Communications of the Association for Information Systems

Volume 57 Paper 58

10-31-2025

Implementing AI into ERP Software

Siar Sarferaz
SAP SE, Research & Development, siar.sarferaz@sap.com

Recommended Citation
Sarferaz, S. (2025). Implementing AI into ERP Software. Communications of the Association for Information Systems, 57, 1396-1426. https://doi.org/10.17705/1CAIS.05758

Implementing AI into ERP Software

Cover Page Footnote

This manuscript underwent editorial review. It was received 06/16/2025 and was with the authors for two months for two revisions. Daniel O'Leary served as Associate Editor.

Communications of the Association for Information Systems

Practitioner Article DOI: 10.17705/1CAIS.05758 ISSN: 1529-3181

Implementing AI into ERP Software

Siar Sarferaz

SAP SE, Research & Development
Germany
siar.sarferaz@sap.com
0009-0006-3142-8589

Abstract:

Enterprise Resource Planning (ERP) systems digitalize business processes across organizations to enhance automation and optimization. They integrate data and workflows from sales, marketing, finance, supply chain, manufacturing, services, procurement, and human resources, thereby serving as a central system of record for many organizations. ERP systems typically support tens of thousands of business processes and maintain data across thousands of tables, which creates significant potential for embedding Artificial Intelligence (AI) to enhance automation and optimization. However, incorporating AI into ERP systems is challenging due to their complexity, which involves hundreds of millions of lines of code and the need to support diverse industry- and region-specific requirements. Accordingly, this paper addresses the following research question: *How can organizations systematically develop and operate (DevOps) AI business applications in ERP systems?* To answer this question, we conduct a gap analysis, derive business requirements, design and implement a DevOps framework, and evaluate it using real-world ERP use cases.

Keywords: Enterprise Resource Planning, ERP, Artificial Intelligence, AI, Enterprise AI, Business AI, Business Applications, Software Integration, AI Development, AI Operations.

This manuscript underwent editorial review. It was received 06/16/2025 and was with the authors for two months for two revisions. Daniel O'Leary served as Associate Editor.

1 Introduction

Enterprise Resource Planning (ERP) systems are comprehensive software suites that integrate and manage core business processes across an organization’s entire operational spectrum (Rashid et al., 2002). These systems serve as the central nervous system for critical businesses, providing a unified platform that consolidates data and workflows from disparate functional areas, including finance, human resources, supply chain management, manufacturing, customer relationship management, and procurement. ERP systems have evolved from basic material requirements planning (MRP) tools into sophisticated enterprise-wide solutions that support real-time data sharing, process standardization, and cross-functional visibility (Rashid et al., 2002; Mhaskey, 2024). Modern ERP systems are built on modular architectures, enabling organizations to implement specific functional components as needed while maintaining enterprise-wide data consistency (Sadrzadehrafiei et al., 2013). These systems typically manage vast repositories of transactional data, often encompassing thousands of database tables and processing tens of thousands of distinct business processes (Fox, 1986; Hau & Aparício, 2008). The inherent complexity of ERP systems, comprising hundreds of millions of lines of code and supporting diverse industry-specific and regional requirements, presents both opportunities and challenges for technological enhancement.

Artificial Intelligence (AI), in the context of enterprise applications, refers to computer systems equipped with the capability to perform tasks that traditionally require human cognitive functions, including pattern recognition, decision-making, natural language processing, and predictive analysis (Kaniyar et al., 2019). The AI paradigm encompasses various sophisticated technologies, including machine learning algorithms, deep learning networks, natural language processing capabilities, computer vision systems, and predictive analytics frameworks (Batra & Santhanam, 2017). According to Pokala (Pokala, 2024), AI technologies in enterprise environments are characterized by their ability to process vast datasets, identify complex patterns, and generate actionable insights that enhance organizational decision-making processes. Machine learning, as a subset of AI (also referred to as Narrow AI), enables systems to automatically learn and improve from data without explicit programming, making it particularly valuable for ERP environments where historical transactional data can inform future business processes (Jawad & Balázs, 2024). The integration of AI capabilities into ERP systems represents a paradigmatic shift from traditional rule-based automation toward intelligent, adaptive business process management (El Hairech & Lyhyaoui, 2022). This convergence is theoretically grounded in several key principles that enhance the traditional ERP value proposition. Daily operations in ERP systems generate vast amounts of structured and semi-structured data. AI technologies, particularly machine learning algorithms, can extract meaningful patterns from this data to support predictive analytics and intelligent automation (Ivanović & Marić, 2021). Jawad and Balázs (2024) argue that machine learning–driven optimization enables organizations to shift from reactive to proactive management, thereby improving both operational efficiency and strategic planning. Traditional ERP systems excel at process standardization and data consistency but require explicit programming for decision logic. AI integration introduces adaptive intelligence that can learn from historical patterns, optimize workflows dynamically, and automate complex decision-making processes (Bergdahl, 2022). Research by Mhaskey (2024) demonstrates that organizations implementing AI-enhanced ERP solutions achieve an average 30% increase in user satisfaction and 25% improvement in productivity through intelligent process automation. Whereas conventional ERP systems mainly provide descriptive analytics through reports and dashboards, AI integration enables predictive and prescriptive analytics (Mediavilla et al., 2021). These advanced analytics capabilities allow organizations to anticipate future trends, identify potential issues before they occur, and receive automated recommendations for optimal actions (Sustrova, 2016).

Recent systematic reviews of AI in ERP systems reveal significant research activity and practical adoption across various industrial sectors. Aktürk (2021) conducted a comprehensive bibliometric analysis of 837 publications related to AI in ERP systems. The empirical evidence supporting AI-ERP integration is substantial. Research by Godbole (2023) involving 300 enterprises demonstrates that AI-infused ERP systems correlate with significant operational efficiency gains, including an average 27% reduction in task processing times and 35% enhancement in accuracy across business functions. Furthermore, analysis of 50 companies implementing AI-driven predictive analytics within ERP platforms shows an 18% decrease in maintenance costs and 22% increase in overall equipment effectiveness (Godbole, 2023). Manufacturing industries have been early adopters of AI-ERP integration, particularly in areas of predictive maintenance, quality control, and supply chain optimization (Kreuzer et al., 2024). Financial services sectors leverage AI capabilities for fraud detection, risk assessment, and automated regulatory compliance (Bandla, 2023). Retail organizations utilize AI-enhanced ERP systems for demand forecasting, inventory optimization, and personalized customer experiences (Lin, 2021). Current AI-ERP implementations demonstrate varying levels of sophistication. Basic implementations focus on process automation and data analytics, while advanced implementations incorporate machine learning models for predictive analytics, natural language processing for user interaction, and computer vision for quality control (Srinivasan et al., 2004). Research indicates that over 50% of organizations plan to incorporate AI capabilities into their ERP systems within the next two years, signifying a notable shift toward intelligent enterprise systems (Mhaskey, 2024).

Despite the promising potential, AI integration into ERP systems faces several significant challenges that require systematic attention. These challenges span technical, organizational, and operational dimensions. ERP systems' inherent complexity, involving millions of lines of code and intricate interdependencies, makes AI integration technically challenging (Ribeiro et al., 2021). The need to maintain system stability while introducing intelligent capabilities requires sophisticated development and operational approaches that can manage both traditional ERP functionality and AI model lifecycles (Themistocleous et al., 2001). AI algorithms are fundamentally dependent on high-quality, consistent data for effective performance. ERP systems, while containing vast amounts of data, often suffer from data quality issues, including inconsistencies, missing values, and varying data formats across different modules (Karamitsos et al., 2020). Establishing robust data governance frameworks becomes critical for successful AI implementation (Kumar, 2018). Traditional ERP development follows established software engineering practices, while AI implementations require specialized approaches encompassing machine learning operations (MLOps) (Arachchi et al., 2015). Research by Karamitsos et al. (Karamitsos et al., 2020) highlights the need for integrated DevOps practices that can accommodate both traditional software development and AI model development, training, and deployment cycles. AI integration necessitates significant changes in business processes, user workflows, and organizational capabilities (Tallón-Ballesteros, 2020). Resistance to change, skills gaps, and cultural barriers represent substantial challenges that must be addressed through comprehensive change management strategies (Patel, 2021).

While existing literature provides valuable insights into the potential and challenges of AI-ERP integration, there remains a significant gap in systematic approaches for developing and operating AI applications within ERP environments. Current research primarily focuses on specific AI applications or theoretical frameworks but lacks comprehensive methodologies for the entire AI development (Dev) and operations (Ops) lifecycle in ERP contexts (Costin et al., 2020; Rojek & Jagodziński, 2022; Zheng et al., 2022). The objective of a DevOps framework in the context of this paper is to standardize the development and operations (DevOps) of AI applications for ERP systems. The complexity of modern ERP systems, combined with the specialized requirements of AI development (Dev) and operations (Ops), necessitates a systematic DevOps framework that can bridge traditional ERP software engineering practices with AI-specific development and operational requirements.

This research gap is particularly evident in the lack of empirical studies evaluating comprehensive DevOps frameworks specifically designed for AI-ERP integration. Furthermore, existing AI-ERP implementations often follow ad-hoc approaches, leading to inconsistent outcomes, technical debt, and scalability challenges. The absence of standardized frameworks for AI-ERP DevOps results in increased efforts, higher implementation risks, and suboptimal utilization of AI capabilities within enterprise environments. Our research addresses the identified gap by developing and evaluating a comprehensive DevOps framework specifically designed for AI implementation in ERP systems. The primary research question guiding this investigation is: How to systematically develop and operate (DevOps) AI business applications in ERP software? The specific objectives of this research include:

Gap Analysis: Conducting a comprehensive analysis of current DevOps practices for AI-ERP integration, identifying specific gaps and challenges that hinder effective implementation and operations.

