

The EARBr Framework: A Decision-Theoretic Model for AI-Driven Process Optimization and Scalable System Design in Enterprise Product Management

Mazdul Hasan Choudhury

Independent Researcher, USA

Abstract: Enterprise product managers face a sequencing problem that most AI governance frameworks ignore. It is not that organizations lack AI capability - it is that they apply it in the wrong order, to the wrong processes, at the wrong level of investment. Teams are rewarded for shipping new AI systems; they are not rewarded for eliminating the processes those systems would have automated, or for reusing the AI infrastructure already sitting in another team's codebase. The result is an enterprise AI landscape that is sophisticated in individual components and redundant, fragmented, and expensive to maintain in aggregate. The EARBr framework - Eliminate, Automate, Reuse, Build-to-Reuse - provides a four-layer decision model for correcting this. Its central governance rule is precedence: advance to the next layer only after genuinely exhausting the prior one. The framework's contribution is not the four categories themselves, which individually are familiar to most practitioners, but the enforced precedence between them: a team may not automate a process it has not first tested for elimination, and may not build new infrastructure for a capability it has not first tried to reuse. This ordering constraint is what existing AI governance guidance omits, and its absence is what allows redundant automation to accumulate at enterprise scale. Three practitioner cases from large-scale retail demonstrate the framework in action, with documented outcomes of approximately 90% efficiency gain (Layer 1), 80% onboarding time reduction across 250,000 annual users (Layer 2), and a modular platform serving four operational domains from a single Build-to-Reuse investment (Layers 3-4). The paper also outlines the organizational adoption conditions - incentive redesign, cross-team visibility, and governance checkpoints - required for the precedence rule to hold under delivery pressure. The intended audience is enterprise product managers, AI governance leads, and platform engineering leaders allocating finite capacity across competing automation demands.

Keywords: EARBr Framework, Enterprise Product Management, AI Decision Framework, Process Automation, Software Reuse, Technical Debt, Scalable System Design, Automation Governance, Build-to-Reuse

1. Introduction

The real problem in enterprise AI adoption is not a capability problem. Organizations have access to capable models, mature platforms, and experienced engineering teams. The problem is sequencing. Every organization has finite engineering capacity and an essentially unlimited list of operational problems that could theoretically benefit from some form of AI or automation. The organizations that generate the best return on AI investment are not the ones with the most sophisticated models - they are the ones with the clearest judgment about which processes to touch, in what order, and at what level of investment. That judgment is currently exercised informally, inconsistently, and under organizational incentive pressures that systematically distort it toward new development regardless of whether new development is actually the right answer [1].

The distortions are structural and familiar to anyone who has worked in enterprise product management. Teams are rewarded for shipping new systems, not for eliminating processes those systems would have automated. Reuse of



existing systems requires cross-team coordination that organizational structures rarely support. Process elimination is perceived as subtraction - a smaller scope - rather than as the highest-leverage intervention available, which is what it often is. And automation is routinely applied to processes that should have been redesigned or discontinued first, locking operational inefficiencies into maintained AI code [2].

The EARBr framework provides a structured answer to the sequencing question. Four principal contributions: (1) a decision taxonomy with four sequential layers - Eliminate, Automate, Reuse, Build-to-Reuse - and explicit precedence relationships; (2) specification of when and how AI and automation are appropriate at each layer; (3) three practitioner case illustrations with quantified outcomes; and (4) an organizational adoption model identifying the institutional changes required for durable EARBr implementation at enterprise scale.

The paper proceeds as follows: Section II establishes the theoretical context; Section III introduces the four EARBr layers; Section IV presents practitioner cases; Section V addresses organizational adoption; Section VI presents evaluation and outcomes; Section VII concludes.

2. Theoretical Context

Three research streams ground EARBr. Enterprise AI governance literature establishes that sequencing quality - the degree to which AI is applied to the right processes in the right order - is a primary determinant of AI return on investment. Organizations without structured governance consistently report lower-than-expected AI returns driven by misaligned sequencing rather than model inadequacy. By 2027, Gartner projects that half of all business decisions will be AI-augmented - which makes the governance of AI adoption sequencing an increasingly consequential organizational capability [3], [4].

