SE4AI: Structured AI/ML Enabled Software Development Process

Felipe Sonntag Manzoni
fsm2@icomp.ufam.edu.br
Federal University of Amazonas - UFAM
Institute of Computing - IComp
Manaus, Amazonas, Brazil
SiDi Intelligence & Innovation Center
AI R&D department
Manaus, Amazonas, Brazil

Abstract

Most organizations developing software that depends on AI/ML models often rely on *ad hoc* or unstructured development practices. In many cases, the model and the surrounding software are developed separately, leading to several problems—particularly regarding the quality of the model, the compatibility of the system in which the model is embedded, and the overall suitability of the resulting system to the problem domain. Although prior research has addressed specific aspects of this process and proposed improvements for isolated phases, a comprehensive, quality-oriented development framework is still lacking. This doctoral research aims to propose an enhanced development process that addresses these gaps and opportunities, ensuring both methodological rigor and the delivery of reliable, high-quality AI/ML-enabled systems.

CCS Concepts

• **Software and its engineering → Agile software development**; *Software development techniques.*

Keywords

Software Development, Agile Process, Artificial Inteligence, Machine Learning, Process Quality

ACM Reference Format:
Felipe Sonntag Manzoni. 2026. SE4AI: Structured AI/ML Enabled Software Development Process. In *2026 IEEE/ACM 48th International Conference on Software Engineering (ICSE-Companion '26), April 12–18, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3774748.3787624

1 Motivation

AI/ML-enabled systems are increasingly embedded in software products across industry and day-to-day sectors, becoming central to decision-making, automation, and user-facing functionalities as well as development and technical processes. Despite this growth, the engineering practices used to build these systems remain highly ad hoc and often poorly structured. Nascimento et al. (2020) [8] show that AI/ML development frequently occurs outside formalized processes and in disconnected cycles, where data collection, experimentation, model training, and deployment are performed with ad hoc practices and minimal alignment with software engineering principles. Similar findings are reported by Lorenzoni et al. (2021) [4], who highlight that model development life cycles diverge significantly from traditional software processes, especially in activities related to data handling, experimental management, and continuous evaluation.

According to these reports on literature these gaps produce well-known problems in industry: difficulties in scoping AI/ML projects, inconsistent interpretation of requirements, rework due to misaligned expectations, poor data readiness, and integration failures between trained models and production systems. These industry problems are often closely related to the overall software quality overview and metrics that are often overlooked due to the ad hoc and research first process adopted by these types of projects [1]. Through industrial empirical observations this research could verify the evaluated scenario: companies tend to underestimate the effort required for data-centric tasks and validation activities, often treating model development as an isolated component. As a result, teams face project overruns, quality issues, and models that are misaligned with business rules or system constraints [9].

Given the increasing dependency of software systems on AI/ML components as well as the intrinsic necessity of model development and integration for new systems, and the practical consequences of adopting ad hoc development practices, the software engineering community urgently needs a systematic, holistic, and empirically founded development process for AI/ML-enabled software.

2 Problem Statement and Hypothesis

2.1 Problem Statement

The current state of practice in AI/ML software development suffers from a lack of integrated, well-defined software engineering processes that explicitly account for AI/ML-specific characteristics such as data life cycle management, experimental iteration loops, model validation, and technical constraints. According to Serban et. al (2020) [9] even though some of these gaps are solved through previously defined agile processes, the industry currently have low adoption of them, either because of compatibility with project steps or because single solutions cannot integrate the development process as needed for full project development.

Existing approaches tend to focus on individual challenges (e.g., data quality, model testing, or deployment), but fail to provide end-to-end guidance spanning the entire development life cycle or to update rather ad hoc practices to more agile processes needs and activities. Nascimento et al. (2020) [8] show that practitioners rely heavily on undocumented "team knowledge", while Lorenzoni et al. (2021) [4] demonstrate that even researchers lack shared understanding of coherent development stages for AI/ML model creation.

In a more recent research Castelli et al. (2025) [1] have developed a maturity framework focused on the Machine Learning (ML) quality and governance evaluation process that may support industry practitioners to identify underlying issues in the developed systems. However, their focus is oriented on the quality assessment and evaluation of post developed or under development ML systems, which may arise the underlying issues, but not clearly show the solution path for current or future projects [5].

This fragmentation on process step solutions leads to misalignment between model development and software engineering practices, resulting in brittle integrations, insufficient quality assurance, and limited traceability of decisions [7]. More importantly, there is no actionable framework that project managers and developers can use to scope, size, and execute AI/ML projects with predictable quality and delivery confidence.