Requirements Derivation: Establishing detailed business and technical requirements for a DevOps framework that can support the entire AI lifecycle within ERP environments.

Framework Design: Developing a systematic DevOps framework that integrates AI development methodologies with traditional ERP software engineering practices.

Implementation and Evaluation: Realizing the proposed framework through real-world ERP use cases and evaluating its effectiveness in terms of development efficiency, deployment reliability, and operational performance.

The contribution of this research extends beyond theoretical framework development to provide practical guidance for organizations seeking to implement AI capabilities within their ERP systems. By addressing the systematic development and operation of AI applications in ERP environments, this research provides valuable insights for both academic researchers and industry practitioners navigating the complex landscape of AI-ERP integration. Low-level technologies for DevOps, like continuous integration and continuous delivery (CI/CD) are taken for granted and are out of scope of this research, which focuses on the higher-level methodologies and frameworks specific to AI-ERP integration challenges.

While ERP vendors increasingly deliver pre-built AI scenarios (e.g., predictive invoice matching, demand forecasting, or anomaly detection), these scenarios typically address standardized and widely recurring business processes. In practice, ERP customers frequently need to configure and extend the ERP vendor's scenarios or develop their own AI use cases to meet highly specific organizational requirements that cannot be fully covered by vendor-delivered functionality. For instance, organizations often enrich the training data sources with additional fields unique to their operations, or design bespoke AI applications that align with specialized workflows outside the ERP core. Thus, our proposed DevOps framework can be utilized by the ERP vendors, but also by the ERP customers to complement and extend vendor offerings by providing a systematic, lifecycle-oriented approach that organizations can adopt to manage AI-enabled ERP scenarios in a sustainable way. Specifically, while ERP vendors typically provide the technical capability to embed AI models (e.g., via APIs, pretrained models, or embedded analytics), they do not provide comprehensive methods for managing the full AI lifecycle - including data preparation, model development, validation, deployment, continuous monitoring, retraining, and governance. Our framework therefore, delineates responsibilities between the vendor and the customers. The DevOps framework we propose offers structured guidance for this organizational side of the integration process, ensuring that AI capabilities can be scaled, adapted, and governed in a manner consistent with both technical requirements and information systems (IS) concerns such as user adoption, knowledge transfer, and sustainable system evolution.

Prior research highlights that ERP implementation and customization projects are characterized by high complexity, contested power relations, and uncertain outcomes. For example, Dechow and Mouritsen (2005) show how ERP systems not only embed technical functionality but also embody organizational routines, control structures, and power struggles. Similarly, Heinzelmann (2017) emphasizes the challenges of over-customization and the ways in which pre-understandings of stakeholders shape ERP outcomes. These insights are critical for our study: integrating AI into ERP does not occur in a vacuum but interacts with the same socio-technical complexities that have long been central to IS scholarship. We argue that AI integration intensifies these challenges. Unlike traditional ERP customization, AI models evolve dynamically, rely on probabilistic outputs, and require ongoing retraining and monitoring. This introduces new tensions around transparency, accountability, and governance that directly resonate with established IS debates on control, standardization, and organizational alignment. By situating AI-ERP integration within these debates, our study contributes to the IS literature by extending existing knowledge on ERP complexity into the emerging domain of enterprise AI. We not only identify technical and organizational gaps in AI-ERP integration but also highlight how these gaps reflect, amplify, or reconfigure well-known issues in ERP implementation research.

2 Literature Review

We applied the systematic literature review technique (Kitchenham & Charters, 2007), aiming to identify, evaluate, and interpret available scientific results related to our research question of how to develop and operate AI business applications in ERP systems. For this research question, we concluded the search term "(DevOps OR Lifecycle Management OR Development OR Operations) AND (AI OR Artificial Intelligence) AND (ERP OR Enterprise Resource Planning)". Based on selection criteria like relevant content or available license, we chose the literature databases ACM Digital Library, AIS Electronic Library, DBLP Computer Science Bibliography, IEEE Xplore/IEEE-IET Library, SpringerLink, Google Scholar, Wiley Online Library, and ScienceDirect. We applied the previously defined search term to those literature databases. However, as we obtained no results, we relaxed the search string by removing some of the AND conditions and received 811 papers. After screening the identified publications based on the title, abstract, and full text, we determined seven papers which we will discuss in this section.

Lwakatare et al. (2020) examine the challenges of developing software systems that incorporate artificial intelligence and machine learning (ML), with particular attention to the tensions between modern software engineering practices and ML workflows. The strengths of the paper lie in its thorough research methods and use of industrial case studies to provide practical, real-world examples of AI application development. The clear structuring of the stages of the ML process and identification of five central challenges provide a good basis for further research. The comparison between software system development and AI/ML model development is particularly insightful. However, some weaknesses lie in the broad coverage of issues, which may result in some areas lacking depth. The paper would have benefited from a more focused discussion on possible solutions to the identified challenges. While it identifies potential areas of concern in the development and integration of AI applications, explicit suggestions for overcoming these issues are sparse, as the focus is on the process view. The argumentation in the paper is logically structured and well substantiated with empirical evidence and relevant citations. The authors acknowledge the rapidly changing landscape of AI and ML, show a clear understanding of the complexities involved in developing software systems that incorporate these components, and recognize the nascent stage of many AI-based systems.

Cheng (2022) introduces the concept of enterprise information system modeling and analyzes the foundational technologies that underpin it - including Enterprise Resource Planning (ERP), Supply Chain Management System (SCM), Customer Relationship Management (CRM), and Product Data Management System (PDM) - as the basis for the proposed solution. The proposed solution is an internet-based enterprise information system architecture. The paper does a commendable job showcasing the uses of the state-of-the-art artificial intelligence technology in corporate information systems. The research's most significant strength lies in its approach to blend machine learning techniques with business operations to increase productivity and efficiency. The research strongly supports its claims by providing a solution, citing concrete examples of using the backpropagation neural network algorithm, and integrating the model with an ERP system. The paper emphasizes the primary significance of undertaking such research, incorporating the ever-evolving machine learning technology to address issues in the enterprise information systems. Despite these strengths, the paper has significant weaknesses as well. The paper shifts too quickly from defining enterprise information system modeling to proposing machine learning integration in ERP solutions, lacking a smooth transition and detailed reasoning. Although the paper does explore various related technologies, the connections with artificial intelligence and machine learning aspects could have been explored in more detail. The paper makes valid contributions to scientific knowledge by proposing a new design for integrating artificial intelligence and machine learning technology into the enterprise information system. However, the validity of the projected result of increasing productivity is only theoretical without empirical evidence or case studies to support it. Herein lies another weakness of the paper – the absence of profound discussions and empirical evidence to support the proposition. Despite providing a blueprint of the system, an absolute dearth of substantial experimental results, data collection, and analysis significantly reduces the paper's credibility.

Muthusamy et al. (2018) explore both the opportunities and risks linked to applying artificial intelligence in business operations. The authors developed an approach to utilizing artificial intelligence methods in unveiling aspects of artificial intelligence applications and controlling their use in production applications. The research paper is essential in contributing to the scientific discourse on artificial intelligence integration in business processes. By dissecting the potential risks of artificial intelligence models and suggesting mitigation measures, the researchers provide valuable insights. The study exhibits an inclusive approach to tackling the complex dynamics of artificial intelligence model use in business, which serves as a strong foundation for further research. However, the paper exhibits a limited empirical approach to support the theories, indicating a significant weakness. While conceptual discussions are insightful, they would have been significantly enhanced with some empirical evidence or practical case studies. Furthermore, the researchers establish their arguments coherently and logically, contributing to the knowledge on the formulation of artificial intelligence models and their use in business applications. The paper details a systematic approach to the safe deployment of artificial intelligence models, providing insights into the various techniques to mitigate the negative effects of artificial intelligence models in business applications. Yet, though the paper provides useful insights into model drift in enterprise applications, it does not comprehensively address how to develop such a capability.

Paul et al. (2022) propose a conceptual architecture for integrating artificial intelligence into supply chain risk management (SCRM), advancing both theoretical understanding and practical application in this field. The paper highlights the need for architecture capable of enterprise-scale implementation of artificial intelligence in SCRM, assimilating data architectures and artificial intelligence model deployment for maximum insight and decision-making utility. Despite the previous research embodied in the paper encompassing areas from data mining to artificial intelligence cloud architecture, the authors critically affirm the need for the paper – the field remains absent of a comprehensive reference architecture for artificial intelligence in SCRM. It is this gap that this paper seeks to fill. The inheritance of knowledge from previous studies and application of findings through expert opinion culminate in conceptual architecture for artificial intelligence in SCRM that both reimagine the traditional IT stack within organizations and ignite an optimum foundation for implementing artificial intelligence. The authors convey several commendable strengths in their research, notably the focus on creating an optimal balance and effective interaction between artificial intelligence and humans. The necessity of designing artificial intelligence that corroborates human habits and tendencies is reflected in the idea of a human-centered architecture. Here, recognition that artificial intelligence should augment and assist humans, not replace them, is paramount to successful artificial intelligence development and future adoption. Arguably, the most significant limitation of this paper is the difficulty in approaching standardization. Artificial intelligence technologies are ripe with complexities and frequent updates, posing challenges to the alignment with static concepts. The eventual replication and application of the model in realistic scenarios may be harder to implement in real-life scenarios than initially anticipated. The paper also offers valuable insights that contribute to the body of scientific knowledge in artificial intelligence and SCRM. For instance, the consistent emphasis on balancing the role of artificial intelligence and humans in SCRM presents a new viewpoint not commonly approached in conventional artificial intelligence research. The paper offers logical and valid arguments about the necessity of a comprehensive architecture for artificial intelligence, citing previous research reinforces its points.