Technical debt literature provides the second anchor. Moreschini et al. [2] document that data-related, infrastructure-related, and pipeline-related technical debt are particularly prevalent in ML systems, and that ML Operations (MLOps) practices are essential for managing debt accumulation in AI-enabled environments. The EARBr framework's emphasis on eliminating and simplifying before automating is directly responsive: automating a poorly designed process encodes it in maintained AI code, where it is more expensive to resolve and more damaging to operational reliability. Tashkinov [5] demonstrates that lean manufacturing principles applied systematically alongside AI implementation address this problem, eliminating process waste before digital interventions are applied.

Software reuse literature provides the third strand. Make/Buy/Reuse decision frameworks in software engineering [6] demonstrate that systematic evaluation of reuse options before new development reduces development cost, improves cross-system consistency, and prevents proliferation of functionally overlapping tools. EARBr extends this principle from software product line engineering to the enterprise AI investment domain, establishing reuse evaluation as a mandatory governance step before any Build-to-Reuse commitment.

Table 1. Theoretical foundations of the EARBr framework and their contribution to enterprise AI governance

Theoretical Stream	Core Contribution to EARBr	Key Evidence
Enterprise AI Governance	Sequencing quality determines AI ROI; organizations without governance consistently underperform	By 2027, 50% of business decisions AI-augmented; unstructured AI investment generates lower returns [3],[4]
Technical Debt in AI Systems	Premature automation encodes process inefficiency in maintained AI code - more expensive to resolve than in original process	Data, infrastructure, pipeline TD prevalent in ML; MLOps essential for debt management [2]

Lean + AI Digital Management	Process elimination before automation improves both AI investment return and operational quality	Lean+AI integration framework demonstrates prerequisite relationship [5]
Software Reuse and Product Lines	Systematic reuse evaluation before new development reduces cost and prevents functional overlap	Make/Buy/Reuse framework; reuse governance reduces TCO and delivery time [6]

3. The EARBr Framework: Four Sequential Decision Layers

A. Layer 1 - Eliminate: Challenge Whether the Process Should Exist

The Eliminate layer is the highest-leverage and most consistently underutilized evaluation in enterprise product decision-making. It asks a question that feels obvious but is organizationally difficult to ask in practice: should this process exist at all? Organizations accumulate legacy processes whose original justification has expired - because they were built for an earlier scale, an earlier competitive environment, or an earlier set of operational constraints that no longer apply. These processes persist not because they continue to deliver value proportionate to their cost, but because questioning their existence requires organizational permission that most product managers do not routinely exercise. Every downstream layer of EARBr - Automate, Reuse, Build-to-Reuse - invests organizational resources in perpetuating a process that should have been discontinued. Automating an unnecessary process produces a sophisticated, maintained AI system that delivers zero net organizational value while consuming engineering capacity and accumulating technical debt [1].

Three diagnostic questions at Layer 1: Does this process serve a current organizational objective - and is that objective still valid? Do the downstream consequences of eliminating the process exceed its own operational cost, or are those consequences artifacts of the process's existence? Has the process been evaluated for elimination in the current environment, or is it persisting by default? When elimination is viable, the efficiency gain is 100% of the process's operational cost, achieved immediately and without ongoing maintenance obligation. Partial elimination - removing redundant stages while preserving stages that still add value - is a frequently applicable variant.

The critical discipline of Layer 1 is not that all processes should be eliminated, but that all processes should be actively evaluated for elimination before any investment in their optimization or automation is made. Product managers who have done this evaluation explicitly are in a materially stronger analytical position when they proceed to Layer 2, because they are advancing on evidence rather than assumption.

B. Layer 2 - Automate: Reduce Manual Intervention in Validated Processes

When a process has been explicitly evaluated for elimination and retained, Layer 2 asks whether manual intervention can be reduced through targeted automation. The governance principle here is proportionality: scope the automation to the specific bottleneck, not the entire process. Full process automation is a Build-to-Reuse decision. Conflating Layer 2 with Layer 4 produces overengineered systems - costly to build and maintain, delivering no more operational value than a narrowly scoped Layer 2 intervention would have [5].

GenAI and AI-driven automation are appropriate at Layer 2 when the specific bottleneck involves classification, pattern recognition, NLP, or prediction tasks. An AI classifier that routes incoming user access requests by role taxonomy - auto-provisioning standard cases, escalating non-standard requests for human review - is proportionate Layer 2 automation. The AI component is scoped to the classification task; governance, escalation handling, and non-standard access management remain under human oversight. Partial automation - AI generates recommendation, human validates - is a valid Layer 2 intermediate state that builds data quality and organizational trust simultaneously.

