

## Article

# CAPTURE: A Stakeholder-Centered Iterative MLOps Lifecycle

Michał Slupczynski <sup>1,\*</sup> , René Reiners <sup>1,2</sup>  and Stefan Decker <sup>1,2</sup> 

<sup>1</sup> Information Systems and Databases, RWTH Aachen University, Ahornstr. 55, 52074 Aachen, Germany; rene.reiners@fit.fraunhofer.de (R.R.); decker@dbis.rwth-aachen.de (S.D.)

<sup>2</sup> Fraunhofer FIT, Schloss Birlinghoven, 53757 Sankt Augustin, Germany

\* Correspondence: slupczynski@dbis.rwth-aachen.de

## Abstract

Current ML lifecycle frameworks provide limited support for continuous stakeholder alignment and infrastructure evolution, particularly in sensor-based AI systems. We present CAPTURE, a seven-phase framework (Consult, Articulate, Protocol, Terraform, Utilize, Reify, Evolve) that integrates stakeholder-centered requirements engineering with MLOps practices to address these gaps. The framework was synthesized from four established standards (ISO/IEC 22989, ISO 9241-210, CRISP-ML(Q), SE4ML) and validated through a longitudinal five-year case study of a psychomotor skill learning system alongside semi-structured interviews with ten domain experts. The evaluation demonstrates that CAPTURE supports governance of iterative development and strategic evolution through explicit decision gates. Expert assessments confirm the necessity of the intermediate stakeholder-alignment layer and substantiate the participatory modeling approach. By connecting technical MLOps with human-centered design, CAPTURE reduces the risk that sensor-based AI systems become ungoverned, non-compliant, or misaligned with user needs over time.

**Keywords:** MLOps; requirements engineering for AI; stakeholder alignment; human-centered AI; AI lifecycle management; participatory modeling; sensor-based systems; intelligent software engineering

## 1. Introduction

As Artificial Intelligence (AI) systems are increasingly used in complex environments, they create social and technical challenges that require oversight from various stakeholders whose needs often go beyond standard business or technical Machine Learning (ML) goals [1,2]. This paper addresses stakeholder-centered ML lifecycle governance as its primary concern. We define *stakeholder-centered* as an approach that places stakeholder needs, constraints, and feedback at the center of lifecycle decisions, extending human-centered design (ISO 9241-210) [3] beyond end-users to encompass all parties affected by or influencing system outcomes. Unlike Machine Learning Operations (MLOps) maturity models that focus on automation capabilities and deployment velocity [4], CAPTURE prioritizes governance accountability and stakeholder alignment as primary maturity indicators.

Sensor-based AI systems, which rely on physical data captured from hardware sensors to perform in real-world environments [5], represent a prominent domain where stakeholder-centered approaches are essential. We therefore use them as the exemplar to drive framework design, while the resulting governance mechanisms apply more broadly.

Sensor-based systems are deployed across domains, including medical diagnostics [6,7], immersive learning [8], human-robot collaboration [9], Industry 5.0 [10], and



Academic Editors: Mirko Perkusich and Emilia Mendes

Received: 20 December 2025

Revised: 10 January 2026

Accepted: 20 January 2026

Published: 26 January 2026

**Copyright:** © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

smart cities [11]. They exhibit operational characteristics that demand sustained stakeholder engagement: strict temporal constraints requiring (near)-real-time processing [12], multimodal sensor fusion across heterogeneous data sources [13], contextual metadata dependencies such as calibration parameters and environmental conditions [14,15], and safety-critical contexts where failures carry significant consequences [5]. These characteristics align with Industry 5.0's emphasis on human-centeredness (prioritizing worker and stakeholder well-being over pure efficiency) and resilience (enabling systems to adapt and recover from disruptions through structured governance). Such operational demands amplify the need for continuous stakeholder engagement throughout deployment and evolution, rather than limiting involvement to initial requirements [1].

CAPTURE is specifically designed for systems where at least two of the following sensor-specific properties hold: (1) continuous or high-frequency data streams, (2) physically grounded uncertainty from calibration, placement, or environmental factors, (3) temporal coupling between data acquisition and action, or (4) safety or liability exposure from sensor failure. For systems with static, curated inputs lacking strict real-time constraints or those driven by singular business Key Performance Indicators (KPIs) without multi-stakeholder alignment, CAPTURE reduces to a form equivalent to CRISP-ML(Q) [2] or ISO/IEC 22989 [16] with standard ML engineering integration.

The framework's applicability spans a spectrum calibrated through decision gate thresholds (see Section 3.3.8). Setting low gate thresholds enables high-velocity projects where gates serve as lightweight reminders and decision traces rather than blocking checkpoints. Conversely, high threshold values suit safety-critical or regulatory-sensitive applications requiring thorough governance at each transition. CAPTURE allows practitioners to record bypassed provenance steps as deliberate meta-decisions, ensuring technical debt remains explicit rather than untracked. While increasing coordination and documentation overhead, the framework trades upstream rigor for reduced downstream failure, rework, and regulatory exposure. The contribution of this work lies in specifying what must be reasoned about before addressing who or what performs the reasoning.

#### *Problem Statement and Contributions*

While established frameworks provide valuable foundations (such as ISO/IEC 22989 [16] for AI lifecycle governance, ISO 9241-210 [3] for Human-Centered Design (HCD) principles, CRISP-ML(Q) [2] for quality-assured ML workflows, and Software Engineering for Machine Learning (SE4ML) [17] for ML engineering practices [18–20]), they exhibit substantial gaps in stakeholder-centered governance and neglect traceable transitions from stakeholder intent to data-driven system evolution, particularly evident when applied to sensor-based AI systems. Current lifecycle models conflate *operational human involvement* (labeling, model evaluation) with *foundational stakeholder engagement* (requirements elicitation, governance, acceptance criteria). This obscures critical questions: Who defines system constraints? What counts as acceptable behavior? How are lifecycle decisions justified? Section 3.2 details the specific gaps in each framework that CAPTURE addresses. This work makes the following contributions to Intelligent Software Engineering. We (RO1) establish stakeholder engagement as a foundational, continuous activity distinct from operational Human-in-the-loop (HITL) approaches, and (RO2) operationalize lifecycle management for continuous sensor dataflows handling temporal, multimodal, and safety constraints. Furthermore, we (RO3) embed Verification and Validation (V&V) obligations across all lifecycle phases, while (RO4) formalizing infrastructure as a governed lifecycle entity with explicit versioning. Finally, to enable evidence-based lifecycle decisions, (RO5) provides machine-readable decision provenance and traceability infrastructure supporting human- and tool-assisted decision-making throughout the lifecycle.

Concretely, we present **CAPTURE**, a seven-phase framework (CONSULT, ARTICULATE, PROTOCOL, TERRAFORM, UTILIZE, REIFY, EVOLVE) that operationalizes ISO/IEC 22989, extends CRISP-ML(Q), and adapts ISO 9241-210 and SE4ML principles, as validated through conceptual cognitive walkthroughs, a longitudinal case study, and expert interviews. While we demonstrate CAPTURE through a sensor-based psychomotor-learning infrastructure case study, the framework's stakeholder-centered governance mechanisms (RO1, RO3, RO4, RO5) could apply broadly to AI systems requiring multi-stakeholder alignment, with continuous dataflow management (RO2) addressing streaming data scenarios increasingly common across AI deployments.

This paper is organized as follows: Section 3.2 describes the CAPTURE framework design, including background on foundational models, the seven-phase process, and decision gate mechanisms. Section 4 evaluates the CAPTURE framework using conceptual cognitive walkthroughs, a longitudinal case study, and expert interviews. Section 5 discusses implications for Intelligent Software Engineering (ISE) practice, limitations, and applicability considerations. Section 6 concludes with a summary of contributions and future research directions.

## 2. Background and Related Work

Modern sensor-based applications increasingly leverage pre-trained foundation models [21], shifting focus from training to model selection and integration. Sensor-based AI systems require governance frameworks that integrate responsible AI principles, accommodate diverse deployment scenarios, and treat infrastructure as an evolving entity rather than a static substrate. We ground CAPTURE in the FATES principles [22] (Fairness, Accountability, Transparency, Explainability, Safety) and explainability dimensions [23] (data, model, post hoc, assessment), integrating them systematically across all lifecycle phases. However, core governance responsibilities persist: data pipelines, contracts, and stakeholder requirements must still be managed even when using inference-only endpoints. Effective governance relies on formal artifacts, including KPIs [24], data contracts [25,26], and quality metrics [27,28], to encode stakeholder expectations into machine-verifiable specifications. Finally, viewing infrastructure as a dynamic sociotechnical system [29,30] rather than a static substrate allows us to govern it as a first-class lifecycle entity. Specifically, CAPTURE manages infrastructure's "installed base" inertia, the path dependencies created by existing organizational, technical, and financial investments that constrain and shape system design [31,32]. By embedding these considerations and accommodating both training and inference-only workflows systematically, CAPTURE positions governance challenges as core components rather than peripheral concerns.

### *Related Work*

The recognition of "hidden technical debt" [31] in ML systems catalyzed disciplined engineering practices for production AI.

Lwakatare et al. [33] systematically document challenges differentiating ML from traditional software engineering [34], including data dependency management, reproducibility, and testing non-deterministic systems. MLOps bridges ML development and production by automating workflows, enabling versioning, tracking metadata, and ensuring reproducibility [20,24,35]. Hanchuk and Semerikov [20] analyze these capabilities across various platforms, while Zarour et al. [36] categorize them using maturity models. Serban et al. [37] synthesize software engineering practices for ML, establishing recommendations for version control, testing, and deployment. While conventional MLOps establishes versioning for technical artifacts such as data and models, Singh et al. [38] introduce *Decision Provenance* to view design decisions as critical artifacts requiring systematic traceability.

For human–AI interaction, Amershi et al. [39] provide guidelines for collaborative interfaces, while van der Stappen and Funk [40] extend this with guidance for HITL training interfaces, balancing automation with human oversight. Cheruvu [41] further distinguishes HITL, where humans act as active collaborators in labeling and refinement, from Human-on-the-loop (HOTL), where they provide supervisory oversight with the capacity to intervene. While HITL is operational and model-centric, stakeholder engagement provides the system-centric foundation for modeling objectives, data governance, and permissible trade-offs. Shin et al. [42] demonstrate that stakeholder motivation in HITL systems requires explicit formalization through clear value propositions and transparent feedback mechanisms. Ahmed [43] provides experimental evidence that *communication* warrants explicit status as a ninth phase in iterative workflows. Domain-specific best practices, such as cybersecurity considerations for IoT systems [44], address sector-specific requirements. Table 1 summarizes the capabilities and limitations of established AI/ML lifecycle frameworks (ISO/IEC 22989 [16], ISO 9241-210 [3], CRISP-ML(Q) [2], SE4ML [17], NIST AI RMF) regarding sensor-based lifecycle requirements. To our knowledge, no framework addresses sensor-based stakeholder-centered AI systems holistically.

Despite these advances, existing frameworks exhibit critical gaps for sensor-based AI systems. Generic MLOps presumes that requirements and stakeholder alignment are external inputs, offering no systematic elicitation or governance mechanisms. Frameworks do not formalize re-evaluation criteria, retirement conditions, or stakeholder-driven evolution triggers for models experiencing drift. These gaps span stakeholder elicitation, evolution criteria, and governance mechanisms. Industry surveys confirm a growing gap between AI governance ambition and maturity, with 53% of organizations rating governance capabilities as only moderately effective [4]. Emerging regulatory requirements (such as the EU AI Act) drive integration of Governance-as-Code within MLOps pipelines [45], while ISO/IEC 42001 [46] establishes organizational requirements for AI management systems. Data-centric AI lifecycles [47] emphasize data quality over model complexity, and continuous learning pipelines [18] address model evolution, yet neither systematically integrates stakeholder governance.

Existing approaches provide limited mechanisms for bridging stakeholder-driven design with continuous feedback loops, particularly where multiple stakeholder groups must remain engaged throughout operation. CAPTURE addresses these gaps by integrating MLOps as an operational automation layer within a governed lifecycle, adding upstream stakeholder engagement (CONSULT, ARTICULATE, PROTOCOL) and downstream evaluation phases (REIFY, EVOLVE). We describe CAPTURE’s synthesis and extension of these foundations in the following section.

**Table 1.** Comparative analysis of AI/ML lifecycle frameworks.

Framework	Key Elements	Strengths	Gaps and Limitations
ISO/IEC 22989 [16]: Artificial intelligence	<ol style="list-style-type: none"> <li>1. Inception</li> <li>2. Design and development</li> <li>3. Verification and validation</li> </ol>	<ul style="list-style-type: none"> <li>• Full lifecycle perspective</li> <li>• Mandates governance and ethics</li> </ul>	<ul style="list-style-type: none"> <li>• No operational guidance for infrastructure lifecycle</li> </ul>
Standardized system-level AI lifecycle framework	<ol style="list-style-type: none"> <li>4. Deployment</li> <li>5. Operation and monitoring</li> <li>6. Re-evaluate</li> <li>7. Retirement</li> </ol>	<ul style="list-style-type: none"> <li>• Provides terminology and conceptual structure</li> </ul>	<ul style="list-style-type: none"> <li>• High-level stakeholder mention without elicitation/traceability mechanisms</li> </ul>

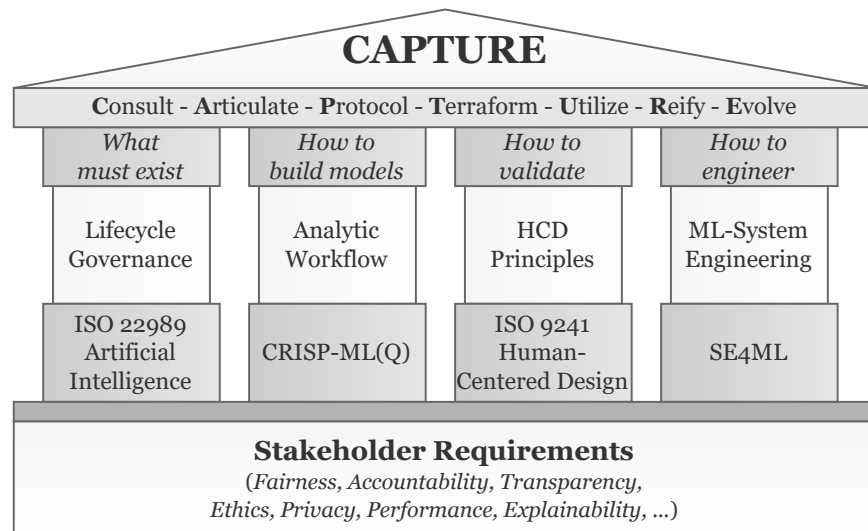
**Table 1.** *Cont.*

Framework	Key Elements	Strengths	Gaps and Limitations
ISO 9241 [3]: Human-Centered Design (HCD)  HCD process standard for interactive systems	<ol style="list-style-type: none"> <li>1. Understand and specify context of use</li> <li>2. Specify user requirements</li> <li>3. Produce design solutions</li> <li>4. Evaluate solutions against requirements</li> </ol>	<ul style="list-style-type: none"> <li>• Emphasis on user needs</li> <li>• Iterative evaluation integrated into design</li> <li>• Applicable across domains and system types</li> </ul>	<ul style="list-style-type: none"> <li>• Not AI/ML-specific</li> <li>• Lacks mechanisms for requirements, provenance, or traceability</li> <li>• Provides principles, not engineering processes or artifacts</li> </ul>
CRISP-ML(Q) [2]  Quality-assured ML workflow extending CRISP-DM	<ol style="list-style-type: none"> <li>1. Business and data understanding</li> <li>2. Data engineering</li> <li>3. ML model engineering</li> <li>4. QA (Quality Assurance)</li> <li>5. Deployment</li> <li>6. Monitoring and maintenance</li> </ol>	<ul style="list-style-type: none"> <li>• Quality-aware, iterative process</li> <li>• Structured model development</li> </ul>	<ul style="list-style-type: none"> <li>• Assumes static datasets</li> <li>• Business-, but not stakeholder centric</li> <li>• Model-, but not infrastructure lifecycle</li> <li>• No stakeholder governance</li> </ul>
SE4ML [17]  Empirically observed industrial ML practices	<ol style="list-style-type: none"> <li>1. Model requirements</li> <li>2. Data collection, cleaning, labeling</li> <li>3. Feature engineering</li> <li>4. Model training, evaluation</li> <li>5. Deployment, monitoring</li> </ol>	<ul style="list-style-type: none"> <li>• Highlights iterative ML workflow</li> <li>• Technical pipeline focus</li> <li>• Describes inter-team dependencies</li> </ul>	<ul style="list-style-type: none"> <li>• HITL limited to operational improvement</li> <li>• Observational, not prescriptive</li> <li>• Pipeline-centric, not system-level</li> <li>• Minimal governance/traceability</li> </ul>
NIST AI RMF  Risk-centric governance framework for trustworthy AI	<p>Non-sequential risk functions:</p> <ul style="list-style-type: none"> <li>• Govern</li> <li>• Map</li> <li>• Measure</li> <li>• Manage</li> </ul>	<ul style="list-style-type: none"> <li>• Stakeholder-aware</li> <li>• Governance-first</li> <li>• Trustworthy AI focus</li> </ul>	<ul style="list-style-type: none"> <li>• Risk functions are not lifecycle phases</li> <li>• No temporal sequencing, transition criteria, or engineering guidance</li> </ul>

### 3. Methods—Deriving the CAPTURE Framework

CAPTURE synthesizes four established frameworks (ISO/IEC 22989, CRISP-ML(Q), ISO 9241-210, SE4ML) to address documented gaps in stakeholder engagement, infrastructure governance, continuous V&V, and sensor-specific lifecycle support (see Figure 1).

This section presents the framework overview (Section 3.1), design rationale (Section 3.2), and detailed phase obligations (Section 3.3).



**Figure 1.** CAPTURE’s four-pillar positioning as operational stakeholder-centered integration layer. Foundation: Stakeholder requirements. Framework: ISO/IEC 22989 (governance), CRISP-ML(Q) (workflow), ISO 9241-210 (HCD validation), SE4ML (engineering).

3.1. Framework Overview

CAPTURE (CONSULT, ARTICULATE, PROTOCOL, TERRAFORM, UTILIZE, REIFY, EVOLVE) is a seven-phase stakeholder lifecycle framework for sensor-based ML systems. Decision gates between phases ensure evidence-based transitions: teams proceed only when mandatory criteria are satisfied and stakeholder approvals obtained.

Gate scoring functions ( $\Theta_{Gx}$ ) aggregate quality criteria using project-specific weights, permitting transition when scores exceed risk-calibrated thresholds ( $\tau_x$ ). This structure supports both forward progression and backward iteration when limitations are discovered. Each phase addresses a specific governance challenge:

- CONSULT Stakeholder identification and mapping: who should be involved and what do they need?
- ARTICULATE Requirements formalization: translating stakeholder needs into SMART specifications, data contracts, and KPIs.
- PROTOCOL Decision provenance: documenting how and why lifecycle decisions are made.
- TERRAFORM Infrastructure lifecycle: treating infrastructure as a governed entity with explicit versioning.
- UTILIZE Model development: building, training, and validating ML components.
- REIFY Deployment and evidence collection: observing system behavior in real stakeholder contexts.
- EVOLVE Evaluation and iteration: assessing impact and selecting appropriate evolution paths.

3.2. Framework Design Rationale

Each foundational framework contributes essential capabilities while exhibiting gaps that CAPTURE addresses. ISO/IEC 22989 [16] provides a governance structure defining *what must exist*, but lacks operational stakeholder mechanisms and sensor-ML specificity.

