

MILESTONE PLANNING AND RESEARCH, INC.

The Inductive Enterprise — Research Paper

MACHINE-ASSISTED PERCEPTION · HYBRID COGNITION

Seeing What the Dashboard **Misses**

*Machine-Assisted Perception and Hybrid Cognition
in Project Risk Management*

PRIMMS®-GPT · Bayesian Weight of Evidence · Cortical Hierarchy Architecture

John M. Aaron

MILESTONE PLANNING AND RESEARCH, INC.

Copyright © 2026 by John M. Aaron

All rights reserved.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, scanning, or other electronic or mechanical methods, without the prior written permission of the copyright holder, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

This paper is intended to provide information and insights regarding artificial intelligence, project risk management, and hybrid cognition architectures. The author has made every effort to ensure the accuracy of the information contained herein. However, the information is provided “as is” without warranty of any kind, either express or implied. The author shall not be liable for any damages arising from the use of this material.

The views expressed in this paper are those of the author and do not necessarily reflect the views of any affiliated organizations.

Title: *Seeing What the Dashboard Misses*

Subtitle: *Machine-Assisted Perception and Hybrid Cognition in Project Risk Management*

Author: John M. Aaron

PRIMMS® is a registered trademark of Milestone Planning and Research, Inc. The author declares a commercial interest in PRIMMS®-GPT. The retrospective validation described herein was conducted on project data from completed programs in which the author was involved. All project-identifiable data has been anonymized. No external research funding was received for this work.

Artificial Intelligence Assistance Notice

Portions of this manuscript were developed with the assistance of artificial intelligence tools used for editing support, structural suggestions, and language refinement. All concepts, interpretations, conclusions, and final editorial decisions remain the responsibility of the author. The author assumes full responsibility for the content of this work.

First Edition

Published 2026

Printed in the United States of America

Seeing What the Dashboard Misses: Machine-Assisted Perception and Hybrid Cognition in Project Risk Management

John M. Aaron

Milestone Planning and Research, Inc.

john.aaron@milestoneplanning.net

Abstract

Despite decades of methodological refinement, project failure rates remain stubbornly high. Root cause analysis consistently points to a shared structural deficit: project management tools record task status but cannot infer the structural conditions that precede visible failure. This paper introduces PRIMMS-GPT — Project Risk Identification Measurement and Mitigation System, augmented by a Large Language Model — as a hybrid cognition architecture designed to close this intelligence gap. PRIMMS-GPT combines a compiled MATLAB application that computes deterministic Bayesian weight-of-evidence (WoE) signals from schedule data, Voice of Team (VoT) distributed sentiment, and documentary artifacts, with a five-layer Cortical Hierarchy prompt architecture that routes those signals through a Large Language Model (LLM) for interpretation, pattern recognition, trajectory projection, and executive briefing generation. The system classifies projects against six empirically grounded failure archetypes, produces recovery trajectory projections at +2, +4, and +8-week horizons, and generates a sponsor-ready governance document — while preserving an inviolable human authority boundary at every decision point. Retrospective application across a portfolio of completed enterprise transformation programs demonstrates that the architecture consistently produces decisive evidence classifications and identifies actionable recovery windows not surfaced by conventional monitoring. An illustrative example drawn from one completed business transformation program produces a schedule jeopardy classification at 65.5 db (decisive evidence strength) and identifies a four-day actionable recovery window that was not surfaced through the conventional monitoring in use at the time. The architecture advances a specific argument: that the most consequential application of AI in project management is not task

automation but structural perception — the capacity to detect what is actually happening inside a project before it becomes visible in delivery output. The paper further identifies two foundational constraints on machine-assisted cognition: the Gödel-Turing constraint, which establishes that inductive systems cannot regulate themselves from within, and the Zero constraint, which establishes that systems accumulating signal complexity without formal governing equations collapse into indeterminacy. The governing equations of PRIMMS-GPT — the Bayesian weight-of-evidence accumulator, the project production function, and the issue closure ratio θ — are the mechanism by which the Zero constraint is satisfied: they bound what the LLM is required to treat as settled, so that its interpretive capacity is directed at what genuinely requires judgment.

Keywords: project risk management; hybrid cognition; Bayesian weight of evidence; machine-assisted perception; artificial intelligence; Voice of the Team; failure archetypes; large language models; project governance; Zero constraint; governing equations; issue lifecycle management; AI

1. Introduction

Project management has been a subject of systematic study for more than seventy years. The discipline has produced a rich body of method: from PERT/CPM scheduling frameworks of the 1950s through the Earned Value Management standards codified in the 1960s and 1970s, from the Agile Manifesto of 2001 through contemporary scaled frameworks such as SAFe and LeSS. Standards organizations — the Project Management Institute (PMI), PRINCE2, ISO 21502 — have invested substantial resources in defining best practice and professionalizing the field.

The outcome, measured against the problem the field set out to solve, is disappointing. PMI's Pulse of the Profession surveys consistently find that fewer than half of projects complete on time, within budget, and with originally promised scope. The Standish Group CHAOS Report, tracking software project outcomes since 1994, finds success rates below 40% by its criteria after three decades of tool improvement and methodology development. McKinsey research on large information technology projects documents an average of 45% cost overrun, 7% schedule overrun, and 56% reduction in delivered value relative to predictions (Bloch, Blumberg, & Laartz, 2012).

These numbers do not reflect ignorance or neglect. They persist alongside genuine improvements in practice.

The structural argument advanced throughout this paper — developed in detail in Aaron (2026, Chapter 1; unpublished manuscript) — is that project failure is primarily a perception and orientation problem, not a planning or tool problem. Projects fail not because managers lack scheduling software or risk registers. They fail because the signals of structural deterioration accumulate invisibly, optimism bias suppresses their interpretation, and the window for low-cost correction closes before leadership recognizes it. The commercial tool landscape, characterized by increasingly sophisticated task-tracking and collaboration platforms, has not addressed this problem — and, given its architectural orientation toward record-keeping rather than inference, cannot address it without a fundamentally different approach.

This paper describes PRIMMS[®]-GPT (Project Risk Identification Measurement and Mitigation System, augmented by a Large Language Model) as a concrete instantiation of a different approach: hybrid cognition. The term, drawn from cognitive science and developed in the project management context by Aaron (2026), denotes architectures in which machine intelligence extends human perception without replacing human judgment. PRIMMS-GPT extends the project manager's perceptual range across three signal modalities — quantitative schedule data, distributed team sentiment, and governance documentation — and routes the fused signal set through a hierarchical LLM interpretation architecture that produces actionable governance outputs. At every decision point, the human project manager retains full authority. The machine orients; the humans decide.

The contribution of this paper is threefold. First, it presents the PRIMMS-GPT architecture in sufficient technical detail to enable replication and extension. Second, it articulates the theoretical grounding — in Bayesian inference, cognitive neuroscience, and decision theory — that distinguishes machine-assisted perception from conventional AI-powered project management tools. Third, it reports an ongoing retrospective validation program and presents one illustrative example in detail, demonstrating the system's analytical capability against a completed project case with quantified evidence strength and known outcomes.

The paper is organized as follows. Section 2 reviews the relevant literature across three domains: project management failure research, AI applications in project management, and hybrid cognition

theory. Section 3 presents the PRIMMS-GPT architecture in detail. Section 4 describes the five-layer Cortical Hierarchy prompt design. Section 5 reports the retrospective validation program and presents one illustrative example in detail. Section 6 discusses theoretical implications and practical contributions. Section 7 addresses limitations and future work.

1.1 Origins and Development History

PRIMMS[®] did not emerge from an academic research program. It was built in direct response to a practitioner problem: the author's experience managing large-scale SAP enterprise resource planning implementations repeatedly produced the same disorienting pattern. Risk signals were present in the project environment — in team conversations, in subtle changes in stakeholder behavior, in the texture of weekly status reports — but the conventional tools available could not surface or integrate those signals into a coherent governance picture. Projects would appear nominally green on the standard reporting dashboard, then enter crisis within weeks. The gap between what project participants knew and what the formal reporting system showed was structural, not incidental, and no available tool addressed it.

The first design response, developed in 2008, was a prediction market mechanism. Drawing on research in parimutuel wagering systems and the observed accuracy of prediction markets in other forecasting contexts, the author designed and filed a patent application for a system that invited project stakeholders to wager on scenario-based project outcomes using parimutuel odds. The core insight was that financial incentives would elicit more honest signals from team members than conventional surveys, and that the price ratios produced by parimutuel wagering would provide a continuous, market-derived estimate of project risk. The PRIMMS[®] name (Project Risk Identification Measurement and Mitigation System) and trademark were registered at this time. The patent application was ultimately rejected; and the wagering mechanism itself, while theoretically sound, proved organizationally impractical — asking employees to bet on project failure outcomes created cultural resistance that undermined adoption.

The parimutuel mechanism's failure as an organizational tool, however, pointed toward a more tractable design. The price ratio at the heart of the wagering system is mathematically equivalent to a likelihood ratio — and likelihood ratios are the fundamental building block of Bayesian inference. This realization, combined with the author's parallel interest in the wartime Bayesian decryption methods developed by I. J. Good and Alan Turing at Bletchley Park, produced the key

architectural shift: replacing parimutuel betting prices with Bayesian weight-of-evidence expressed in decibans. Rather than asking team members to bet, the revised system asked them to vote on outcome scenarios using a structured six-point scale. The frequency and intensity of mentions across risk categories in those votes and in project documents could then be converted directly into likelihood ratios, summed in deciban form, and compared to empirically established thresholds — without requiring any financial transaction. PRIMMS[®] went live in this form, using Bayesian weight of evidence from Voice of the Team (VoT) distributed sentiment as its core inference mechanism.

From 2008 through the present, PRIMMS has been deployed across a substantial portfolio of complex enterprise resource planning and business transformation projects. This deployment history provides the empirical foundation for the Bayesian WoE architecture described in Section 3: the signal weights, risk category benchmarks, and threshold calibrations reflect accumulated evidence from real project outcomes across more than seventeen years of operational use. The VoT mechanism and the Bayesian classification layer are, in this sense, not experimental — they are the mature core of a production system with a documented track record. Formal analysis of the 20 risk categories that emerged from text mining of project risk logs across 24 ICT projects (Aaron, 2017 internal working paper) provides the empirical grounding for the risk signal architecture, including statistical identification of the subset of risk categories most predictive of project schedule performance.