Layer 2 also requires explicit evaluation of partial automation as a deployment option. In many enterprise contexts, full automation is technically feasible but organizationally premature - because data quality is insufficient

or organizational trust in automated decisions has not been established. Partial automation builds both simultaneously, creating the conditions under which full automation can be introduced later with confidence.

C. Layer 3 - Reuse: Evaluate Existing Organizational Capabilities

Layer 3 addresses the most costly failure mode in enterprise product development: building new systems that duplicate existing capabilities because no one knew they were there. In large enterprises that have invested in technology across multiple years and organizational units, the probability that a new operational requirement can be served by an existing system with moderate adaptation is substantially higher than most product investment processes acknowledge. The Reuse layer requires an explicit audit of existing organizational capabilities before any new development commitment. AI reuse takes several forms: model reuse (applying validated models to new domains with similar problem structure), platform reuse (deploying new capabilities on existing AI infrastructure), and workflow reuse (adapting existing automation logic with input-output modifications) [6].

The organizational barriers to reuse are primarily informational and incentive-related. Product managers who are unaware of an existing system cannot consider reusing it. Teams rewarded for new system launches have structural incentives to build rather than adapt. Addressing these barriers requires organizational infrastructure: shared capability registries, documented APIs for existing systems, cross-team reuse agreements, and performance frameworks that credit reuse contributions [9].

D. Layer 4 - Build-to-Reuse: Design New Systems for Organizational Leverage

When elimination, automation, and reuse have all been genuinely evaluated and found insufficient, new system development is appropriate. Build-to-Reuse specifies how the new system should be designed: organizational reusability is a first-order design criterion, not a secondary consideration. A system designed at Layer 4 is architected for the broadest class of organizationally relevant use cases that the same foundation can serve, with modular components, documented interfaces, and extension mechanisms that make future reuse straightforward rather than incidental [8].

For AI and ML systems specifically: model architecture should be modular; data pipelines designed as organizational assets with standardized schemas; AI governance artifacts designed as reusable templates; and deployment infrastructure abstracted from model-specific logic. The first deployment must also invest in the documentation, API design, and organizational communication required for other teams to discover and reuse the system. A system that is architecturally reusable but organizationally invisible will not be reused - it will be duplicated.

Table 2. EARBr decision layers: AI and automation role, governance criteria, and advancement conditions

EARBr Layer	Primary Question	AI and Automation Role	Advance to Next Layer When
1. Eliminate	Should this process exist at all?	None - this is a process existence question not a technology question	Process has valid current purpose not achievable via discontinuation
2. Automate	Can manual intervention be reduced in this validated process?	AI for classification, NLP, prediction within the process; partial automation is a valid outcome	Full process-level automation required; targeted automation insufficient for the objective
3. Reuse	Can an existing system or AI capability serve this need?	Model reuse, platform reuse, workflow adaptation; existing	No existing capability serves the need within acceptable cost and quality thresholds

		capability evaluated before new AI initiated	
4. Build-to-Reuse	Design new systems for organizational reusability?	New AI/ML with modular architecture, documented APIs, reusability as first-order design criterion	Terminal layer - all prior layers genuinely evaluated and found insufficient

Figure 1. EARBr Framework Decision Flow
Four sequential layers in strict precedence order. AI applies at Layers 2-4 only.