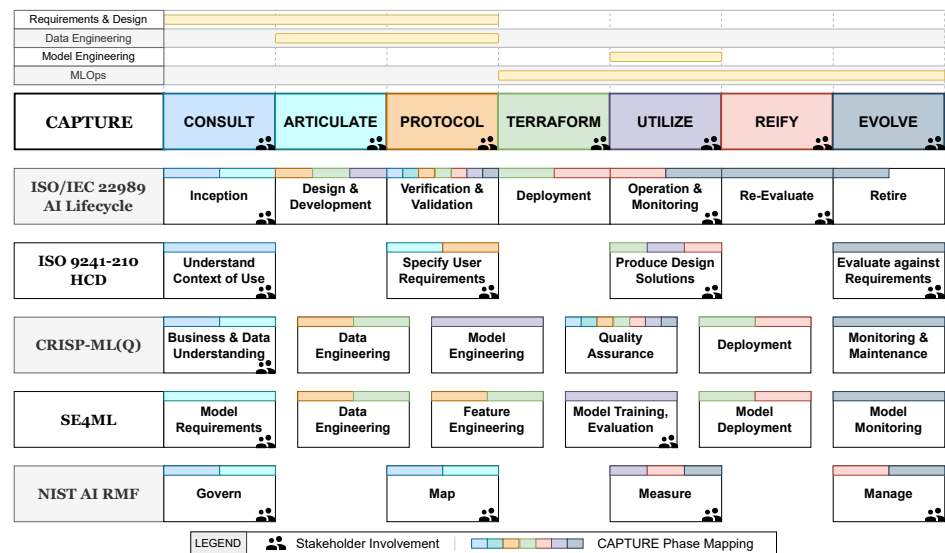
CRISP-ML(Q) [2] defines an analytic workflow specifying *how to build models*, but assumes stable datasets and terminal deployment, lacking infrastructure governance and structured iteration paths. ISO 9241-210 [3] establishes HCD principles for *how to validate*

with stakeholders, but does not address ML-specific concerns (drift, retraining, sensor fusion, continuous dataflows). SE4ML [17] contributes engineering practices for *how to engineer systems*, but focuses HITL on downstream labeling rather than upstream stakeholder engagement, omitting infrastructure lifecycle and decision provenance.

CAPTURE addresses these gaps through four contributions: (1) stakeholder-centered requirements engineering [48], (2) infrastructure as lifecycle entity [35,49], (3) continuous V&V [17,50], and (4) sensor-specific lifecycle support [12,51,52]. These contributions are operationalized through seven concrete phases (CONSULT, ARTICULATE, PROTOCOL, TERRAFORM, UTILIZE, REIFY, EVOLVE) with phase-specific obligations, decision gates, and sensor-ML-specific mechanisms.

### 3.2.1. Phase Alignment with Established Frameworks

Figure 2 visualizes how CAPTURE unifies the four foundational frameworks through seven phases. Above CAPTURE, software-engineering dimensions map to phases: Requirements and Design (CONSULT, ARTICULATE, PROTOCOL), Data and Feature Engineering (PROTOCOL, TERRAFORM), Model Engineering (UTILIZE), and MLOps (TERRAFORM, UTILIZE, REIFY, EVOLVE). Below CAPTURE, established frameworks align via color-coding to corresponding phases. The visual mapping demonstrates that CAPTURE does not replace these frameworks but operationalizes them through a unified stakeholder-centered structure.



**Figure 2.** Alignment of CAPTURE’s lifecycle phases (CONSULT (blue), ARTICULATE (teal), PROTOCOL (orange), TERRAFORM (green), UTILIZE (purple), REIFY (red), and EVOLVE (gray)) as an extension of ISO/IEC 22989, ISO 9241-210, SE4ML, and CRISP-ML(Q). Each phase of the reference models is marked with a color indicating their mapping to one of the CAPTURE phases. Additionally, explicit stakeholder involvement in one of the phases is indicated with an appropriate icon.

The phase alignment reflects deliberate design choices for how and why CAPTURE extends each framework. CAPTURE addresses ‘how’ by providing concrete activities, artifacts, and decision gates for each phase. CAPTURE addresses ‘why’ by linking each phase to documented gaps in stakeholder engagement, infrastructure governance, and continuous validation. Specifically: CONSULT operationalizes ISO 9241-210’s “understand context” principle by mandating stakeholder mapping before any technical work begins, preventing the data-scientist-centered bias documented in RE4ML surveys [53].

TERRAFORM addresses the infrastructure blindness in CRISP-ML(Q)’s “Business and Data Understanding” by treating infrastructure as a versioned lifecycle entity rather than an assumed constant. REIFY separates deployment from evaluation because ISO 22989’s

“Operation and Monitoring” conflates system release with feedback collection, while EVOLVE provides the structured iteration paths that CRISP-ML(Q)’s single “Deployment” phase cannot express.

The seven phases are summarized as follows: CONSULT (*engage and define*) establishes systematic engagement with domain experts, data subjects, operators, and regulators, identifying sensor-specific constraints such as latency envelopes [12], privacy boundaries [54], and environmental limits. ARTICULATE (*formalize requirements*) translates stakeholder goals into SMART requirements [55], data contracts [25,26], and measurable KPIs [24], making stakeholder intent computationally verifiable. PROTOCOL (*document decisions*) introduces decision provenance [38,56,57] and traceability matrices, with provenance artifacts structured in machine-readable formats (e.g., W3C PROV) to enable gate scoring automation. TERRAFORM (*build infrastructure*) establishes infrastructure as a versioned, auditable, governance-enforcing entity requiring its own lifecycle management [35,49]. UTILIZE (*implement ML modules*) accommodates both training workflows and inference-only scenarios, maintaining traceability between requirements, data, models, and validation [24,58]. REIFY (*apply and obtain feedback*) establishes deployment as a deliberate feedback-gathering activity, with temporal separation ensuring iteration decisions are evidence-based rather than reactive [24]. EVOLVE (*evaluate and iterate*) provides structured decision gates with five explicit iteration paths: minor iteration, major iteration, model update, continuous monitoring, and retirement [16,52].

### 3.2.2. Phase Selection Rationale

While CAPTURE aligns with several established frameworks, it employs a seven-phase structure to provide greater operational granularity. This section presents the rationale for this architectural choice.

#### Stakeholder Engagement as Scarce Resource

CAPTURE treats stakeholder engagement as a constrained resource requiring explicit budgeting (see Figure 3). The framework distinguishes three role classes: *decision authorities* who participate in gates and authorize transitions, *domain contributors* who provide requirements and feedback asynchronously, and *affected parties* who are observed or informed but not consulted at gates. Early phases (CONSULT, ARTICULATE) favor breadth by sampling many stakeholders, while later phases (REIFY, EVOLVE) favor depth by focusing on decision authorities. This design acknowledges that unlimited stakeholder involvement is neither feasible nor desirable, yet systematic involvement at critical decision points is essential.

<b>Phase Alignment</b>	CONSULT	ARTICULATE	...	EVOLVE	REIFY
	<i>early phases</i>			<i>later phases</i>	
<b>Strategy</b>	<b>BREADTH</b> <i>Sample many stakeholders</i>			<b>DEPTH</b> <i>Focus on decision authorities</i>	
<b>Role Classes</b>	<b>Affected parties</b> <i>(observed / informed)</i>		<b>Domain contributors</b> <i>(async feedback)</i>	<b>Decision authorities</b> <i>(gate participation)</i>	

**Figure 3.** Stakeholder engagement as scarce resource: engagement intensity transitions from breadth-focused (many stakeholders, early phases) to depth-focused (decision authorities, later phases).

#### Upstream Stakeholder Engagement

The three-phase upstream structure (CONSULT, ARTICULATE, PROTOCOL) operationalizes ISO 9241-210’s [3] first three HCD principles. This design addresses a documented gap in Requirement Engineering for Machine Learning (RE4ML) [48,53]: existing approaches

remain data-scientist-centered rather than stakeholder-centered. CONSULT establishes stakeholder mapping and constraint identification. ARTICULATE transforms preliminary requirements into SMART specifications [55], data contracts [25,26,59], and KPIs [24]. PROTOCOL implements decision provenance [38] for tracking lifecycle decisions.

Collapsing these phases would conflate stakeholder identification with requirement formalization, or requirement formalization with decision documentation, obscuring accountability boundaries. Table 2 maps each phase to its HCD principle grounding and documents the accountability risks that would arise from phase conflation.

**Table 2.** Upstream stakeholder engagement: HCD principle mapping and separation rationale.

Phase	HCD Principle	Focus	Risk if Conflated
CONSULT	Understand context	Stakeholder mapping	Unidentified stakeholders excluded from governance
ARTICULATE	Specify requirements	SMART formalization	Unmeasurable requirements evade validation
PROTOCOL	Produce solutions	Decision provenance	Untraceable decisions undermine accountability

**Separating Observation from Evaluation.** The distinction between REIFY and EVOLVE addresses traditional approaches that conflate deployment observation with iteration decision-making. REIFY focuses on deployment and operation in stakeholder contexts and systematic evidence collection. EVOLVE conducts impact assessment and drift analysis [51,52], determining what action the evidence warrants. This separation provides three benefits: temporal distinction (observation periods complete before evaluation), stakeholder role distinction (observed subjects in REIFY versus decision authorities in EVOLVE), and governance checkpoint alignment with ISO 22989’s distinction between “Operation and Monitoring” and “Re-evaluate.” Table 3 contrasts these phases and identifies the governance failures that arise when observation and evaluation are merged.

**Table 3.** Observation-evaluation separation: REIFY vs. EVOLVE with failure risks.

Aspect	REIFY	EVOLVE
Purpose	Deployment, evidence collection	Impact assessment, drift analysis
Stakeholder role	Observed/feedback providers	Decision authorities
ISO 22989 alignment	Operation/Monitor	Re-evaluate
Risk if conflated	Premature decisions before sufficient evidence; reactive rather than evidence-based iteration	

### Necessity of the Seven-Phase Structure

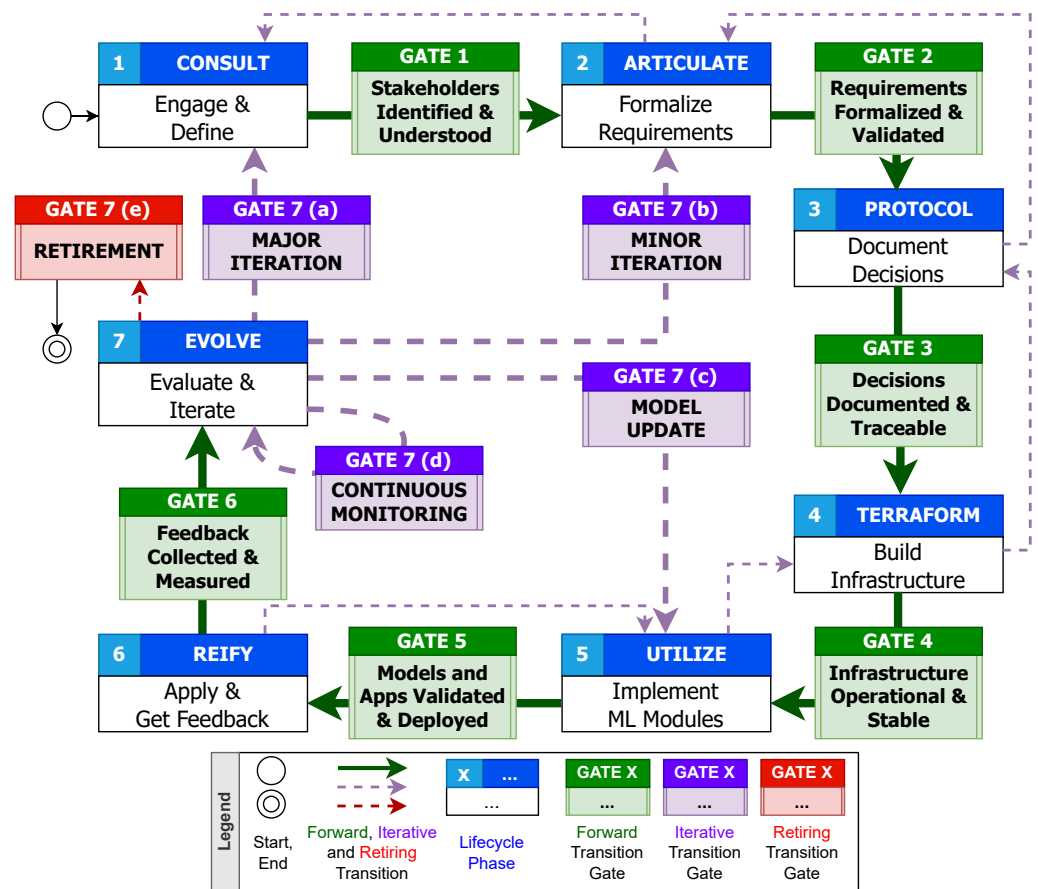
Each phase addresses gaps that persist across all four foundational frameworks. Fewer phases would conflate distinct stakeholder roles, validation criteria, and governance checkpoints. More phases would fragment workflow coherence and obscure the logical progression from stakeholder engagement through operational iteration. The seven-phase structure represents the minimum granularity necessary to maintain stakeholder-centered governance throughout the entire ML lifecycle.

### 3.3. Phase Transition Model

CAPTURE defines seven phases along with decision gates regulating phase transitions (see Figure 4). Each phase description includes a summary table with framework

extensions, data context, core activities, and key outputs. Gates enable evidence-based governance: transitions proceed only when mandatory requirements are satisfied, quality metrics exceed defined thresholds, and required stakeholder approvals are obtained. Each decision gate applies a quantitative scoring function  $\Theta_{Gx}$  that aggregates quality criteria using project-specific weights, permitting transition only when the score meets or exceeds a risk-calibrated threshold  $\tau_x$ .

CAPTURE externalizes lifecycle intelligence to structured socio-technical decision points rather than embedding it in automated agents; decision gates govern when and why transitions occur based on accumulated evidence. Decision gates in CAPTURE define observable system states, admissible transitions, and evidence thresholds, which together form a decision-theoretic model of the ML lifecycle. Stakeholder absence should default to previously ratified constraints unless explicitly reopened in EVOLVE, treating silence as consent within bounded risk.



**Figure 4.** CAPTURE phase transition model. Phases (blue) are linked by forward progression (solid green), iterative refinement (dashed purple), and retirement (dashed red). Decision gates (green/purple) regulate transitions from CONSULT through to strategic evaluation outcomes (7a–7e) in EVOLVE. The lifecycle initiates at CONSULT and follows solid green arrows for forward progression through sequential phases. Each decision gate (1–7) enforces quality criteria satisfaction before transition, while purple dashed arrows represent permitted backward transitions for iterative refinement. Gate 7 provides five strategic paths (7a–7e) determined by evaluation outcomes.

The gate scoring functions ( $\Theta_{Gx}$ ) adopt a linear weighted summation for three design reasons. First, linearity provides interpretability: stakeholders can trace how individual criteria contribute to overall scores without statistical expertise. Second, weighted aggregation accommodates domain-specific priorities through configurable weights. Third, the formulation supports incremental calibration based on empirical evidence from EVOLVE

phases. Mandatory criteria function as veto gates: if any mandatory requirement is unsatisfied, the gate blocks regardless of aggregate scoring performance, ensuring critical requirements cannot be traded off against strong performance in other areas.

#### Iterative Refinement

Backward transitions (purple dashed arrows in Figure 4) enable context-driven iteration [3] when validation failures or new insights emerge. ARTICULATE → CONSULT occurs when stakeholder landscape understanding proves incomplete. PROTOCOL → ARTICULATE addresses infeasible or conflicting requirements. TERRAFORM → PROTOCOL handles unimplementable design decisions. UTILIZE → TERRAFORM addresses infrastructure inadequacies. REIFY → UTILIZE manages deployed models failing production requirements. EVOLVE → REIFY occurs when evaluation reveals insufficient evidence for decision-making. No direct EVOLVE → TERRAFORM transition exists; infrastructure changes require returning to PROTOCOL or ARTICULATE to revise requirements and design decisions first. Although the framework defines backward transitions as single-step adjacencies, practitioners may chain multiple backward transitions when issues propagate across phases, requiring a return to earlier phases for corrective revision.

##### 3.3.1. CONSULT: Engage and Define

CONSULT (see Table 4) establishes foundational understanding of stakeholder needs, system constraints, and collaborative requirements guiding the ML lifecycle, positioning **data as a collaborative artifact** [60]. Comprehensive stakeholder analysis [24,61] identifies ML-generic and domain-specific stakeholders, distinguishing active and passive roles [62]. Stakeholder mapping approaches [63,64] assess influence [65], power [66], and interest levels, producing stakeholder maps documenting roles, information needs, and engagement strategies. Constraint identification encompasses ethical considerations [67], explainability requirements [68,69], legal compliance (GDPR), domain-specific constraints [70], and latency requirements [12]. Data requirements address ownership [71], permissions [72], privacy [54], transformations [73,74], anonymization [75], accountability [76], provenance [77], and governance [78]. Preliminary KPIs measure system success from stakeholder perspectives [24], including accuracy thresholds, latency targets, satisfaction metrics, and business value measures. RE4ML approaches [79,80] map explainability requirements to stakeholders' information needs [81], employing model-based requirement integration [82]. Critical architectural decisions determine whether the system performs training, fine-tuning, or inference only, and whether data processing uses streaming, batching, or one-shot training.

Deliverables include stakeholder maps, design principle collections [83], and comprehensive system and data requirements lists. Failure to address sensing objectives in CONSULT manifests as mis-specified sensor placement, blind spots in coverage, or unmapped environmental constraints.

**Table 4.** Overview of the CONSULT phase. Header color follows Figure 2 for visual distinction.

Phase	CONSULT–Engage and Define
Extends	<ul style="list-style-type: none"> <li>– ISO/IEC 22989: Inception;</li> <li>– CRISP-ML(Q): Business understanding (+sensor-derived constraints);</li> <li>– ISO 9241-210: Understand and specify context of use (Principle 1);</li> <li>– SE4ML: Problem formulation (+multi-stakeholder elicitation)</li> </ul>
Data Context	Data as a collaborative artifact [60]; stakeholder context defining who stakeholders are, what constraints govern data collection/use, and what trade-offs are acceptable

**Table 4.** *Cont.*

Phase	CONSULT–Engage and Define
Core Activities	Stakeholder analysis and mapping; constraint identification (ethics, legal, domain-specific, latency); model input/output requirements; preliminary KPI definition; collaborative requirements derivation
V&V Focus	Verification conditions framing; Explicit mapping to ISO/IEC 22989 inception phase
Decision Question	Who are the stakeholders and what are their requirements, information needs, constraints, and success criteria?
Key Outputs	Stakeholder maps; design principle collection; system and data requirements list

**Decision Gate 1: Transition to ARTICULATE**

Gate 1 (see Table 5) operationalizes comprehensive stakeholder landscape mapping and ML-specific constraint discovery before requirements formalization. It addresses sensor-specific constraints (calibration [14], mounting, synchronization [15]), data ownership complexities, multi-stakeholder structures, and AI-specific ethical considerations that standard Requirements Engineering processes do not systematically capture [1,48].