2.2 Research Hypothesis

The main research hypothesis for this work is as follows:

"*A structured and empirically grounded software development process life cycle that can measurably improve the quality, predictability, and alignment of AI/ML-enabled software systems.*"

The proposed process of this research should be explicitly designed to integrate AI/ML-specific activities into traditional SE life cycles. The main hypothesis from this research is supported by early industrial evidence from a in-company case study, where the inclusion of quality-oriented activities across data, modeling, and deployment stages improved alignment with business rules and reduced model rework. This initial industrial evidence is being published as industrial evidence with the needs from a specific case and results from the usage of a quality oriented development process and further needs that are needed for this field.

3 Research Objectives

The overarching objective of this doctoral research is to design, evaluate, and validate a comprehensive software development process tailored for AI/ML-enabled systems, grounded in empirical evidence drawn from literature and industry.

To achieve this, the research is structured into the following objectives:

(1) **Synthesize** the scattered knowledge on AI/ML development life cycles through a rigorous Systematic Literature Review and identify crucial gaps and needs on the development process for AI/ML enabled systems;
(2) **Verify**, through a wide and effective survey, that synthesized knowledge on industry needs are currently faced by global practitioners and researchers;
(3) **Identify essential process activities** for AI/ML software development—especially those overlooked by traditional SE life cycles from both past results. This includes data-related tasks, model iteration loops, validation strategies, and integration constraints.
(4) **Design a development process** that harmonizes AI/ML activities with software engineering stages, explicitly ensuring quality assurance integration throughout the life cycle.
(5) **Construct a practical framework** enabling industry practitioners to use and validate the process for project scoping, estimation, role definition, and quality planning.
(6) **Empirically evaluate the proposed process** through feasibility studies, observation studies, life cycle case studies, and industrial case studies, following an evidence-based SE research methodology ensuring both internal and external validity.

4 Contributions and Impact

4.1 Scientific Contributions

The overall expected contributions of this research regarding the scientific view of the Software Engineering for AI/ML field are as follows:

- Providing an empirically grounded, end-to-end development process specifically designed for AI/ML-enabled software systems, bridging the gap between scattered research insights and coherent life cycle guidance;
- Clarifying the essential activities and dependency relationships in AI/ML development, offering a structured vocabulary and process definition for researchers and practitioners;
- Integrating quality assurance principles into all phases of AI/ML development, addressing concerns raised in past researches [4, 7–9];
- Delivering a reusable framework that can support future empirical studies on AI/ML project management, process refinement, and SE/ML integration practices.

4.2 Industrial Impact

This research have an expected impact on the industry directly to practitioners and AI/ML teams as follows:

- Improved project scoping and estimation, enabling realistic expectations about effort, complexity, and resource needs;
- Higher alignment between models and business rules, reducing rework and integration issues;
- Earlier and more systematic quality assurance, decreasing defects related to data, model behavior, and software integration;
- A shared process framework that aligns cross-functional teams (software engineers, ML engineers, QA analysts, product owners and others);
- Transferability across projects, providing organizations with a repeatable, scalable way to develop AI/ML-enabled systems.

Industrial feedback from early observations already indicates that structured quality activities during dataset creation, model evaluation, and integration help prevent common delivery issues and tangibly increase confidence in the final system or at least on model development and evaluation phases.

5 Research Methodology

This research adopts a multi-method, evidence-based approach grounded in Software Engineering research practices. The goal is to systematically build and refine a comprehensive development process for AI/ML-enabled software systems, integrating insights

from the literature, industrial practice, and empirical observations. The methodology draws on three pillars: (1) a Systematic Literature Review, (2) Grounded Theory analysis, and (3) iterative empirical evaluation in industrial environments.

5.1 Systematic Literature Review (SLR)

Firstly a rigorous SLR, following Kitchenham's (2009) [3] guidelines, and using the mapping produced by Nascimento et al. (2020) [8] and Lorenzoni et al. (2021) [4] as foundational anchors, is currently under development. These two surveys reveal significant fragmentation in existing development approaches and highlight the dominance of ad hoc practices, insufficient documentation of processes, and the lack of comprehensive life cycle definitions. The SLR will extend and deepen these findings by extracting process elements, dependencies, and quality concerns across all stages of AI/ML development—from data acquisition to deployment and maintenance as well as update rather "old" findings to current state of the field.

5.2 Grounded Theory for Process Construction

To build an empirically grounded development process, Grounded Theory (GT) techniques will be applied to analyze qualitative data from industry observations, interviews, documents, and case studies [11]. GT is well suited for emerging domains where formal theory is lacking and where practice varies widely — a situation that directly aligns with AI/ML software engineering [2].