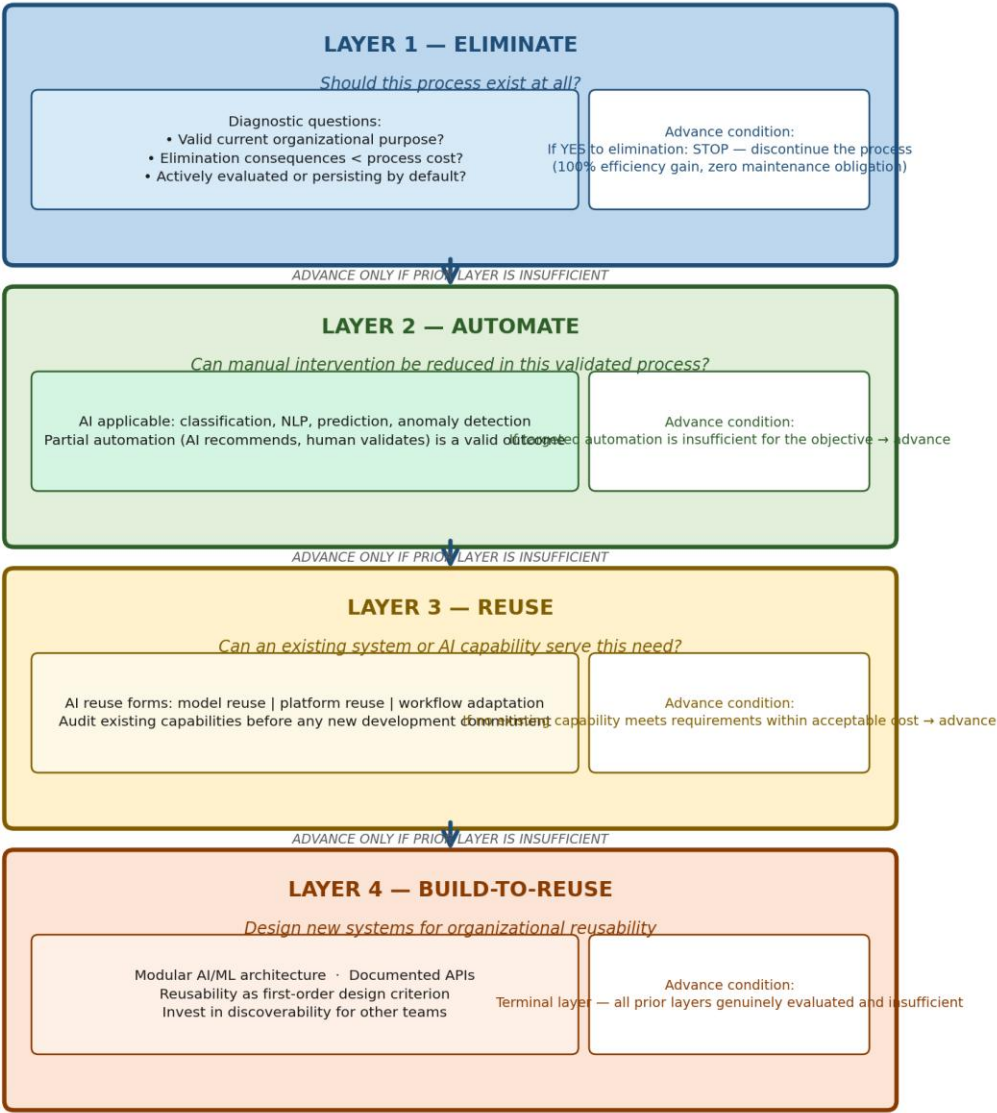


Figure 1. EARBr framework decision flow. Four sequential layers in strict precedence order. Each layer has an explicit advance condition: proceed to the next layer only when the prior layer has been genuinely evaluated and

found insufficient. AI and automation interventions apply across Layers 2-4; Layer 1 is a process existence question that precedes all technology decisions.

4. Practitioner Case Illustrations

A. Case 1 - User Access Management Onboarding: Layers 1 and 2

A large retail organization managed user access to enterprise operational systems through a manual onboarding process: requests submitted through a third-party ticketing platform, multi-step approvals from multiple functional owners, and access provisioned manually by a dedicated operations team. The initial product request was to build an alerting and monitoring system to improve ticket processing throughput - a Layer 2 solution applied to the existing process architecture.

Applying the Layer 1 Eliminate evaluation reframed the problem. The question was not how to make ticket processing faster. The question was whether the ticketing and approval process should exist in its current form. Analysis showed that the vast majority of onboarding events involved users with well-defined standard roles, for which access requirements were fully predictable from existing HR system data and required no human approval. A self-service system was implemented for standard users; non-standard requests routed through a streamlined approval workflow. Result: approximately 90% efficiency improvement, onboarding lead time reduced from days to hours - achieved through Layer 1 process redesign, before any AI investment [7].

The Layer 2 automation followed. A specific user population - those whose role, location, and access requirements were fully specified in HR records - was identified as eligible for fully automated provisioning through direct HR system integration. Result: 80% onboarding time reduction for approximately 250,000 users annually, zero ongoing manual intervention for standard-role provisioning. The automation was scoped precisely to the classification task for the eligible population - all governance and non-standard access management remained under human oversight [7].

