICSS.2022.876.

[24] Project Management Institute. (2026) *The Standard for Artificial Intelligence in Portfolio, Program, and Project Management.* Project Management Institute. ISBN 978-1-62825-891-2 (paperback) / 978-1-62825-892-9 (ebook).