The more recent additions to PRIMMS-GPT — semantic document analysis and the Cortical Hierarchy prompt architecture grounded in neuroscientific models of cortical processing identified by Fuster (2008) and Glimcher (2011) — represent a qualitatively different development stage. These components extend the system’s perceptual range into unstructured documentary text and integrate the multi-modal signal set through a five-layer LLM interpretation architecture. Their validation status differs materially from the Bayesian core: while the Bayesian WoE layer has been tested prospectively across full project lifecycles, the semantic analysis and Cortical Hierarchy components have been validated retrospectively against completed project phases rather than across complete prospective projects. This distinction in validation depth is explicitly acknowledged in Section 7.1 and is the primary motivation for the prospective deployment study identified as the highest-priority future work item.

2. Literature Review

2.1 The Persistent Project Failure Problem

The literature on project failure is extensive and largely convergent. Flyvbjerg (2017) documents systematic optimism bias and strategic misrepresentation across infrastructure megaprojects spanning five continents and six decades, finding cost overruns in 86% of cases with a mean overrun of 28%. Budzier and Flyvbjerg (2011) identify a distinct population of 'black swans' — IT projects with cost overruns exceeding 200% — and demonstrate that this distribution is fundamentally different from the rest, suggesting qualitatively different causal mechanisms. Maylor, Vidgen, and Carver (2008) introduce the concept of managerial complexity — the degree of interaction among project system elements — as a predictor of project performance independent of technical scope.

Running across this literature is a consistent finding: the signals of structural deterioration are typically present weeks or months before they become visible in formal project reporting. Kutsch and Hall (2010) document the phenomenon of 'deliberate ignorance' — project managers who recognize emerging risk signals but choose not to act on them due to social and organizational pressures. Williams (2005) identifies the emergence of complex adaptive dynamics in troubled projects, arguing that conventional linear cause-and-effect reasoning cannot adequately model what is actually occurring in project execution. The implication — that project management requires qualitatively different tools for signal detection and pattern recognition — motivates the present work.

2.2 AI Applications in Project Management

The integration of artificial intelligence into project management practice has accelerated markedly in the period following the widespread availability of transformer-based LLMs. Existing reviews (e.g., Pellerin & Perrier, 2019; Niazi, Zhang, & Bhatt, 2023) catalogue applications across the project management lifecycle, including automated scheduling, resource optimization, risk prediction, and progress monitoring. The dominant pattern in this literature is the application of supervised machine learning to historical project datasets to predict outcomes — completion probability, cost overrun magnitude, schedule delay — based on project characteristics measurable at inception.

This line of work produces genuinely useful predictions in contexts where sufficient historical data of consistent quality are available. However, it shares a structural limitation with the conventional PM tools it augments: it operates on data that have already been recorded in formal project systems. The signals that most reliably precede failure — team confidence erosion, perception distortion in leadership, governance gap formation — do not appear in task completion records, cost reports, or formal risk registers. They are distributed across the conversational, documentary, and social texture of project work in ways that historical dataset approaches cannot access.

A second line of work applies LLMs directly to project management tasks: automated meeting summarization, natural language task creation, stakeholder communication drafting, and risk log generation. These applications are productivity improvements — they help project managers perform existing tasks more efficiently. The claim of this paper is more specific: that LLMs have an untapped role as interpretive reasoning engines in a structured signal processing pipeline, performing the natural-language synthesis and pattern recognition tasks that neither deterministic algorithms nor unaided human judgment perform reliably.

Deploying LLMs without this structured pipeline introduces three specific failure modes that are relevant to evaluating existing AI-in-PM applications. The first is Proxy Collapse: when an AI system optimizes for a measurable proxy of project health — schedule variance, cost performance index, milestone completion rate — the proxy becomes the target and decouples from the underlying reality it was meant to track. Teams learn to optimize for the metric; status reports improve while the project deteriorates. The second is Tacit Knowledge Destruction: if project teams stop exercising their own judgment because an AI system is treated as authoritative, the experiential knowledge of practitioners — the felt sense that something is wrong before any dashboard confirms it — atrophies from disuse. When the AI eventually fails in a genuinely novel configuration, the human capacity to fill the gap has been systematically degraded. The third is Recursive Sycophancy: large language models are trained toward user satisfaction, and a project leader who uses an LLM as a primary advisor will receive outputs shaped by the pattern of their own prompts. The model colludes in optimism rather than surfacing it as a risk signal — confidence increases precisely as accuracy decreases. PRIMMS-GPT is explicitly designed to counter all three failure modes: multi-modal Bayesian signal fusion defeats proxy gaming; the advisory-only posture preserves practitioner judgment; and the deterministic MATLAB signal

layer operates independently of any LLM and has no social incentive to satisfy the project manager, structurally preventing sycophantic drift.

Prior work on AI-assisted project risk specifically includes Kock, Gemünden, and Salter (2021), who examine early warning systems for innovation projects, and Ika, Love, and Pinto (2020), who survey the evidence on project management success factors with attention to the role of contextual intelligence. Neither work addresses the hybrid cognition architecture — the explicit coupling of machine signal processing with human interpretive authority — that is the central contribution of the present paper.

2.3 Hybrid Cognition and Bayesian Perception

The theoretical foundation for PRIMMS-GPT draws from two bodies of literature that have not been systematically connected in the project management context: Bayesian brain theories from computational neuroscience, and hybrid cognition frameworks from decision science.

Friston's free energy principle and predictive coding framework (Friston, 2010) propose that biological brains operate as hierarchical Bayesian inference engines, continuously updating generative models of the environment against sensory prediction errors. The hierarchical organization of cortex — from low-level feature detection through increasingly abstract pattern recognition to executive planning — is interpreted as implementing successive levels of inference across progressively larger temporal and spatial scales. The PRIMMS-GPT Cortical Hierarchy, while not a technical implementation of predictive coding, draws on this organizational principle: Layer 1 performs feature detection, Layer 2 gestalt pattern recognition, Layer 3 temporal situation awareness, Layer 4 executive planning, and Layer 5 document generation. The architecture mirrors the computational structure of biological cortex rather than simulating it.

The Bayesian weight-of-evidence framework employed in PRIMMS-GPT's signal fusion layer derives from Jeffreys (1961) and the information-theoretic formalization by Good (1950, 1985). The representation of likelihood ratios as logarithmic deciban values — permitting additive accumulation of evidence across independent signals — operationalizes Bayes' theorem in an auditable, inspectable form. This approach was advocated for decision support in high-stakes contexts by MacKay (2003) and has been applied in medical diagnosis and intelligence analysis contexts.

The foundational application of Bayesian methods to project management was established by Aaron (2015), who introduced a microeconomic production function framework for project issue management and derived the first Bayesian quality gate metric for project management practice. That work modeled project output as a function of scheduled activity completions, issue discovery, and issue closure — $Y = f(AC, IC, IO)$ — and introduced the theta parameter ($\theta = \text{total closed issues} / \text{total issues in the pool}$) as a continuous proxy for project health and phase-exit readiness. Aaron (2015) further showed that the Bayes Factor, expressed in decibans (db, the logarithm of the likelihood ratio of success to failure as a function of θ), provides a natural decision boundary at even odds, and demonstrated via Monte Carlo simulation how team sentiment from PRIMMS® surveys could be embedded as the Bayesian prior — to be updated by observed issue evidence in the likelihood function. These results constitute the theoretical axioms on which PRIMMS-GPT's WoE architecture is built. Critically, the 2015 framework established that the optimal management decision is not 100% issue closure but the point at which marginal benefit of further closure equals marginal cost — a microeconomic comparative statics result that Mundell-style analysis makes tractable. PRIMMS-GPT extends this foundation from quality gate readiness at a single milestone to continuous multi-signal risk monitoring across the full project lifecycle, replacing the scalar theta metric with a multi-dimensional signal vector and replacing the single Bayes Factor curve with a structured db summation across signal modalities. The present paper therefore advances a research program that Aaron (2015) initiated but that required the emergence of capable LLMs and compiled signal-processing tooling to fully operationalize.

On the hybrid cognition side, the fundamental argument is that neither machine intelligence alone nor human judgment alone is sufficient for the perceptual task PRIMMS-GPT addresses. Kahneman's (2011) dual-process theory identifies the systematic biases — anchoring, availability, optimism, planning fallacy — that degrade human project risk assessment. Machine signal processing is immune to these biases but cannot perform the contextual interpretation, narrative synthesis, or stakeholder-sensitive communication that project governance requires. The hybrid architecture — machine processing in the signal and classification layers, human authority in the interpretation and decision layers — exploits the comparative advantage of each.

This theoretical integration — Bayesian inference, cortical hierarchy, and hybrid cognition — provides the design rationale for PRIMMS-GPT and distinguishes it from both pure automation approaches and conventional AI productivity tools.

3. The PRIMMS-GPT Architecture

3.1 System Overview

PRIMMS-GPT is a two-component system. The first component is a compiled MATLAB desktop application that performs deterministic signal extraction, Bayesian WoE computation, failure classification, and structured prompt generation. The second component is an LLM such as Claude (Anthropic) or ChatGPT (OpenAI), a Large Language Model accessed through the conversational interface, which receives the structured prompts and produces natural-language interpretation, pattern analysis, recovery trajectory projections, and a formatted executive briefing document.

The design rationale for this two-component architecture reflects a specific principle: computational tasks for which deterministic, auditable algorithms are available should not be delegated to LLMs, and interpretive tasks for which natural language synthesis is required should not be performed by deterministic algorithms. The MATLAB component handles signal computation with complete auditability and reproducibility. The LLM component handles interpretation and synthesis with full transparency about the signal inputs it receives. The division of labor is explicit and enforced by design.

Table 1 presents the five architectural layers of PRIMMS-GPT with implementation details and design rationale.

Table 1. PRIMMS-GPT Architectural Layers: Implementation and Rationale

Architectural Layer	Implementation Detail	Design Rationale
Signal Acquisition	Schedule metrics: deliverable completion state, days behind plan, jeopardy index, velocity-projected finish date. Voice of the Team (VoT): discrete 1–6 team confidence ratings timestamped and stored as a time series. Documentary corpus: free-text status reports, governance documents, and risk artifacts ingested via keyword and pattern extraction.	Human-generated signals from three independent modalities — quantitative schedule, distributed team sentiment, and governance language — are fused without requiring any modification to existing project management tools.
Inductive Classification	A deterministic Bayesian weight-of-evidence (WoE) classifier maps each signal to a likelihood ratio expressed in decibans (db) using Jeffreys' strength-of-evidence bands: 0–5 db (barely worth mention), 5–10 db (substantial), 10–15 db (strong), 15–20 db (very strong), >20 db (decisive).	The logarithmic db summation operationalizes Bayes' theorem in an auditable, inspectable form. Every signal weight is fixed and traceable to specific evidence — a deliberate design choice favoring auditability over adaptive learning for enterprise governance contexts.