B. Case 2 - Modular Task Management Platform: Layers 3 and 4

A retail organization facing supply chain disruptions needed a task-based application for associate management of customer interactions related to delayed inventory. Before initiating new development, the Layer 3 Reuse evaluation found an existing in-store application with generalizable task management capabilities that could be adapted to the new use case with interface modifications rather than full redevelopment. This approach avoided full development cost and delivered the capability faster [7].

The Reuse evaluation also revealed the existing system had not been designed for reusability, requiring substantial integration effort to adapt. This informed the subsequent Layer 4 Build-to-Reuse decision: when a second requirement emerged (proactive outreach and upselling), the system was redesigned as a modular task management platform capable of serving multiple use cases without full redevelopment. The Build-to-Reuse investment produced a platform subsequently serving four operational domains from a single development commitment - a compound return not achievable from a single-use architecture [7, 19].

C. Case 3 - Associate-Facing Bakery Application Redesign: Layer 4

The redesign of an associate-facing bakery application at a large retail organization illustrates Layer 4 Build-to-Reuse applied to AI-adjacent product decisions. The original application had been built for a specific operational context with no reuse architecture: catalog updates required engineering dependency, online and in-store offerings were managed in separate systems, and the application could not support new business models without full redevelopment. The Build-to-Reuse redesign implemented a merchant-managed, single-source-of-truth catalog with extensible architecture. Measured outcomes: reduced order cancellations, improved fulfillment accuracy, eliminated engineering dependency for catalog updates, and a platform subsequently extended to additional retail categories without new development investment [7].

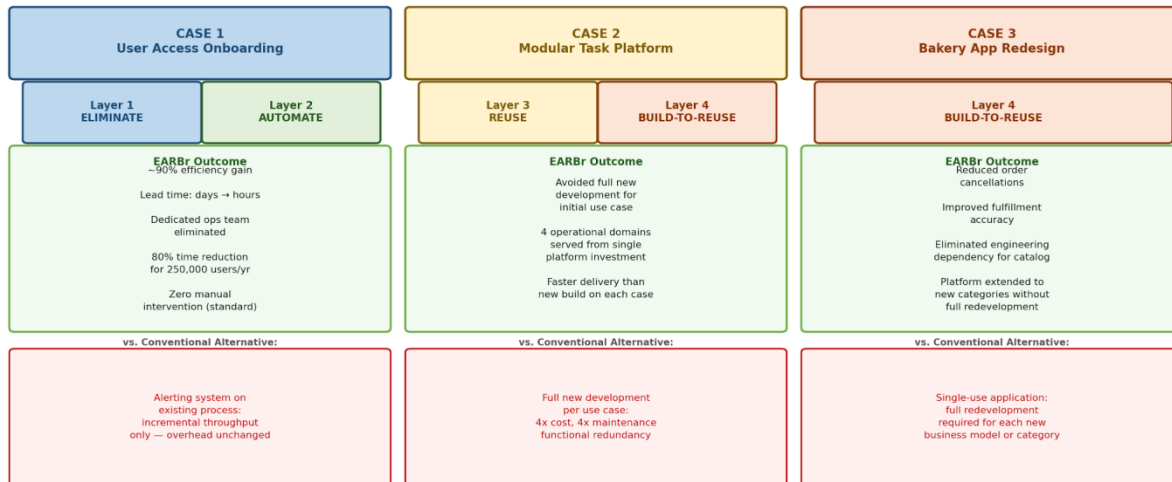


Figure 2. EARBr Framework Applied to Three Practitioner Case Illustrations
 Each case demonstrates the efficiency differential between EARBr-consistent decisions and conventional practice alternatives.

Figure 2. EARBr framework applied to three practitioner case illustrations. Each case demonstrates the efficiency differential between EARBr-consistent decisions and conventional practice alternatives. Cases 1 and 2 demonstrate multi-layer application; Case 3 demonstrates Layer 4 Build-to-Reuse applied to associate-facing product architecture.

5. Organizational Adoption

Making EARBr work at enterprise scale requires embedding it in organizational processes that make the evaluation mandatory rather than optional. The primary mechanism is PRD integration: every new product investment request includes an EARBr evaluation section documenting the evidence reviewed at each layer and the reasoning for advancing past each layer found insufficient. Without this documentation requirement, EARBr is applied by the teams inclined to apply it and bypassed by those who are not - which is exactly the incentive dynamic the framework is designed to correct. PRD integration transforms EARBr from a decision heuristic into an organizational governance standard with audit visibility.