**Table 5.** Overview of Gate 1: CONSULT → ARTICULATE. Header color follows Figure 2.

Transition	CONSULT → ARTICULATE
ISE Challenge	Traditional Requirements Engineering lacks systematic capture of ML-specific constraints (sensor calibration, data ownership, multi-stakeholder structures, AI ethics)
Gate Trigger Criteria	<ul style="list-style-type: none"> <li>– Mandatory: All stakeholder groups identified, representatives consulted, domain constraints catalogued; project initiation approved</li> <li>– Quality: Stakeholder power mapping completed, KPIs identified for majority of groups</li> <li>– Documentation: Stakeholder map, design principle collection, requirements list</li> </ul>
Decision Question	Do we understand the stakeholder landscape sufficiently to formalize requirements?
Approvers	Project lead + senior stakeholder representative

The gate score  $\Theta_{G1}$  combines stakeholder mapping completeness  $c_{stakeholder\_map}$ , KPI coverage  $c_{KPI\_coverage}$ , and documentation completeness  $c_{doc\_complete}$ :

$$\Theta_{G1} = w_1 \cdot c_{stakeholder\_map} + w_2 \cdot c_{KPI\_coverage} + w_3 \cdot c_{doc\_complete}$$

where

- $c_{stakeholder\_map} \in [0, 1]$  measures stakeholder mapping completeness;
- $c_{KPI\_coverage} \in [0, 1]$  represents the fraction of stakeholder groups with identified KPIs;
- $c_{doc\_complete} \in [0, 1]$  assesses documentation completeness; and
- $w_j$  are project-specific weights satisfying  $\sum w_j = 1$ .

The gate proceeds when  $\Theta_{G1} \geq \tau_1$ , all mandatory criteria are satisfied, and required documentation artifacts are complete. Mandatory criteria include the following: project initiation approved, stakeholder mapping established, key stakeholder groups identified with representatives, and domain constraints catalogued. Required artifacts include the following: stakeholder map, design principle collection, and requirements list. Higher  $\tau_1$  values increase rigor but may delay transitions (Type II error: false negative, stifling

innovation); lower  $\tau_1$  values enable faster iteration but risk premature transitions (Type I error: false positive, allowing incomplete stakeholder understanding). Approval requires sign-off from the project lead and a senior stakeholder representative. The central decision question is: as follows: “Do we understand the stakeholder landscape sufficiently to formalize requirements?”

### 3.3.2. ARTICULATE: Formalize Requirements

ARTICULATE (see Table 6) transforms stakeholder-defined constraints into measurable, testable acceptance criteria, positioning **data as a SMART requirement** [55].

**Table 6.** Overview of the ARTICULATE phase. Header color follows Figure 2.

Phase	ARTICULATE—Formalize Requirements
Extends	<ul style="list-style-type: none"> <li>– ISO/IEC 22989: Early design;</li> <li>– CRISP-ML(Q): Data understanding (+formalization layer);</li> <li>– ISO 9241-210: Specify user requirements (Principle 2);</li> <li>– SE4ML: Data quality assessment (+formal requirements and contracts)</li> </ul>
Data Context	Data as a requirement; translation layer converting stakeholder-defined constraints into measurable, testable acceptance criteria (SMART [55])
Core Activities	Formalization of design principles and requirements; translation of stakeholder goals to SMART requirements; trade-off optimization; data contract establishment; KPI formalization
V&V Focus	Requirement verifiability; KPI definition for data, code, and models; technical and non-technical metrics specification
Decision Question	Are the requirements SMART? What are the trade-offs? What are the KPIs?
Key Outputs	Requirement matrices (SWOT, DSM, decision matrices, QFD); trade-off models (accuracy vs. interpretability, robustness vs. fairness, accuracy vs. privacy/energy)

Building on CONSULT, requirements are formalized, ensuring data management meets established desiderata (e.g., relevance, completeness, balance, accuracy) [50]. Design principles and requirements are captured using templates documenting rationale, including ethical principles [67] and explainability requirements [84]. Stakeholder goals translate into formal requirements [80], addressing RE4ML gaps, including lack of validated techniques [53] and challenges formalizing AI behaviors and explainability [48]. Systematic requirement trade-off optimization [85] resolves conflicts between stakeholders through explicit negotiation and documentation rather than implicit technical decisions. System and tool selection aligns with stakeholder priorities [86], producing documented tool choices and configuration specifications. Data contracts [25,26] formally specify structure, schema, format, and semantic expectations of data exchanged between systems or pipeline stages, serving as executable specifications for automated validation. However, data contracts often fail due to lack of automated validation or unclear ownership [59,87].

CAPTURE addresses this by coupling ARTICULATE contracts with executable validation rules in PROTOCOL, ensuring automated enforcement rather than manual compliance. Preliminary KPIs from CONSULT are formalized as measurable success criteria [24] with baseline values, target thresholds, measurement methodologies, and stakeholder ownership. Verification criteria are established for data, code, and models, including technical metrics (e.g., performance, reliability, precision, accuracy, drift [51,52], latency [12], safety) and non-technical metrics (e.g., explainability [23,81], fairness [88], reproducibility, telemetry).

For interactive ML systems, the Interactive Machine Learning Interaction Quality (IMLIQ) score [40] may measure human-model collaboration quality. Deliverables include requirement matrices employing methodologies such as SWOT analysis [89], Design Structure Matrices [90], or Quality Function Deployment [91], and trade-off models documenting inherent tensions (e.g., accuracy vs. interpretability [92], robustness vs. accuracy vs. fairness [93], accuracy vs. privacy [94], accuracy vs. energy consumption [95]). Failure in ARTICULATE manifests as unmeasurable latency requirements, undocumented synchronization assumptions, or calibration validity windows specified only implicitly.

Decision Gate 2: Transition to PROTOCOL

Gate 2 (see Table 7) supports verification that ML-specific requirements are formalized with stakeholder validation and documented trade-off models. It addresses RE4ML gaps in formalizing explainability [81], fairness [88], and data quality expectations [28,58]. This ensures ML system trade-offs (latency vs. accuracy, accuracy vs. interpretability, privacy vs. utility) are explicitly negotiated with stakeholders rather than optimized unilaterally.

The gate score  $\Theta_{G2}$  combines stakeholder validation fraction  $v_{stakeholder}$ , trade-off documentation completeness  $c_{tradeoff\_doc}$ , and requirement matrix quality  $c_{matrix\_complete}$ :

$$\Theta_{G2} = w_1 \cdot v_{stakeholder} + w_2 \cdot c_{tradeoff\_doc} + w_3 \cdot c_{matrix\_complete}$$

where

- $v_{stakeholder} \in [0, 1]$  represents the fraction of stakeholders validating requirements;
- $c_{tradeoff\_doc} \in [0, 1]$  measures trade-off documentation completeness;
- $c_{matrix\_complete} \in [0, 1]$  assesses requirement matrix quality; and
- $w_j$  are project-specific weights.

**Table 7.** Overview of Gate 2: ARTICULATE → PROTOCOL. Header color follows Figure 2.

Transition	ARTICULATE → PROTOCOL
ISE Challenge	RE4AI gaps in formalizing ML-specific requirements (explainability, fairness, data quality); trade-offs require explicit stakeholder negotiation
Gate Trigger Criteria	– Mandatory: Requirements are SMART, data contracts established, KPIs formalized with measurement methods, trade-offs documented – Quality: Requirements validated by decision authorities, data contracts include schema and quality thresholds – Documentation: Requirement matrix, trade-off models, data quality metric definitions
Decision Question	Can we commit to these requirements for this iteration, understanding they may evolve in EVOLVE phase?
Approvers	Requirements owners + data governance lead + stakeholder representatives

The gate proceeds when  $\Theta_{G2} \geq \tau_2$ , all mandatory criteria are satisfied, and required documentation artifacts are complete. Mandatory criteria include the following: all requirements satisfy SMART criteria [55], data contracts established for all sources, KPIs formalized with measurement methods [24], and trade-offs documented with resolution strategies. Required artifacts include requirement matrix, trade-off models, and data quality metric definitions. Higher  $\tau_2$  values ensure strong stakeholder consensus before committing resources to implementation but may delay design work; lower  $\tau_2$  values enable faster progression but risk proceeding with incompletely validated requirements. Approval requires

sign-off from requirements owners, data governance lead, and stakeholder representatives. The central decision question is as follows: “Can we commit to these requirements for this iteration, understanding they may evolve in EVOLVE phase?”

### 3.3.3. PROTOCOL: Document Decisions

PROTOCOL (see Table 8) establishes a decision provenance infrastructure [38] enabling evidence-based lifecycle governance, positioning **data as a traceable artifact**. Decision provenance tracks lifecycle decisions, their rationale, and their downstream effects using semantic data structures to extract [96], encode, and track [97,98] requirements and decisions throughout the ML lifecycle.

**Table 8.** Overview of the PROTOCOL phase. Header color follows Figure 2

Phase	PROTOCOL—Document Decisions
Extends	<ul style="list-style-type: none"> <li>– ISO/IEC 22989: Design and documentation;</li> <li>– CRISP-ML(Q): New capability (traceability not in CRISP-ML(Q));</li> <li>– ISO 9241-210: Produce design solutions (Principle 3);</li> <li>– SE4ML: New capability (explicit traceability and decision tracking)</li> </ul>
Data Context	Data as a traceable artifact
Core Activities	Decision tracking using semantic data structures; traceability maintenance across artifacts; template-based communication (model cards, dataset cards); metadata and label synchronization; data quality metrics definition; data provenance and lineage tracking
V&V Focus	Requirements verification; traceability validation
Decision Question	Are design decisions documented and traceable to stakeholder requirements?
Key Outputs	Versioned design documentation; traceability links to stakeholders and requirements; defined data rules enforcing data contracts

Beyond decision provenance, complementary artifact traceability [57] links stakeholder requirements to design decisions, data transformations, model architectures, validation results, and deployment configurations.

Provenance systems such as W3C PROV (<https://www.w3.org/TR/prov-overview/>, accessed on 18 December 2025) and data lineage tools such as DVC can provide machine-readable support for both decision provenance and artifact traceability. Standardized communication templates such as model cards [99] and dataset cards ensure consistent documentation patterns. Personalized architectural documentation addresses stakeholder-specific information needs [100], recognizing different stakeholders require different system views. Audit trails enable retrospective analysis of why specific architectural or algorithmic choices were made [101]. Appendix A provides a template for documenting gate transition decisions, ensuring systematic capture of decision rationale and approval chains. Comprehensive data quality metrics [28,58] cover completeness, accuracy, consistency, timeliness, validity, and uniqueness, with specified measurement methods for format compliance, schema adherence, labeling quality, and versioning practices.

Data provenance [77] tracks dataset origins, transformations, and processing, while data lineage [102,103] tracks dataset flow over time; model lineage combines both [35], enabling comprehensive impact analysis. Data governance frameworks [26,78] define policies, rules, and individual stewards responsible for implementation.

For sensor-based applications, physical-world attributes (calibration parameters, mounting configurations, spatial layouts [14,15]) are included in data descriptions. A Sensor Assumption Contract should explicitly document the following: expected sampling rates and acceptable jitter, calibration validity windows and recalibration triggers,

environmental operating bounds (temperature, humidity, lighting), and failure tolerances specifying graceful degradation behavior. This artifact has no equivalent in non-sensor ML systems. Deliverables include versioned design documentation [97], traceability links forming a comprehensive knowledge graph [104], and executable data rules enforcing contracts established in ARTICULATE.

Decision Gate 3: Transition to TERRAFORM

Gate 3 (see Table 9) supports verification that design decisions are documented with traceability links. This includes data quality metrics, data rules, versioning strategies, and provenance requirements, with non-trivial maintenance burdens. It addresses decision provenance tracking gaps in ML lifecycle frameworks [97,98] where CRISP-ML(Q) and SE4ML lack explicit mechanisms to document design rationale, creating reproducibility and auditability challenges.

**Table 9.** Overview of Gate 3: PROTOCOL → TERRAFORM. Header color follows Figure 2.

Transition	PROTOCOL → TERRAFORM
ISE Challenge	Decision provenance tracking gap in ML frameworks; lack of explicit design rationale documentation
Gate Trigger Criteria	– Mandatory: Decision tracking mechanism established, data quality metrics defined, data rules specified, provenance requirements documented – Quality: Traceability links created, sensor physical-world attributes documented, versioning schema designed – Documentation: Versioned design docs, traceability matrix (stakeholders → requirements → decisions), data governance policies
Decision Question	Do we have sufficient design documentation to build infrastructure that enforces our data contracts and governance policies?
Approvers	System architect + data governance lead

The gate score  $\Theta_{G3}$  combines six quality criteria measuring traceability, versioning, governance, and data management completeness:

$$\Theta_{G3} = w_1 \cdot c_{\text{traceability\_complete}} + w_2 \cdot c_{\text{versioning\_designed}} + w_3 \cdot c_{\text{governance\_doc}} + w_4 \cdot c_{\text{quality\_metrics}} + w_5 \cdot c_{\text{data\_rules}} + w_6 \cdot c_{\text{provenance\_doc}}$$

where

- $c_{\text{traceability\_complete}} \in [0, 1]$  measures the completeness of stakeholder-requirement-decision links;
- $c_{\text{versioning\_designed}} \in [0, 1]$  assesses versioning schema quality;
- $c_{\text{governance\_doc}} \in [0, 1]$  evaluates governance documentation completeness;
- $c_{\text{quality\_metrics}} \in [0, 1]$  quantifies data quality metric coverage across all six dimensions;
- $c_{\text{data\_rules}} \in [0, 1]$  evaluates completeness of executable data rules enforcing contracts;
- $c_{\text{provenance\_doc}} \in [0, 1]$  assesses provenance tracking specification completeness;
- $w_j$  are project-specific weights.

The gate proceeds when  $\Theta_{G3} \geq \tau_3$ , all mandatory criteria are satisfied, and required documentation artifacts are complete. Mandatory criteria include the following: decision tracking mechanism established, data quality metrics cover all six dimensions [28,58], data rules specified as executable constraints, and provenance tracking requirements documented [77]. Required artifacts include the following: versioned design documentation,

traceability matrix, and data governance policies. Higher  $\tau_3$  values ensure comprehensive traceability before infrastructure implementation but may delay development; lower  $\tau_3$  values enable faster progression but risk proceeding with incomplete documentation, undermining auditability. Approval requires sign-off from the system architect and data governance lead.

The central decision question is as follows: “Do we have sufficient design documentation to build infrastructure that enforces our data contracts and governance policies?”

### 3.3.4. TERRAFORM: Build Infrastructure

TERRAFORM phase (see Table 10), whose phase name derives from the verb “to terraform” (to shape terrain for habitation)—not the HashiCorp infrastructure tool, establishes the infrastructure foundation supporting ML development, deployment, and operations throughout the system lifecycle, positioning **data as an infrastructural flow**. Traditional ML frameworks conflate infrastructure setup with model development, leading to technical debt [31] and deployment challenges when infrastructure evolves independently of models. TERRAFORM treats infrastructure as a first-class lifecycle entity with its own requirements, versioning, and governance, acknowledging the “inertia of installed base” as organizational, technical, and financial investments in existing infrastructure that constrain and shape ML system design [31].

**Table 10.** Overview of the TERRAFORM phase. Header color follows Figure 2.

Phase	TERRAFORM—Build Infrastructure
Extends	<ul style="list-style-type: none"> <li>– ISO/IEC 22989: Development and integration;</li> <li>– CRISP-ML(Q): No analogue (infrastructure lifecycle not specified);</li> <li>– ISO 9241-210: Produce design solutions (Principle 3);</li> <li>– SE4ML: New capability (sensor pipelines and MLOps infrastructure)</li> </ul>
Data Context	Data as an infrastructural flow; focus on data pipeline architecture and movement rather than model-specific processing
Core Activities	MLOps automation setup (CI/CD); data pipeline architecture design; ETL pipeline setup; hardware and software provisioning; monitoring infrastructure establishment; governance framework implementation
V&V Focus	Dataflow integrity verification; automated governance flows
Decision Question	Is the infrastructure ready to support ML development and deployment according to stakeholder requirements?
Key Outputs	Functional infrastructure prototypes; versioning schema; deployment pipelines; benchmark data (latency, scalability)

Core MLOps principles [35,105] establish automation for recurring tasks, including data ingestion, artifact versioning, development workflows, deployment processes, testing procedures, and validation activities.

Example MLOps platforms with CI/CD support include Tekton (<https://tekton.dev>, accessed on 18 December 2025), Argo (<https://argoproj.io>, accessed on 18 December 2025), and MLflow (<https://mlflow.org>, accessed on 18 December 2025) with review workflows. Hardware and software are provisioned to fulfill performance requirements [106,107], ensuring infrastructure capabilities align with system needs. Modular, scalable data pipeline architectures [49] are built, with particular attention to sensor-based systems where data originates from physical devices with specific calibration, mounting, and environmental characteristics. ETL pipelines implement data rules and feature engineering strategies documented in PROTOCOL as executable constraints, ensuring data quality violations are detected automatically.

Audit trails [108], versioning systems, and monitoring infrastructure are integrated and enforced through governance frameworks. Monitoring is established for both KPIs [24] and data quality metrics [58], with potential triggers for automated retraining [18] when drift or degradation is detected. Infrastructure is designed for experimentation and rollout strategies through reproducible, interoperable design. Deliverables include functional prototypes, universal versioning schemas, automated deployment pipelines, and performance benchmarks (e.g., latency, throughput, scalability). Failure in TERRAFORM manifests as infrastructure-induced drift, resampling artifacts, or compromised temporal fidelity in edge deployments.

Decision Gate 4: Transition to UTILIZE

Gate 4 (see Table 11) supports verification that MLOps infrastructure is operational and stable before model development. This includes data pipelines, versioning systems, audit trails, and KPI monitoring. It addresses temporal misalignment where existing ML lifecycle frameworks treat infrastructure as an implicit precondition rather than an explicit phase [19,35,105], causing governance and quality controls to be implemented reactively rather than proactively.

The gate score  $\Theta_{G4}$  combines infrastructure stability, versioning system functionality, and performance benchmark completeness:

$$\Theta_{G4} = w_1 \cdot s_{\text{stability}} + w_2 \cdot c_{\text{versioning\_operational}} + w_3 \cdot c_{\text{benchmarks\_collected}}$$

where

- $s_{\text{stability}} \in [0, 1]$  measures infrastructure uptime stability over the observation period;
- $c_{\text{versioning\_operational}} \in [0, 1]$  assesses versioning system functionality;
- $c_{\text{benchmarks\_collected}} \in [0, 1]$  evaluates performance benchmark completeness;
- $w_j$  are project-specific weights.

**Table 11.** Overview of Gate 4: TERRAFORM → UTILIZE. Header color follows Figure 2.

Transition	TERRAFORM → UTILIZE
ISE Challenge	ML frameworks treat infrastructure as implicit precondition; governance implemented reactively rather than proactively
Gate Trigger Criteria	– Mandatory: Data pipeline functional, data rules enforced, KPI monitoring operational, data quality metrics tracked, infrastructure passes dataflow integrity tests – Quality: Versioning operational, audit trails functional, benchmark data collected, sensor streaming/buffering/synchronization operational – Documentation: Infrastructure docs, deployment pipelines, operational runbooks
Decision Question	Can this infrastructure reliably support ML model training and/or inference at required scale/performance?
Approvers	Infrastructure lead + MLOps engineer + data governance lead

The gate proceeds when  $\Theta_{G4} \geq \tau_4$ , all mandatory criteria are satisfied, and required documentation artifacts are complete. Mandatory criteria include data pipeline functional, data rules enforced in pipeline, KPI monitoring operational [24], data quality metrics tracked [58], and infrastructure passes dataflow integrity tests. Required artifacts include infrastructure documentation, deployment pipelines, and operational runbooks. Higher  $\tau_4$  values ensure robust infrastructure before model development but may delay experimentation; lower  $\tau_4$  values enable faster model iteration but risk building on unstable

foundations. Approval requires sign-off from the infrastructure lead, MLOps engineer, and data governance lead. The central decision question is as follows: “Can this infrastructure reliably support ML model training and/or inference at required scale/performance?”