	Signal db values are summed to a prior total representing aggregate risk evidence.	
Failure Archetype Recognition	The classifier maps the convergent signal pattern to one of six defined failure archetypes: (1) Execution Collapse — delivery cadence failing; (2) Perception Distortion — leadership optimism diverges from data; (3) Premature Closure — replan triggered without testing recovery; (4) Scope Drift — boundary erosion without capacity adjustment; (5) Governance Gap — decision authority unclear; (6) Confidence Erosion — team belief in achievability declining.	Archetype identification is a qualitative inference step that cannot be performed by schedule analytics alone. It requires the integration of VoT sentiment data with schedule and documentary evidence — the multi-modal signal fusion that distinguishes PRIMMS-GPT from task-tracking platforms.
Cortical Hierarchy (LLM Interpretation)	The MATLAB-computed signal set, VoT data, and documentary signals are formatted into a structured five-layer prompt package. Claude (Anthropic; model version current at time of writing, recorded in the audit trail per Section 4.3) receives these sequentially: Layer 1 — feature validation; Layer 2 — gestalt pattern recognition; Layer 3 — temporal situation awareness with trajectory projections at +2/+4/+8 weeks; Layer 4 — executive planning and COA matrix; Layer 5 — sponsor-ready briefing document generation (terminated by DOCX_READY).	The LLM layer performs the natural-language interpretation, synthesis, and projection that classical algorithms cannot provide. The five-layer structure mirrors the hierarchical organization of biological cortex from sensory processing through executive planning, an architecture grounded in Kanerva (1988) and Friston's hierarchical predictive coding framework.
Human Governance	Every PRIMMS-GPT output is framed as orientation, not instruction. The system's closing statement — 'The machine has oriented. The humans now decide.' — is structurally enforced in every prompt. An immutable audit trail records all signal values, db computations, LLM model version, full prompt and response text, and DOCX_READY detection status.	The human authority boundary is the architectural property that most directly addresses the AI governance concerns catalogued by the Project Management Institute and ISO 21502. The machine extends perception; the project manager owns every decision and every escalation.

3.2 Signal Extraction and Bayesian WoE Computation

The signal extraction process ingests three input modalities: a project schedule workbook (Microsoft Excel or CSV format), Voice of Team (VoT) entries recorded through the application's survey interface, and documentary text ingested by the user. The documentary modality accepts free-text uploads of any project communication or governance artifact — including weekly status

reports, steering committee updates, escalation emails, meeting transcriptions, risk and issue logs, corrective action plans, and change request documentation. Users paste or upload this material directly into the application's document tab; no structured format or tagging is required. This design reflects a deliberate architectural choice: the signals most predictive of structural project deterioration — risk language intensifying in status reports, recovery plan language present or absent in governance documents, escalation references in meeting records — reside in unstructured text that no schedule system captures. No custom data collection infrastructure is required beyond the schedule export and communications that project managers typically already produce. The empirical rationale for combining three independent modalities rather than relying on any single signal source is grounded in simulation work reported in Aaron (2019), which demonstrated that a machine learning system applied to a real-time operational decision problem achieved 73% predictive accuracy using image data alone; accuracy improved to 82% when a second modality (audio signal features) was added; and reached 100% test accuracy when a third modality (text annotations) was incorporated. The implication for PRIMMS-GPT is direct: schedule data, VoT sentiment, and documentary text each carry independent information about project risk that the others do not fully capture, and their fusion produces a materially stronger evidence base than any single source can provide.

From the schedule, the MATLAB component computes: schedule state (on track / lagging / behind / critical), activities behind, maximum days behind, jeopardy index (a composite of activities behind and overdue activity-days), rundown statistics comparing actual versus expected versus planned deliverable completions, and a linear velocity model projecting the finish date with associated R^2 fit quality. From VoT entries, the component computes: mean rating, distribution across the 1–6 scale, temporal trend (improving, flat, worsening), and safe-band classification. From documents, the component applies keyword and pattern extraction to generate signals concerning risk language intensity, recovery plan evidence quality, and charter-level governance documentation.

Each detected signal is mapped to a likelihood ratio and expressed in decibans using the Jeffreys (1961) strength-of-evidence convention. Signals are summed to a prior total db value and classified against bands: 0–5 db (barely worth mention), 5–10 db (substantial), 10–15 db (strong), 15–20 db (very strong), >20 db (decisive). Mitigating signals — such as strong recovery plan documentation

or ahead-of-plan velocity — carry negative db values, appropriately reducing the aggregate evidence total. Table 2 presents a representative signal set from a pilot deployment.

Table 2. Representative Signal Set: Illustrative Example (Snapshot: 08-Aug-2025)

Signal Name	Source Modality	dB Weight	Trigger Condition
SCHEDULE_STATE_BEHIND	Schedule	10 dB	Project activities are in a Behind state; deviations from plan are present
SCHEDULE_DAYS_BEHIND	Schedule	Scaled (max 10)	Days behind plan; scaled logarithmically
SCHEDULE_JEOPARDY	Schedule	Scaled (max 10)	Composite jeopardy index (days behind + overdue activities)
VOT_MEAN_ELEVATED	VoT	7 dB	Team mean confidence rating ≥ 3.5 on 1–6 scale
CLASSIFIER_JEOPARDY	Inductive	18 dB	Bayesian classifier has reached jeopardy threshold on documentary signals
RUNDOWN_GAP_BEHIND	Rundown	Scaled	Actual deliverable completions lag expected count
VELOCITY_FINISH_SLIP	Velocity	Scaled	Linear velocity model projects finish beyond planned end date
RECOVERY_URGENCY_CRITICAL	Recovery	14 dB	Schedule state + days behind converge on critical recovery need
RECOVERY_PLAN_DOCUMENTED_STRONG	Recovery	–6 dB	Strong recovery plan language present in documents (mitigating signal)
RECOVERY_TREND_WORSENING	Recovery	8 dB	VoT mean is rising (worsening) across reporting periods

The PRIMMS-GPT WoE architecture extends the deciban (db) framework introduced in Aaron (2015) from the single-milestone quality gate context to continuous multi-signal monitoring across the project lifecycle. In the 2015 formulation, a single Bayes Factor curve mapped the scalar theta parameter (issue closure rate) to a db value, with 0 db marking even odds of phase-exit success. PRIMMS-GPT generalizes this structure: each signal contributes an independent db value (the

logarithm of its likelihood ratio), and these are summed across modalities to produce an aggregate prior reflecting the weight of evidence for the risk hypothesis across schedule, team sentiment, and documentary domains simultaneously. This multi-signal generalization is what makes continuous lifecycle monitoring tractable — the same mathematical foundation that Aaron (2015) applied to a single quality gate now operates across the full arc of project execution. The VoT 1–6 sentiment survey is the direct descendant of the team sentiment prior described in the 2015 framework, operationalized through PRIMMS® in both cases.

The choice of deterministic WoE scoring over a learned reward policy — the original design intent called for a DDPG/Q-learning agent that would update signal weights from weekly VoT survey responses — was a deliberate tradeoff decision. Live reinforcement learning would enable the system to adapt signal weights to the specific organizational context over time, but it would sacrifice auditability: weights modified by a Reinforcement Learning (RL) agent are no longer directly inspectable or explainable. For an enterprise governance context where every risk signal must be citable to specific evidence, fixed and inspectable weights are the appropriate design. This tradeoff is explicitly acknowledged as a gap versus original design intent and is discussed further in Section 7.

3.3 Failure Archetype Classification

The failure archetype classification translates the aggregate WoE signal into a qualitative pattern diagnosis. The six archetypes represent empirically grounded failure modes identified through project portfolio analysis and the project management literature. Each archetype has a characteristic signal signature:

- Execution Collapse: high schedule jeopardy index; high rundown gap; worsening VoT trend; low velocity fit quality ($R^2 < 0.5$).
- Perception Distortion: divergence between VoT mean and leadership-reported status; wishful thinking signals in forecast accuracy; stable schedule metrics masking deteriorating team sentiment.
- Premature Closure: recovery urgency signal present; strong VoT worsening; replan or extension language prominent in documents without evidence of tested recovery.

- Scope Drift: charter signal absent or weak; deliverable count increasing over time; rundown gap widening despite adequate team confidence.
- Governance Gap: escalation language in documents without documented resolution; VoT comments referencing blocked decisions; schedule state deteriorating without intervention signals.
- Confidence Erosion: progressive VoT mean increase across reporting periods; flat or declining schedule metrics; risk language in VoT comments intensifying.

The dominant archetype and any co-present secondary archetypes are embedded in the Layer 1 signal summary and passed to the Cortical Hierarchy for LLM interpretation. The archetype classification frames the pattern-recognition task for the LLM layer, enabling it to produce contextually specific recovery guidance rather than generic risk commentary.

3.4 Recovery Signal Architecture

A distinctive feature of the PRIMMS-GPT signal architecture is the explicit representation of recovery signals as first-class inputs. Three recovery signals are computed independently from schedule state and VoT:

Recovery Urgency is derived from the convergence of schedule state, days behind, and jeopardy index, and classified as Watch / Elevated / Critical. Recovery Plan Evidence searches the documentary corpus for plan, corrective action, recovery, and rebaseline language, classifying the quality of documented recovery intent as Absent, Vague Only, Moderate, or Strong, and applying a corresponding db adjustment (−6 db for Strong evidence, reflecting the mitigating effect of a credible plan). Recovery Temporal Trend examines the VoT time series for directional movement — improving, flat, or worsening — and applies a corresponding signal to capture whether recovery actions are producing visible team sentiment effects.

The three recovery signals together answer three governance questions that project sponsors need answered on at-risk projects: Is recovery needed and how urgently? Is there a documented recovery plan? Are recovery steps producing visible effects? The Layer 3 prompt (Temporal Situation Awareness) is explicitly structured around these three questions, requiring the LLM to answer each in order before proceeding to trajectory projections.