Capability visibility is the infrastructure prerequisite that most organizations discover they are missing only after attempting Reuse evaluation. Product managers cannot evaluate reuse options they are unaware of, and organizational leaders cannot identify redundant investments that are not visible across team boundaries. Building and maintaining shared capability registries - structured, searchable, kept current as systems are deployed and deprecated - is unglamorous infrastructure work. It is also the single most important enabler of Layer 3 adoption. Organizations with this infrastructure consistently report faster delivery timelines and lower portfolio total cost of ownership [9, 10].

The incentive problem requires leadership to actively redesign how engineering and product teams are evaluated. As long as teams are measured primarily on new system delivery, EARBr-consistent decisions that reduce rather than increase new development will be organizationally disadvantaged. Layer 1 decisions look like a smaller scope. Layer 3 decisions look like slower delivery. These appearances are real in the short term and misleading in the long term - but short-term metrics drive short-term decisions. Performance frameworks that credit process elimination, recognize reuse contributions, and evaluate investment efficiency alongside delivery velocity require senior leadership commitment, not just endorsement in a strategy document [4, 13, 14, 15, 16].

Table 3. EARBr organizational adoption model: requirements, enablers, and success indicators

Adoption Dimension	Implementation Requirement	Key Enabler	Success Indicator
--------------------	----------------------------	-------------	-------------------

PRD Integration	EARBr evaluation section mandatory in all investment PRDs	Standard PRD template with EARBr checklist; leadership review gates	All investment PRDs include documented EARBr layer evaluations with cited evidence
Capability Visibility	Current organizational capability registry accessible to all product teams	Shared registry platform; ownership assignment; quarterly maintenance	Reuse evaluation completable under one week; no post-hoc missed reuse discoveries
Incentive Alignment	Performance frameworks credit elimination and reuse alongside new delivery	Leadership-level metric redesign; engineering recognition for reuse architecture	Voluntary reuse selection rate 30%+; Layer 1 outcomes valued in performance reviews
Justification Requirements	Documented rationale required to advance past each EARBr layer	Review board gate at Layer 3 to Layer 4 transition	Layer 4 decisions reviewed at appropriate level; redundant development incidents decreasing year-over-year

6. Evaluation and Measured Outcomes

EARBr is evaluated against three criteria: whether it correctly identifies the highest-return intervention for a given operational challenge; the efficiency gains achieved relative to the conventional practice alternative; and whether Build-to-Reuse architectural decisions generate measurable compound reuse over time. Across three case illustrations, EARBr-consistent decisions substantially outperformed what conventional practice would have produced. The Layer 1 process redesign in Case 1 achieved approximately 90% efficiency improvement, against the incremental throughput gain that the originally requested alerting system would have delivered. The Layer 2 automation achieved 80% onboarding time reduction for approximately 250,000 annual users. The Layers 3-4 decisions in Case 2 produced a platform serving four operational domains from a single development investment [7, 11].

The compounding value of consistent EARBr application across a product portfolio is the framework's most important long-term property and the one most difficult to evaluate in individual project assessments. Organizations applying EARBr systematically accumulate a progressively better-structured capability landscape: fewer redundant systems, more reusable platforms, lower total cost of ownership per operational capability, and faster delivery timelines for new requirements as the reusable component pool grows with each Build-to-Reuse investment. This compounding is invisible to evaluation frameworks that assess investment decisions in isolation [2, 12-14].

Limitations of the current evaluation are acknowledged [15]. Three cases from a single organizational context; practitioner judgment rather than controlled experimental conditions; efficiency figures are reported operational outcomes. Broader empirical validation across diverse enterprise contexts, industry verticals, and AI technology maturity levels is required to establish the framework's generalizability [16, 17-19].