### 3.3.5. UTILIZE: Implement ML Modules

UTILIZE (see Table 12) implements ML-specific functionality using infrastructure established in TERRAFORM, positioning **data as training input or inference input**. ML components and user-facing applications are built based on requirements formalized in ARTICULATE and infrastructure provided by TERRAFORM. Interfaces between end-users and ML components are provided by traditional software applications.

**Table 12.** Overview of the UTILIZE phase. Header color follows Figure 2.

Phase	UTILIZE—Implement ML Components
Extends	<ul style="list-style-type: none"> <li>– ISO/IEC 22989: Development and V&amp;V;</li> <li>– CRISP-ML(Q): Model engineering (+embedded V&amp;V);</li> <li>– ISO 9241-210: Evaluate solutions against requirements (Principle 4, part 1: model validation);</li> <li>– SE4ML: Model evaluation (+continuous V&amp;V across infrastructure and data)</li> </ul>
Data Context	Data as training input or inference input; focus on model-specific data processing
Core Activities	ML component development; model selection, training, fine-tuning, or inference; data management for ML tasks; traceability maintenance
V&V Focus	Model behavior validation; KPI verification; data quality monitoring during training and inference; drift detection
Decision Question	Do the ML models and applications meet the specified requirements and KPIs?
Key Outputs	Integrated dataflow pipelines; ML models, components, and decision mechanisms

These are developed using established software engineering and HCD best practices, with explicit versioning of linkages between application and ML component artifacts. Data is managed specifically for training and inference tasks [109], with attention to distinct requirements of each mode: training data supports model learning and generalization, while inference data matches the distribution and format expected by deployed model versions.

With infrastructure operational, models are developed through multiple pathways: training from scratch, fine-tuning pre-trained models, or deploying inference-only models. The model creation process distinguishes between HITL execution (direct human participation in labeling, adjudication, and refinement), HOTL oversight (supervisory oversight with the ability to intervene), and stakeholder requirement decisions (shaping fundamental system constraints and success criteria) [40,41]. Participatory modeling approaches [110] facilitate collaborative requirement determination by engaging stakeholders in iterative model conceptualization, enabling domain experts to contribute tacit knowledge without requiring deep technical ML expertise. Traceability links are maintained between data, code, models, applications, and requirements, ensuring every design decision can be traced to stakeholder needs and every requirement can be traced to implementation artifacts. Comprehensive validation verifies model behavior against requirements and KPIs established in earlier phases.

KPIs are verified systematically [24], checking adherence to accuracy thresholds, latency constraints, and fairness metrics. Data quality is monitored during training (completeness, consistency) and inference (drift, distribution shifts) [28]. Deliverables

include integrated dataflow pipelines, validated ML models, and deployment-ready decision mechanisms.

Decision Gate 5: Transition to REIFY

Gate 5 (see Table 13) addresses both training scenarios and inference-only scenarios with stakeholder approval, ensuring alignment with HCD principles [3]. Training scenarios include model development, validation, and KPI achievement. Inference-only scenarios include pre-trained model selection, integration testing, and domain-specific performance validation. This addresses how traditional ML lifecycles assume training-centric workflows, failing to accommodate model selection as a primary activity.

The gate score  $\Theta_{G5}$  combines KPI achievement, data quality, traceability, and reproducibility:

$$\Theta_{G5} = w_1 \cdot \min_i p_{KPI,i} + w_2 \cdot q_{data} + w_3 \cdot c_{traceability} + w_4 \cdot r_{reproducible}$$

where

- $p_{KPI,i} \in [0, 1]$  is the achievement level for KPI  $i$  in validation (fraction of target met);
- $q_{data} \in [0, 1]$  measures data quality during model development;
- $c_{traceability} \in [0, 1]$  assesses requirement-to-model traceability completeness;
- $r_{reproducible} \in \{0, 1\}$  indicates successful reproducibility verification;
- $w_j$  are project-specific weights.

**Table 13.** Overview of Gate 5: UTILIZE → REIFY. Header color follows Figure 2.

Transition	UTILIZE → REIFY
ISE Challenge	Traditional ML lifecycles assume training-centric workflows; fail to accommodate pre-trained model selection and domain-specific validation
Gate Trigger Criteria	– Mandatory: ML models trained/fine-tuned/selected and validated, KPIs met (or deviations documented with stakeholder approval), data quality metrics acceptable, model behavior validated, reproducibility verified – Quality: Traceability links established (requirements → data → code → models), model cards created, explainability and fairness requirements satisfied – Documentation: Integrated dataflow pipelines, ML model documentation, test reports
Decision Question	Is this model ready for deployment in real stakeholder contexts, understanding that real-world feedback may require iteration?
Approvers	ML engineer + domain expert + stakeholder representative

The gate proceeds when  $\Theta_{G5} \geq \tau_5$ , all mandatory criteria are satisfied, and required documentation artifacts are complete. Mandatory criteria include the following: ML models trained/fine-tuned/selected and validated, KPIs met or consciously accepted deviations documented with stakeholder approval [24], data quality metrics within acceptable ranges [58], model behavior validated through tests, and reproducibility verified. Required artifacts include integrated dataflow pipelines, ML model documentation, and test reports. Higher  $\tau_5$  values indicate more rigorous validation but delay stakeholder feedback; lower values permit faster testing but risk deploying under-validated models. Approval requires sign-off from the ML engineer, domain expert, and stakeholder representative. The central decision question is as follows: “Is this model ready for deployment in real stakeholder contexts, understanding that real-world feedback may require iteration?”

### 3.3.6. REIFY: Apply and Obtain Feedback

REIFY (see Table 14) deploys ML components in real-world stakeholder contexts and systematically collects evidence about system performance, user experience, and stakeholder satisfaction, positioning **data as a feedback artifact**. Applications are deployed in stakeholder contexts where they can be observed under realistic operating conditions, enabling collection of data reflecting actual usage patterns, edge cases, and stakeholder interactions. Automated insights are captured through monitoring infrastructure established in TERRAFORM, including system performance metrics, error rates, resource utilization, and behavioral patterns. Stakeholder and user feedback is collected systematically, recognizing the distinction between HITL execution, HOTL oversight, and stakeholder feedback.

HITL activities in REIFY include operational labeling and adjudication of ambiguous cases. HOTL activities include supervisory monitoring and human overrides when automated decisions are inappropriate. These activities are governed by stakeholder-defined rules from CONSULT and ARTICULATE. Insights from both automated monitoring and stakeholder feedback are fed back into the decision tracking infrastructure established in PROTOCOL, ensuring deployment insights are preserved and accessible.

**Table 14.** Overview of the REIFY phase. Header color follows Figure 2.

Phase	REIFY—Apply and Obtain Feedback
Extends	<ul style="list-style-type: none"> <li>– ISO/IEC 22989: Deployment and monitoring;</li> <li>– CRISP-ML(Q): Deployment (+feedback loops);</li> <li>– ISO 9241-210: Evaluate solutions against requirements (Principle 4, part 2: real-world feedback);</li> <li>– SE4ML: Monitoring (+systematic real-world feedback collection)</li> </ul>
Data Context	Data as feedback artifact [24,58]; real-world evidence collection
Core Activities	Stakeholder context deployment; automated insights capture; stakeholder/user feedback collection; real-world KPI measurement; production data quality monitoring
V&V Focus	Real-world outcome confirmation
Decision Question	Do we have sufficient real-world evidence to evaluate system performance and stakeholder satisfaction?
Key Outputs	Case studies and use cases; feedback datasets; updated stakeholder requirements

Real-world KPIs are measured in production [24], including user satisfaction scores, business outcomes, operational efficiency improvements, and other stakeholder-defined success metrics. Production data quality metrics [58] are monitored continuously to detect emerging issues before they compromise system reliability. Deliverables include case studies and use cases documenting real-world system deployment, stakeholder interactions, and observed outcomes, feedback datasets capturing both automated metrics and stakeholder input, and updated stakeholder requirements emerging as stakeholders interact with the deployed system and develop refined understanding of their needs. Failure in REIFY manifests as emergent sensor behavior under real-world environmental variance (e.g., lighting, temperature, vibration) not captured in controlled validation.

#### Decision Gate 6: Transition to EVOLVE

Gate 6 (see Table 15) ensures the collection of sufficient real-world observations before evaluation decisions are finalized in EVOLVE. This phase prioritizes both quantitative metrics and qualitative stakeholder feedback, even at the cost of delayed iteration. By doing so, it addresses a common gap in ML frameworks that conflate deployment with evaluation,

missing the critical separation between observation and decision-making. This separation is necessary for statistically valid conclusions and aligns with HCD principles [3] for iterative evaluation in authentic contexts.

The gate score  $\Theta_{G6}$  combines observation duration, interaction count, KPI compliance in production, and stakeholder feedback completeness:

$$\Theta_{G6} = w_1 \cdot r_{\text{obs}} + w_2 \cdot r_{\text{interactions}} + w_3 \cdot \min_i p_{\text{KPI},i} + w_4 \cdot f_{\text{feedback}}$$

where

- $r_{\text{obs}} = t_{\text{obs}}/t_{\text{required}}$  is the ratio of observation duration to required period;
- $r_{\text{interactions}} = n_{\text{interactions}}/n_{\text{required}}$  is the ratio of interactions to required count;
- $p_{\text{KPI},i} \in [0, 1]$  is the compliance level for KPI  $i$  in production (fraction of target met);
- $f_{\text{feedback}} \in [0, 1]$  measures stakeholder feedback completeness; and
- $w_j$  are project-specific weights.

**Table 15.** Overview of Gate 6: REIFY → EVOLVE. Header color follows Figure 2.

Transition	REIFY → EVOLVE
ISE Challenge	ML frameworks conflate deployment with evaluation; missing separation between feedback collection (observation) and impact assessment (decision-making)
Gate Trigger Criteria	– Mandatory: System operational in at least one stakeholder context, real-world KPIs measured over statistically significant period, stakeholder feedback gathered – Quality: Automated insights captured, feedback in decision tracking system, at least one complete use case documented – Documentation: Feedback datasets, real-world performance reports, updated stakeholder requirements (if emerged)
Decision Question	Do we have enough real-world data to evaluate system success and plan next iteration?
Approvers	Product owner + stakeholder representatives

The gate proceeds when  $\Theta_{G6} \geq \tau_6$ , with mandatory criteria satisfied and required documentation artifacts complete. Mandatory criteria include the following: system operational in at least one stakeholder context, real-world KPIs measured over a statistically significant period [24], production data quality metrics collected [58], and stakeholder feedback gathered. Required artifacts include feedback datasets, real-world performance reports, and updated stakeholder requirements. Higher  $\tau_6$  values ensure robust evidence before evaluation but may delay iteration decisions; lower  $\tau_6$  values enable faster iteration planning but risk premature conclusions from insufficient data. Approval requires sign-off from the product owner and stakeholder representatives. The central decision question is as follows: “Do we have enough real-world data to evaluate system success and plan the next iteration?”

### 3.3.7. EVOLVE: Evaluate and Iterate

EVOLVE (see Table 16) systematically evaluates real-world evidence collected during REIFY and makes informed decisions about iteration strategies, positioning **data as a learning opportunity**.

**Table 16.** Overview of the EVOLVE phase. Header color follows Figure 2.

Phase	EVOLVE—Evaluate and Iterate
Extends	<ul style="list-style-type: none"> <li>– ISO/IEC 22989: Re-evaluate (+triggers retirement);</li> <li>– CRISP-ML(Q): Monitoring and maintenance (+systematic re-evaluation);</li> <li>– ISO 9241-210: Evaluate solutions against requirements (Principle 4, part 3: iteration decisions);</li> <li>– SE4ML: Retraining (+drift detection and structured iteration paths)</li> </ul>
Data Context	Data as a learning opportunity; evidence for iteration decisions
Core Activities	Data-driven evaluation (technical and social impact); KPI achievement assessment; data quality trend analysis; requirements gathering for next iteration
V&V Focus	Model and concept drift assessment; decision impact evaluation
Decision Question	What action does the evidence warrant (major/minor iteration, model update, steady-state, or retirement)?
Key Outputs	Evaluation reports (quantitative and qualitative); lessons learned and best practices; updated models and data/system requirements

Using REIFY evidence, a comprehensive data-driven evaluation assesses both technical impact (latency, scalability, resource utilization, error rates) and social impact (stakeholder trust, user agency, satisfaction, adoption patterns). KPI achievement is evaluated systematically [24] to determine whether the system met stakeholder-defined success criteria, identifying unmet KPIs and analyzing root causes to inform adjustments. Lessons learned are documented for KPI refinement, recognizing that initial KPI definitions may require adjustment as stakeholders develop deeper understanding of system capabilities and limitations. Data quality metrics are analyzed for trends over time [58], identifying data quality issues that emerged during operations. Data contracts are assessed to determine whether they require revision based on real-world usage patterns, and data rules are updated based on observed validation failures or newly identified requirements. Based on evaluation findings, requirements for the next iteration are gathered from stakeholders, incorporating lessons learned from deployment. Updated KPIs reflect evolved stakeholder needs and refined understanding of system capabilities. Data contracts are refined to address identified gaps, and new or modified data quality metrics and data rules are specified to prevent recurring issues. This requirement gathering feeds directly back into CONSULT and ARTICULATE for subsequent iterations. It ensures CAPTURE operates as a true spiral model, where each iteration builds upon accumulated knowledge. Stakeholder sets should be treated as versioned artifacts, where changes trigger partial backward transitions scoped to affected phases rather than full lifecycle resets. Model drift [51] (changes in model performance over time) and concept drift [52] (changes in underlying relationships between inputs and outputs) are assessed. Complex dynamic systems may require a transition from data representation to operator representation as models evolve [111]. Deliverables include comprehensive evaluation reports combining quantitative metrics with qualitative stakeholder feedback, lessons learned and best practices, updated models reflecting refinements based on deployment experience, and updated data and system requirements incorporating evolved stakeholder needs.

**Decision Gate 7: Strategic Iteration Path Selection**

Gate 7 (see Table 17) provides five strategic options based on impact evaluation, enabling context-appropriate lifecycle decisions. It addresses unique ML lifecycle dynamics that traditional software maintenance models do not capture: concept drift requiring

retraining [51,52], incremental stakeholder requirement evolution, model updates [18,35], and graceful retirement scenarios when fitness thresholds fail [16].

**Table 17.** Overview of Gate 7: EVOLVE → multiple strategic paths. Header color follows Figure 2.

Transition	EVOLVE → Multiple strategic paths
ISE Challenge	Traditional software maintenance models do not capture multi-modal ML iteration strategies (continuous drift, automated updates, graceful retirement)
Gate Trigger Criteria	– Mandatory: Impact evaluation completed (technical + social), KPI achievement assessed, data quality trends analyzed, drift analysis performed – Quality: Lessons learned documented, stakeholder satisfaction assessed – Documentation: Evaluation reports (quantitative and qualitative), updated models and requirements (if iterating), retirement plan (if retiring)
Strategic Options	– 7(a): Major Iteration → CONSULT – 7(b): Minor Iteration → ARTICULATE – 7(c): Model Update → UTILIZE – 7(d): Continuous Monitoring – 7(e): Retirement
Decision Question	What iteration strategy does the evidence warrant?
Approvers (varies by option)	7(a): Steering committee; 7(b): Product owner + stakeholders; 7(c): ML engineer + operations lead; 7(d): Operations lead; 7(e): Steering committee + compliance officer

Path selection employs a two-stage decision process. First, system viability is assessed:

$$Viable = \begin{cases} \text{Yes} & \text{if } s_{\text{stakeholder}} \geq \tau_{\text{min}} \text{ and } r_{\text{KPI}} \geq \tau_{\text{critical}} \\ \text{No} & \text{otherwise} \end{cases}$$

If viable, iteration strategy is selected based on change scope  $\Delta r$  and drift magnitude  $d_{\text{drift}}$ :

$$Path = \begin{cases} 7a \text{ (Major Iteration)} & \text{if } \Delta r > \tau_{\text{major}} \\ 7b \text{ (Minor Iteration)} & \text{if } \tau_{\text{minor}} < \Delta r \leq \tau_{\text{major}} \\ 7c \text{ (Model Update)} & \text{if } d_{\text{drift}} > \tau_{\text{drift}} \\ 7d \text{ (Continuous Monitoring)} & \text{if drift and changes negligible} \\ 7e \text{ (Retirement)} & \text{if not viable} \end{cases}$$

where

- $s_{\text{stakeholder}} \in [0, 1]$  is the overall stakeholder satisfaction from REIFY feedback;
- $r_{\text{KPI}} = k_{\text{met}}/k_{\text{total}}$  is the ratio of KPIs meeting thresholds to total KPIs;
- $\Delta r \in [0, 1]$  measures fraction of requirements requiring modification;
- $d_{\text{drift}} \in [0, 1]$  quantifies model or concept drift magnitude [51,52]; and
- thresholds  $\tau_{\text{min}}, \tau_{\text{critical}}, \tau_{\text{major}}, \tau_{\text{minor}},$  and  $\tau_{\text{drift}}$  are project-specific, reflecting organizational risk tolerance and resource availability.

Higher thresholds increase rigor for iteration decisions but may delay necessary adaptations; lower thresholds enable faster response but risk premature or unnecessary changes. Mandatory prerequisites include completed impact evaluation across technical

and social dimensions [24], documented KPI achievement assessment, analyzed data quality trends [58], and performed drift analysis [51,52].

Note: Potential automation of certain EVOLVE→UTILIZE model-update transitions (Option 7c) can occur in mature MLOps pipelines, but automation is an operational option rather than the research contribution of this framework; when present, automated transitions should be governed by pre-defined quality gates and human oversight [18,35].

**Option 7a: Major Iteration (EVOLVE →CONSULT)** Returns to CONSULT when fundamental changes to system scope are warranted (new stakeholder groups, technology landscape shifts, use case pivots, or invalid core assumptions). Major iteration is *reactive* (addressing discovered deficiencies) versus proactive feature expansion (addressing new opportunities); both require returning to CONSULT. Approval requires the project steering committee sign-off. Decision question: “Do required changes warrant a full lifecycle restart with comprehensive stakeholder re-consultation?” Operational implications: large capital expenditure, re-budgeting, contract renegotiation, and extended stakeholder re-engagement [35].

**Option 7b: Minor Iteration (EVOLVE→ARTICULATE)** Returns to ARTICULATE for incremental refinements without fundamental stakeholder re-consultation when KPIs require adjustment, data contracts need refinement (not fundamental redesign), or requirements were partially misspecified but stakeholder groups remain stable. Addresses discovered issues rather than new feature requests (which should initiate a separate CAPTURE cycle at CONSULT). Approval requires product owner and stakeholder representative to sign-off. Decision question: “Can we iterate by refining requirements without re-consulting all stakeholders?” Operational implications: incremental operational expenditure, faster stakeholder sign-off loops, limited procurement, feature flags, and A/B experiments [18].

**Option 7c: Model Update (EVOLVE→UTILIZE)** Returns to UTILIZE when drift is detected or new training data was gathered but infrastructure and requirements remain valid (e.g., new training data available, scheduled routine retraining, or gradual performance degradation). May be automated in mature MLOps environments [18,35] with predefined quality gates and human oversight. Approval requires the ML engineer and operations lead to sign-off. Decision question: “Can we address issues through model retraining without infrastructure or requirement changes?” Operational implications: predictable recurring operational expenditure (e.g., data labeling, retraining compute, storage, monitoring) [18,35], potential SLA renegotiation, observable cost growth with data volume/update frequency, compliance and audit overhead in regulated domains [16].