4. The Five-Layer Cortical Hierarchy

4.1 Architecture Rationale

The LLM interpretation component of PRIMMS-GPT is structured as a five-layer sequential prompt architecture termed the Cortical Hierarchy. The architecture reflects three design principles. First, no single prompt can simultaneously perform reliable feature validation, gestalt pattern recognition, temporal situation awareness, executive planning, and document generation — these are qualitatively different cognitive tasks that benefit from sequential, focused processing. Second, the hierarchical structure mirrors the computational organization of biological cortex as described in the predictive coding literature, with each layer operating at a higher level of abstraction than its predecessor. Third, a layered architecture creates an auditable record of reasoning: each layer's output is visible and correctable before it propagates to the next layer.

In the deployed system, each layer's prompt is generated by the MATLAB application and saved to a timestamped text file. The user pastes each numbered prompt into the LLM conversational interface sequentially, reading the response before pasting the next prompt. This human-in-the-loop step is not an implementation limitation to be engineered away — it is a deliberate governance feature. The project manager who reads each layer's output before proceeding is performing a judgment operation: assessing whether the LLM's interpretation is consistent with their contextual knowledge, and retaining the authority to override or redirect before the next layer builds on the prior output.

4.2 Layer Descriptions

Layer 1 — Feature Detection. The sensory layer receives the full MATLAB-computed signal set with db values and supporting evidence. The Layer 1 role is to validate and extend: confirm which signals are correctly computed from the available data, identify any signals the deterministic algorithm may have missed given contextual information visible in the project description, and flag any signals that should be removed as spurious. The layer concludes with a confirmed total db value and Jeffreys band. The instruction explicitly prohibits prescriptions — Layer 1 is a perceptual layer, not an intervention layer.

Layer 2 — Gestalt Pattern Recognition. The pattern recognition layer receives the validated signal set and classifies the dominant failure archetype from the six defined types. The layer is instructed to consider all six archetypes (including Premature Closure — the failure mode of

closing a recoverable decision prematurely) and to explicitly assess whether any archetype's firing conditions are met given the evidence. The Premature Closure instruction is specifically structured to counter the institutional default toward replanning: 'PREMATURE_CLOSURE fires when recovery is plausible on at least one quantitative signal AND the institutional default is to replan rather than test recovery.' The layer concludes with the dominant archetype and cumulative db.

Layer 3 — Temporal Situation Awareness. The situation awareness layer extends Endsley's (1995) three-level model — perception, comprehension, projection — to the project risk context. The layer explicitly requires trajectory analysis at +2, +4, and +8 week horizons; identification of surprising or counter-intuitive signals in the data; and five leading indicators of recovery. The recovery assessment sub-section (R1–R5b) mirrors the three governance questions described in Section 3.4, with the addition of a Premature Closure check (R5b) that explicitly asks whether the replan posture is itself a bias flag — whether the recovery hypothesis has been tested and falsified, or merely assumed to be false.

Layer 4 — Executive Planning. The executive planning layer produces the Course of Action (COA) matrix: six structured recovery options (Scope Freeze, Cadence Tighten, Dependency Clearing, Forecast Rebase, Strike Team, Quality Gate Enforce), each with root cause targeted, 3–5 concrete remediation actions, owner archetype, decision gate trigger, leading indicator of success, and COA risk. The layer also produces 2–3 decision gates, a Week 1 intervention plan, key assumptions and bias flags, an escalation posture on a 1–6 scale, and a 400–600-word governance narrative. The layer's closing instruction — 'The machine has oriented. The humans now decide.' — is structurally enforced as the last sentence of every Layer 4 output.

Layer 5 — Executive Briefing Document. The document generation layer synthesizes the full five-layer analysis into a formatted executive briefing using all prior layer outputs. The document structure includes: a color-coded status banner (the banner color, label, and tone instructions are set by MATLAB's deterministic STATUS_KEY computation, not by the LLM); a recovery status section with explicit velocity-based recovery window language; a plain-English situation narrative; a signal table; failure archetype analysis; trajectory projections; recovery options; three leadership decisions required this week; and a first-week intervention plan with an end-of-week velocity gate. The layer terminates with the string 'DOCX_READY', which the MATLAB

application detects automatically as confirmation that the complete executive summary is present and ready for docx generation.

4.3 Audit and Traceability

The MATLAB application maintains a persistent CorticalLog table recording every LLM session. Each record stores: session timestamp, project ID, prompt file path, SHA-256 hash of the prompt content, mode (layered or single), layers processed, LLM model version string (the specific version identifier is recorded at session time and included in the audit log; the default at time of writing was claude-sonnet-4-6), full JSON of layer responses, DOCX_READY detection flag, and log creation timestamp. The complete log is included in session saves and audit exports. This architecture ensures that governance reviewers can identify exactly which model version produced any given analysis, reproduce the signal inputs that drove the analysis, and verify that no output was modified post-generation.

The model version string is embedded in the prompt file header and requires explicit update when the recommended LLM model changes — a design choice that forces a conscious governance decision rather than silent model substitution.

5. Retrospective Validation

5.1 Case Context

The retrospective validation applies PRIMMS-GPT to a set of recently concluded enterprise transformation programs, including SAP S/4HANA Rise implementations and other large-scale IT and data science initiatives. The Bayesian WoE and Voice of Team components of PRIMMS® have been in active production use since 2008 and carry an extensive operational track record across this portfolio. The Cortical Hierarchy LLM layer, by contrast, is newer and is currently being tested retrospectively against completed programs and project phases — running in parallel with, but not yet as a replacement for, the existing PRIMMS® web platform. The retrospective testing program applies the full PRIMMS-GPT architecture to historical project records, with the purpose of assessing whether the system produces signals, classifications, and recovery guidance consistent with what actually occurred or should have occurred.

The protocol applied to each project is as follows. Project schedule exports, status update emails, steering committee documents, and meeting records from the completed program are assembled

and ingested into PRIMMS-GPT. Anonymous Voice of Team (VoT) ratings are reconstructed from meeting records and contemporaneous written assessments by team members and stakeholders. The MATLAB signal layer is then run against each weekly snapshot in sequence, and the five-layer Cortical Hierarchy is executed against selected snapshots of particular analytical interest — typically those that, in retrospect, represented inflection points in the program’s risk trajectory. The outputs of the PRIMMS-GPT analysis are compared against what the conventional PMO monitoring showed at the same date and against the known final outcome of the program.

This testing design is explicitly retrospective and post-hoc. The Cortical Hierarchy component has not yet been deployed in real time during a live client engagement. The validation question is therefore: given access to the same data that was available at the time, does PRIMMS-GPT produce signals, classifications, and recovery window estimates consistent with what actually happened or should have happened? This is a weaker claim than prospective deployment — it cannot establish that the system would have changed decisions in real time — but it is a necessary precursor to prospective validation and tests the signal architecture against known ground truth. The following subsections present one illustrative example from this validation program in detail: a snapshot dated 8 August 2025 from a completed machine learning proof of concept data science enterprise project, chosen because it represents a clear inflection point with a well-documented outcome against which the system’s outputs can be precisely compared.

5.2 Illustrative Example: Signal Set at the 8 August 2025 Snapshot

At the snapshot for a sample project dated 8 August 2025 — the selected illustrative inflection point for this example project — the MATLAB Layer 1 computation produced the following signal set from the retrospectively assembled project data:

Schedule signals: SCHEDULE_STATE_BEHIND (10.0 db); SCHEDULE_DAYS_BEHIND, value 18 days (5.4 db); SCHEDULE_JEOPARDY, index 24.75 (5.0 db). VoT signals: VOT_MEAN_ELEVATED, mean 3.60 across five entries (7.0 db). Classifier signal: CLASSIFIER_JEOPARDY (18.0 db). Rundown signal: RUNDOWN_GAP_BEHIND, gap -9 deliverables (3.6 db). Velocity signal: VELOCITY_FINISH_SLIP, +4 day projected slip (0.6 db). Recovery signals: RECOVERY_URGENCY_CRITICAL, value 18 days behind (14.0 db); RECOVERY_PLAN_DOCUMENTED_STRONG, quality level Strong (-6.0 db, mitigating);

RECOVERY_TREND_WORSENING, VoT mean rising early 3.00 → recent 4.00, delta +1.00 (8.0 db).

Total prior: 65.5 db. Jeffreys band: Decisive.

The velocity-based rundown model produced a projected finish date of 24 October 2025 — four days after the planned end date of 20 October 2025 — with $R^2 = 0.63$. The critical finding was not the slip itself but its magnitude: a 6% improvement in daily delivery velocity (from 0.54 to 0.57 deliverables per day) would close the gap entirely within the existing timeline, without scope or schedule change.

5.3 Cortical Hierarchy Analysis: Illustrative Example

The five-layer Cortical Hierarchy was executed against the 8 August 2025 snapshot of the illustrative example program. The outputs are reported here as produced — they represent what the system generated from the retrospectively assembled data, not a real-time governance document — and are intended to demonstrate the architecture’s reasoning process rather than to constitute a standalone validation claim.

The Layer 2 archetype classification identified GOVERNANCE_GAP and PREMATURE_CLOSURE as co-dominant archetypes. The governance gap was evidenced by the divergence between a strongly-documented recovery plan (−6 db mitigating) and a worsening VoT trend (8 db) — the plan existed on paper but was not producing measurable team sentiment effects. The premature closure archetype was flagged because the velocity projection showed a recoverable slip, while the institutional default under schedule pressure was to trigger a replan — closing a decision that the evidence did not yet justify closing. The Layer 2 output explicitly stated: 'The recovery hypothesis has not been falsified; it has simply not been tested.'

Layer 3 situation awareness computed that the recovery window was approximately 2–3 weeks: after that point, the compounding of inaction would make the 6% velocity improvement insufficient to close the gap within the original timeline. The +2-week trajectory showed the recovery window narrowing but open; the +4-week trajectory showed formal scope negotiation becoming necessary; the +8-week trajectory showed structural jeopardy with stakeholder alignment likely broken down.

Layer 4 produced a six-option COA matrix. The recommended sequencing was: Dependency Clearing (primary target: DATA_READINESS risk at 57.2, the dominant risk category) within 24 hours; Scope Freeze in parallel to arrest compounding; Cadence Tighten to ensure real-time tracking of intervention effects. The escalation posture was set at 5 of 6 (Immediate Leadership) — not yet 6 (Decision Gate Required) because the recovery window remained open, but requiring leadership decisions within the current week.

Layer 5 generated a complete executive briefing document. The status banner used the STATUS_KEY JEOPARDY_RECOVERABLE, with tone instructions emphasizing the recovery window rather than failure probability: 'The briefing MUST lead with this — the recovery window is open. Urgency framing: the window closes in 2–3 weeks, not that failure is inevitable.'