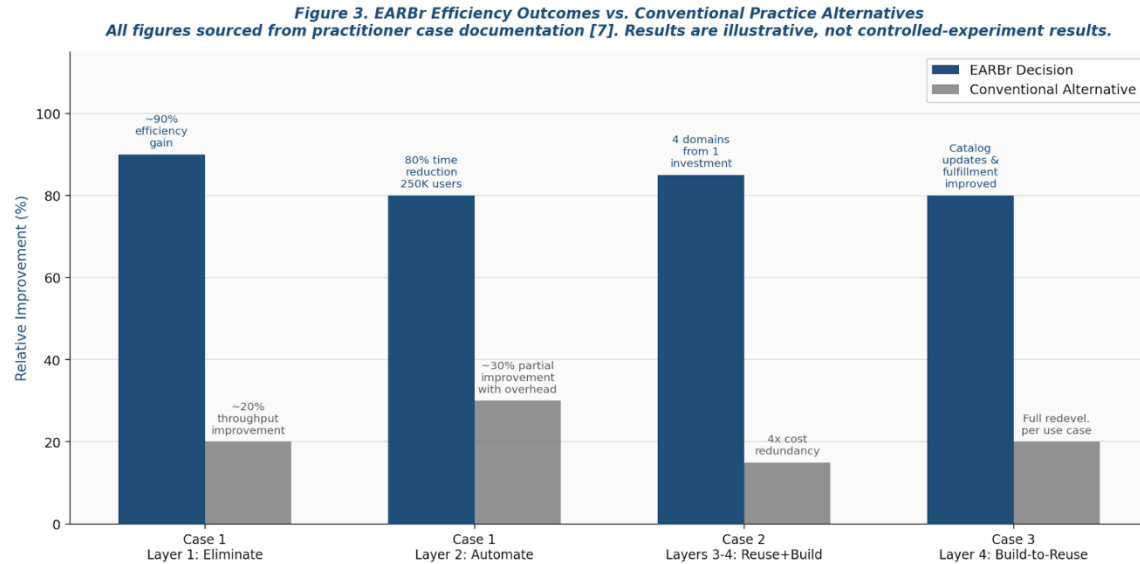


Figure 3. EARBr efficiency outcomes vs. conventional practice alternatives across three practitioner cases. EARBr-consistent decisions (dark bars) substantially outperform conventional alternatives (grey bars) in every case. All figures sourced from practitioner case documentation [7].

7. Conclusion

The EARBr framework addresses a problem that most AI governance literature treats as peripheral: the sequencing quality of enterprise AI adoption decisions. Most AI investment failures are not model failures - the models often work exactly as designed. They are sequencing failures: AI applied to the wrong processes, before simpler and higher-return options were genuinely evaluated, producing systems that perform admirably in technical benchmarks and deliver disappointing organizational returns. EARBr corrects this by making sequencing explicit, mandatory, and auditable - four layers with documented advancement criteria, organized in a governance model that works with organizational incentive structures rather than against them.

The practitioner cases demonstrate that EARBr-consistent decisions generate efficiency outcomes that substantially exceed conventional practice in comparable contexts. Approximately 90% process efficiency gains from Layer 1 elimination. Layer 2 targeted automation achieves 80% reductions in automation time for 250,000 annual users. Four-domain platforms arise from a single Build-to-Reuse investment. These outcomes are attributable to sequencing discipline - knowing what to do in what order - not to superior AI technology.

Three future research directions carry the most urgency: empirical validation of EARBr application across diverse enterprise contexts to establish generalizability; longitudinal measurement of EARBr adoption on AI ROI and technical debt accumulation to demonstrate compounding value; and integration of EARBr with established AI governance frameworks - OECD AI Principles, NIST AI Risk Management Framework - to position EARBr as a complementary operational governance layer within the broader enterprise AI governance architecture.

8. Funding

This research received no external funding.

9. Author Contribution

M. H. Choudhury is the sole author of this work and is responsible for its conception, analysis, drafting, and final approval.

10. Conflict of Interest

The author declares no conflict of interest.

11. Data Availability Statement

No new datasets were generated for this study. The practitioner case outcomes reported in Section IV are drawn from the author's professional experience; no proprietary organizational data are available for public release.