Omar and Rachid (2023) investigate how artificial intelligence can be incorporated into accounting practices in Morocco, particularly in relation to compliance with International Financial Reporting Standards (IFRS). The strength of this paper lies in its comprehensive exploration of the opportunities and challenges faced when integrating artificial intelligence with IFRS in the accounting system, specifically using Morocco as a case study. It shines a spotlight on the importance of compliance with these standards, the role of accounting education, corporate governance, financial reporting quality, and investment efficiency, and the specific intricacies concerning the Moroccan context. On the other hand, one of the weaknesses of this paper is the lack of quantitative data, which could provide empirical evidence to support the assertions made. The paper indeed contributes to scientific knowledge by exploring the implications of advanced technological integration into the accounting industry. The arguments provided in the paper are valid as they are based on well-researched evidence from relevant literature and actual experiences in the Moroccan context. It gives insights into the possibilities and challenges of artificial intelligence integration in accounting systems and tries to provide a mitigation strategy against potential ethical and legal implications.

Lin (2021) analyzes how artificial intelligence contributes to strengthening the global value chains (GVC) of manufacturing enterprises. The paper is strong in its depth of analysis, acknowledging the position of manufacturing at the GVC's low-end, then proposing solutions for ascension to higher levels. Addressing the advantages of artificial intelligence in manufacturing and logistics offers a new perspective on how these technologies can be integrated. While addressing complex issues, the paper falls short of providing case studies or practical examples where such implementations have been successful or are in progress. The formulations tend to be theoretical, and it may be challenging for the reader well-versed in the manufacturing sector to ascertain the feasibility of the presented propositions. The paper adds scientific knowledge, as the author explores the potential of artificial intelligence in industrial manufacturing, a somewhat uncharted domain. Thus, it could potentially open avenues for more research on merging these two fields. The article does not present definitive arguments; rather, it speculates on the potential advantages. Thus, while valid, they may be tested further for concrete proof.

Wang et al. (2022) examine the intersection of artificial intelligence and management accounting, highlighting pathways for integrating AI into accounting processes. The authors delve into the applications of artificial intelligence technology, describing the evolution and projected future of intelligent accounting. Strengths of the paper lie in its comprehensive discussion on the intersection of artificial intelligence technology and management accounting. It highlights crucial facets of intelligent accounting, effectively showcasing analysis tools as novel applications of artificial intelligence in accounting. The authors also present an all-encompassing view of intelligent accounting's future, recognizing its importance in modern enterprises. However, a significant weakness is the lack of a detailed evaluation of the differences between traditional and intelligent accounting methodologies. Additionally, the paper could benefit from a more in-depth exploration of the potential challenges and impacts of artificial intelligence implementation in accounting practices, providing a holistic depiction of the current technological shift. The paper significantly contributes to scientific conversations surrounding artificial intelligence's integration into accounting practices. The validity of the authors' arguments is substantiated through logical reasoning and referencing prevailing technological trends, ensuring their conclusions are grounded in observable evidence. The research underlines that the fusion of artificial intelligence technology and accounting calls for a revolutionary change in the financial management sector, enhancing its potential for strategic decision-making. The authors detail the operation mechanism of intelligent accounting systems, touching upon the intelligent operation of financial processes.

The systematic literature review reveals that while the seven identified papers address AI applications in business contexts (Table 1), they do not systematically address the development and operations challenges of embedding AI into ERP systems. This gap is significant given fundamental differences between implementing AI capabilities and traditional ERP functionality. To understand this gap, we must examine two research streams that have evolved largely independently:

- **ERP Implementation Literature - Established Challenges:** ERP implementation research has extensively documented socio-technical complexities inherent in enterprise system adoption. Dechow and Mouritsen (2005) demonstrate that ERP systems embody organizational routines, control structures, and power relations, with implementations becoming sites of contested meanings. Heinzelmann (2017) extends this by examining how stakeholders' preunderstandings shape ERP customization, often leading to over-customization or under-customization. These studies establish enduring challenges: (1) tension between standardization and organizational specificity, (2) maintaining system stability while enabling evolution, (3) power dynamics shaping configurations, and (4) user adoption and knowledge transfer. Importantly, this literature treats ERP systems as fundamentally deterministic - once configured, business rules execute predictably.
- **AI Operations Literature - Emerging Challenges:** The AI operations literature addresses fundamentally different challenges stemming from the probabilistic and continuously evolving nature of intelligent systems. Lwakatare et al. (2020) identify five central challenges: misalignment between software engineering and ML workflows, data quality issues, model drift, testing non-deterministic systems, and maintaining explainability. Muthusamy et al. (2018) emphasize risks in deploying AI models, including drift and opacity in decision-making. Unlike traditional software, where bugs can be fixed, AI systems require continuous monitoring and retraining as data distributions evolve.

When we examine the intersection of these streams - implementing AI within ERP environments - we find a significant gap. The challenges identified in ERP research are amplified and reconfigured when embedding AI capabilities:

1. **Configuration vs. Continuous Evolution:** Traditional ERP customization involves relatively stable configuration decisions that can be managed through established change control processes. AI models, however, require continuous retraining as business conditions evolve, creating ongoing configuration challenges that existing ERP governance frameworks are not designed to address.
2. **Deterministic Rules vs. Probabilistic Intelligence:** ERP business rules execute predictably - invoice matching either succeeds or fails based on defined tolerances. AI-based invoice matching, by contrast, produces probabilistic assessments that may vary as models are retrained, creating challenges for organizational accountability structures designed around deterministic processes.
3. **Static Expertise vs. Dynamic Competency Requirements:** ERP implementations require expertise in business process configuration, but once configured, systems operate relatively autonomously. AI-enabled ERP requires continuous organizational capability in evaluating model performance, interpreting accuracy metrics, and making retraining decisions - competencies that most ERP user organizations lack.
4. **One-Time Integration vs. Lifecycle Orchestration:** Traditional ERP change management treats system modifications as discrete events (configuration changes, transportation to production, stabilization). AI integration requires orchestrating continuous lifecycle activities - data preparation, training, validation, deployment, monitoring, retraining - that span both ERP and external AI platforms, creating integration complexities not addressed in existing literature.
5. **Transparency vs. Algorithmic Opacity:** ERP systems' business logic is typically inspectable and auditable - users can trace why a particular calculation occurred. AI models, particularly deep learning

approaches, introduce opacity that conflicts with organizational needs for auditability and regulatory compliance in ERP-managed business processes.

The absence of systematic approaches for managing these intersecting challenges represents a significant gap. Our study contributes to this intersection by proposing and evaluating a DevOps framework tailored to AI in ERP systems, thereby addressing a gap in the literature where these debates have so far remained largely separate.

Table 1. Lessons from Prior Work and Implications for AI in ERP

Study / Stream	Key Insights	Implications for AI in ERP
(Dechow & Mouritsen, 2005); (Heinzelmann, 2017) – ERP implementation studies	ERP projects are complex, often face over/under-customization, and are shaped by power struggles, legacy systems, and user preunderstandings.	AI integration inherits these complexities: AI models may reinforce or challenge existing power structures and require careful governance to avoid misalignment with organizational practices.
(Lwakatare et al., 2020) – AI/ML development challenges	Misalignment between software engineering practices and ML workflows; five central challenges identified (e.g., data quality, model drift).	Highlights the need for ERP-specific DevOps frameworks that integrate AI lifecycle management (training, monitoring, retraining) with established ERP development practices.
(Cheng, 2022); (Lin, 2021) – AI in enterprise architectures and manufacturing	AI can enhance productivity and efficiency, but prior work is often conceptual and lacks empirical validation.	Demonstrates the need for real-world, empirically validated frameworks – addressed in our study through SAP case-based evaluation.
(Muthusamy et al., 2018) – Risks of AI in business	Identified risks such as model drift, lack of transparency, need for mitigation measures.	Reinforces importance of monitoring, retraining, and governance mechanisms within ERP-integrated AI.
(Paul et al., 2022) – AI in supply chain risk management	Proposed conceptual architecture integrating AI and human-centered design.	Supports need for socio-technical integration; AI in ERP must augment user decision-making, not replace it.
(Omar & Rachid, 2023); (Wang et al., 2022) – AI in accounting	AI improves compliance and efficiency, but lacks empirical data and raises regulatory/ethical concerns.	Suggests ERP AI frameworks must ensure compliance, auditability, and explainability – directly motivating requirements in section 4 such as [REQ 5] Deploy and [REQ 6] Monitor .