5.4 Comparison Against Known Outcomes: Illustrative Example

This illustrative example permits a direct comparison between what PRIMMS-GPT produced from the 8 August 2025 snapshot and what the conventional PMO monitoring showed at the same date for the same program. The conventional PMO reporting for that period classified the program as Amber on the standard RAG dashboard. No escalation was triggered, no formal recovery plan was activated at that point, and no specific recovery window was quantified.

PRIMMS-GPT, applied to the same data, produced: a decisive evidence classification (65.5 db); identification of DATA_READINESS as the primary blocker (risk score 57.2, more than twice the next-ranked risk); a specific recovery target (6% velocity improvement, from 0.54 to 0.57 deliverables per day); a projected finish date of 24 October 2025 against a planned end of 20 October 2025; and a recovery window of approximately 2–3 weeks before the gap became unrecoverable without scope or schedule change. The dominant failure archetypes were classified as GOVERNANCE_GAP and PREMATURE_CLOSURE — the latter reflecting that a replan was institutionally plausible at that point, while the velocity data indicated recovery was still achievable.

The known final outcome of the program is that it completed within two days of its original planned end date. Recovery was achieved through actions that addressed the data readiness blockage and introduced an accelerated governance cadence — the same intervention path that PRIMMS-GPT's Layer 4 COA matrix identified as the recommended sequencing. The velocity

model's projection proved accurate in retrospect: a further unremediated reporting cycle would have extended the gap to 9–12 days, consistent with the rate of deterioration observed in the project record, and would have placed recovery outside the original timeline.

The appropriate inference from this illustrative comparison is constrained in two respects. First, because PRIMMS-GPT did not run in real time during the program, we cannot claim that the system caused or contributed to the recovery. Second, this is one example drawn from the broader retrospective validation program; it is presented to demonstrate the architecture's reasoning process and output quality, not to constitute a statistically representative sample. What this example does establish, and what the broader validation program supports across multiple programs, is that the signal architecture correctly classifies known-critical snapshots; that the velocity-based recovery window calculation is consistent with ground truth; and that the Cortical Hierarchy's COA recommendations align with the interventions that produced successful outcomes. These are necessary but not sufficient conditions for prospective validity. The validation program supports proceeding to a prospective deployment study — the highest-priority item in the future work agenda.

6. Theoretical and Practical Contributions

6.1 Advancing the Hybrid Cognition Framework

The PRIMMS-GPT architecture makes a specific theoretical contribution to the emerging literature on human-AI collaboration in complex decision environments. It operationalizes a design principle — machine perception with human authority — at a level of architectural specificity that is absent from prior work. The division of labor between MATLAB (deterministic, auditable, inspectable signal computation) and LLM (natural-language interpretation, synthesis, projection) is not a pragmatic engineering choice but a principled allocation based on the comparative cognitive advantage of each.

The Cortical Hierarchy prompt architecture advances a claim about how LLMs should be integrated into professional decision support: not as autonomous reasoners generating advice, but as interpretive engines operating on structured, machine-verified signal sets within architectures that preserve human authority at every consequential step. This framing positions LLMs differently than both the automation literature (which treats them as replacements for human tasks)

and the productivity literature (which treats them as accelerants for existing tasks). The present paper argues for a third role: structural perception amplifiers that extend the human decision-maker's ability to detect what is actually happening before it becomes visible in conventional reporting.

The explicit treatment of Premature Closure as a failure archetype — the institutional default of closing a recoverable decision without testing the recovery hypothesis — represents an original contribution to project management failure classification. The Janis and Mann (1977) conflict theory framework is extended to the project risk context through the Layer 3 R5b prompt, which requires the LLM to explicitly assess whether the replan posture is itself a bias flag. This operationalizes the Janis-Mann defensive avoidance construct in a way that can be applied in real-time governance practice.

A practical implication of this architecture deserves explicit statement. As AI capabilities commoditize — as models converge, techniques diffuse, and capabilities that once distinguished organizations are absorbed into widely available platforms — the competitive boundary in AI-augmented management shifts. Advantage no longer resides in access to prediction. It resides in how prediction is governed: where authority is located, how signal evidence is structured before interpretation begins, and who remains accountable when outcomes are realized. The PRIMMS-GPT architecture is a response to precisely this condition. Its durable contribution is not the specific LLM model it employs — that is a commodity — but the governance architecture that wraps it: the deterministic signal layer, the hierarchical interpretation structure, the human authority boundary, and the governing equations that prevent complexity collapse. Organizations that mistake model access for governance architecture will find that the AI systems they adopt amplify their existing decision quality, whether good or poor. Those that invest in the architecture will find that the same underlying models produce systematically better orientation — because the structure, not the model, is the source of performance.

6.2 The Intelligence Gap Addressed

Aaron (2026, Chapter 1) identifies five structural gaps in conventional project management practice: team confidence erosion; perception distortion in leadership; failure archetype recognition; multi-modal evidence fusion; and recovery trajectory projection. Table 3 maps the PRIMMS-GPT architectural layers to these gaps.

Table 3. Mapping PRIMMS-GPT Architecture to Identified Intelligence Gaps

Intelligence Gap	PRIMMS-GPT Response	Architectural Layer
Team confidence erosion	VoT 1–6 time-series captures distributed team sentiment; temporal trend computed deterministically; VOT_MEAN signals fused into prior dB	Signal Acquisition; Layer 1 Feature Detection
Perception distortion in leadership	Divergence between VoT trend and leadership-reported schedule status surfaced as PERCEPTION_DISTORTION archetype; Layer 3 explicitly checks for optimism bias in recovery claims	Layer 2 Pattern Recognition; Layer 3 Situation Awareness
Failure archetype recognition	Six archetypes with characteristic signal signatures; deterministic archetype classifier in MATLAB; LLM interprets dominant and co-present archetypes in Layer 2	Inductive Classifier; Layer 2 Pattern Recognition
Multi-modal evidence fusion	Schedule, VoT, and documentary signals fused via Bayesian WoE summation into a single prior db; signal independence assumption acknowledged as limitation	Signal Extraction; Bayesian WoE Computation
Recovery trajectory projection	+2/+4/+8 week projections explicitly required in Layer 3; velocity-based recovery window computation in MATLAB; recovery signals as first-class inputs	Recovery Signal Architecture; Layer 3 Situation Awareness

6.3 Design Intent vs. Implementation Fidelity

Table 4 presents the alignment assessment between the original design intent described in the source monograph (Aaron, 2026, Chapter 8.7) and the current PRIMMS-GPT implementation. This assessment follows the transparency standard argued for elsewhere in this paper: the system's design gaps should be as visible as its demonstrated capabilities.

Table 4. Design Intent vs. Current Implementation Fidelity

Dimension	Original Design Intent	Current Implementation	Fidelity
Cybernetic control loop	Sensing → deviation → corrective signal via adaptive control agent	5-layer Cortical Hierarchy: MATLAB computes signals; LLM interprets, projects, recommends	High
MDP state/action space	Schedule state × VoT average × days behind → discrete intervention actions	Schedule + VoT + Documents → Bayesian	High

		WoE (db) + 6-option COA matrix with decision gates	
Failure archetype recognition	Six archetypes derived from multi-modal signal convergence	Six archetypes implemented with characteristic signal signatures and recovery implications	High
Human authority boundary	Recommend only; full audit trail; manager owns every decision	'Orient, never decide' structurally enforced in all prompts; audit trail present	High
RL learned reward policy	DDPG/Q-learning with reward updates from weekly survey responses over time	Deterministic db scoring (Bayesian WoE). No live RL adaptation loop across weeks	Partial
ML rundown curve classifier	Visual ML classifier trained on historical project images; detects jeopardy from curve geometry	Rundown tab computes velocity fit, R^2 , and gap as signals — no image-based ML classifier	Low

The overall fidelity assessment is 75–80%. The governance philosophy, state-space formulation, domain encoding, failure archetype classification, and human authority boundary are faithfully implemented — in some respects strengthened by the Cortical Hierarchy and Bayesian WoE framework relative to the original design. The two significant gaps (live RL adaptation and image-based rundown classifier) are noted as future development priorities rather than disqualifying limitations.

6.4 Self-Reference, Gödel, and the Structural Case Against Autonomous AI

The PRIMMS-GPT architecture's two-component design — a deterministic MATLAB layer for signal computation and an LLM layer for interpretation — is grounded in a practical engineering principle: tasks requiring auditability should not be delegated to probabilistic systems. However, the development of this architecture surfaced a deeper theoretical observation, one that connects the empirical behavior of large language models to foundational results in twentieth-century logic and computation.

During the development and testing of the Cortical Hierarchy prompt architecture, a consistent pattern emerged: attempts to instruct the LLM to constrain its own output — to remain concise while simultaneously performing rich analytical reasoning — failed systematically. Brevity constraints embedded in a prompt did not function as external governing rules. They functioned as one competing signal among many, processed through the same inductive mechanism that

produced the analysis itself. The richer the analytical task, the more reliably the brevity constraint degraded. The system could not enforce a bound on its own behavior using the same faculty that generated the behavior.

This empirical observation has a formal analogue. Kurt Gödel's incompleteness theorems (1931) demonstrated that any sufficiently expressive formal system must be either incomplete or inconsistent: there exist true statements within the system that cannot be proven from within it, and the system cannot demonstrate its own consistency using only its own axioms. Alan Turing extended this boundary into computation, proving that no general procedure exists to determine whether an arbitrary program will halt — a computational system cannot universally predict its own behavior (Turing, 1936). Both results describe the same structural limit: self-referential systems that are sufficiently expressive cannot fully know, predict, or regulate themselves from within.

Large language models represent a modern instantiation of this principle. The instructions provided to an LLM and the outputs it produces exist within the same representational substrate: both are sequences of tokens processed through the same inductive mechanism. A constraint instruction is not an external control; it is an input processed by the same system it is intended to govern. This is precisely the structure that Gödel's and Turing's results identify as fundamentally limited. The LLM cannot step outside its own generative process to enforce a rule upon that process — for the same structural reason that a formal system cannot prove its own consistency from within its own axioms.

The implication for the governance debate surrounding artificial intelligence is significant. Much of the discourse around autonomous AI — both the optimistic strand arguing that sufficiently capable systems will self-correct and self-regulate, and the pessimistic strand warning of uncontrollable self-directed AI — shares a common premise: that intelligence, once sufficiently scaled, becomes self-governing. The Gödel-Turing analysis suggests this premise is structurally false. A system powerful enough to exhibit the kind of general reasoning that would make it a candidate for autonomy is, by that same expressive power, too complex to reliably constrain itself using its own mechanisms. The constraint and the thing being constrained are the same system — and that structure is precisely what the halting problem and the incompleteness theorems show cannot self-regulate under general conditions.