**Option 7d: Continuous Monitoring** Maintains system in production without active changes when all KPIs are met, no significant drift is detected, stakeholders remain satisfied, and no new requirements have emerged. Approval requires operations lead’s sign-off. Decision question: “Can the system continue in production with routine monitoring?” Option 7d explicitly allows deliberate inaction when detected issues do not warrant intervention cost, signaling economic rationality over reflexive remediation. Operational implications: baseline monitoring costs (infrastructure uptime, automated alerting, periodic stakeholder feedback collection), active governance (monitoring dashboards, compliance obligations, incident response capabilities).

**Option 7e: Retirement** Implements ISO/IEC 22989 “Retirement” phase [16] when safety/maintainability thresholds are violated, fitness thresholds fail persistently (KPIs consistently unmet, persistent stakeholder dissatisfaction), business/project closure occurs, or technology obsolescence makes continued operation untenable. Approval requires project steering committee and compliance officer (if safety-critical) sign-off. Decision question: “Should this system be retired rather than iterated?” Operational implications: decommissioning costs (data migration, system archival, stakeholder communication, compliance documentation), retirement planning addressing data retention obligations,

model explainability requirements for historical decisions, and handover procedures for replacement systems.

### 3.3.8. Decision Gate Threshold Determination

Gates employ illustrative threshold values (qualified stakeholder validation, multi-day infrastructure stability tests, statistically significant evaluation periods) serving as **examples to guide thinking**, not prescriptive requirements. Unresolved conflicts should escalate to a designated lifecycle owner with documented rationale rather than requiring full consensus. Determining appropriate sensitivity for decision gates is a critical governance challenge. Gates that are too strict can stifle innovation and delay deployment (Type II error), while gates that are too loose can allow defective or harmful models into production (Type I error). Type I error (false positive) incorrectly concludes a positive effect/alarm when none exists, with operational costs including unnecessary retraining/rollouts, wasted budget, and stakeholder fatigue.

Type II error (false negative) fails to detect a true effect, with operational costs including missed degradations, latent failures, and regulatory/safety exposure. For example, a medical image classifier with a Gate 6 threshold ( $\tau_6$ ) set too loosely (e.g., requiring only 70% accuracy) may pass evaluation despite systematic misclassification of rare conditions, leading to patient harm. Conversely, an excessively strict threshold (e.g., 99.9% accuracy) on a consumer recommendation system may block beneficial updates indefinitely, causing a competitive disadvantage. Gate implications: choose thresholds based on domain risk—safety-critical systems should set very high  $\tau_x$  values with larger sample sizes/longer observation windows [24], while low-risk consumer features tolerate low threshold values to enable faster iteration.

*Practical rule:* explicitly record the statistical thresholds used for Gate 6 evaluation in the decision tracking system (from PROTOCOL) to ensure reproducibility of evaluation decisions. Actual threshold values must be determined through **domain-specific research**, considering organizational context (e.g., maturity level, risk tolerance, budget flexibility), domain requirements (e.g., safety-criticality, regulatory constraints, performance demands), project characteristics (e.g., complexity, temporal constraints, resource availability), and empirical validation (e.g., historical project data, industry benchmarks, pilot studies, expert judgment).

When implementing CAPTURE, teams should analyze context (e.g., identify factors influencing threshold requirements for their specific project), research precedents (e.g., review similar projects, domain standards, and organizational policies), engage stakeholders (e.g., collaborate with domain experts and governance bodies to establish appropriate thresholds), document rationale (e.g., record threshold choices and justifications for traceability and future refinement), and iterate based on evidence (e.g., adjust thresholds based on empirical outcomes from REIFY and EVOLVE phases).

CAPTURE suggests a risk-based approach to threshold determination, adapted from the NIST AI Risk Management Framework (<https://www.nist.gov/itl/ai-risk-management-framework>, accessed on 18 December 2025): Contextual risk scoring during CONSULT categorizes systems by impact severity (e.g., financial, safety, societal). Gate configurations scale dynamically: lower-risk tiers prioritize velocity through automated validation, while high-criticality systems mandate redundant Human-in-the-loop oversight and strict transition thresholds. Initial conservative thresholds are iteratively recalibrated using empirical evidence from EVOLVE to optimize the safety-velocity trade-off. This adaptive architecture enables CAPTURE to support both rapid research iterations and safety-critical industrial applications within a unified governance framework.

### 3.3.9. Lifecycle Transition Tracking

Systematic documentation of phase transitions ensures traceability, governance, and enables retrospective analysis, addressing the lack of decision provenance in existing ML frameworks [97,98]. By recording the rationale behind design choices and iteration triggers, teams maintain a ‘living artifact’ that evolves through decision gates. Transitions recorded in the PROTOCOL decision tracking system (see Section 3.3.3) should specify the following: (1) the transition decision and timestamp; (2) the rationale (e.g., exit criteria fulfillment or stakeholder feedback); (3) the scope of subsequent work; (4) authorized approvers; and (5) affected artifacts. Such an audit trail facilitates root cause analysis for backward transitions and clarifies strategic trajectories for EVOLVE multi-path decisions. Furthermore, automated MLOps transitions must log trigger conditions and quality gate results to maintain accountability and human oversight [18,35]. This structured approach transforms documentation from a passive record into an active governance mechanism that preserves the system evolution.

## 4. Results

This section provides empirical validation complementing the normative framework design presented in Section 3.2 through three methods. First, conceptual cognitive walkthroughs use hypothetical scenarios to validate the framework’s decision gate logic and phase structure (Section 4.1).

Second, a longitudinal case study applies CAPTURE retrospectively to a five-year sensor-based ML project (Section 4.2).

Third, semi-structured interviews with  $n = 10$  practitioners, conducted independently from the case study, assess CAPTURE’s design principles from an external perspective (Section 4.3).

### 4.1. Conceptual Cognitive Walkthrough

The following scenarios illustrate how CAPTURE’s decision gates guide lifecycle progression in an exemplary ML project context. Scenarios 1–4 demonstrate the framework’s completeness through diverse iteration patterns; Scenarios 5–7 validate phase granularity through conflation failure modes.

#### (1) Smooth Forward Progression

A computer vision model for defect detection passes all gates sequentially.

*Interpretation:* This ideal case occurs when CONSULT feasibility analysis accurately predicts technical and organizational constraints.

Gates 1–3	Requirements stable, data contracts verified.
Gate 4	Infrastructure stress test passed (72 h stability).
Gate 5	Model F1-score exceeds 0.95 on held-out test set.
Gate 6	Deployment successful, user feedback positive.
Gate 7	System transitions to continuous monitoring.

#### (2) Infrastructure Inadequacy Discovered

During UTILIZE, training repeatedly exceeds allocated memory, causing job failures.

*Interpretation:* Gate 5 rejection addresses the root cause (compute capacity) through backward transition to TERRAFORM.

Decision	Rejected due to mandatory criterion violation.
Transition	Return to TERRAFORM.
Action	Infrastructure team provisions GPUs with higher memory capacity.
Re-entry	Gate 4 re-evaluated for new infrastructure, then UTILIZE repeated.

- (3) **Requirements Gap Discovered in Production**  
 After Gate 6 approval, field usage reveals that lighting conditions are degrading performance.  
*Interpretation:* Gate 7 triggers minor iteration (7b) because the issue stems from an incomplete requirement, not model degradation. Multi-phase traversal demonstrates traceability.
- |           |   |
|-----------|---|
| Trigger   | Gate 7 evaluation shows a KPI dip due to unforeseen conditions.               |
| Decision  | Option 7b (minor iteration) returns to ARTICULATE.                            |
| Rationale | Low-light robustness was not specified; this is a requirement gap, not drift. |
| Action    | New requirement “low-light robustness” added.                                 |
| Flow      | ARTICULATE → PROTOCOL → UTILIZE (fine-tune on dark images).                   |
- (4) **System Retirement Due to Persistent Drift**  
 A prediction model degrades over 6 months despite two retraining cycles.  
*Interpretation:* Gate 7 identifies concept drift acceleration where retraining no longer maintains KPI thresholds.
- |          |  |
|----------|--|
| Trigger  | Gate 7 fitness threshold failure (KPI below 0.7 for 3 consecutive months). |
| Decision | Retirement path selected.  |
| Action   | System decommissioned, data archived, stakeholders notified.               |
- (5) **Observation–Evaluation Conflation (REIFY + EVOLVE)**  
 A recommendation system is deployed and the team evaluates early feedback.  
*Interpretation:* Without distinct REIFY and EVOLVE phases, teams make iteration decisions before statistically valid evidence accumulates.
- |             |  |
|-------------|--|
| Issue       | Team decides to retrain after 48 h of initial user data.   |
| Consequence | Early adopter behavior is non-representative; model oscillates through multiple retraining cycles. |
| Root Cause  | No temporal separation between evidence collection and evaluation.                                 |
- (6) **Stakeholder–Specification Conflation (CONSULT + ARTICULATE)**  
 Stakeholder feedback is gathered and immediately formalized into requirements.  
*Interpretation:* Without distinct CONSULT and ARTICULATE phases, informal statements are prematurely hardcoded into requirements.
- |             |  |
|-------------|--|
| Issue       | Stakeholder says “we need fast predictions”; formalized as “latency below 100 ms”.                                 |
| Consequence | Accuracy sacrificed for latency; stakeholder actually meant “fast enough for interactive use” (500 ms acceptable). |
| Root Cause  | No trade-off analysis or stakeholder validation before formalization.  |
- (7) **Specification–Documentation Conflation (ARTICULATE + PROTOCOL)**  
 Requirements are captured as informal specifications without explicit traceability.  
*Interpretation:* Without distinct ARTICULATE and PROTOCOL phases, requirements lack traceable design rationale.
- |             |   |
|-------------|---|
| Issue       | Model performs poorly after 6 months; team cannot trace which requirement drove problematic key design decisions. |
| Consequence | Root cause analysis impossible; requires re-interviewing stakeholders.  |
| Root Cause  | No versioned decision tracking linking requirements to design rationale.  |

### Synthesis: Validating Phase Granularity

These seven scenarios demonstrate that CAPTURE’s seven-phase structure provides appropriate granularity for stakeholder-aligned MLOps. Scenarios 1–4 illustrate a few exemplary iteration patterns: forward progression, backward transitions, multi-phase regression, and retirement. Scenarios 5–7 validate specific phase separations through conflation failure modes. Scenario 5 demonstrates premature iteration decisions when observation and evaluation are merged (REIFY/EVOLVE), while Scenario 6 reveals requirement misalignment when stakeholder engagement and formalization are conflated (CONSULT/ARTICULATE).

Scenario 7 shows traceability loss when specification and documentation lack explicit separation (ARTICULATE/PROTOCOL). Each phase addresses a specific ISE challenge with unique validation obligations and stakeholder configurations (see Section 3.2.2).

#### 4.2. Empirical Case Study: Longitudinal Framework Application

This case study shows how CAPTURE was applied in a five-year (2021–2025) sensor-based ML project developing a psychomotor learning system. This project was selected as an evaluation case due to its high sensor density and heterogeneous data streams. These factors necessitate complex multi-stakeholder coordination across technical and domain-specific roles. The project spanned multiple development cycles, each exercising CAPTURE’s decision gates, thus demonstrating the framework’s capacity to structure iterative refinement based on stakeholder feedback. Table 18 presents the chronological progression of activities mapped to CAPTURE phases.

**Table 18.** Chronological progression of research activities mapped to CAPTURE phases for the Multimodal Sensing and Immersive Feedback project (2021–2025).

Year	Publication	CAPTURE Phase(s)	Key Activities
2021	[112,113]	CONSULT, ARTICULATE, PROTOCOL, TERRAFORM	Initial stakeholder consultation, infrastructure conceptualization, security standards analysis, technology evaluation
	[114]	TERRAFORM, REIFY	MLOps pipeline implementation, real-time communication infrastructure
2022	[115]	ARTICULATE, REIFY	Feedback model design guidelines, ILS prototype deployment
	[116]	UTILIZE, EVOLVE	Motion comparison algorithm development for whole-skeleton similarity, limitation identification
	[117]	ARTICULATE	Conceptual semantic motion notation proposal (not implemented)
2023	[118]	TERRAFORM, REIFY	Session management infrastructure, ML-based feedback generation
	[119]	ARTICULATE, EVOLVE	Visualization techniques research and comparative analysis
	[120]	UTILIZE, EVOLVE	Motion comparison algorithm validation in sports, key pose limitation identification
2024	[121]	UTILIZE, REIFY, EVOLVE	Rule-based feedback engine implementation, keypoint detection integration
2025	[122]	REIFY, EVOLVE	Peer-assisted exergame prototype, gamification integration

##### 4.2.1. Cycles 1–2 (2021–2022): Foundation and Infrastructure

The project’s initial cycles established foundational infrastructure and stakeholder requirements [112,113]. Comprehensive stakeholder consultation identified privacy preservation, real-time feedback, and scalability as critical constraints.

**Decision Gates 1–3** validated stakeholder consensus, requirement completeness, and protocol readiness. The second cycle deployed an MLOps pipeline with data ingestion, processing, and feedback delivery components [114,115]. Data contracts formalized in **ARTICULATE** specified multimodal sensor data schemas, addressing fusion challenges that would otherwise emerge as implicit technical debt. **Decision Gate 4** confirmed infrastructure operational readiness. Qualitative feedback revealed limitations in initial visual overlay approaches, triggering **Gate 7 Option 7c** to pursue algorithmic refinement.

#### 4.2.2. Cycles 3–5 (2023–2025): Iterative Refinement and CAPTURE-Enabled Success

These cycles demonstrate CAPTURE's capacity to structure iterative refinement based on stakeholder feedback, guiding a transition from aggregate similarity metrics to targeted, rule-based assessment. The initial algorithmic approach produced aggregate similarity scores across entire motion sequences. CAPTURE's stakeholder engagement mechanisms in EVOLVE surfaced a critical gap: trainers could identify that movements differed but could not pinpoint which configurations required correction [120].

This feedback was not a feature request but a fundamental requirement gap, triggering **Gate 7 Option 7b** for rigorous validation rather than immediate expansion. Empirical validation in Cycle 4 confirmed the limitation: aggregate scores were not actionable for specific corrective guidance [119,120]. **Gate 7 Option 7c** authorized fundamental algorithm redesign. The final cycle implemented a rule-based feedback engine where domain experts defined validation constraints (joint angles, relative distances, timing) without programming expertise [121]. The REIFY phase integrated this through a web-based interface empowering teachers to translate domain expertise into executable rules.

User evaluation in EVOLVE confirmed successful generalization across multiple exercise domains. **Gate 7** managed divergent paths: production maintenance (**7b**), domain expansion (**7a**), with explicit rejection of retirement (**7e**) due to continued stakeholder value.

#### Validation Synthesis

The case study validates CAPTURE's three key mechanisms. First, stakeholder-centered engagement ensured domain requirements shaped technical decisions from inception. The evolution from aggregate similarity metrics to rule-based feedback exemplified responsive adaptation: trainers' expressed need for key-pose specification drove fundamental algorithm redesign rather than incremental parameter tuning. Second, decision gates provided explicit checkpoints for iteration decisions, with Gate 7 enabling strategic choices across multiple cycles. Gate 7's multi-path structure (Options 7a–7e) prevented premature retirement while enabling targeted iteration scoped to specific phases. Third, infrastructure and model development evolved on separate cadences, confirming the value of treating infrastructure as a distinct lifecycle entity. Infrastructure established in Cycle 2 supported all subsequent model iterations without fundamental redesign, validating TERRAFORM's separation principle. Without CAPTURE, stakeholder roles would likely have been conflated; this algorithmic limitation was framed as a visualization problem rather than a requirement gap, and iteration decisions were made without structured evaluation. The phase structure proved supportive rather than prescriptive: different cycles emphasized different phases based on project needs.

#### 4.3. Expert Interview Validation

Independent of the case study, semi-structured interviews with  $n = 10$  practitioners assessed CAPTURE's design principles. Participants included senior data scientists, ML researchers, HCI specialists, and industrial practitioners. The structured questionnaire covered eight topics: stakeholder engagement, role definition, stakeholder identification, participatory modeling, requirement verification, disagreement handling, action recording, and accessibility. Interview questions probed specific CAPTURE design decisions; responses were synthesized using structured topic-based analysis organized by framework phase. Analysis revealed strong alignment with CAPTURE's design and identified practical considerations.

### Stakeholder Engagement (CONSULT)

All ten experts affirmed that end-user involvement supports ML processes. Experts identified an “intermediate layer” between technical teams and stakeholders as essential for translating requirements across expertise boundaries. This validates CAPTURE’s separation of stakeholder requirement decisions from HITL execution. Stakeholder identification strategies emphasized impact analysis and role-based representation, with experts noting identification failure as a critical project risk.

### Participatory Modeling (ARTICULATE)

Nine of ten experts endorsed participatory modeling for determining requirements, noting it provides common discussion grounds and reveals emergent requirements. Experts uniformly acknowledged that complete requirement identification cannot be verified a priori. One expert emphasized the following: “Requirements should be co-authored in multiple iterations; no single session captures the full picture.” This supports CAPTURE’s spiral lifecycle model where EVOLVE feeds refined requirements back to ARTICULATE.

### Decision Tracking (PROTOCOL)

Experts identified multiple recording models: dedicated roles, distributed responsibility, and hybrid approaches. One practitioner advocated the following: “Log everything—automated systems should capture decisions without adding burden to developers.” This validates CAPTURE’s emphasis on decision tracking infrastructure established in PROTOCOL.

### Accessibility (UTILIZE)

Experts emphasized that model building requires data science expertise, but stakeholder involvement can be structured around inputs and outputs. Proposed mechanisms included explainable AI interfaces, visualization tools, and rule-based explanations. These findings support CAPTURE’s distinction between stakeholder governance and HITL execution.

### Iteration and Disagreement (EVOLVE)

Disagreement handling strategies ranged from value proposition alignment to A/B testing. Expert consensus that requirements verification is iterative supports CAPTURE’s positioning of EVOLVE as a distinct phase from deployment.

### Summary

Expert interviews validate CAPTURE’s core principles: stakeholder-centered engagement [1], explicit role definitions, participatory requirements with iterative refinement, decision tracking infrastructure, separation of governance from HITL execution, and structured iteration mechanisms.

## 5. Discussion

The longitudinal validation and expert interviews demonstrate how CAPTURE addresses the gaps outlined in Section 3.2.

### 5.1. Interpretation of Results

The longitudinal validation across five development cycles (see Section 4.2) and expert interviews (see Section 4.3) demonstrate how CAPTURE’s extensions address requirements engineering, sensor-specific challenges, continuous verification, infrastructure governance, and traceability. The following sub-subsections synthesize how each research objective was addressed through a framework application.

### 5.1.1. Stakeholder-Centered Foundation (RO1)

Stakeholder consultation shaped technical decisions from project inception rather than treating needs as post hoc constraints. The evolution from aggregate similarity metrics to rule-based feedback (Cycles 3–5) exemplified responsive adaptation to trainer-expressed needs for actionable, pose-specific guidance [120,121]. Expert interviews validated stakeholder engagement necessity across all ten respondents, emphasizing intermediate coordination layers bridging technical teams and domain experts. The participatory rule creation interface (Cycle 5) operationalized domain expertise without programming skills, enabling teachers to specify key poses, angle ranges, and timing requirements [121].

CAPTURE's distinction between stakeholder requirement decisions (governance) and operational HITL activities (labeling, refinement) prevents inappropriate delegation of consent boundaries and safety constraints to operational teams (see Section 3.3.1).

### 5.1.2. Sensor-Specific Engineering (RO2)

The longitudinal application revealed sensor-specific challenges that generic ML frameworks do not address systematically.