3 Research Methodology

As the systematic literature review did not yield substantial work regarding implementing artificial intelligence in ERP software, for understanding the problem space, we analyzed artificial intelligence use cases (SAP SE, 2025a; SAP SE, 2025b) in the ERP domain from analysis to development and operations phase. Iteratively, we identified gaps, proposed solutions, and evaluated the improvements with the next use cases. The key questions we considered in the context of the use case analysis were: Should the underlying problem be resolved with artificial intelligence, or could rule-based techniques be a reasonable alternative? What technical functionality is required to implement the artificial intelligence use case? What were the shortcomings of already developed use cases? To have a broad coverage, the use cases originated from the ERP core business processes idea to market, source to pay, plan to fulfill, lead to cash, recruit to retire, acquire to decommission, governance, and finance. From the use case analysis, we derived the business requirements and validated them additionally with customers and domain experts. We identified 20 gaps [GAP 1–20] through a structured coding and analysis process. First, we examined each use case to document encountered challenges in development and operations. Second, we categorized these challenges into thematic clusters (e.g., configuration, training, deployment, monitoring). Third, we validated these clusters through internal workshops with domain experts and ERP customers, ensuring that the gaps reflected real-world practice rather than isolated experiences. Where possible, we triangulated our findings with existing literature to determine whether a gap represented a known ERP challenge (e.g., data quality issues) or a novel AI-specific requirement (e.g., model retraining or safeguarding upgrades). This iterative process aligns with Design Science Research (Peffers et al., 2007), which emphasizes cycles of problem identification, artifact design, demonstration, and evaluation. We recognize that using SAP use cases introduces limitations in terms of generalizability; however, given SAP's position as a leading ERP provider, these cases offer insights into challenges faced by a wide range of industries and regions. We explicitly reflect on these limitations in the conclusion section. The primary gap we identified was the absence of a framework for development and operations (DevOps) of AI features in the context of ERP. By AI features, we mean functionality realized with AI techniques that are integrated into business processes and user interfaces of ERP systems. Initially, the AI features were developed and operated heterogeneously, which resulted in a high total cost of development and operations. Therefore, we proposed a DevOps framework to close this gap. We refer to the DevOps framework, a software solution providing the functionality specified as [REQ-01] to [REQ-06] in the next section. The DevOps framework harmonizes the implementation and operations of artificial intelligence use cases and reduces the total cost of ownership. For validating our concepts for the DevOps framework, we introduced a concrete implementation as a feasibility proof based on the ERP platform of SAP as the market-leading vendor. We utilized the DevOps framework for the implementation of a real-world use case to prove the feasibility of the suggested solution. Figure 1 summarizes our methodology. The outer circle depicts our iterative process model, and the inner circle the applied methods.

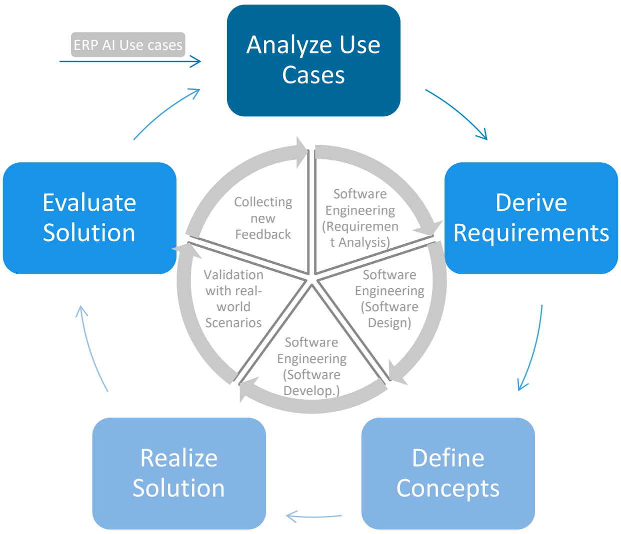

Figure 1. Process Model (outer circle) with Applied Methods (inner circle)

We utilized computer science methods taken from the field of software engineering (Broy & Kuhrmann, 2021). To derive requirements from the gaps, we applied the software engineering techniques for requirement analysis (Dick et al., 2017) for defining solution concepts from the requirements. We facilitated the software engineering practices for software design (Dooley & Kazakova, 2024) to implement the concepts. We used the software engineering procedures for software development (Dooley & Kazakova, 2024). These software engineering methodologies have already achieved a high level of maturity, are widely accepted in computer science, and therefore not further explained. To ensure verifiability of the deduction of our results, we applied the design science research methodology (Peffers et al., 2007) and operationalization through iterative action research cycles (Baskerville & Wood-Harper, 1996). Thus, according to information systems theory taxonomy (Gregor, 2006), for design and action builds the underlying theoretical foundation for this elaboration.

Our empirical material was collected through a combination of documentation, expert workshops, customer pilots, and direct observation of ERP AI implementations. Specifically, we conducted six domain expert workshops (with ERP architects, data scientists, and DevOps engineers, n = 18 in total) and four customer pilot workshops (with business administrators, ERP consultants, and operations managers, n = 22 in total). These workshops were structured around the identification of gaps [GAP 1–20] and validation of requirements [REQ 1–6], which will be detailed in section 4. Short, anonymized comments were systematically documented; for example, one expert emphasized that “monitoring must include, in addition to technical aspects, also a business process view, otherwise adoption will remain superficial,” while a customer pilot noted that “the prerequisite checks can save days of manual assessment.” We complemented workshop data with documentation analysis of ERP AI projects (SAP SE, 2025a; SAP SE, 2025b) and observations of implementation and operations processes. The combined dataset provided both technical and organizational perspectives on AI lifecycle challenges. Analysis proceeded in iterative coding cycles, where workshop transcripts and observation notes were thematically coded against the categories of configuration, training, deployment, and monitoring. Triangulation across data sources (experts, customers, documentation) increased validity by ensuring that identified gaps reflected recurrent patterns rather than isolated experiences. Reliability was supported by the structured format of workshops and coding protocols, while replicability is fostered through transparent reporting of our data sources, coding scheme, and evaluation process. Embedding these cycles in Design Science Research (Peffers et al., 2007) and Action Research (Baskerville & Wood-Harper, 1996) ensured that findings were not only rigorously derived but also iteratively validated in practice. Each cycle of diagnosis, planning, action, and reflection was informed by both theoretical insights and empirical evidence, reinforcing the robustness of our framework development. This methodological approach is particularly suited to studying AI in ERP systems, as it captures the performative and evolving character of AI (Orlikowski & Scott, 2025) while providing structured design and evaluation of artifacts.

The design science research approach consists of stages and is particularly suitable for research that aims to develop innovative artifacts addressing practical and theoretical challenges, such as the integration of AI into ERP systems:

1. Problem identification and motivation. Rather than relying exclusively on prior literature, which provides limited insights into AI-specific ERP challenges, we conducted an in-depth analysis of numerous AI use cases across different ERP modules. This analysis allowed us to surface concrete issues depicted in Section 4 as [GAP 1-20], such as insufficient training data fields, and complex system setup.
2. Define the objectives for a solution. From this use case analysis, we derived six core requirements listed in Section 4 as [REQ 1–6] that a DevOps framework for ERP must address. These requirements span both technical aspects and socio-organizational dimensions.
3. Design and development. Based on the identified requirements, we designed in Section 5 a domain-specific DevOps framework for AI in ERP systems, extending traditional DevOps practices with mechanisms for AI model management and development.
4. Demonstration. We instantiated the proposed framework in section 6 based on SAP ERP as the market leader that develops and operates AI scenarios, and its customers extend vendor-delivered AI scenarios (e.g., by adding organization-specific fields to training data) and developing novel AI use cases that go beyond the ERP core. This allowed us to observe the framework in action within real-world contexts.
5. Evaluation. The framework was evaluated through case studies with ERP, as depicted in section 6. Effectiveness was assessed based on the framework’s ability to support continuous integration, deployment, monitoring, and retraining of AI models, as well as its support for organizational adoption.
6. Communication. The outcomes of this research are communicated in this paper, which contributes to the academic discourse on AI in enterprise systems and provides practitioners with a concrete, lifecycle-oriented DevOps framework that can be applied by ERP vendors and customers.

We ground our investigation in the sociomateriality perspective (Orlikowski, 2007; Orlikowski & Scott, 2008, 2025), which posits that materiality (technology) and sociality (human practices, organizational structures) are constitutively entangled. Orlikowski and Scott (2008) argue that the social and the material are considered to be inextricably related - there is no social that is not also material, and no material that is not also social. Traditional ERP configurations represent relatively stable sociomaterial assemblages: business rules execute consistently, embodying organizational policies predictably. AI models represent continuously evolving sociomaterial assemblages. This introduces several implications:

1. **Shifting Loci of Agency**: In traditional ERP systems, agency resides primarily with humans who configure rules that the system then executes. With AI, agency becomes distributed between human decisions (what data to include, when to retrain), algorithmic learning (which patterns the model identifies), and material constraints (available computing resources, data quality). As Orlikowski and Scott (2025) argue in their work on AI performativity, AI systems exhibit forms of agency that emerge from the entanglement of data, algorithms, infrastructures, and human practices in ways that cannot be reduced to any single element.
2. **Temporal Dynamics of Entanglement**: Traditional ERP implementations involve intense sociomaterial negotiation during configuration, followed by relative stability in operation. AI-enabled ERP requires continuous sociomaterial negotiation because model retraining periodically reconfigures what the system does and how it performs organizational work. A procurement AI that learns new patterns may effectively change organizational purchasing practices without explicit human reconfiguration, creating ongoing questions about accountability and control.
3. **Opacity and Interpretive Flexibility**: Traditional ERP configurations are materially inspectable - business users can examine approval workflows or pricing rules. AI models, particularly neural networks, introduce material opacity that requires new forms of sociomaterial practice for interpreting and legitimating their outputs. This creates challenges for organizational accountability: when an AI forecast influences inventory decisions that prove costly, determining responsibility requires navigating the entanglement of human training decisions, algorithmic learning processes, and data quality issues.
4. **Multiplicity of Performances**: While ERP business rules execute identically each time (given the same inputs), AI models may produce varying outputs as they are retrained, creating what is called performative multiplicity. The “same” AI capability performs organizational work differently across time and contexts, complicating efforts to standardize business processes - a core value proposition of traditional ERP systems.

From a sociomateriality perspective, our proposed DevOps framework is not merely a technical solution for managing AI lifecycle activities. Rather, it represents an attempt to create organizational structures and practices that can accommodate the dynamic sociomaterial entanglements introduced by AI in ERP environments.