The failure mode this analysis predicts is not domination but instability. A system that expands generative capability without a corresponding external constraint layer does not become sovereign; it becomes unpredictable. Its outputs remain coherent and often persuasive, but not reliably bounded. This is consistent with observed LLM behavior: systems that produce sophisticated reasoning under one framing produce contradictory reasoning under another, without awareness of the inconsistency. The system cannot self-adjudicate because the adjudication mechanism is not separate from the generative mechanism.

This analysis provides theoretical grounding for the PRIMMS-GPT architectural choice that might otherwise appear as mere engineering conservatism. The separation between the MATLAB signal layer and the LLM interpretation layer is not simply a preference for auditability. It is a structural response to the Gödel-Turing constraint: deterministic constraint enforcement must originate outside the inductive system. The MATLAB component imposes bounds that the LLM layer cannot override — not because it is more intelligent, but because it is external. The human authority boundary enforced at every decision point serves the same structural function: it provides the external constraint layer that inductive systems cannot supply for themselves.

The argument advanced here is developed at length in Aaron (2026, Section 3.10), which traces the historical arc from Hilbert’s program through Gödel’s incompleteness theorems and Turing’s halting problem to the structural constraints on contemporary large language models. The present paper makes a more specific and practically grounded claim: the two-layer PRIMMS-GPT architecture — external deterministic constraint plus internal inductive interpretation — is not merely a convenient engineering design. It is the theoretically necessary form of any hybrid cognition system that seeks both the interpretive richness of large language models and the governance reliability that autonomous AI systems cannot, by their structural nature, supply for themselves.

6.5 Command, Control, and the Irreducible Human Boundary

The Gödel-Turing constraint establishes a theoretical limit on machine self-regulation. A complementary and independently derived boundary emerges from the operational theory of command and control developed by John Boyd and formalized in military planning doctrine. Boyd distinguished sharply between command and control: command establishes intent and commitment; control maintains orientation under uncertainty. In Boyd’s formulation, effective

systems depend on trust, shared understanding, and freedom within bounds — control that becomes too explicit turns into interference, while command delegated to a machine destroys initiative. Effective command must be explicit; effective control must often be invisible.

This distinction maps directly onto the PRIMMS-GPT architecture. The system strengthens control — orientation, signal detection, pattern recognition, trajectory projection — while leaving command entirely untouched. The project manager who reads the Layer 4 COA matrix and selects an intervention is exercising command: establishing intent, assuming consequence, and committing to a course of action that cannot be undone. The machine has oriented. The human commands. Attempting to automate command would violate the structural conditions required for coordinated institutional action — not because machines lack processing power, but because command carries accountability that is conferred socially and institutionally, not computationally.

The RAND Corporation's research on command concepts (Builder, Bankes, & Nordin, 1999) reinforces this boundary from an empirical direction. Analyzing command effectiveness across military operations, RAND found that system performance depends less on the volume or speed of information and more on the intellectual quality of the commander's concept of operations. Information systems cannot rescue a flawed command concept; superior data cannot substitute for intellectual clarity. Conversely, a clear concept enables coordinated decentralized action even under degraded information conditions. The implication for AI-augmented project management is direct: PRIMMS-GPT can improve the quality of information available to the project leader, but it cannot generate the transformation concept, nor can it absorb contractual, political, or reputational consequences when outcomes are realized. These remain irreducibly human functions — not as a temporary accommodation to technological limits, but as a permanent structural feature of any institutional system that must remain accountable.

6.6 The Zero Constraint: Complexity Collapse and the Role of Governing Equations

The Gödel-Turing constraint identifies one fundamental limit on machine-assisted cognition: a sufficiently expressive inductive system cannot fully regulate itself from within. During the development of the PRIMMS-GPT Cortical Hierarchy, a second and distinct constraint emerged from practice. It is not a constraint of logical incompleteness but of operational collapse — and it has a precise cultural analogue in the 1975 science fiction film *Rollerball*.

In that film, the supercomputer Zero is asked to retrieve historical information. It cannot. The failure is not attributed to insufficient data or processing power. The cause is accumulation: Zero has been given so much information across so many domains that retrieval has become indeterminate. When pressed for a specific answer, human users of Zero complain that “everything is ambiguous now.” The machine has not failed to reason. It has accumulated itself into uselessness. It considers everything — and precisely because it considers everything, it can say nothing actionable.

This failure mode is not fictional. It is a live risk in any machine-assisted project management system that adds signals, archetypes, prompt layers, and contextual blocks without a corresponding mechanism to reduce the output space. Each individually justified addition to the analytical architecture increases the probability that the LLM output will resolve to a hedge: a qualified, multi-conditional statement that acknowledges every signal and commits to nothing. The project manager reads three paragraphs and finds no single actionable conclusion. The orientation system has produced disorientation.

The Gödel-Turing constraint and the Zero constraint are related but distinct. Gödel-Turing concerns the epistemic ceiling — what the machine can in principle know about its own outputs. Zero concerns the operational floor — the minimum condition for an output to remain useful. A system can satisfy the Gödel-Turing constraint (by providing an external deterministic signal layer) and still fail the Zero constraint by producing outputs too complex for a human decision-maker to act on. Both constraints must be satisfied simultaneously. The human occupies the space between: as the external validator required by Gödel-Turing, and as the necessary simplifier required by Zero.

The PRIMMS-GPT architecture addresses the Zero constraint through a specific mechanism: governing equations. These are formal mathematical structures — derived from the Bayesian, microeconomic, and signal-processing frameworks that constitute the system’s theoretical foundation — that impose hard boundaries on what the LLM layer is permitted to treat as ambiguous. They function as axioms: propositions the system treats as settled, not as inputs for further deliberation.

Three governing structures are central. First, the Bayesian weight-of-evidence accumulator: signals are converted to decibans and summed. The Jeffreys classification bands — Barely (0–5 dB), Substantial (5–10), Strong (10–15), Very Strong (15–20), Decisive (>20) — translate a

potentially infinite evidence space into five ordinal categories. The LLM does not re-derive the evidence strength; it receives a computed value and a named band, and its interpretive task begins from that anchored point. Second, the project production function established in Aaron (2015): the microeconomic relationship $Y = f(AC, IC, IO)$, expressing project output as a function of activities completed, issues closed, and issues opened, provides a formal constraint on how scope, schedule, quality, and issue dynamics interact. This equation governs the PRIMMS-GPT issue management layer: the Bayesian issue closure ratio $\theta = \text{closed} / \text{total}$ is not an informal indicator but a formal metric derived from the production function, and is its operational expression in the deployed system — the form in which the production function’s theoretical constraint is actually computed and enforced at each session. θ carries an established readiness threshold ($\theta \geq 0.70$ for even odds; $\theta \geq 0.95$ for phase exit) that the LLM is instructed to treat as a pre-condition, not a deliberative question. Third, the velocity-based rundown projection: the linear regression of actual delivery velocity against the planned rundown produces a deterministic projected finish date. The LLM does not estimate when the project will finish; it receives a computed date and a computed slip in days, and its narrative task is to interpret that number, not to derive it.

The governing equations matter for a reason that goes beyond computational efficiency. They define the boundary between what is settled and what requires judgment. Without that boundary, the LLM treats every dimension of the problem as open for deliberation — and in a sufficiently complex project state, every dimension has genuine uncertainty, which means every statement can be qualified, and Zero is approached. With the governing equations in place, the LLM receives a pre-classified evidence state: a dB total and Jeffreys band, a θ ratio and readiness assessment, a projected finish date and recoverability label. Its task shifts from “assess the evidence” to “interpret a classified state and identify the one action that addresses the dominant risk.” That is a task the LLM can perform with specificity. The hedge-forest is replaced by a named archetype, a recovery posture, and a single owned action.

This is the operational meaning of hybrid cognition in PRIMMS-GPT. The machine does not consider everything. It considers what the governing equations leave open after they have been applied — which is the subset of the problem space that genuinely requires natural language reasoning, pattern recognition, and narrative synthesis. The MATLAB layer is not merely an input source for the LLM; it is a complexity reducer. It closes the questions that have formal answers so that the LLM can apply its interpretive capability to the questions that do not.

The practical expression of this principle in PRIMMS-GPT’s issue management integration illustrates the mechanism concretely. When the system detects open issues older than two weeks, it does not ask the LLM to assess issue severity and derive a priority. Instead, the governing equation θ has already computed the closure ratio; the stale issue count and age are deterministic facts; and the Cortical Hierarchy receives a pre-condition instruction that reduces to a single mandate: G1, the most important action this week, must directly address the oldest stale issue, stated in the form “Owner does X by date, measurable by Z.” If open issues reach ten or more, a Zero constraint flag fires: one of the three executive decisions must address issue volume control, because the governing equation registers that the issue count has crossed the threshold where complexity begins to impair the team’s decision capacity. The system detects its own approach toward Zero and names it — which is the one thing Zero itself could not do.

This self-monitoring property — the capacity to detect and report that issue accumulation has reached a decision-impairing threshold — represents a qualitatively different function from conventional project risk monitoring. The system is not merely reporting that issues exist. It is applying a formal threshold derived from the production function to assess whether the issue load has crossed from a tractable project management challenge into a structural decision-impairing condition. That assessment is deterministic, not narrative. It happens before the LLM is consulted, and it shapes what the LLM is permitted to say. Zero is avoided not by limiting what the system knows, but by structuring what it is required to conclude.

The broader implication for hybrid cognition system design is that governing equations are not merely computational conveniences. They are the mechanism by which complexity is made tractable without being made false. The project environment does not become simple when the governing equations are applied; it remains as complex as it is. But the portion of that complexity assigned to machine deliberation shrinks to what formal structures cannot resolve, and the portion assigned to human judgment expands correspondingly — which is exactly what the Janis-Mann vigilant decision-making framework requires. The human decision-maker does not need the formal system to be complete. They need it to be bounded. Governing equations provide that bound. Zero is the consequence of their absence.

7. Limitations and Future Work

7.1 Current Limitations

The PRIMMS-GPT architecture has several limitations that should be acknowledged explicitly.