REFERENCES

1. Valeria Jannelli et al., "Agentic LLMs in the supply chain: towards autonomous multi-agent consensus-seeking," *Int. J. Prod. Res.*, 2025. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/00207543.2025.2604311>
2. S. Moreschini et al., "The evolution of technical debt from DevOps to generative AI: A multivocal literature review," *J. Syst. Softw.*, vol. 231, p. 112599, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0164121225002687>
3. Zied Bahroun et al., "A systematic analysis of generative artificial intelligence for supply chain transformation," 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2949863525000883>
4. P. Jooss et al., "Artificial intelligence and work design: implications for frontline service employees and future research," *J. Service Manag.*, 2025. [Online]. Available: <https://www.emerald.com/josm/article/doi/10.1108/JOSM-12-2024-0535/1318276/Artificial-intelligence-and-work-design>
5. A. G. Tashkinov, "The implementation of lean and digital management techniques using artificial intelligence in industrial settings," *Discov. Artif. Intell.*, vol. 4, p. 94, 2024. [Online]. Available: <https://link.springer.com/article/10.1007/s44163-024-00186-5>
6. Juha Savolainen; Mike Mannion, "A make/buy/reuse feature development framework for product line evolution," in *Proc. IEEE Conf. Softw. Eng.*, 2015. [Online]. Available: <https://ieeexplore.ieee.org/document/7384227/>
7. M. Passalacqua et al., "Human-centred AI in industry 5.0: a systematic review," *Int. J. Prod. Res.*, vol. 63, no. 7, pp. 2638-2669, 2024. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/00207543.2024.2406021>
8. Sunaina Sridhar et al., "A comprehensive framework for human-AI collaborative decision making in intelligent retail environments," *Expert Syst. Appl.*, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417425036292>
9. Y. Yang et al., "Multi-agent deep reinforcement learning for integrated demand forecasting and inventory optimization in sensor-enabled retail supply chains," *Sensors*, vol. 25, no. 8, p. 2428, 2025. [Online]. Available: <https://www.mdpi.com/1424-8220/25/8/2428>
10. A. Walter, K. Ahsan, and S. Rahman, "Application of artificial intelligence in demand planning for supply chains," *Int. J. Logistics Manag.*, vol. 36, no. 3, pp. 672-719, 2025. [Online]. Available: <https://www.emerald.com/insight/content/doi/10.1108/IJLM-02-2024-0120/full/html>
11. S. Ahn, J. Park, and S. Ye, "How AI enhances employee service innovation in retail," *J. Retailing Consumer Serv.*, vol. 84, p. 104207, 2025. [Online]. Available: <https://www.sciencedirect.com/article/abs/pii/S0969698924005034>
12. D. Grewal et al., "Leveraging in-store technology and AI," *J. Retailing*, vol. 99, no. 4, pp. 487-504, 2023. [Online]. Available: <https://www.sciencedirect.com/article/pii/S0022435923000507>
13. Paris Avgeriou et al., "Technical debt management: the road ahead," *arXiv:2403.06484*, 2024. [Online]. Available: <https://arxiv.org/html/2403.06484v1>
14. S. Kesavan et al., "Doing well by doing good: improving retail store performance with responsible scheduling practices at the Gap, Inc.," *Manage. Sci.*, vol. 68, no. 11, pp. 7818-7836, 2022. [Online]. Available: <https://pubsonline.informs.org/doi/10.1287/mnsc.2021.4291>
15. H. Gaggar, "MemGov: Policy-Governed Memory Lifecycle Management for Enterprise LLM Agents," *J. Intell. Decis. Mak. Inf. Sci.*, vol. 3, no. 11s, pp. 2199-2214, 2026.
16. K. Rekulapalli, H. Nilagiri, and M. Vakkanthula, "A Comparative Evaluation of National FHIR Implementation Guides: Strategic Methodologies for Enhancing Interoperability, Regulatory Alignment, and Clinical Comprehensiveness in Global Health IT," *Health Systems*, pp. 1-13, 2026. [Online]. Available: <https://doi.org/10.1080/20476965.2026.2728621>.
17. C. T. Bobba, A. G. Vijay, V. Aher, J. Singh, A. Kejriwal, K. Srinivasan, and A. Ranjan, "Radiographic Evaluation of Periapical Healing Following Single vs Multiple Visit Root Canal Treatment: A PAI Based Study," *Journal of Pharmacy & Bioallied Sciences*, vol. 18, Suppl. 1, art. S206, 2026.
18. A. Bhatnagar and V. R. K. Bonagiri, "Operational Excellence through Cloud-Native Distributed Systems," *Int. J. Comput. Exp. Sci. Eng.*, vol. 12, no. 2, 2026. [Online]. Available: <https://doi.org/10.22399/ijcesen.5207>.
19. H. Gaggar, "Architecting Scalable Enterprise AI Agent Infrastructure for Distributed Intelligent Systems," *Int. J. Adv. Signal Image Sci.*, pp. 193-208, 2025. [Online]. Available: <https://doi.org/10.29284/da9w0117>.