4 Identified Requirements

During the development of the ERP AI use cases (SAP SE, 2025a; SAP SE, 2025b), we faced various challenges, for example, provisioning high-quality data during the data science exploration phase or ensuring compliance with laws for AI applications. However, critical key gaps existed in lifecycle management concerning the development and operations phases. The gaps listed in this section give rise to business requirements, which we address in the next section by proposing concepts for resolving them. While some of the identified gaps may appear intuitive (e.g., the need for AI expertise), our analysis demonstrates that they have profound implications in ERP contexts and are insufficiently addressed in current literature. For example, although prior work has acknowledged the skills gap in ERP implementation projects (Heinzelmann, 2017), the extension of this problem to AI lifecycle management introduces new challenges, such as the interpretation of model accuracy metrics or the governance of retraining cycles. By systematically documenting these gaps across multiple ERP AI use cases, we contribute empirical evidence that these issues are not merely theoretical but recurrent and costly obstacles in practice. To illustrate the gaps and requirements, we utilize the following concrete example throughout the descriptions: A manufacturing company wants to implement AI-powered demand forecasting in its ERP system to optimize inventory management. The AI model should predict product demand based on historical sales data, seasonal patterns, marketing campaigns, and external factors like weather data.

[GAP 1] Complex Setup: The configuration of AI applications is daunting due to the lack of automation and standardization. This necessitates extensive support from ERP vendors, which might be feasible for pilot programs but not scalable for mass adoption. Heterogeneous individual solutions and tools increase the total cost of ownership (TCO). Example: The IT team struggles for weeks to configure the demand forecasting AI application. They need to manually set up data connections between the ERP system, external weather APIs, and the AI platform. Each configuration requires specialized knowledge of different systems, making it impossible to scale to other AI use cases without vendor support.

[GAP 2] Need for AI Expertise: Successful deployment of artificial intelligence solutions requires the customer to evaluate the quality and applicability of the trained models. This ranges from technical integrations to complex personalized models for business process automation. The customers, however, often lack this specific knowledge, leading to the delegation of these tasks back to the ERP vendor. Example: The business users cannot evaluate whether the forecasting model's 85% accuracy is acceptable for their inventory decisions. They don't understand concepts like MAE (Mean Absolute Error) or seasonality detection, forcing them to rely entirely on the ERP vendor's recommendations.

[GAP 3] Uncertain Benefits: Lifecycle management challenges must be navigated before the customer can ascertain if the artificial intelligence scenario delivers the anticipated benefits. Customers cannot assess the potential return on investment (ROI) before starting the implementation process. This uncertainty makes it difficult for customers to justify the investment and results in low adoption rates for the AI application. Example: Management cannot determine if investing €200,000 in AI demand forecasting will reduce inventory costs sufficiently. Without clear ROI predictions before implementation, the project approval process stalls for months.

[GAP 4] Missing Technical Metrics: There are no metrics to indicate the technical suitability for implementing the artificial intelligence scenario. These metrics should include, for example, the minimal amount and quality of training data in the ERP system, which must be sufficient to achieve the appropriate performance of the artificial intelligence model. Particularly, as customers do not have direct access to application and configuration tables, the availability of technical metrics is crucial. Example: The system cannot automatically determine if the company has enough sales history (e.g., minimum 2 years of daily sales data for 1000+ products) to train an effective forecasting model. This discovery happens only after failed training attempts.

[GAP 5] Heterogeneous Development: There are use cases involving the same artificial intelligence pattern being implemented differently by application teams. These are typically handled by developers using self-defined technical frameworks, with each use case being developed differently. The root cause of this is the various available artificial intelligence libraries and technologies, which require individual handling by the developers. However, the potential for synergy and reuse across applications is often overlooked. This results in decreased development efficiency. Example: The supply chain team implements demand forecasting using TensorFlow, while the sales team develops price optimization using scikit-learn. Both solve similar time-series prediction problems but with incompatible frameworks, duplicating development effort.

[GAP 6] No Prerequisite Checks: Customer request evaluating the prerequisites for the artificial intelligence application, for example, which configuration for the underlying business process is required. Customer system readiness and data sufficiency for the intended artificial intelligence use case are crucial factors to consider, but prerequisite checks on the customer's ERP system before they invest in a license, configuration, or send data for model training are not supported. Example: The system cannot verify that the required master data (product hierarchies, customer segmentation) and business process configurations (sales order processing workflows) are properly set up before attempting to train the forecasting model.

[GAP 7] Missing Scenario Discovery: There is no central place in the ERP system where the available artificial intelligence scenarios are listed so that customers can find the relevant use cases for them. The lack of clear, comprehensive documentation leaves customers doubtful and confused about adopting artificial intelligence scenarios. Thus, customers should be provided with information to understand the available artificial intelligence functionality and the required infrastructure. Example: Business users browse through scattered documentation trying to understand what AI scenarios are available. They cannot easily discover that demand forecasting could be combined with automatic reorder point optimization.

[GAP 8] Complex Connectivity Configuration: Initial communication configuration on the ERP system is required to access the artificial intelligence services on the AI technology platforms. Customers must create several remote destinations in the ERP system with adequate security measures. Additionally, customers need to set up communication arrangements. These steps can be error-prone due to a lack of specialized knowledge. Furthermore, this process is mainly manual and needs automation. Example: The IT team spends days manually creating multiple remote function calls and destination configurations to connect the ERP system to the AI platform, with frequent authentication and network configuration errors.

[GAP 9] Insufficient Data Replication: Training data replication occurs through manually executed programs, which trigger asynchronous training processes on the AI technology platform. The customer sends their data and then opens a ticket to request a training model for their company. Once model training is complete, they are notified via the same ticket. The main issue is the lack of automation and transparency in the data sending process. Customers lack clear information about the data they are sending, when the data exchange has ended, whether the data transfer was successful, and what will happen to their data after it is sent. Example: The monthly sales data extraction for model training requires manual execution of customer programs. The team has no visibility into whether the data transfer is completed successfully and when the new model will be ready.

[GAP 10] Inadequate Activation Mechanism: The process of activation does not support more than one phase, which poses a challenge when the artificial intelligence model needs to be trained in production after activation. This leads to a period where the application is active but unable to call the artificial model for inference. To overcome this challenge, customers require a two-phased activation process. In the first phase, all the necessary functionalities needed to train the initial model should be provided. This includes the capabilities to transfer training data and to train and deploy the artificial intelligence model. The second phase should activate the entire business process using the model trained in the first phase. Example: Two weeks after the forecasting model had been deployed, the team recognized that the accuracy of the model had been insufficient. Unfortunately, the demand planning module immediately utilized the model and generated poor predictions with negative implications on the inventory.

[GAP 11] Lacking Training and Retraining: Customers need a more efficient and transparent process for training new artificial intelligence models for applications or retraining of present models. The existing process is slow, costly, and lacks transparency and adequate feedback because the procedure requires manual intervention from the DevOps team. Issues with this process include delays in model readiness, a lack of transparency for the customer, and the performance information being insufficient for informed decision-making. The model training should be a self-service for the customer for all scenarios involving customer-trained models. Example: When seasonal patterns change (e.g., COVID-19 impact), retraining the demand forecasting model requires submitting tickets to the DevOps team and waiting 3-4 days for completion, during which forecasting accuracy degrades significantly.

[GAP 12] Challenging Global Models: Global models are trained by the ERP vendor and serve multiple customers. The responsibility of retraining these global models lies with the ERP vendor, but there are unresolved issues like identifying when retraining is needed, acquiring the appropriate data for retraining, and evaluating the model's performance. It is unclear whether customers can choose to retain older model versions if the new ones perform poorly in their specific use cases. Thus, the ERP vendor must announce new model versions in advance, allowing customers to prepare. Example: The ERP vendor updates the global demand forecasting algorithm, but the new version performs poorly for the company's specific industry. There's no way to revert to the previous model version or to understand what changed.

[GAP 13] Negative Load Impact: Training jobs can occasionally require excessive resources, which could negatively affect the performance of the productive ERP system. Thus, careful management of these training jobs, especially when they run on the productive ERP, is necessary. Detecting critical training jobs, handling the load balance with background tasks, and assigning sufficient resources is not optimal and requires advanced techniques. Example: Monthly model retraining jobs consume excessive CPU resources during business hours, causing the ERP system to slow down and affect critical order processing operations.

[GAP 14] Lacking Model Management: Customers require the ability to deploy multiple models concurrently for the same scenario but different segments (e.g., countries, product types) to achieve higher prediction accuracy. However, this feature is not supported. Another need for customers is the ability to manage models with ease. Presently, models can be versioned, which allows customers to deploy a model of their choice if they know the model number. However, there is no support for enhanced model management functionality, for example, rolling back to the last model version if the performance unexpectedly degrades after deploying a new model version. Example: The company needs separate forecasting models for different product categories (electronics vs. seasonal goods), but the system can only deploy one model at a time. They cannot switch the inference calls to the corresponding model for more accurate results.

[GAP 15] Insufficient Model History and Validation: For compliance regulations, tracking and presenting the history of active models is required. However, this information is accessible only to the DevOps teams, but not to customers. The history of active models, which includes the model version used for a specific inference call, is essential for auditing purposes but also for monitoring. Performance evaluation is carried out on the training data by the DevOps team, and this information is conveyed to the user through a program on model key performance indicators (KPIs). However, there is a gap: manual testing of the model or a performance comparison between the new and currently active model is not possible based on this program, although this is a customer requirement. Example: During an audit, the company cannot provide evidence of which model version generated specific forecasts used for regulatory inventory reporting. Manual testing of new models against business KPIs is impossible.