Temporal constraints in real-time feedback delivery informed architectural decisions during TERRAFORM, distinguishing edge processing for immediate corrective feedback from cloud processing for post-session analysis [114]. Multimodal fusion challenges across XSense IMU suits, Perception Neuron sensors, and Azure Kinect vision were addressed through data contracts establishing skeletal keypoint schemas, joint angle representations, and temporal sequences [113,123]. The PROTOCOL phase requirement to document physical-world attributes such as sensor calibration parameters, mounting configurations, and spatial layouts [14,15] exemplifies sensor-specific obligations absent from traditional ML frameworks.

### 5.1.3. Continuous Verification and Validation (RO3)

CAPTURE distributes V&V across all lifecycle phases rather than treating validation as a terminal gate. The longitudinal application illustrates phase-specific verification. This includes requirement validation in CONSULT/ARTICULATE, data contract verification in PROTOCOL, infrastructure integrity testing in TERRAFORM, model behavior validation in UTILIZE, real-world outcome assessment in REIFY, and drift analysis in EVOLVE. Decision gates provided explicit quality governance checkpoints, with Gate 7's multi-path structure enabling targeted iteration scoped to specific phases rather than binary continue/abandon decisions (see Section 'Decision Gate 7: Strategic Iteration Path Selection').

### 5.1.4. Infrastructure Lifecycle and Evidence-Based Governance (RO4, RO5)

Infrastructure evolution tracking through TERRAFORM's infrastructure-as-lifecycle-entity positioning prevented undocumented technical debt as message queuing, storage infrastructure, and MLOps pipelines evolved [114]. Traceability serves as the substrate for decision-making, not the intelligent element itself; the intelligence resides in the structured gates that interpret accumulated evidence.

Separating infrastructure provisioning from model development reflected fundamental differences in stakeholder roles, validation criteria, and evolution cadences. Decision tracking infrastructure established during PROTOCOL enabled audit trails across architectural evolution, algorithm redesigns, and deployment contexts. Lifecycle transition tracking maintained provenance linking stakeholder feedback through evaluation findings to subsequent design changes, addressing decision provenance gaps in existing frameworks [97,98].

Traceability proved essential for retrospective validation, enabling comparative analysis during EVOLVE to confirm algorithm selection rationale given the data requirements and dataset constraints.

### *5.2. Implications for Intelligent Software Engineering*

CAPTURE enables evidence-based lifecycle governance by structuring decisions explicitly, binding them to accumulated evidence, and maintaining full traceability. These properties are required for any form of intelligent decision support, whether human-centered or machine-assisted. Without an explicit lifecycle model, any “intelligent” tooling would merely automate ad hoc practices rather than improve decision quality.

### *5.3. Limitations and Threats to Validity*

This section addresses framework applicability boundaries, evaluation scope constraints, and validity considerations.

#### Construct Validity

Determining whether documented artifacts (stakeholder maps, data contracts, decision logs) represent stakeholder engagement quality rather than bureaucratic compliance poses measurement challenges. The case study demonstrates artifact utility through documented design decisions and iteration rationales. However, this does not empirically validate that artifact creation caused improved outcomes, as results may correlate with project maturity or team expertise.

#### Internal Validity

Definitive attribution of project success to CAPTURE adoption is prevented by confounding factors. These include research team ML expertise, stakeholder commitment, institutional support, and domain favorability. The lack of a controlled comparison between CAPTURE-guided development and alternative lifecycle approaches limits causal claims about framework effectiveness. The longitudinal case study prioritized documenting qualitative evolution patterns over quantitative process metrics such as deployment cycle times or rework rates. This evaluation establishes CAPTURE’s feasibility and stakeholder acceptance rather than quantitative superiority claims. Comparative studies measuring governance overhead, iteration velocity, and stakeholder satisfaction across CAPTURE-guided versus traditional ML development remain a priority for future empirical validation.

#### External Validity

The primary validation relies on a single longitudinal case study in psychomotor skill assessment (see Section 4.2). While the five-cycle evolution spanning 2021–2025 demonstrates CAPTURE’s capacity to guide multi-year development [112,114,120,121], the sensor-based focus may not generalize to purely data-driven ML contexts. Expert interviews (see Section 4.3) provide complementary validation through  $n = 10$  practitioners, yet this sample size limits statistical power for quantitative claims. Chronological documentation spanning multiple publication cycles provides external validation of the project timeline. CAPTURE’s applicability to higher-criticality domains (such as medical diagnostics, autonomous vehicles, or industrial control) remains theoretically grounded but empirically unvalidated. The framework design draws on principles applicable across sensor-based AI systems, but domain-specific calibration and regulatory compliance verification require targeted future studies.

### Calibration Burden

Decision gate thresholds require domain-specific calibration rather than universal prescription (see Section 3.3.8). This demands expertise and empirical validation to determine appropriate quality criteria, weights, and mandatory requirements. Organizations lacking mature MLOps practices may face adoption barriers.

### Comparison with Existing Approaches

CAPTURE's decision gate structure shares conceptual ancestry with Stage-Gate product development methodologies [124], which structure projects into phases separated by formal review points. Traditional Stage-Gate assumes linear progression with binary continue/stop decisions, whereas CAPTURE supports multi-path branching (Gate 7 Options 7a–7e) and backward iteration when limitations are discovered.

The gate scoring functions ( $\Theta_{Gx}$ ) formalize what Stage-Gate literature terms “gate criteria,” providing machine-readable specifications that enable potential automation of governance checks. Where participatory design emphasizes stakeholder involvement in specific design activities, CAPTURE operationalizes this principle across the entire ML lifecycle through explicit engagement phases (CONSULT, ARTICULATE) and feedback loops (REIFY, EVOLVE). Value Sensitive Design (VSD) [125] similarly integrates stakeholder values through conceptual, empirical, and technical investigations, though CAPTURE extends this with machine-readable gate criteria and explicit lifecycle phase transitions.

### 5.4. Applicability and Adoption Considerations

This section outlines the contexts where CAPTURE provides value, when existing frameworks may suffice, and practical considerations for organizational adoption.

#### Sensor-Specific Scope Criteria

CAPTURE is designed for systems where at least two of the following properties hold: continuous or high-frequency data streams, physically grounded uncertainty from calibration or environmental factors, temporal coupling between data acquisition and action, or safety/liability exposure from sensor failure.

Projects integrating multimodal sensor data, requiring temporal synchronization, and documenting physical-world attributes benefit most from CAPTURE's structured governance. Safety-critical and regulatory compliance contexts justify the framework's continuous V&V approach and decision provenance infrastructure.

#### Threshold-Calibrated Applicability

CAPTURE's applicability spans a spectrum calibrated through decision gate thresholds (see Section 3.3.8). Low threshold settings enable high-velocity projects where gates serve as lightweight reminders and decision traces rather than blocking checkpoints. High threshold values suit safety-critical or regulatory-sensitive applications requiring thorough governance at each transition. Bypassed provenance steps can be recorded as deliberate meta-decisions, ensuring technical debt remains explicit rather than untracked.

Table 19 summarizes how CAPTURE scales across three implementation tiers. Resource-constrained teams and small organizations can adopt Tier 1, progressively maturing to higher tiers as governance needs increase. Tier 1 activities represent the mandatory core for any CAPTURE adoption, while Tier 2 and Tier 3 activities are optional enhancements scaled to project criticality.

**Table 19.** CAPTURE implementation tiers calibrated by project context and risk profile.

Phase	Tier 1: Lightweight (Research Prototypes)	Tier 2: Standard (Industrial-Grade)	Tier 3: Full (Safety-Critical)
CONSULT	Informal mapping	Documented mapping	Formal audit trail
ARTICULATE	Lightweight reqs	Data contracts	Full V&V criteria
PROTOCOL	Minimal tracking	Decision tracking	Rigorous provenance
TERRAFORM	Ad hoc setup	CI/CD pipelines	Full MLOps governance
UTILIZE	Experimental	Validated models	Certified models
REIFY	Pilot deployment	Monitored ops	Audited operations
EVOLVE	Informal review	Structured gates	Governed transitions
Gate thresholds	$\tau = \text{Low}$	$\tau = \text{Medium}$	$\tau = \text{High}$

Table 20 illustrates how gate weight parameters ( $w_i$ ) and thresholds ( $\tau$ ) vary across these tiers. Using Gate 1’s criteria ( $c_{\text{stakeholder\_map}}$ ,  $c_{\text{KPI\_coverage}}$ ,  $c_{\text{doc\_complete}}$ ) as an example: research prototypes may prioritize documentation completeness ( $w_3$ ) over stakeholder coverage ( $w_1$ ) to enable rapid iteration, while safety-critical medical systems require comprehensive stakeholder mapping before proceeding.

**Table 20.** Example weight configurations for Gate 1 ( $\Theta_{G1}$ ) across implementation tiers.

Domain	Tier	$w_1$ (Stakeholder)	$w_2$ (KPI)	$w_3$ (Doc)	$\tau_1$ (Threshold)
Research prototype	Lightweight	0.2	0.3	0.5	0.5
Industrial analytics	Standard	0.4	0.3	0.3	0.7
Medical rehabilitation	Full	0.5	0.3	0.2	0.85

### Reduction to Simpler Frameworks

For systems with static, curated inputs lacking strict real-time constraints, or those driven by singular business KPIs without multi-stakeholder alignment, CAPTURE reduces to a form equivalent to CRISP-ML(Q) or ISO/IEC 22989. Projects with single stakeholders, static datasets, or minimal governance needs may find these established frameworks adequate without CAPTURE’s overhead. Prototyping phases before stakeholder commitment can appropriately defer comprehensive governance until deployment decisions solidify.

### Trade-Off Acknowledgment

CAPTURE increases coordination and documentation overhead. The framework trades upstream rigor for reduced downstream failure, rework, and regulatory exposure. Organizations must weigh governance benefits against velocity costs based on their specific risk profiles and stakeholder complexity. While CAPTURE does not provide a formal cost model, the tiered approach (Table 19) enables organizations to calibrate governance investment to project risk. Higher-tier implementations incur greater upfront documentation and coordination costs but reduce downstream failure rates and stakeholder rework. The case study illustrates this: transitioning to rule-based feedback (Cycles 3–5) required additional ARTICULATE documentation but eliminated a requirement gap that would have persisted without stakeholder-centered governance. Without CAPTURE’s structured CONSULT phase, this algorithmic limitation would likely have been framed as a visualization problem rather than a requirement gap, resulting in continued iteration without addressing the root cause.

### Generalizability

While validation focuses on sensor-based systems, CAPTURE's stakeholder-centered foundations (RO1, RO3–RO5) address governance challenges documented across diverse ML domains [48,53]. Non-sensor systems could adopt CAPTURE while simplifying sensor-specific TERRAFORM elements. The framework intentionally remains model-agnostic, accommodating diverse paradigms without prescribing how model structure should connect to lifecycle phases. Multi-agent system governance remains outside the current scope. Cross-domain empirical validation remains necessary.

### Organizational Readiness

Successful adoption requires MLOps infrastructure maturity, stakeholder engagement culture, and documentation practices [24,58]. Incremental deployment beginning with CONSULT and ARTICULATE can establish engagement patterns before introducing decision tracking complexity.

Training and change management interventions addressing the paradigm shift from data-scientist-centered to stakeholder-centered development are essential, particularly for teams accustomed to traditional ML workflows [1]. Pilot projects can help to enable empirical threshold calibration before widespread deployment [1]. Appendix B provides an adoption checklist to guide incremental deployment, reducing artifact creation overhead. Automated governance tools can support CAPTURE adoption by implementing gate scoring functions ( $\Theta_{G_x}$ ) as executable checks within MLOps pipelines, reducing manual compliance burden while maintaining structured oversight.

## 6. Conclusions

This paper presented CAPTURE, a seven-phase stakeholder-aligned lifecycle framework for sensor-based AI systems. CAPTURE addresses documented gaps in four established frameworks through stakeholder-centered requirements engineering, infrastructure lifecycle management, continuous V&V, and sensor-specific support (see Section 3.2). Longitudinal validation through a psychomotor skill assessment system (2021–2025) exercised all seven phases across five development cycles (see Section 4.2).

The case study demonstrated backward transitions, infrastructure evolution that is independent of model development, and simultaneous pursuit of divergent evolution paths. Expert interviews with ten ML practitioners validated the framework's stakeholder-centered foundations and identified the necessity of intermediate coordination layers between technical teams and domain experts (see Section 4.3).

### 6.1. Contributions

CAPTURE contributes to Intelligent Software Engineering by externalizing tacit lifecycle knowledge into explicit decision structures and traceability substrates, providing risk containment mechanisms for lifecycle decisions under uncertainty in sensor-based AI systems through four contributions. First, the CONSULT, ARTICULATE, and PROTOCOL phases operationalize ISO 9241-210's HCD principles with sensor-ML-specific mechanisms. These include data contracts, multimodal provenance requirements, and temporal constraint specifications.

The framework extends established standards to enable incremental adoption, prioritizing stakeholder alignment before introducing infrastructure governance (see Section 3.2). Second, twelve distinct lifecycle transition points provide systematic quality governance through seven primary gates for sequential progression and five strategic evolution options structuring iteration paths. Lifecycle transition tracking enables decision provenance linking stakeholder feedback to subsequent design changes (see Section 3.3).

Third, CAPTURE provides a systematic basis for managing sensor-specific challenges like temporal synchronization and multimodal fusion through dedicated artifacts and validation criteria (see Section 4).

## 6.2. Future Work

Validation of CAPTURE's core principles provides a foundation for the following targeted extensions. These directions represent scope expansions rather than corrections to identified weaknesses. Future development should address formal artifact models specifying linkage rules and propagation logic to enable automated consistency checking across decision provenance chains. Extending the adoption checklist (Appendix B) and gate template (Appendix A) into validated, domain-specific instantiations would further lower adoption barriers. Furthermore, tool implementations supporting CAPTURE phases and decision gate automation will enable operationalization of the framework for widespread application. Finally, maturity models for adoption assessment could assist organizations in gauging readiness and planning incremental deployment.

CAPTURE provides a foundation for several research directions. The decision provenance infrastructure [38] established through PROTOCOL creates the substrate for AI-assisted lifecycle governance. Intelligent agents could recommend iteration paths, support documentation efforts, flag potential requirement conflicts, or predict gate outcomes based on historical patterns. The explicit gate structures with quantitative scoring functions enable formal model-based verification of ML lifecycle decisions, supporting compliance demonstration in regulated domains. Extended case studies across higher-criticality sensor-based systems such as medical diagnostics, autonomous vehicles, Industry 5.0 manufacturing and smart city infrastructure would validate CAPTURE's scalability to safety-critical contexts.

These studies would also inform threshold calibration for these domains. While CAPTURE's stakeholder-centered governance principles apply broadly, the sensor-specific artifacts primarily address continuous dataflow scenarios. Adaptation to non-sensor-based but high-risk AI systems requires substituting sensor-specific artifacts with domain-appropriate equivalents while preserving the underlying governance mechanisms. Comparative studies between CAPTURE-guided and traditional ML development across diverse domains would provide empirical evidence for the framework's governance benefits relative to its coordination costs.

**Author Contributions:** Conceptualization: M.S.; data curation: M.S.; formal analysis: M.S.; funding acquisition: S.D.; investigation: M.S.; methodology: M.S.; project administration: M.S.; resources: M.S.; software: M.S.; supervision: R.R. and S.D.; validation: M.S. and R.R.; visualization: M.S.; writing—original draft: M.S.; writing—review and editing: M.S., R.R., and S.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by BAFA grant number 46SKD0233E.

**Institutional Review Board Statement:** Ethical review and approval were waived for this study due to its use of fully anonymous, voluntary interviews on a non-sensitive topic, which posed no risks beyond daily life and collected no personal or sensitive data as defined by GDPR. The research did not involve medical procedures, thus falling outside the scope of the Declaration of Helsinki and relevant national regulations in Germany and the U.S., which exempt such non-medical, low-risk studies from ethics committee oversight.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

**Acknowledgments:** During the preparation of this manuscript, the author(s) used ChatGPT 5 and Google Gemini 3 for the purposes of language improvement. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
CI/CD	Continuous Integration, Continuous Deployment
HCD	Human-Centered Design
HITL	Human-in-the-loop
HOTL	Human-on-the-loop
ISE	Intelligent Software Engineering
KPI	Key Performance Indicator
ML	Machine Learning
MLOps	Machine Learning Operations
RE4ML	Requirement Engineering for Machine Learning
SE4ML	Software Engineering for Machine Learning
V&V	Verification and Validation
XAI	Explainable AI

## Appendix A. Decision Gate Template

---

**Phase Transition:** \_\_\_\_\_ → \_\_\_\_\_

**Date:** \_\_\_\_\_ **Approvers:** \_\_\_\_\_

---

### Mandatory Criteria

- \_\_\_\_\_
  - \_\_\_\_\_
- 

### Quality Gates

- \_\_\_\_\_
- \_\_\_\_\_

*Warnings/Deviations:* \_\_\_\_\_

---

### Documentation Gates

- \_\_\_\_\_
  - \_\_\_\_\_
- 

### Decision

- APPROVED** - Proceed to next phase
  - CONDITIONAL** - Proceed with documented risks/deviations
  - REJECTED** - Return to phase [\_\_\_\_] for [reason]
- 

**Rationale:** \_\_\_\_\_

---

## Appendix B. CAPTURE Adoption Checklist

### CONSULT—Engage and Define

- Map stakeholder groups (roles, influence, information needs)
- Define constraints (ethical, legal, latency, privacy, explainability)
- Establish preliminary KPIs (technical + social impact metrics)

### ARTICULATE—Formalize Requirements

- Formalize SMART requirements with acceptance criteria
- Establish data contracts (e.g., structure, schema, quality expectations)
- Document trade-offs (e.g., accuracy vs. interpretability, privacy vs. utility)
- Formalize KPIs with baselines, thresholds, measurement methods

### PROTOCOL—Document Decisions

- Establish decision tracking (e.g., ADRs, rationale, alternatives)
- Define traceability links (stakeholders ↔ reqs ↔ design)
- Specify data provenance and quality metrics
- Implement data rules as executable specifications

### TERRAFORM—Build Infrastructure

- Design modular, scalable pipeline architecture
- Establish versioning for infrastructure, data, pipelines, models
- Implement MLOps automation (CI/CD)
- Configure monitoring infrastructure and automated alerting
- Test end-to-end dataflow integrity

### UTILIZE—Implement ML Modules

- Develop ML components or integrate pre-trained models
- Build application interfaces with explainability features
- Maintain artifact versioning (apps ↔ models ↔ data ↔ requirements)
- Validate model behavior against KPI thresholds
- Implement drift detection and data quality monitoring

### REIFY—Apply and Obtain Feedback

- Deploy in real-world stakeholder contexts
- Establish automated insights capture and feedback mechanisms
- Measure production KPIs with sufficient observation periods
- Monitor data quality and drift in production

### EVOLVE—Evaluate and Iterate

- Assess technical and social impact comprehensively
- Analyze KPI achievement and conduct root cause analysis
- Evaluate drift indicators and data quality trends
- Select iteration path (major/minor iteration, update, or retirement)
- Document iteration decision rationale

### Cross-Cutting Guidance

- Design transition gates with mandatory/quality criteria
- Schedule stakeholder review cadence aligned with decision gates
- Instrument for automated traceability
- Start incrementally with CONSULT and ARTICULATE
- Adapt to project context (simplify where non-applicable)