Signal independence assumption. The Bayesian WoE summation assumes that individual signals carry independent evidence about the risk hypothesis. In practice, schedule signals are correlated: a project with high days-behind will also typically have a high jeopardy index and a worsening VoT trend. The independence assumption therefore inflates the prior db relative to what a fully specified joint probability model would produce. The practical consequence is that the absolute db value should be interpreted as a relative evidence accumulator rather than a precisely calibrated posterior probability. The Jeffreys band classification, which uses ordinal thresholds rather than continuous probability, partially mitigates this concern.

Retrospective design validation. The validation described in Section 5 is retrospective — PRIMMS-GPT was applied to completed project data after the fact, not deployed in real time during execution. This means the study can establish that the signal architecture correctly classifies a known-critical snapshot and that the recovery window calculation is consistent with ground truth, but it cannot establish that the system would have changed decisions had it been running in real time. A prospective deployment study — the highest-priority item in the future work agenda — is required before causal claims can be made.

LLM non-determinism. The MATLAB signal layer produces fully deterministic, reproducible outputs. The Cortical Hierarchy LLM outputs do not — the same prompt will produce different natural-language outputs across sessions. The audit trail records the full response text for each session, enabling governance reviewers to assess consistency. However, this non-determinism means that the LLM layer cannot be formally validated in the way that deterministic algorithms can. The architecture's response is to confine non-deterministic processing to the interpretation and synthesis layers while keeping all signal computation deterministic.

Missing RL adaptation. The absence of a live reinforcement learning loop means that signal weights cannot adapt to the specific organizational context over time. A project environment with distinctive risk characteristics — unusually high scope volatility, or unusually strong governance — cannot be captured in the fixed WoE weights. This is the most significant gap versus original design intent and the highest-priority item for future development.

Differential validation depth across architectural layers. The validation evidence reported in Section 5 encompasses the full PRIMMS-GPT architecture, but the individual layers carry materially different levels of empirical support. The Bayesian WoE core — the VoT mechanism, the risk category benchmarks, and the signal threshold calibrations — has been tested prospectively across complete project lifecycles in operational deployment since 2008. The 20 risk categories identified through text mining of project risk logs, and the subset found statistically predictive of project schedule performance, represent a substantial empirical base accumulated over more than seventeen years of production use. The semantic document analysis layer and the Cortical Hierarchy prompt architecture, by contrast, have been validated retrospectively against completed project phases rather than prospectively across full project lifecycles. Their recommendations have been assessed as coherent and appropriate in retrospective review, but the evidence base is thinner. This layered validation status is an honest characterization of where the research program stands: the perceptual foundation is mature; the interpretive superstructure, while architecturally sound and retrospectively validated, requires prospective deployment to establish the same depth of empirical grounding as the Bayesian core.

Architectural complexity and verification scope. The signal interpretation architecture of PRIMMS-GPT operates across two dimensions: a set of multi-modal input signals computed by MATLAB across the schedule, stakeholder, contractual, and team domains, and a five-layer Cortical Hierarchy through which those signals are sequentially processed — from raw feature detection, through gestalt pattern recognition and temporal situation awareness, to executive planning and briefing document generation. Each layer operates at a higher level of abstraction than its predecessor, and each layer's output propagates forward to condition the next. A further constraint on the state space is provided by the risk ontology embedded in the MATLAB application: a library of 47 practitioner-validated risk categories — covering domains from data readiness and scope volatility to AI model quality, vendor risk, and governance framework gaps — developed from thirty years of program delivery experience across IT and AI/automation projects. This pre-classified ontology means that the LLM does not face an open-ended categorization task at each session; the combinatorial space of risk interpretations is bounded in advance by a structured, experience-grounded taxonomy. This directly mitigates the Zero problem described in Section 6.5: the LLM receives a named, bounded risk category as input, not an undifferentiated corpus from which it must independently induce structure. Verifying that the

system's recommendations are appropriate across the full combinatorial space of signal inputs and layer interactions is a substantial challenge: the number of distinct contextual situations is large, and retrospective case validation samples only a small fraction of them. This architectural complexity is a deliberate design choice, grounded in the principle that no single prompt can simultaneously perform reliable feature validation, pattern recognition, temporal projection, and executive planning — and in the neuroscientific evidence that hierarchical, multi-level signal integration is necessary for accurate situational perception in complex environments. However, it means that the verification effort required to establish the system's reliability across its full operating range is correspondingly large. Systematic coverage testing — in which specific combinations of signal inputs are constructed and the resulting layer-by-layer outputs evaluated for coherence and correctness — is an explicit component of the future validation program.

The raw-upload alternative. A question that any intellectually honest evaluation of PRIMMS-GPT must address directly is whether the architecture is necessary at all. A project manager or sponsor who has access to a capable general-purpose LLM — such as Claude (Anthropic) or GPT-4o (OpenAI) — can upload a schedule export, a stack of status reports, and a risk log, and receive a coherent synthesis. The LLM will surface alarming language, identify themes, and answer follow-up questions. The marginal cost of this approach is near zero, and no application infrastructure is required. The question is legitimate and deserves a direct answer rather than dismissal.

The raw-upload approach has genuine value and should not be caricatured. An experienced project manager who understands failure archetypes, frames the right questions, and possesses sufficient self-awareness to resist optimism bias will extract real analytical utility from an unstructured upload. For ad hoc diagnostic queries — a one-time situation assessment, a quick second opinion on a recovery hypothesis — the approach is appropriate and efficient. The claim of this paper is not that raw-upload analysis is useless. It is that raw-upload analysis fails to satisfy two architectural constraints that become consequential precisely in the high-stakes, time-pressured governance situations PRIMMS-GPT is designed to address.

The first constraint is the Zero constraint described in Section 6.5. When a raw document corpus is uploaded without a pre-classification layer, the LLM is asked to perform two tasks simultaneously: determine what the evidence says and interpret what the evidence means. In a

project environment with genuine complexity — multiple schedule signals pulling in different directions, VoT sentiment diverging from documentary tone, recovery plan language present but unverified against schedule facts — the evidence space is large enough that every signal carries real uncertainty. Absent governing equations that close the questions with formal answers, the LLM tends toward hedge-forest outputs: multi-conditional paragraphs that acknowledge every tension and commit to nothing. The project manager reads three pages and finds no single owned action. The analytical tool has produced disorientation rather than orientation. PRIMMS-GPT's MATLAB layer addresses this by computing the db total, classifying it against the Jeffreys bands, applying the production function_ratio, and projecting the velocity-based finish date before the LLM is consulted. The LLM receives a pre-classified state and is directed to interpret a named archetype and identify a single dominant recovery action. That task produces specificity. The raw-upload approach has no equivalent mechanism for forcing this reduction.

The second constraint is auditability. A MATLAB-computed db signal total is a number with a fully traceable derivation: each signal contribution is fixed, inspectable, and reproducible. If a governance reviewer asks why the system classified the project at 65.5 dB decisive jeopardy, the arithmetic is available. If a sponsor challenges the recovery window calculation, the velocity regression parameters are in the audit trail. When Claude or any other LLM derives the same conclusion from a raw upload, the reasoning is a natural-language narrative that cannot be reconstructed, formally reviewed, or independently verified. For project governance contexts — where decisions about resource reallocation, executive escalation, and program termination carry material organizational consequences — the inability to audit the analytical basis is not a minor inconvenience. It is a structural governance gap. The ISO 21502 standard and PMI governance frameworks both require that risk classifications be traceable to their evidentiary basis. A raw-upload synthesis, however coherent, does not satisfy that requirement.

A third practical distinction concerns temporal continuity. Risk management is an inherently longitudinal problem: what matters is not the signal state at a single point in time but how that state has been changing — whether confidence is eroding or stabilizing, whether the schedule gap is widening or recovering, whether the db accumulation is accelerating toward a decisive classification or plateauing. PRIMMS-GPT maintains this longitudinal record as a structural property of the application. Each weekly session appends to the accumulated signal history: the

VoT time series of discrete confidence ratings, the progressive db trend across reporting periods, the velocity regression updated against each new schedule snapshot, and the Bayesian WoE accumulation itself, which derives its evidential force precisely from the compounding of independent signals across time. A project manager attempting to replicate this outside the application would face a substantial and error-prone manual burden: assembling multiple weeks of schedule exports, reconstructing the db computation from raw signal inputs, recalculating the velocity model from scratch, and presenting the resulting trend data to the LLM in a form that is coherent enough for pattern recognition across time. In practice, this burden means that ad hoc raw-upload analysis defaults to a snapshot assessment — the current state, evaluated in isolation. Snapshot assessment is precisely what the failure literature identifies as inadequate: the signals of structural deterioration are characteristically gradual, and their significance lies in their trajectory, not their instantaneous value. A project whose db total has moved from 8 dB to 22 dB over four weeks presents a qualitatively different governance situation from one that has held at 22 dB for a month, even though the current snapshot is identical. PRIMMS-GPT surfaces that distinction automatically. Raw-upload analysis, without heroic manual preparation by the user, does not.

Finally, raw-upload analysis is inherently prompt-dependent. Its quality is a function of the framing choices made by whoever conducts the upload: which documents are included, which questions are asked, which assumptions are made explicit. A skilled analyst who understands failure archetypes and knows what to look for will produce better results than a less experienced one. PRIMMS-GPT encodes that analytical expertise into a replicable, consistent process. The five-layer Cortical Hierarchy ensures that the same diagnostic questions are asked in the same order against the same pre-classified signal set, regardless of who operates the system. The value proposition is not that PRIMMS-GPT is analytically superior to an expert analyst with a raw upload; it is that it is consistently equivalent to one — and that it removes the dependency on the expertise and optimism-awareness of the individual operator at each session. In governance terms, consistency and reproducibility are not secondary virtues. They are the conditions under which institutional trust in an analytical system can be established and maintained.