[GAP 16] Lacking Model Extensibility: Customers need to extend the artificial intelligence models, for example, by considering additional attributes in the training data. However, they can only use the models as delivered or replace them with a customer-developed model, which requires substantial effort and expertise. To make AI applications more adaptable to customer needs, proper lifecycle management of these extensions needs to be guaranteed. This includes preserving customer adjustments when a new model or training script is released, and ensuring all new releases are compatible with all supported customer adjustments. Example: The business wants to include promotional data and competitor pricing in the forecasting model, but they must either accept the standard model or invest months in developing a complete customer solution.

[GAP 17] Insufficient Model Monitoring: Technical monitoring of artificial intelligence models is necessary but not sufficient in the domain of ERP, where monitoring on the level of business processes is required but missing. Model monitoring is complex and requires solving two main problems: collecting the right data and automatically calculating performance metrics based on this data. Determining model degradation in terms of decreased prediction power is an important but missing capability of the traditional monitoring approach. Model degradation can be detected (e.g., through model monitoring or by observing changes in the distribution of inference data). Example: Technical monitoring shows the AI service is running, but business users don't know that forecasting accuracy has dropped from 85% to 65% due to changing market conditions, leading to significant inventory overstock.

[GAP 18] Safeguarding Upgrades: The update and upgrade process for the artificial intelligence applications are not sufficiently realized regarding managing the lifecycle and dependencies of new artifacts like models. Updates must respect customers' investments in artificial intelligence functionality, guaranteeing not to disrupt productive artificial intelligence models or related business processes. For the ERP system, updates should keep existing artificial intelligence scenarios running without necessitating retraining. Furthermore, customer extensions must be protected during all update scenarios. Example: A quarterly ERP update breaks the demand forecasting integration. The AI application stops working, and restoring functionality requires retraining models and reconfiguring business processes.

[GAP 19] Inadequate Availability Mechanism: Customers require a systematic protocol that can be followed when a deployed model becomes unavailable. At present, no such workflow is established. Although the infrastructure is designed to prevent model unavailability, it is acknowledged that a protocol should be in place for such situations. Customers also want to have the ability to know when a model becomes unreachable. Typically, customers discover a model is unreachable only when an inference call fails, and this information rests with the business user, who may not have the skills or responsibility to remedy the problem. The issues with model availability can range from broken configurations to network problems to malfunctioning containers. It is vital to notify an administrator when such problems occur. Example: The forecasting model becomes unavailable due to network issues, but planning managers only discover this when generating weekly demand plans. There's no proactive alerting or fallback mechanism.

[GAP 20] Lacking Supportability: There is a need for improvements in the supportability of AI applications. Usually, DevOps teams are involved for troubleshooting, but the complexity of the infrastructure and the non-intuitive nature of logs and error messages make this process challenging and time-consuming. To address these concerns, a revamp of the quality process chain and professionalization of the support process to ensure end-to-end supportability is required. The goal should be to empower customers by improving error messages, which in turn would facilitate self-help. Example: When forecasts become unrealistic (e.g., predicting negative demand), the support team struggles to diagnose the issue because error logs are scattered across different systems with cryptic technical messages.

We used the gap analysis in the previous section to assess the difference between the current state and the desired future state regarding the development and operations of AI applications. We now specify this target state with business requirements which are derived from the identified gaps by utilizing software engineering methods for requirement analysis as outlined in the methodology section.

[REQ 1] Create: Developers for artificial intelligence applications integrate the models seamlessly into the ERP business processes and user interfaces. However, as various technologies and libraries are available for artificial intelligence, the development task can result in multiple heterogeneous approaches. Therefore, a standardized programming model is required that hides the technical differences and provides a uniform abstraction to keep the application coding decoupled from the underlying artificial intelligence infrastructure. Thus, the developer can focus on implementing business logic for the AI application. Standardizing the programming model also reduces the total cost of development (TCD), as for example, the training efforts decrease, and ensure uniformity across all artificial intelligence applications of the ERP system. [GAP 5, 16]. Example: The development team uses a unified AI framework that abstracts different machine learning libraries behind a common interface, enabling both supply chain and sales teams to implement forecasting solutions using identical development approaches. This standardization eliminates the need for teams to learn multiple AI technologies and ensures consistent implementation patterns across all demand forecasting applications in the ERP system.

[REQ 2] Check: Customers frequently encounter difficulties in identifying the necessary technical and business prerequisites for training and using artificial intelligence applications. A significant data volume is crucial for effectively training AI algorithms, and the relevant business processes must be initiated and configured to create a valid framework for the training activities. As the number of AI applications increases, conducting manual assessments becomes unfeasible due to the substantial total cost of ownership (TCO) and overwhelming complexity involved. Therefore, an automated system for checking prerequisites is essential to ensure that the conditions needed for training and utilizing each AI application are satisfied. [GAP 2, 3, 4, 6]. Example: The system automatically validates that minimum data requirements are met, including verifying 24 months of sales history, complete product master data, and proper business process configurations before allowing AI model training to commence. This automated validation prevents failed implementations and provides clear feedback on system readiness, eliminating weeks of manual assessment and reducing project risks significantly.

[REQ 3] Configure: Prior to utilizing artificial intelligence scenarios, customers need to establish connectivity with the AI technology platform. The onboarding process involves creating a customer account with service entitlement and a service key. The content of the service key provides the necessary information for the ERP system to initially communicate with the AI services on the technology platform. This manual process is labor-intensive and challenging for customers. Therefore, it would be beneficial to automate this setup using a wizard that guides users through the steps to provision and link to the AI technology platform. Additionally, the setup should support customizing hyperparameters and filtering training data. [GAP 1, 7, 8]. Example: A guided configuration wizard enables business administrators to select demand forecasting from an AI scenario catalog and automatically establishes secure connections to the AI platform without manual technical setup. The wizard provides business-friendly parameter configuration options and integrates data sources through intuitive interfaces, reducing setup time from hours to minutes.

[REQ 4] Train: Customers must conduct training for AI models to be utilized effectively. However, this training is often manual, lengthy, and lacks transparency. The KPIs related to model accuracy are inadequate for decision-making purposes. Thus, it is recommended that the training process be available as a self-service, allowing customers across all AI applications to adjust parameters for optimal training results. Errors and warnings should be articulated in language accessible to individuals without AI expertise. Support should be in place for fully automated training runs scheduled as jobs or triggered by specific events. [GAP 9, 11 ,12 ,13]. Example: Business administrators can initiate model training through a simplified interface that provides real-time progress updates in plain language and allows scheduling of automated retraining cycles based on business calendar events. The self-service approach eliminates dependency on technical teams and enables rapid iteration on model parameters to optimize forecasting accuracy for specific business contexts.

[REQ 5] Deploy: Customers desire control over when trained models are deployed and activated. Typically, deployment is handled by the DevOps team, characterized by laborious manual procedures that are costly and time-intensive. Automating this process would allow customers to manage deployment as a self-service feature, including the simultaneous launch of multiple models for activities like A/B testing before activation. It should also be possible to deactivate models, for instance, those lacking accuracy. A log of model activation and deactivation events should be maintained for purposes of monitoring and auditing. [GAP 10, 14, 15]. Example: The system provides controlled deployment capabilities, including A/B testing functionality that allows new models to be validated on subsets of products before full activation, with one-click rollback options if performance degrades. Complete audit trails track all deployment activities and model version changes, ensuring compliance requirements are met while giving business users control over when AI predictions become active in production processes.

[REQ 6] Monitor: Customers are interested in evaluating the quality of models in use during the runtime. However, AI infrastructure typically offers only technical monitoring without incorporating business process perspectives. Monitoring should provide customers with a comprehensive understanding of employed AI models via a centralized interface for all artificial intelligence scenarios. This should include model status, accuracy KPIs, inference call conditions, and data processing volume. In cases of problems, alerts should notify administrators for action. Recommendations for problem-solving should be offered to administrators. Statistics like error rates, resource usage, and associated costs should be available. [GAP 17, 18, 19, 20]. Example: A centralized monitoring dashboard presents forecast accuracy trends and business impact metrics in executive-friendly formats, with automated alerting when model performance drops below acceptable thresholds. The monitoring system provides actionable recommendations for performance improvements and tracks cost-benefit metrics, enabling continuous optimization of AI investments and proactive management of the model lifecycle.

To improve transparency and traceability, we have added a mapping between the identified gaps [GAP 1–20] and the six requirements [REQ 1–6] that underpin our proposed DevOps framework. This mapping demonstrates how each requirement directly addresses one or more observed gaps, and where applicable, how these align with insights from prior IS and ERP literature. Table 2 summarizes the gaps and requirements. This table illustrates that while some requirements address well-known ERP challenges (e.g., customization complexity, user training), others specifically tackle AI-related lifecycle concerns such as model retraining, monitoring, and safeguarding upgrades.