## References

1. Yalley, P.P. Limitations of Maximizing Shareholder Value in the Era of Artificial Intelligence and Machine Learning. *World J. Adv. Res. Rev.* **2025**, *28*, 1816–1825. <https://doi.org/10.30574/wjarr.2025.28.1.3600>.
2. Studer, S.; Bui, T.B.; Drescher, C.; Hanuschkin, A.; Winkler, L.; Peters, S.; Müller, K.R. Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 392–413. <https://doi.org/10.3390/make3020020>.
3. *ISO 9241-210:2019; Human-Centred Design for Interactive Systems*. ISO Publishing: Geneva, Switzerland, 2019.
4. ModelOP. Responsible AI-Benchmark Report 2024. 2024. Available online: <https://www.modelop.com/resources-ebooks/responsible-ai-report-2024> (accessed on 18 December 2025).
5. Sharma, A.; Sharma, V.; Jaiswal, M.; Wang, H.C.; Jayakody, D.N.K.; Basnayaka, C.M.W.; Muthanna, A. Recent Trends in AI-Based Intelligent Sensing. *Electronics* **2022**, *11*, 1661. <https://doi.org/10.3390/electronics11101661>.
6. Rogan, J.; Bucci, S.; Firth, J. Health Care Professionals' Views on the Use of Passive Sensing, AI, and Machine Learning in Mental Health Care: Systematic Review with Meta-Synthesis. *JMIR Ment. Health* **2024**, *11*, e49577. <https://doi.org/10.2196/49577>.
7. Henzler, D.; Schmidt, S.; Koçar, A.; Herdegen, S.; Lindinger, G.L.; Maris, M.T.; Konopka, M.J. Healthcare professionals' perspectives on artificial intelligence in patient care: A systematic review of hindering and facilitating factors on different levels. *BMC Health Serv. Res.* **2025**, *25*, 633. <https://doi.org/10.1186/s12913-025-12664-2>.
8. Mat Sanusi, K.A.; Iren, D.; Fanchamps, N.; Geisen, M.; Klemke, R. Virtual virtuoso: A systematic literature review of immersive learning environments for psychomotor skill development. *Educ. Technol. Res. Dev.* **2025**, *73*, 909–949. <https://doi.org/10.1007/s11423-025-10449-2>.
9. Karampatzakis, D.; Fominykh, M.; Fanchamps, N.; Firssova, O.; Amanatidis, P.; Van Lankveld, G.; Klemke, R. Educational Robotics at Schools Online with Augmented Reality. In Proceedings of the 2024 IEEE Global Engineering Education Conference (EDUCON), Kos Island, Greece, 8–11 May 2024; IEEE: New York, NY, USA, 2024; pp. 1–10. <https://doi.org/10.1109/educon60312.2024.10578583>.
10. Chander, B.; Pal, S.; De, D.; Buyya, R. Artificial Intelligence-based Internet of Things for Industry 5.0. In *Artificial Intelligence-Based Internet of Things Systems*; Pal, S., De, D., Buyya, R., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2022; pp. 3–45. [https://doi.org/10.1007/978-3-030-87059-1\\_1](https://doi.org/10.1007/978-3-030-87059-1_1).
11. Nguyen, T.T.H.; Nguyen, P.T.L.; Wachowicz, M.; Cao, H. MACeIP: A Multimodal Ambient Context-Enriched Intelligence Platform in Smart Cities. In Proceedings of the 2024 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Danang, Vietnam, 3–6 November 2024; IEEE: New York, NY, USA, 2024; pp. 1–4. <https://doi.org/10.1109/ICCE-Asia63397.2024.10774046>.
12. Chen, G.; Wang, X. Performance Optimization of Machine Learning Inference under Latency and Server Power Constraints. In Proceedings of the 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), Bologna, Italy, 10–13 July 2022; IEEE: New York, NY, USA, 2022; pp. 325–335. <https://doi.org/10.1109/icdcs54860.2022.00039>.
13. Lin, K.; Li, Y.; Sun, J.; Zhou, D.; Zhang, Q. Multi-sensor fusion for body sensor network in medical human–robot interaction scenario. *Inf. Fusion* **2020**, *57*, 15–26. <https://doi.org/10.1016/j.inffus.2019.11.001>.
14. Marotta, L.; Buurke, J.H.; van Beijnum, B.J.F.; Reenalda, J. Towards Machine Learning-Based Detection of Running-Induced Fatigue in Real-World Scenarios: Evaluation of IMU Sensor Configurations to Reduce Intrusiveness. *Sensors* **2021**, *21*, 3451. <https://doi.org/10.3390/s21103451>.
15. Höschler, L.; Halmich, C.; Schranz, C.; Koelewijn, A.D.; Schwameder, H. Evaluating the Influence of Sensor Configuration and Hyperparameter Optimization on Wearable-Based Knee Moment Estimation During Running. *Int. J. Comput. Sci. Sport* **2025**, *24*, 80–106. <https://doi.org/10.2478/ijcss-2025-0014>.
16. *ISO/IEC 22989:2022; Artificial Intelligence Concepts and Terminology*. ISO Publishing: Geneva, Switzerland, 2022.
17. Amershi, S.; Begel, A.; Bird, C.; DeLine, R.; Gall, H.; Kamar, E.; Zimmermann, T. Software Engineering for Machine Learning: A Case Study. In Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Montreal, QC, Canada, 25–31 May 2019; IEEE: New York, NY, USA, 2019; pp. 291–300. <https://doi.org/10.1109/icse-seip.2019.00042>.
18. Lima, A.; Monteiro, L.; Furtado, A. MLOps: Practices, Maturity Models, Roles, Tools, and Challenges—A Systematic Literature Review. In Proceedings of the 24th International Conference on Enterprise Information Systems, Virtual Event, 25–27 April 2022; SCITEPRESS-Science and Technology Publications: Setubal, Portugal, 2022; pp. 308–320. <https://doi.org/10.5220/0010997300003179>.
19. Diaz-de Arcaya, J.; Torre-Bastida, A.I.; Zárate, G.; Miñón, R.; Almeida, A. A Joint Study of the Challenges, Opportunities, and Roadmap of MLOps and AIOps: A Systematic Survey. *ACM Comput. Surv.* **2024**, *56*, 1–30. <https://doi.org/10.1145/3625289>.
20. Hanchuk, D.; Semerikov, S. Automating Machine Learning: A Meta-Synthesis of MLOps Tools, Frameworks and Architectures. In Proceedings of the 7th Workshop for Young Scientists in Computer Science & Software Engineering (CS&SE@SW 2024). CEUR Workshop Proceedings, Kryvyi Rih, Ukraine, 27 December 2024; Volume 3917, pp. 362–414.
21. Strong, B.; Boyda, E.; Kruse, C.; Ingold, T.; Maron, M. Digital applications unlock remote sensing AI foundation models for scalable environmental monitoring. *Front. Clim.* **2025**, *7*, 1520242. <https://doi.org/10.3389/fclim.2025.1520242>.

22. Memarian, B.; Doleck, T. Fairness, Accountability, Transparency, and Ethics (FATE) in Artificial Intelligence (AI) and higher education: A systematic review. *Comput. Educ. AI* **2023**, *5*, 100152. <https://doi.org/10.1016/j.caeai.2023.100152>.
23. Ali, S.; Abuhmed, T.; El-Sappagh, S.; Muhammad, K.; Alonso-Moral, J.M.; Confalonieri, R.; Herrera, F. Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Inf. Fusion* **2023**, *99*, 101805. <https://doi.org/10.1016/j.inffus.2023.101805>.
24. Eken, B.; Pallewatta, S.; Tran, N.; Tosun, A.; Babar, M.A. A Multivocal Review of MLOps Practices, Challenges and Open Issues. *ACM Comput. Surv.* **2025**, *58*, 1–35. <https://doi.org/10.1145/3747346>.
25. Truong, H.L.; Comerio, M.; de Paoli, F.; Gangadharan, G.R.; Dustdar, S. Data contracts for cloud-based data marketplaces. *Int. J. Comput. Sci. Eng.* **2012**, *7*, 280. <https://doi.org/10.1504/ijcse.2012.049749>.
26. Wider, A.; Harrer, S.; Dietz, L.W. AI-Assisted Data Governance with Data Mesh Manager. In Proceedings of the 2025 IEEE International Conference on Web Services (ICWS), Helsinki, Finland, 7–12 July 2025; IEEE: New York, NY, USA, 2025; pp. 963–965. <https://doi.org/10.1109/icws67624.2025.00126>.
27. Rangineni, S. An Analysis of Data Quality Requirements for Machine Learning Development Pipelines Frameworks. *Int. J. Comput. Trends Technol.* **2023**, *71*, 16–27. <https://doi.org/10.14445/22312803/ijctt-v71i8p103>.
28. ISO/IEC 5259:2024; Data Quality for Analytics and Machine Learning (ML). ISO Publishing: Geneva, Switzerland, 2024.
29. Star, S.L. The Ethnography of Infrastructure. *Am. Behav. Sci.* **1999**, *43*, 377–391. <https://doi.org/10.1177/00027649921955326>.
30. Pipek, V.; Wulf, V. Infrastructuring: Toward an Integrated Perspective on the Design and Use of Information Technology. *J. Assoc. Inf. Syst.* **2009**, *10*, 447–473. <https://doi.org/10.17705/1jais.00195>.
31. Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Dennison, D. Hidden Technical Debt in Machine Learning Systems. In Proceedings of the 29th International Conference on Neural Information Processing Systems-Volume 2, Montreal, QC, Canada, 7–12 December 2015; MIT Press: Cambridge, MA, USA, 2015; pp. 2503–2511.
32. Volmar, A. From Systems to “Infrastructuring”: Infrastructure Theory and Its Impact on Writing the History of Media. In *Rethinking Infrastructure Across the Humanities*; Pinnix, A., Volmar, A., Esposito, F., Binder, N., Eds.; Transcript Verlag: Bielefeld, Germany, 2023; pp. 51–64. <https://doi.org/10.1515/9783839469835-003>.
33. Lwakatara, L.E.; Raj, A.; Bosch, J.; Olsson, H.H.; Crnkovic, I. A Taxonomy of Software Engineering Challenges for Machine Learning Systems: An Empirical Investigation. In *Agile Processes in Software Engineering and Extreme Programming*; Kruchten, P., Fraser, S., Coallier, F., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; Volume 355, pp. 227–243. [https://doi.org/10.1007/978-3-030-19034-7\\_14](https://doi.org/10.1007/978-3-030-19034-7_14).
34. Martínez-Fernández, S.; Bogner, J.; Franch, X.; Oriol, M.; Siebert, J.; Trendowicz, A.; Wagner, S. Software Engineering for AI-Based Systems: A Survey. *ACM Trans. Softw. Eng. Methodol.* **2022**, *31*, 1–59. <https://doi.org/10.1145/3487043>.
35. Kreuzberger, D.; Köhl, N.; Hirschl, S. Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access* **2023**, *11*, 31866–31879. <https://doi.org/10.1109/access.2023.3262138>.
36. Zarour, M.; Alzabut, H.; Al-Sarayreh, K.T. MLOps best practices, challenges and maturity models: A systematic literature review. *Inf. Softw. Technol.* **2025**, *183*, 107733. <https://doi.org/10.1016/j.infsof.2025.107733>.
37. Serban, A.; van der Blom, K.; Hoos, H.; Visser, J. Software engineering practices for machine learning—Adoption, effects, and team assessment. *J. Syst. Softw.* **2024**, *209*, 111907. <https://doi.org/10.1016/j.jss.2023.111907>.
38. Singh, J.; Cobbe, J.; Norval, C. Decision Provenance: Harnessing Data Flow for Accountable Systems. *IEEE Access* **2019**, *7*, 6562–6574. <https://doi.org/10.1109/access.2018.2887201>.
39. Amershi, S.; Weld, D.; Vorvoreanu, M.; Fournay, A.; Nushi, B.; Collisson, P.; Horvitz, E. Guidelines for Human-AI Interaction. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019; ACM: New York, NY, USA, 2019; pp. 1–13. <https://doi.org/10.1145/3290605.3300233>.
40. van der Stappen, A.; Funk, M. Towards Guidelines for Designing Human-in-the-Loop Machine Training Interfaces. In Proceedings of the 26th International Conference on Intelligent User Interfaces, Paphos, Cyprus, 23–26 March 2026; ACM: New York, NY, USA, 2021; pp. 514–519. <https://doi.org/10.1145/3397481.3450668>.
41. Cheruvu, R. Human-in/on-the-Loop Design for Human Controllability. In *Handbook of Human-Centered Artificial Intelligence*; Xu, W., Ed.; Springer Nature: Singapore, 2025; pp. 1–47. [https://doi.org/10.1007/978-981-97-8440-0\\_75-1](https://doi.org/10.1007/978-981-97-8440-0_75-1).
42. Shin, H.; Park, J.; Yu, J.; Kim, J.; Kim, H.Y.; Oh, C. Looping In: Exploring Feedback Strategies to Motivate Human Engagement in Interactive Machine Learning. *Int. J. Hum.-Comput. Interact.* **2025**, *41*, 8666–8683. <https://doi.org/10.1080/10447318.2024.2413293>.
43. Ahmed, A.M.A. Exploring MLOps Dynamics: An Experimental Analysis in a Real-World Machine Learning Project. *arXiv* **2023**, arXiv:2307.13473. <https://doi.org/10.48550/arxiv.2307.13473>.
44. Boukerche, A.; Coutinho, R.W.L. Design Guidelines for Machine Learning-based Cybersecurity in Internet of Things. *IEEE Netw.* **2021**, *35*, 393–399. <https://doi.org/10.1109/mnet.011.2000396>.
45. Cigoj, M. Achieving EU AI Act Compliance by Integrating Governance as Code within MLOps. 2025. Available online: <https://htec.com/insights/whitepapers/achieving-eu-ai-act-compliance/> (accessed on 18 December 2025).

46. ISO/IEC 42001:2023 ; Artificial Intelligence—Management System. International Organization for Standardization: Geneva, Switzerland, 2023.
47. Singh, P. Systematic Review of Data-Centric Approaches in Artificial Intelligence and Machine Learning. *Data Sci. Manag.* **2023**, *6*, 144–157. <https://doi.org/10.1016/j.dsm.2023.06.001>.
48. Habiba, U.e.; Haug, M.; Bogner, J.; Wagner, S. How mature is requirements engineering for AI-based systems? A systematic mapping study on practices, challenges, and future research directions. *Requir. Eng.* **2024**, *29*, 567–600. <https://doi.org/10.1007/s00766-024-00432-3>.
49. Munappy, A.R.; Bosch, J.; Olsson, H.H. Data Pipeline Management in Practice: Challenges and Opportunities. In *Product-Focused Software Process Improvement*; Morisio, M., Torchiano, M., Jedlitschka, A., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; Volume 12562, pp. 168–184. [https://doi.org/10.1007/978-3-030-64148-1\\_11](https://doi.org/10.1007/978-3-030-64148-1_11).
50. Ashmore, R.; Calinescu, R.; Paterson, C. Assuring the Machine Learning Lifecycle. *ACM Comput. Surv.* **2022**, *54*, 1–39. <https://doi.org/10.1145/3453444>.
51. Nelson, K.; Corbin, G.; Anania, M.; Kovacs, M.; Tobias, J.; Blowers, M. Evaluating model drift in machine learning algorithms. In Proceedings of the 2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), Verona, NY, USA, 26–28 May 2015; IEEE: New York, NY, USA, 2015; pp. 1–8. <https://doi.org/10.1109/cisda.2015.7208643>.
52. Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 2346–2363. <https://doi.org/10.1109/tkde.2018.2876857>.
53. Villamizar, H.; Escovedo, T.; Kalinowski, M. Requirements Engineering for Machine Learning: A Systematic Mapping Study. In Proceedings of the 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Palermo, Italy, 1–3 September 2021; IEEE: New York, NY, USA, 2021; pp. 29–36. <https://doi.org/10.1109/seaa53835.2021.00013>.
54. Demir, E. The protection of human biodata: Is there any role for data ownership? *Comput. Law Secur. Rev.* **2023**, *51*, 105905. <https://doi.org/10.1016/j.clsr.2023.105905>.
55. Doran, G.T. There's a SMART Way to Write Managements's Goals and Objectives. *Manag. Rev.* **1981**, *70*, 35.
56. Kruchten, P. An Ontology of Architectural Design Decisions in Software Intensive Systems. In Proceedings of the 2nd Groningen Workshop on Software Variability, Groningen, The Netherlands, 20–24 August 2012; pp. 54–61.
57. Arif, S.; Amjad, M.U.; Faisal, M. AI-Driven Decision Support Systems for Software Architecture: A Framework for Intelligent Design Decision-Making (2025). *J. Cloud Artif. Intell.* **2025**, *3*, 1–32. <https://doi.org/10.69591/jcai.3.1.1>.
58. Zhou, Y.; Tu, F.; Sha, K.; Ding, J.; Chen, H. A Survey on Data Quality Dimensions and Tools for Machine Learning Invited Paper. In Proceedings of the 2024 IEEE International Conference on Artificial Intelligence Testing (AITest), Shanghai, China, 15–18 July 2024; IEEE: New York, NY, USA, 2024; pp. 120–131. <https://doi.org/10.1109/AITest62860.2024.00023>.
59. Wasser, J.; Kumara, I.; Monsieur, G.; Van Den Heuvel, W.J.; Tamburri, D.A. Data Contracts in Data Mesh: A Systematic Gray Literature Review. In *Business Modeling and Software Design*; Shishkov, B., Ed.; Springer Nature: Geneva, Switzerland, 2025; Volume 559, pp. 21–38. [https://doi.org/10.1007/978-3-031-98033-6\\_2](https://doi.org/10.1007/978-3-031-98033-6_2).
60. Heltweg, P.; Riehle, D. A Systematic Analysis of Problems in Open Collaborative Data Engineering. *J. Data Inf. Qual.* **2023**, *6*, 1–30. <https://doi.org/10.1145/3629040>.
61. Leventon, J.; Fleskens, L.; Claringbould, H.; Schwilch, G.; Hessel, R. An applied methodology for stakeholder identification in transdisciplinary research. *Sustain. Sci.* **2016**, *11*, 763–775. <https://doi.org/10.1007/s11625-016-0385-1>.
62. Miller, G.J. Stakeholder roles in artificial intelligence projects. *Proj. Leadersh. Soc.* **2022**, *3*, 100068. <https://doi.org/10.1016/j.plas.2022.100068>.
63. Brugha, R.; Varvasovszky, Z. Stakeholder analysis: A review. *Health Policy Plan.* **2000**, *15*, 239–246. <https://doi.org/10.1093/heapol/15.3.239>.
64. Suresh, H.; Gomez, S.R.; Nam, K.K.; Satyanarayan, A. Beyond Expertise and Roles: A Framework to Characterize the Stakeholders of Interpretable Machine Learning and their Needs. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, 8–13 May 2021; ACM: New York, NY, USA, 2021; pp. 1–16. <https://doi.org/10.1145/3411764.3445088>.
65. Walker, D.H.T.; Bourne, L.M.; Shelley, A. Influence, stakeholder mapping and visualization. *Constr. Manag. Econ.* **2008**, *26*, 645–658. <https://doi.org/10.1080/01446190701882390>.
66. Ninan, J.; Mahalingam, A.; Clegg, S.; Sankaran, S. ICT for external stakeholder management: Sociomateriality from a power perspective. *Constr. Manag. Econ.* **2020**, *38*, 840–855. <https://doi.org/10.1080/01446193.2020.1755047>.
67. Poszler, F.; Portmann, E.; Lütge, C. Formalizing ethical principles within AI systems: Experts' opinions on why (not) and how to do it. *AI Ethics* **2025**, *5*, 937–965. <https://doi.org/10.1007/s43681-024-00425-6>.
68. Das, S.; Hossain, M.; Shiva, S.G. Requirements Elicitation and Stakeholder Communications for Explainable Machine Learning Systems: State of the Art. In Proceedings of the 2023 International Conference on Information Technology (ICIT), Amman, Jordan, 9–10 August 2023; IEEE: New York, NY, USA, 2023; pp. 561–566. <https://doi.org/10.1109/icit58056.2023.10225934>.
69. Imrie, F.; Davis, R.; van der Schaar, M. Multiple stakeholders drive diverse interpretability requirements for machine learning in healthcare. *Nat. Mach. Intell.* **2023**, *5*, 824–829. <https://doi.org/10.1038/s42256-023-00698-2>.