Using open, axial, and selective coding, the GT analysis will enable the identification of recurrent patterns, development activities, common pain points, and stakeholder roles across real-world AI/ML projects [11]. These analytic insights form the basis for the initial version of the process, ensuring that it reflects practical needs rather than solely theoretical expectations. This is critical because the challenges observed in practice — such as unclear requirements, insufficient data readiness, experimental drift, and late-stage integration failures — are often missing or underrepresented in academic models [2].

5.3 Iterative Process Refinement

The initial process version derived from the SLR + GT analysis would be refined through iterative cycles involving:

- **Feasibility studies** to test applicability in small, controlled scenarios;
- **Observation studies** during real AI/ML development tasks to examine alignment with team workflows and identify shortcomings;
- **Life cycle case studies** applying the process end-to-end on actual in-house projects;
- **Industry case studies** to evaluate adoption challenges, integration issues, and scalability in live industrial environments.

This structured methodology follows the evidence-based technology maturation model proposed by Shull et al. (2001) [10] and extended by Mafra et al. (2006) [6], and it ensures that the process evolves pragmatically and empirically across increasing levels of realism and complexity.

Acknowledgments

This work was carried out with the support of the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES-PROEX) – Funding Code 001. This work was partially supported by Amazonas State Research Support Foundation - FAPEAM - through the POSGRAD project 2025/2026. Additionally, we would like to thank the funding granted by CNPq/MCTI/FNDCT Nº 44/2024 - UNIVERSAL through the project 406506/2025-6 that supported the development of this project.

References

[1] Angelantonio Castelli, Georgios Christos Chouliaras, and Dmitri Goldenberg. 2025. Maturity Framework for Enhancing Machine Learning Quality. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2* (Toronto ON, Canada) *(KDD '25)*. Association for Computing Machinery, New York, NY, USA, 4296–4307. doi:10.1145/3711896.3737246

[2] Rashina Hoda. 2022. Socio-Technical Grounded Theory for Software Engineering. *IEEE Transactions on Software Engineering* 48, 10 (2022), 3808–3832. doi:10.1109/TSE.2021.3106280

[3] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology* 51, 1 (2009), 7–15. doi:10.1016/j.infsof.2008.09.009

[4] Giuliano Lorenzoni, Paulo Alencar, Nathalia Nascimento, and Donald Cowan. 2021. Machine Learning Model Development from a Software Engineering Perspective: A Systematic Literature Review. arXiv:2102.07574 [cs.SE] https://arxiv.org/abs/2102.07574

[5] Lucy Ellen Lwakatare, Aiswarya Raj, Ivica Crnkovic, Jan Bosch, and Helena Holmström Olsson. 2020. Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions. *Information and Software Technology* 127 (2020), 106368. doi:10.1016/j.infsof.2020.106368

[6] Sômulo Mafra, Rafael Barcelos, and Guilherme Travassos. 2006. Aplicando uma Metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software. In *Anais do XX Simpósio Brasileiro de Engenharia de Software* (Florianópolis). SBC, Porto Alegre, RS, Brasil, 239–254. doi:10.5753/sbes.2006.21216

[7] Silverio Martínez-Fernández, Justus Bogner, Xavier Franch, Marc Oriol, Julien Siebert, Adam Trendowicz, Anna Maria Vollmer, and Stefan Wagner. 2022. Software Engineering for AI-Based Systems: A Survey. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31, 2, Article 37e (April 2022), 59 pages. doi:10.1145/3487043

[8] Elizamary Nascimento, Anh Nguyen-Duc, Ingrid Sundbø, and Tayana Conte. 2020. Software engineering for artificial intelligence and machine learning software: A systematic literature review. arXiv:2011.03751 [cs.SE] https://arxiv.org/abs/2011.03751

[9] Alex Serban, Koen van der Blom, Holger Hoos, and Joost Visser. 2020. Adoption and Effects of Software Engineering Best Practices in Machine Learning. In *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (Bari, Italy) *(ESEM '20)*. Association for Computing Machinery, New York, NY, USA, Article 3, 12 pages. doi:10.1145/3382494.3410681

[10] Forrest Shull, Jeffrey Carver, and Guilherme H. Travassos. 2001. An empirical methodology for introducing software processes. In *Proceedings of the 8th European Software Engineering Conference Held Jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (Vienna, Austria) *(ESEC/FSE-9)*. Association for Computing Machinery, New York, NY, USA, 288–296. doi:10.1145/503209.503248

[11] Klaas-Jan Stol, Paul Ralph, and Brian Fitzgerald. 2016. Grounded theory in software engineering research: a critical review and guidelines. In *Proceedings of the 38th International Conference on Software Engineering* (Austin, Texas) *(ICSE '16)*. Association for Computing Machinery, New York, NY, USA, 120–131. doi:10.1145/2884781.2884833