Table 2. Mapping of Requirements to Addressed Gaps

Requirement	Addressed Gaps
REQ1: Create	GAP5 (Heterogeneous Development), GAP16 (Lacking Model Extensibility)
REQ2: Check	GAP 2 (Need for AI Expertise), GAP3 (Uncertain Benefits), GAP4 (Missing Technical Metrics), GAP6 (No Prerequisite Checks)
REQ3: Configure	GAP1 (Complex Setup), GAP7 (Missing Scenario Discovery), GAP8 (Complex Connectivity Configuration)
REQ4: Train	GAP9 (Insufficient Data Replication), GAP11 (Lacking Training and Retraining), GAP12 (Challenging Global Models), GAP13 (Negative Load Impact)
REQ5: Deploy	GAP10 (Inadequate Activation Mechanism), GAP14 (Lacking Model Management), GAP15 (Insufficient Model History and Validation)
REQ6: Monitor	GAP17 (Insufficient Model Monitoring), GAP18 (Safeguarding Upgrades), GAP19 (Inadequate Availability Mechanism), GAP20 (Lacking Supportability)

5 Solution Architecture

Traditional DevOps practices in ERP environments have primarily focused on application lifecycle management: managing code deployment, system configuration, transport management, and ensuring smooth upgrades within standardized ERP landscapes. These practices are largely deterministic and emphasize stability, compliance, and consistency across organizational units. In contrast, integrating AI into ERP systems introduces qualitatively different lifecycle requirements that go beyond conventional DevOps. AI models are probabilistic rather than deterministic, and their performance depends heavily on data availability, quality, and drift over time. Unlike ERP configuration changes, AI models require continuous retraining, monitoring, and validation to remain effective as business processes, data distributions, and external conditions evolve. This creates a need for lifecycle practices that address issues such as data governance, automated retraining pipelines, model versioning, explainability, and auditability - dimensions that traditional ERP DevOps frameworks do not systematically cover. Our proposed DevOps framework therefore, differs from existing ERP DevOps environments in two important ways:

- Lifecycle focus: It extends the DevOps perspective beyond software deployment to encompass the entire AI lifecycle, including data preprocessing, model training, evaluation, deployment, monitoring, and retraining.
- Organizational alignment: It integrates technical lifecycle activities with business-facing requirements such as transparency for end-users, self-service configurability for administrators, and mechanisms for regulatory compliance.

This dual focus ensures that the framework not only supports the technical embedding of AI into ERP systems but also addresses the socio-technical challenges emphasized in the IS and ERP literature (e.g., Dechow & Mouritsen, 2005; Heinzelmann, 2017). In this way, the framework provides a domain-specific extension of DevOps tailored to the unique complexities of AI-enabled ERP environments, rather than a generic application of DevOps principles.

To address the business requirements [REQ 1] to [REQ 6] we design a coherent DevOps framework [13] for embedding artificial intelligence into ERP software. The objective of the DevOps framework is to standardize the development and operations (DevOps) of artificial intelligence ERP applications. To achieve this goal, we suggest a concept for standardized development [REQ 1] and a concept for standardized operations [REQ 2–6] concerning artificial intelligence ERP applications. The development of the DevOps framework followed an action research logic (Baskerville & Wood-Harper, 1996) integrated with design science principles (Peffers et al., 2007). Each cycle began with the identification of challenges from real-world use cases (diagnosis), followed by the derivation of requirements (planning), the design of framework components (action), and their implementation and evaluation in SAP ERP environments (reflection). Feedback from domain experts and customer pilots informed subsequent iterations, ensuring continuous refinement of the framework. Importantly, each of the six requirements [REQ 1–6] directly maps to specific gaps [GAP 1–20]. For example, [REQ 2] Check addresses [GAP 2-4, 6] and [REQ 6] concerning the absence of systematic prerequisite checks and ROI transparency. [REQ 5] Deploy addresses [GAP 10, 14, 15] regarding inadequate activation, model management, and auditability. This traceability demonstrates that the framework is not a purely technical solution, but a research artifact systematically grounded in empirical problem analysis. By explicitly articulating the action research cycles and linking the framework to both practice-derived challenges and established ERP literature, we show that our contribution is not merely a tool but a rigorously developed design artifact addressing a recognized gap in IS research. We applied well-known software engineering methods for software design, including unified modeling language (UML) for conducting the requirements to software design as depicted in the methodology section. For the conceptual diagrams shown below, we utilized an extension of the UML component diagrams. The diagram models the static structure of a software system using active elements (agent as a rectangle), and passive elements (storage as an oval, channel as lines with arcs), visual elements. Agents represent system components such as software modules. Storage – like databases – hold information that agents can read and modify. Channels facilitate communication between agents, transport but do not store information, and can be unidirectional (arc with an arrow in the direction of the request labeled with R) or bidirectional (arc withoutan arrow).

Figure 2 illustrates the development concept for implementing artificial intelligence into ERP software. Basic tasks like ranking, categorization, and forecasting can be effectively tackled using conventional algorithms such as classification, clustering, regression, and time series analysis. These algorithms generally require minimal memory and CPU resources, making them suitable for direct implementation within the ERP platform where the application data for model training and the AI-driven business processes are located. We refer to this development strategy as embedded artificial intelligence, characterized by a low total cost of ownership (TCO) and low total cost of development (TCD) because it avoids data transfers and additional software requirements. The embedded AI approach leverages AI libraries provided by the database system. During the exploration phase, data scientists identify the appropriate algorithms and necessary application data to solve a particular artificial intelligence challenge. Subsequently, developers build the pipelines needed to train these algorithms using the relevant application data and integrate the derived inferences into business processes or user interfaces. Example: For demand forecasting scenarios involving standard products, basic algorithms like linear regression and ARIMA time series analysis are implemented directly within the ERP database system using built-in machine learning libraries. This approach leverages two years of historical sales data that remains securely stored within existing ERP database tables, enabling model training to occur locally without data extraction or replication to external systems, thereby minimizing the total cost of ownership while ensuring data security and seamless integration into Material Requirements Planning (MRP) calculations.

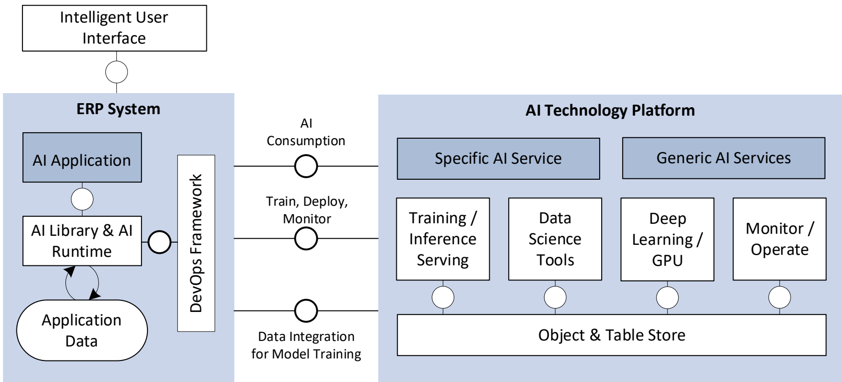

Figure 2. DevOps Framework – Development Concept

Complex tasks such as image recognition, sentiment analysis, and natural language processing require deep learning algorithms powered by neural networks. These algorithms require substantial amounts of data and significant GPU processing for training models. To prevent any negative impact on the transactional ERP system and avoid disrupting business processes, we recommend allocating these scenarios to AI technology platforms, a development pattern we refer to as side-by-side artificial intelligence. Typically, the data necessary for these tasks – such as images, audio files, text documents, historical records, and application logs – are stored in business data lakes rather than ERP systems. The consumption of trained models often involves remote interfaces integrated into business process flows and user interfaces. For large-scale processing needs, these interfaces should support either bulk operations or enable local deployment of inference models. The aim is to provide inference outputs to the right user, at the optimal location and timing – a concept we refer to as built-in artificial intelligence. Ideally, users should remain unaware of whether a feature utilizes AI services. ERP users, predominantly business professionals, generally have limited expertise in data science or statistical methods. Therefore, it is essential to shield these mathematical components and translate inference results into terms familiar to ERP users to ensure effective use and acceptance of AI-driven business applications. Example: Complex forecasting scenarios incorporating external data sources such as weather patterns, social media sentiment, and economic indicators utilize deep learning neural networks hosted on separate AI technology platforms like Google AI or Azure AI. External data from APIs and market indices are combined with internal promotional data, competitor analysis, and market research in business data lakes to train sophisticated ensemble models using LSTM and transformer networks, with GPU-intensive training processes deliberately isolated from the production ERP system to prevent performance impact while handling complex product categories with irregular demand patterns.

For overseeing the lifecycle of an AI application, we recommend adopting the DevOps framework. This framework is designed to streamline and standardize the development and operation of intelligent scenarios within ERP systems. It simplifies lifecycle management tasks – such as prerequisite checks, configuration, training, deployment, monitoring, and inference consumption – across a variety of business domain-specific intelligent scenarios. By offering straightforward functionality and features, the DevOps framework allows individuals without AI expertise to manage lifecycle operations for intelligent scenarios. It breaks down the complexities of managing intelligent scenarios across different platforms, such as the ERP system and AI technology platform, making them accessible. Business administrators can perform lifecycle management for intelligent scenarios via self-service from a centralized control center. The components for handling the operations tasks are illustrated in Figure 3. The DevOps framework provides a unified operational experience for both embedded artificial intelligence and the side-by-side artificial intelligence models by treating the intelligent scenario as a design-time artifact encompassing all development objects. For side-by-side AI scenarios, the DevOps framework includes a consumption client REST API specifically for the AI technology platform. This API is equipped with the necessary logic to invoke platform-specific REST APIs in a native manner. The AI technology platform offers a comprehensive suite of REST APIs for diverse functions, such as managing AI scenarios, training, deployment, and metrics. The DevOps framework orchestrates these APIs to deliver a simplified perspective for non-AI experts when handling side-by-side intelligent scenarios. It assumes that the connection between the AI technology platform and the ERP system is pre-established, with all necessary authorizations and authentications in place, enabling data exchange for model training or batch inference. The DevOps framework includes the following applications for developers and operations personnel:

- Intelligent Scenarios: Using this application, developers can set up intelligent scenarios within the DevOps framework, which involves defining essential details and specifying a class that links to the code responsible for AI logic, such as data transformations. The core design-time component is an intelligent scenario, which includes all the required artifacts for creating an AI-driven solution. Customers and partners can use this application to design and develop their own intelligent scenarios tailored to their needs.
- Intelligent Scenario Management: This application empowers business administrators and operations experts to execute tasks such as prerequisite checks, configuration, training, deployment, and monitoring of intelligent scenarios tailored to specific business domains. It emphasizes accessibility for those without artificial intelligence expertise, allowing them to efficiently manage intelligent scenarios without the need for deep technical knowledge.