70. Cantor, A.; Kiparsky, M.; Hubbard, S.S.; Kennedy, R.; Pecharroman, L.C.; Guivetchi, K.; Bales, R. Making a Water Data System Responsive to Information Needs of Decision Makers. *Front. Clim.* **2021**, *3*, 761444. <https://doi.org/10.3389/fclim.2021.761444>.
71. Asswad, J.; Marx Gómez, J. Data Ownership: A Survey. *Information* **2021**, *12*, 465. <https://doi.org/10.3390/info12110465>.
72. Martens, B. The Importance of Data Access Regimes for Artificial Intelligence and Machine Learning. *SSRN Electron. J.* **2018**, 1–23. <https://doi.org/10.2139/ssrn.3357652>.
73. Maharana, K.; Mondal, S.; Nemade, B. A review: Data pre-processing and data augmentation techniques. *Glob. Transitions Proc.* **2022**, *3*, 91–99. <https://doi.org/10.1016/j.gltp.2022.04.020>.
74. Borrohou, S.; Fissoune, R.; Badir, H. The role of data transformation in modern analytics: A comprehensive survey. *J. Comput. Lang.* **2025**, *84*, 101329. <https://doi.org/10.1016/j.cola.2025.101329>.
75. Shamsinejad, E.; Baniroostam, T.; Pedram, M.M.; Rahmani, A.M. A Review of Anonymization Algorithms and Methods in Big Data. *Ann. Data Sci.* **2025**, *12*, 253–279. <https://doi.org/10.1007/s40745-024-00557-w>.
76. Hutchinson, B.; Smart, A.; Hanna, A.; Denton, R.; Greer, C.; Kjartansson, O.; Mitchell, M. Towards Accountability for Machine Learning Datasets: Practices from Software Engineering and Infrastructure. In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Toronto, ON, Canada, 3–10 March 2021; ACM: New York, NY, USA, 2021; pp. 560–575. <https://doi.org/10.1145/3442188.3445918>.
77. Werder, K.; Ramesh, B.; Zhang, R. Establishing Data Provenance for Responsible Artificial Intelligence Systems. *ACM Trans. Web* **2022**, *13*, 1–23. <https://doi.org/10.1145/3503488>.
78. Janssen, M.; Brous, P.; Estevez, E.; Barbosa, L.S.; Janowski, T. Data governance: Organizing data for trustworthy Artificial Intelligence. *Gov. Inf. Q.* **2020**, *37*, 101493. <https://doi.org/10.1016/j.giq.2020.101493>.
79. Gjorgjevikj, A.; Mishev, K.; Antovski, L.; Trajanov, D. Requirements Engineering in Machine Learning Projects. *IEEE Access* **2023**, *11*, 72186–72208. <https://doi.org/10.1109/access.2023.3294840>.
80. Dey, S.; Lee, S.W. A Multi-layered Collaborative Framework for Evidence-driven Data Requirements Engineering for Machine Learning-based Safety-critical Systems. In Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, Tallinn, Estonia, 27–31 March 2023; ACM: New York, NY, USA, 2023; pp. 1404–1413. <https://doi.org/10.1145/3555776.3577647>.
81. Umm-E-Habiba.; Habibullah, K.M. Explainable AI: A Diverse Stakeholder Perspective. In Proceedings of the 2024 IEEE 32nd International Requirements Engineering Conference (RE), Reykjavik, Iceland, 24–28 June 2024; IEEE: New York, NY, USA, 2024; pp. 494–495. <https://doi.org/10.1109/re59067.2024.00060>.
82. Tun, H.T.; Husen, J.H.; Yoshioka, N.; Washizaki, H.; Fukazawa, Y. Goal-Centralized Metamodel Based Requirements Integration for Machine Learning Systems. In Proceedings of the 2021 28th Asia-Pacific Software Engineering Conference Workshops (APSEC Workshops), Taipei, Taiwan, 6–9 December 2021; IEEE: New York, NY, USA, 2021; pp. 13–16. <https://doi.org/10.1109/apsecw53869.2021.00013>.
83. Alsaggaf, F.; Eijkelenboom, A.; Lugten, M.; Turrin, M. A machine learning framework for early-stage dementia-friendly architectural design evaluation using visual access metrics. *Comput. Intell.* **2025**, *4*, 11. <https://doi.org/10.1007/s44223-025-00087-2>.
84. Kücherer, C.; Gerasch, L.; Junger, D.; Burgert, O. Elicitation and Documentation of Explainability Requirements in a Medical Information Systems Context. In Proceedings of the 27th International Conference on Enterprise Information Systems, Porto, Portugal, 4–6 April 2025; SCITEPRESS-Science and Technology Publications: Setubal, Portugal, 2025; pp. 83–94. <https://doi.org/10.5220/0013470600003929>.
85. Alebrahim, A.; Choppy, C.; Faßbender, S.; Heisel, M. Optimizing Functional and Quality Requirements According to Stakeholders' Goals. In *Relating System Quality and Software Architecture*; Elsevier: Amsterdam, The Netherlands, 2014; pp. 75–120. <https://doi.org/10.1016/b978-0-12-417009-4.00004-1>.
86. Schlegel, M.; Sattler, K.U. Management of Machine Learning Lifecycle Artifacts. *ACM SIGMOD Rec.* **2023**, *51*, 18–35. <https://doi.org/10.1145/3582302.3582306>.
87. Polyzotis, N.; Roy, S.; Whang, S.E.; Zinkevich, M. Data Lifecycle Challenges in Production Machine Learning: A Survey. *ACM SIGMOD Rec.* **2018**, *47*, 17–28. <https://doi.org/10.1145/3299887.3299891>.
88. Ryan, S.; Cai, W.; Bowman, R.; Doherty, G. Fairness Challenges in the Design of Machine Learning Applications for Healthcare. *ACM Trans. Comput. Hum. Interact.* **2025**, *6*, 1–26. <https://doi.org/10.1145/3728368>.
89. Puyt, R.W.; Lie, F.B.; Wilderom, C.P. The origins of SWOT analysis. *Long Range Plan.* **2023**, *56*, 102304. <https://doi.org/10.1016/j.lrp.2023.102304>.
90. Sanaei, R.; Otto, K.; Hölltä-Otto, K.; Luo, J. Trade-Off Analysis of System Architecture Modularity Using Design Structure Matrix. In Proceedings of the Volume 2B: 41st Design Automation Conference, Boston, MA, USA, 2 August 2015; American Society of Mechanical Engineers: New York, NY, USA, 2016; p. V02bt03a037. <https://doi.org/10.1115/detc2015-46403>.
91. Yoshinaga, K.; Kato, T.; Kai, Y. Clustering Method Of Design Structure Matrix For Trade-off Relationships. *J. Sci. Des.* **2017**, *1*, 1\_77–1\_86. [https://doi.org/10.11247/jsd.1.1\\_1\\_77](https://doi.org/10.11247/jsd.1.1_1_77).
92. Luo, Y.; Tseng, H.H.; Cui, S.; Wei, L.; ten Haken, R.K.; El Naqa, I. Balancing accuracy and interpretability of machine learning approaches for radiation treatment outcomes modeling. *BJR Open* **2019**, *1*, 20190021. <https://doi.org/10.1259/bjro.20190021>.

93. Li, J.; Li, G. Triangular Trade-off between Robustness, Accuracy, and Fairness in Deep Neural Networks: A Survey. *ACM Comput. Surv.* **2025**, *57*, 1–40. <https://doi.org/10.1145/3645088>.
94. Pham, N.D.; Phan, K.T.; Chilamkurti, N. Enhancing Accuracy-Privacy Trade-Off in Differentially Private Split Learning. *IEEE Trans. Emerg. Top. Comput. Intell.* **2025**, *9*, 988–1000. <https://doi.org/10.1109/tetci.2024.3485723>.
95. Brownlee, A.E.; Adair, J.; Haraldsson, S.O.; Jabbo, J. Exploring the Accuracy–Energy Trade-off in Machine Learning. In Proceedings of the 2021 IEEE/ACM International Workshop on Genetic Improvement (GI), Madrid, Spain, 30 May 2021; IEEE: New York, NY, USA, 2021; pp. 11–18. <https://doi.org/10.1109/gi52543.2021.00011>.
96. Bhat, M.; Shumaiev, K.; Biesdorf, A.; Hohenstein, U.; Matthes, F. Automatic Extraction of Design Decisions from Issue Management Systems: A Machine Learning Based Approach. In *Software Architecture*; Lopes, A., de Lemos, R., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; Volume 10475, pp. 138–154. [https://doi.org/10.1007/978-3-319-65831-5\\_10](https://doi.org/10.1007/978-3-319-65831-5_10).
97. Leest, J.; Gerostathopoulos, I.; Raibulet, C. Evolvability of Machine Learning-based Systems: An Architectural Design Decision Framework. In Proceedings of the 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C), L'Aquila, Italy, 13–17 March 2023; IEEE: Berlin/Heidelberg, Germany, 2023; pp. 106–110. <https://doi.org/10.1109/icsa-c57050.2023.00033>.
98. Charoenwut, P.; Drobnjakovic, M.; Oh, H.; Nikolov, A.; Kulvatunyong, B. Machine Learning Lifecycle Explorer: Leverage the Lifecycle Metadata (onto) Logically. In *Advances in Production Management Systems. Cyber-Physical-Human Production Systems: Human-AI Collaboration and Beyond*; Mizuyama, H., Morinaga, E., Nonaka, T., Kaihara, T., von Cieminski, G., Romero, D., Eds.; Springer Nature: Geneva, Switzerland, 2025; Volume 769, pp. 370–385. [https://doi.org/10.1007/978-3-032-03550-9\\_25](https://doi.org/10.1007/978-3-032-03550-9_25).
99. Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Geburu, T. Model Cards for Model Reporting. In Proceedings of the Conference on Fairness, Accountability, and Transparency, Atlanta, GA, USA, 29–31 January 2019; ACM: New York, NY, USA, 2019; pp. 220–229. <https://doi.org/10.1145/3287560.3287596>.
100. Nicoletti, M.; Diaz-Pace, J.A.; Schiaffino, S.; Tommasel, A.; Godoy, D. Personalized architectural documentation based on stakeholders' information needs. *J. Enterp. Inf. Manag.* **2014**, *2*, 9. <https://doi.org/10.1186/s40411-014-0009-3>.
101. Parlak, İ.E. Blockchain-assisted explainable decision traces (BAXDT): An approach for transparency and accountability in artificial intelligence systems. *Knowl.-Based Syst.* **2025**, *329*, 114402. <https://doi.org/10.1016/j.knosys.2025.114402>.
102. Karkoskova, S.; Novotny, O. Design and Application on Business Data Lineage as a part of Metadata Management. In Proceedings of the 2021 International Conference on Computers and Automation (CompAuto), Paris, France, 7–9 September 2021; IEEE: New York, NY, USA, 2021; pp. 34–39. <https://doi.org/10.1109/CompAuto54408.2021.00014>.
103. Silva, C.; Abbasi, M.; Martins, P.; Silva, J.; Váz, P.; Mota, D. Enhancing Organizational Data Integrity and Efficiency through Effective Data Lineage. In Proceedings of the 2023 Second International Conference on Smart Technologies for Smart Nation (SmartTechCon), Singapore, 18–19 August 2023; IEEE: New York, NY, USA, 2023; pp. 312–317. <https://doi.org/10.1109/SmartTechCon57526.2023.10391436>.
104. Wohlrab, R.; Knauss, E.; Steghöfer, J.P.; Maro, S.; Anjorin, A.; Pelliccione, P. Collaborative traceability management: A multiple case study from the perspectives of organization, process, and culture. *Requir. Eng.* **2020**, *25*, 21–45. <https://doi.org/10.1007/s00766-018-0306-1>.
105. Stone, J.; Patel, R.; Ghiasi, F.; Mittal, S.; Rahimi, S. Navigating MLOps: Insights into Maturity, Lifecycle, Tools, and Careers. In Proceedings of the 2025 IEEE Conference on Artificial Intelligence (CAI), Santa Clara, CA, USA, 5–7 May 2025; IEEE: New York, NY, USA, 2025; pp. 643–650. <https://doi.org/10.1109/cai64502.2025.00118>.
106. Mavromatis, I.; Katsaros, K.; Khan, A. Computing Within Limits: An Empirical Study of Energy Consumption in ML Training and Inference. In Proceedings of the 1st International Workshop on Artificial Intelligence for Sustainable Development, Pescara, Italy, 3 July 2024. <https://doi.org/10.48550/arxiv.2406.14328>.
107. Tyagi, A.J. Scaling deep learning models: Challenges and solutions for large-scale deployments. *World J. Adv. Eng. Technol. Sci.* **2025**, *16*, 010–020. <https://doi.org/10.30574/wjaets.2025.16.2.1252>.
108. Vijayan, N.E. Building Scalable MLOps: Optimizing Machine Learning Deployment and Operations. *Int. J. Sci. Res. Eng. Manag.* **2024**, *8*, 1–9. <https://doi.org/10.55041/ijrsrem37784>.
109. Du, K.L.; Zhang, R.; Jiang, B.; Zeng, J.; Lu, J. Understanding Machine Learning Principles: Learning, Inference, Generalization, and Computational Learning Theory. *Mathematics* **2025**, *13*, 451. <https://doi.org/10.3390/math13030451>.
110. Voinov, A.; Jenni, K.; Gray, S.; Kolagani, N.; Glynn, P.D.; Bommel, P.; Smajgl, A. Tools and methods in participatory modeling: Selecting the right tool for the job. *Environ. Model. Softw.* **2018**, *109*, 232–255. <https://doi.org/10.1016/j.envsoft.2018.08.028>.
111. Sakovich, N.; Aksenov, D.; Pleshakova, E.; Gataullin, S. A Neural Operator Using Dynamic Mode Decomposition Analysis to Approximate Partial Differential Equations. *SSRN Electron. J.* **2025**, *10*, 22432–22444. <https://doi.org/10.2139/ssrn.5542238>.
112. Slupczynski, M.; Klamma, R. *Vorschlag einer Infrastruktur zur verteilten Datenanalyse von multimodalen psychomotorischen Lerneraktivitäten*; Technical Report; RWTH Aachen University: Aachen, Germany, 2021. <https://doi.org/10.13140/RG.2.2.23664.58886/1>.

113. Slupczynski, M.; Klamma, R. MILKI-PSY Cloud: Facilitating Multimodal Learning Analytics by Explainable AI and Blockchain. In Proceedings of the Multimodal Immersive Learning Systems 2021 (MILeS 2021), Bozen-Bolzano, Italy, 20–24 September 2021; Klemke, R., Di Mitri, D., Ciordas-Hertel, G.P., Eds.; CEUR Workshop Proceedings; Volume 2979, pp. 22–28.
114. Slupczynski, M.; Klamma, R. MILKI-PSY Cloud: MLOps-based Multimodal Sensor Stream Processing Pipeline for Learning Analytics in Psychomotor Education. In Proceedings of the Multimodal Immersive Learning Systems 2022 (MILeS 2022), Toulouse, France, 12–16 September 2022; Sanusi, K.A.M., Limbu, B., Schneider, J., Eds.; CEUR Workshop Proceedings; pp. 8–14.
115. Mat Sanusi, K.A.; Slupczynski, M.; Geisen, M.; Iren, D.; Klamma, R.; Klatt, S.; Klemke, R. IMPECT-Sports: Using an Immersive Learning System to Facilitate the Psychomotor Skills Acquisition Process. In Proceedings of the Multimodal Immersive Learning Systems 2022, Toulouse, France, 12–16 September 2022; Sanusi, K.A.M., Limbu, B., Schneider, J., Di Mitri, D., Klemke, R., Eds.; CEUR Workshop Proceedings; pp. 34–39.
116. Gao, C. Sensor Based Human Motion Comparison. Master Thesis, RWTH Aachen University, Aachen, Germany, 2023.
117. Slupczynski, M.P.; Decker, S.J. *Semantic Motion Learning Analytics and Feedback System*; Universitätsbibliothek der RWTH Aachen: Aachen, Germany, 2024. <https://doi.org/10.18154/RWTH-2024-09936>.
118. Slupczynski, M.; Mat Sanusi, K.A.; Majonica, D.; Klemke, R.; Decker, S. Implementing Cloud-Based Feedback to Facilitate Scalable Psychomotor Skills Acquisition. In Proceedings of the Multimodal Immersive Learning Systems 2023, Aveiro, Portugal, 4–8 September 2023; Sanusi, K.A.M., Limbu, B., Schneider, J., Kravcik, M., Klemke, R., Eds.; CEUR Workshop Proceedings; Volume 3499, pp. 36–43.
119. Li, S. From Still to Dynamic: A Comparative Analysis of Visualization Techniques for Human Movements. Bachelor's Thesis, RWTH Aachen University, Aachen, Germany, 2024.
120. Geisen, M.; Seifriz, F.; Fasold, F.; Slupczynski, M.; Klatt, S. A Novel Approach to Sensor-Based Motion Analysis for Sports: Piloting the Kabsch Algorithm in Volleyball and Handball. *IEEE Sens. J.* **2024**, *24*, 35654–35663. <https://doi.org/10.1109/JSEN.2024.3455173>.
121. Slupczynski, M.; Nekhviadovich, A.; Duong-Trung, N.; Decker, S. Analyzing Exercise Repetitions: YOLOv8-Enhanced Dynamic Time Warping Approach on InfiniteRep Dataset. In *Sensor-Based Activity Recognition and Artificial Intelligence*; Konak, O., Arnrich, B., Bieber, G., Kuijper, A., Fudickar, S., Eds.; Springer Nature: Potsdam, Germany, 2024; Volume 15357, pp. 94–110. [https://doi.org/10.1007/978-3-031-80856-2\\_7](https://doi.org/10.1007/978-3-031-80856-2_7).
122. Slupczynski, M.; Mat Sanusi, K.A.; Klemke, R.; Decker, S. SpaceStriker: A Peer-Assisted Sensor-Based Exergame with Real-Time ML-Driven Feedback. In *Sensor-Based Activity Recognition and Artificial Intelligence*; Konak, O., Fudickar, S., Arnrich, B., Bieber, G., Kuijper, A., Eds.; Springer Nature: Geneva, Switzerland, 2025; Vol. 16292. [https://doi.org/10.1007/978-3-032-13312-0\\_27](https://doi.org/10.1007/978-3-032-13312-0_27).
123. Singh, D.; Slupczynski, M.; Pillai, A.G.; Pandian, V.P.S. Grounding Explainability Within the Context of Global South in XAI. *arXiv* **2022**, arXiv:2205.06919. <https://doi.org/10.48550/ARXIV.2205.06919>.
124. Cooper, R.G. The 5-Th Generation Stage-Gate Idea-to-Launch Process. *IEEE Eng. Manag. Rev.* **2022**, *50*, 43–55. <https://doi.org/10.1109/emr.2022.3222937>.
125. Sadek, M.; Constantinides, M.; Quercia, D.; Mougnot, C. Guidelines for Integrating Value Sensitive Design in Responsible AI Toolkits. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 11–16 May 2024; pp. 1–20. <https://doi.org/10.1145/3613904.3642810>.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.