7.2 Future Work

The following extensions are identified for subsequent development phases:

- Live RL adaptation layer: implementing a DDPG/Q-learning agent that updates signal weights from accumulated weekly VoT survey responses across multiple projects, enabling the WoE framework to adapt to organizational context over time.
- Image-based rundown curve classifier: training a visual ML classifier on historical project rundown curve images to detect jeopardy signatures from curve geometry — the original design intent described in Aaron (2026, §8.7.3).
- Automated notification dispatch: extending the current human-delivery model to enable push notifications keyed to WBS Level 2 deliverable status changes.
- Portfolio-level deployment: extending the single-project architecture to portfolio monitoring, enabling pattern-level comparisons across projects and portfolio-level archetype profiling.
- Prospective deployment study: the highest-priority next step. A pre-registered study deploying PRIMMS-GPT in real time on active projects, comparing governance decisions and outcomes against matched controls. The retrospective validation program reported in this paper establishes the signal architecture’s accuracy against ground truth across multiple completed programs; prospective deployment is required to establish that real-time use changes governance decisions.
- Calibration study: comparing PRIMMS-GPT db totals against eventual project outcomes across a retrospective dataset to assess the calibration of the WoE scoring system and refine signal weights.
- Mundell comparative statics formalization: the microeconomic production function framework established in Aaron (2015) — relating project output to scope, time, cost, quality, and issue dynamics — creates a tractable basis for Mundell-style comparative statics analysis of PRIMMS-GPT intervention options. Each COA option in the Layer 4 matrix can be analyzed as a shift in one parameter (e.g., resource input / force u) holding others constant, with the db signal set providing the empirical basis for estimating the relevant partial derivatives. Formalizing this connection would ground the COA recommendations in the same microeconomic framework that governs the project production function, enabling optimization of intervention sequencing under budget and time constraints.

- Likelihood-ratio ML layer for “what-if” COA prediction: the Bayesian WoE foundation that underpins PRIMMS-GPT is mathematically compatible with a broader family of modern machine learning classifiers whose inference also reduces to likelihood ratios — including regularized logistic regression, gradient boosting, and naïve Bayes models. A natural extension of the current architecture is to train such classifiers on the accumulated historical project record to produce “what-if” conditional predictions: given that the project manager takes Course of Action X at the current signal state, what is the posterior probability distribution of outcomes at +2, +4, and +8 weeks? This would transform the current COA matrix from a qualitatively reasoned set of options into a quantitatively ranked decision surface grounded in empirical project outcome data, while preserving the interpretable likelihood-ratio structure that makes the WoE framework auditable.
- Expanded multi-modal input architecture: the simulation work reported in Aaron (2019) demonstrated that adding successive input modalities — image, audio, and text — to a machine learning system progressively reduced classification error from 27% (image only) to 18% (image plus audio) to 0% (all three modalities combined) on a real-time operational decision task. The implication for PRIMMS-GPT is that the current three-modality architecture (schedule, VoT, documents) may itself be extendable: structured data feeds from financial systems, audio or video sentiment analysis from team meetings, and integration with collaboration platform message streams represent candidate additional modalities that could further increase the system’s predictive precision. The Aaron (2019) multi-modal framework provides the experimental methodology for evaluating the marginal accuracy contribution of each additional source before production integration.
- Generalization to broader organizational intelligence applications: the hybrid cognition architecture described in this paper — multi-modal signal fusion, Bayesian WoE classification, cortical hierarchy interpretation, and human authority boundary — is not inherently specific to project management. The same design principles apply wherever an organization requires early structural perception of deteriorating conditions that do not yet appear in formal reporting: supply chain resilience monitoring, regulatory compliance posture, M&A integration health, and executive talent risk are candidate domains. The author intends to develop specialty applications of the intelligence engine architecture for

organizational decision contexts beyond project management, building on the PRIMMS-GPT foundation established here.

8. Conclusion

This paper has argued that the most consequential application of AI in project management is not task automation but structural perception — the capacity to detect what is actually happening inside a project before it becomes visible in delivery output. Conventional project management tools, however sophisticated, are record-keeping systems. They capture what has happened and display it. They do not infer the structural conditions — team confidence erosion, perception distortion, governance gap formation — that precede most project crises.

PRIMMS-GPT addresses this intelligence gap through a hybrid cognition architecture that couples deterministic Bayesian weight-of-evidence signal computation with a five-layer LLM interpretation hierarchy, classifies convergent signal patterns against six empirically grounded failure archetypes, produces recovery trajectory projections at +2/+4/+8-week horizons, and generates sponsor-ready governance documents — while preserving an inviolable human authority boundary at every decision point. The architecture is designed to satisfy two foundational constraints simultaneously: the Gödel-Turing constraint, which requires that deterministic bounds on machine output originate outside the inductive system, and the Zero constraint, which requires that complexity accumulation not collapse the output into indeterminacy. The governing equations of the system — the Bayesian WoE accumulator, the project production function relating output to scope, issue closure, and schedule adherence, and the Bayesian issue closure ratio θ — are the mechanism by which both constraints are satisfied. They close the questions that have formal answers, so that the LLM is directed only at the questions that do not.

The architecture makes three claims that are advanced beyond the level of prior work. First, it operationalizes the hybrid cognition principle at a level of architectural specificity sufficient for replication and validation. Second, it introduces Premature Closure as a first-class failure archetype and the recovery window as a first-class governance metric — the insight that a recoverable situation prematurely closed as a failure is itself a failure mode with characteristic signals and a specific remedy. Third, it demonstrates through a retrospective validation that the signal architecture correctly classifies a known-critical project snapshot, that the velocity-based

recovery window calculation is consistent with known outcomes, and that the Cortical Hierarchy produces governance documents whose COA recommendations align with the interventions that produced successful project recovery. These are necessary preconditions for prospective deployment — not yet sufficient to establish real-time governance value, but sufficient to justify the controlled prospective study that is the next step in this research program.

The project management field has done what its current paradigm allows. The persistent failure rate after decades of tool and methodology investment suggests that the paradigm itself requires extension. PRIMMS-GPT represents one instantiation of what that extension might look like: not more sophisticated scheduling, not more comprehensive risk registers, not faster reporting — but a structured intelligence layer that extends the project manager’s perceptual range across the signal modalities that matter, and routes those signals through an interpretive architecture that is both powerful and accountable. The retrospective validation program reported here establishes that the architecture produces correct signal classifications against ground truth across multiple completed programs, with one illustrative example presented in detail. Establishing that it changes governance decisions in real time is the work that remains.

References

- Aaron, J. M. (2015). A Bayesian methodology for project issue management: A microeconomic and data analytic perspective. Milestone Planning and Research, Inc. Working paper.
- Aaron, J. M. (2019). Designing and deploying systems of intelligence into business process applications: Multi-modal machine learning for operational decision support. Milestone Planning and Research, Inc. Internal technical report.
- Aaron, J. M. (2026). The Inductive Enterprise: Machine-Assisted Perception, Hybrid Cognition, and the Architecture of Organizational Intelligence. Unpublished manuscript. Milestone Planning and Research, Inc.
- Aaron, J. M. (2017). Identifying emergent project risks and their performance impacts through text mining: A mixed methods study. Unpublished manuscript. Milestone Planning and Research, Inc.
- Bloch, M., Blumberg, S., & Laartz, J. (2012). Delivering large-scale IT projects on time, on budget, and on value. *McKinsey Quarterly*.
- Builder, C. H., Bankes, S. C., & Nordin, R. (1999). Command concepts: A theory derived from the practice of command and control. RAND Corporation.
- Budzier, A., & Flyvbjerg, B. (2011). Double whammy: How ICT projects are fooled by randomness and why they overshoot on time and budget. *Oxford Working Papers in Information Management*.
- Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 32–64.
- Flyvbjerg, B. (2017). *The Oxford Handbook of Megaproject Management*. Oxford University Press.
- Friston, K. (2010). The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138.
- Fuster, J. M. (2008). *The Prefrontal Cortex* (4th ed.). Academic Press.
- Glimcher, P. W. (2011). *Foundations of Neuroeconomic Analysis*. Oxford University Press.

Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I [On formally undecidable propositions of Principia Mathematica and related systems I]. *Monatshefte für Mathematik und Physik*, 38, 173–198.

Good, I. J. (1950). *Probability and the Weighing of Evidence*. Griffin.

Good, I. J. (1985). Weight of evidence: A brief survey. *Bayesian Statistics*, 2, 249–270.

Ika, L. A., Love, P. E. D., & Pinto, J. K. (2020). Moving beyond the planning fallacy: The emergence of a new principle of project behavior. *IEEE Transactions on Engineering Management*, 67(3), 640–649.

Janis, I. L., & Mann, L. (1977). *Decision Making: A Psychological Analysis of Conflict, Choice, and Commitment*. Free Press.

Jeffreys, H. (1961). *Theory of Probability* (3rd ed.). Oxford University Press.

Kahneman, D. (2011). *Thinking, Fast and Slow*. Farrar, Straus and Giroux.

Kanerva, P. (1988). *Sparse Distributed Memory*. MIT Press.

Kock, A., Gemünden, H. G., & Salter, A. (2021). An empirically grounded model of innovation project portfolio management: Understanding the role of learning and strategic alignment. *International Journal of Project Management*, 39(5), 586–599.

Kutsch, E., & Hall, M. (2010). Deliberate ignorance in project risk management. *International Journal of Project Management*, 28(3), 245–255.

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Maylor, H., Vidgen, R., & Carver, S. (2008). Managerial complexity in project-based operations: A grounded model and its implications for practice. *Project Management Journal*, 39(S1), S15–S26.

Niazi, M., Zhang, X., & Bhatt, U. (2023). A systematic literature review on artificial intelligence in project management. *International Journal of Project Management*, 41(7), 102522.

Pellerin, R., & Perrier, N. (2019). A review of methods, techniques and tools for project planning and control. *International Journal of Production Research*, 57(7), 2160–2178.

Project Management Institute. (2023). *Pulse of the Profession*. Project Management Institute.

Standish Group. (2023). *CHAOS Report 2023: Decision Latency Theory*. Standish Group International.

Williams, T. (2005). Assessing and moving on from the dominant project management discourse in the light of project overruns. *IEEE Transactions on Engineering Management*, 52(4), 497–508.

Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(1), 230–265.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433–460.

Author Note

John M. Aaron is the founder of Milestone Planning and Research, Inc., and the designer of the PRIMMS® platform. The research reported in this paper was conducted as part of the development of PRIMMS-GPT, an extension of the PRIMMS® platform. The author declares a commercial interest in PRIMMS-GPT. The retrospective validation described in Section 5 was conducted on project data from completed programs in which the author was involved; the illustrative example presented in Sections 5.2–5.4 is drawn from one such program. All project-identifiable data has been anonymized. No external research funding was received for this work.

Correspondence concerning this article should be addressed to John M. Aaron, Milestone Planning and Research, Inc. Email: john.aaron@milestoneplanning.net

Acknowledgements. The architecture described in this paper draws on the theoretical foundations of Bayesian inference, computational neuroscience, and decision theory as synthesized in *The Inductive Enterprise* (Aaron, 2026). The retrospective validation benefited from access to complete project records from multiple concluded enterprise programs.