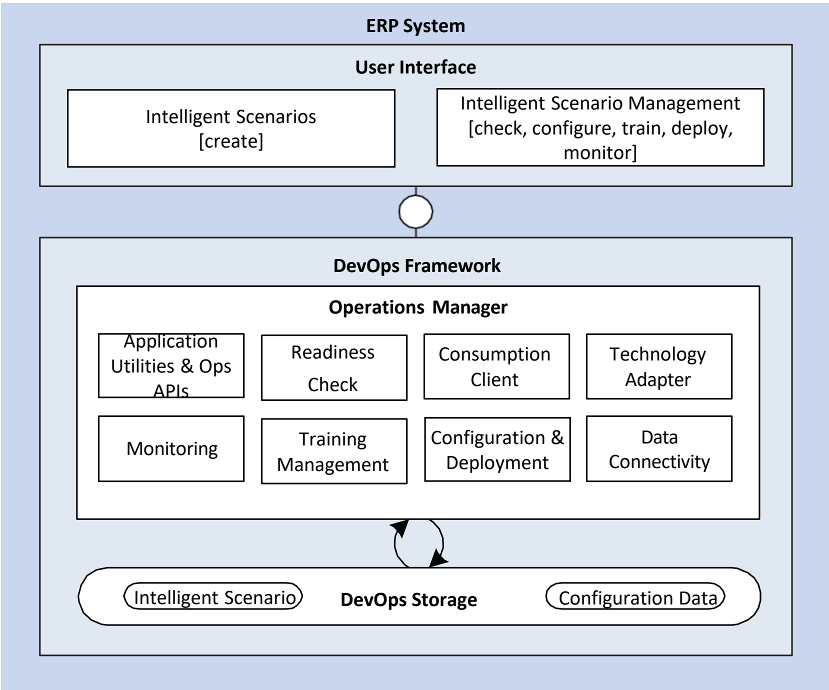

Figure 3. DevOps Framework – Operations Concept

The DevOps framework facilitates conducting prerequisite checks for intelligent scenarios prior to their productive usage in the ERP system, ensuring that high-quality training data of sufficient volume is available to achieve optimal results. Business administrators and operations experts can leverage the framework to assess data quality, volume, and configuration prerequisites, enabling them to make necessary adjustments. The framework maintains detailed records of historical evaluations to ensure traceability and supportability, while defining data structures, exceptions, and interfaces related to prerequisite checks. Additionally, the DevOps framework includes a metamodel comprising interconnected artifacts that verify the requirements of each use case. This encompasses key methods related to metadata, parameters, and execution processes. Through these comprehensive checks and records, the framework ensures robust preparation and validation of intelligent scenarios, paving the way for successful integration and implementation within the ERP system.

Configuration within the DevOps framework involves tailoring systems to meet specific business requirements. The framework is designed to handle configuration parameters efficiently and support multiple models, enhancing prediction accuracy by choosing the most suitable model for a given inference request automatically. It comprises design-time artifacts, such as training data views, and a consumption API, with a model dispatcher that guides inference requests to the appropriate segment model. The management of configuration parameters, including hyperparameters for AI algorithms, is centralized through a universal configuration interface, making it easier to adjust settings to optimize model performance for particular business scenarios. By providing a robust and flexible configuration environment, the DevOps framework enables businesses to fine-tune AI applications to deliver precise and actionable insights tailored to their needs.

The training process within the DevOps framework involves leveraging business data to build AI models. For embedded artificial intelligence, the training data is accessed directly within the local environment, whereas side-by-side artificial intelligence necessitates integrating data from the ERP system to the AI technology platform. The framework employs view operators for managing structured data and object operators for handling unstructured data. Training management accommodates both initial data loads and incremental delta data loads, ensuring flexibility in data processing. The training process is organized around scheduled tasks and asynchronous executions to facilitate efficient data handling and model training. Within this setup, operations experts are tasked with evaluating the outcomes of the training process and making decisions regarding the deployment of trained models. The DevOps framework further enhances the efficiency of this process by supporting the scheduling of training jobs and facilitating notifications to keep administrators informed of progress and results. This structure allows for streamlined management of training activities and effective oversight of model development, ensuring readiness for deployment when optimal performance is achieved.

Deployment within the DevOps framework involves setting up operational server instances to facilitate inferences using trained models. Operations experts are responsible for managing these deployments, ensuring they do not disrupt ongoing business applications. The deployment process encompasses selecting the most suitable models based on their training quality and thoroughly testing deployments before they are activated. The DevOps framework provides robust support for managing deployments, including the capability to roll back active deployments if necessary and recommending the removal of nonactive models to optimize system resources. Once deployed, business applications can seamlessly consume inferences from active deployments, with complexities of the underlying processes abstracted away from users to guarantee smooth and efficient utilization of AI-driven insights. This architecture ensures that new deployments are both effective and minimally invasive, enhancing the agility and responsiveness of business operations while maximizing the benefits derived from intelligent scenarios.

Inferencing integrates AI-generated insights into business operations via inference REST APIs, facilitating seamless interaction between AI models and business applications. Each new trained model results in the creation of a dedicated REST API, enabling specific access to the model's capabilities. The DevOps framework offers utilities that ensure stable and reliable consumption of these APIs, supporting consistent integration into business processes. AI applications interact with the inference REST API, leveraging a generic utility to access the active deployment API efficiently. This utility ensures that the correct API corresponding to the active and most suitable model is used, maintaining the accuracy and relevance of the artificial intelligence insights provided. For embedded artificial intelligence scenarios, a dedicated class is employed to manage AI logic, streamlining the execution and integration of artificial intelligence outcomes directly within the local business environment. This setup guarantees that artificial intelligence capabilities are incorporated smoothly into business workflows, providing actionable insights without disruptive complexity, thus enhancing decision-making and operational processes.

The DevOps framework is designed to continuously oversee trained models and asynchronous training procedures, effectively identifying and addressing operational problems to improve the quality and efficiency of business processes. Its monitoring system not only provides valuable insights into ongoing business activities but also promptly alerts business administrators to any disruptions, enabling automated solutions to potential issues. By integrating both operational and application data, it facilitates comprehensive analysis to proactively tackle anomalies. The workflow includes assessing monitoring data to pinpoint problems, implementing corrective actions, and distributing notifications as required. Additionally, the framework keeps a close watch on the performance of deployed models and tracks inference usage, ensuring consistent and reliable operational functionality.

The six lifecycle stages in our framework (Create, Check, Configure, Train, Deploy, Monitor) can be understood as moments of sociomaterial negotiation (Orlikowski, 2007; Orlikowski & Scott, 2008, 2025) where different configurations of human expertise, organizational authority, and technical agency come together:

- **Create** establishes the initial sociomaterial assemblage, defining which business problems will be addressed through AI and how algorithmic capabilities will be integrated with organizational practices
- **Check** negotiates organizational readiness by assessing whether sufficient data (material) and business process configurations (social-material) exist to support effective AI performance
- **Configure** determines the parameters shaping how AI models learn and perform, distributing agency between human hyperparameter choices and algorithmic learning processes
- **Train** is the moment where algorithms materially encode patterns from organizational data, creating models that embody (and potentially reconfigure) existing practices
- **Deploy** controls when AI models' agency becomes active in business processes, requiring decisions about acceptable performance levels and organizational trust
- **Monitor** creates ongoing practices for assessing how the sociomaterial assemblage is performing and when renegotiation (retraining) is necessary

By providing standardized structures for these lifecycle stages, the DevOps framework attempts to make the dynamic sociomaterial entanglements of AI-enabled ERP more manageable. It does not eliminate the complexities identified in the problematization (Section 2.1) but rather provides organizational mechanisms for continuously navigating them.

6 Evaluation

To demonstrate the effective feasibility of the suggested solution architecture for implementing AI applications into ERP software, a concrete realization of the proposed DevOps framework was developed. A practical realization must be based on an existing ERP platform. We selected SAP ERP (SAP S/4HANA 2023/2024) for this purpose as it is widely used and well known. The evaluation of our DevOps framework follows the logic of action research (Baskerville & Wood-Harper, 1996) integrated into design science (Peffers et al., 2007): Diagnosis - identification of gaps [GAP 1–20] based on AI use cases, Planning - derivation of business and technical requirements [REQ 1–6], Action - design and implementation of DevOps framework, Reflection - evaluation with practitioners and iteration on framework design. We applied the framework to multiple real-world AI use cases within SAP ERP, including Predictive Delivery Delay, Image-Based Buying, and Business Integrity Screening. For each case, we documented the development and operational processes before and after applying the framework, focusing on measures such as implementation effort, deployment time, and model monitoring transparency. This iterative cycle allowed us to refine the framework in practice, demonstrating its feasibility and value in reducing the total cost of ownership and increasing operational reliability. While our evaluation is situated within one ERP platform, the framework's design principles are transferable to other ERP environments facing similar AI integration challenges. Given the extensive functionality of the realized framework incorporating AI into ERP systems, we selectively highlight key components in this section. Figure 4 illustrates some exemplary screenshots of the realized DevOps framework. Figure 4 (A, B) shows the Intelligent Scenarios app for development and the Intelligent Scenario Management app for operations of AI applications with functionality specified in the previous section. The initiation of developing an embedded and side-by-side artificial intelligence use case is depicted in Figure 4 (I), while operations tasks of selecting, training, and activation of AI model are demonstrated in Figure 4 (1, 2, 3).

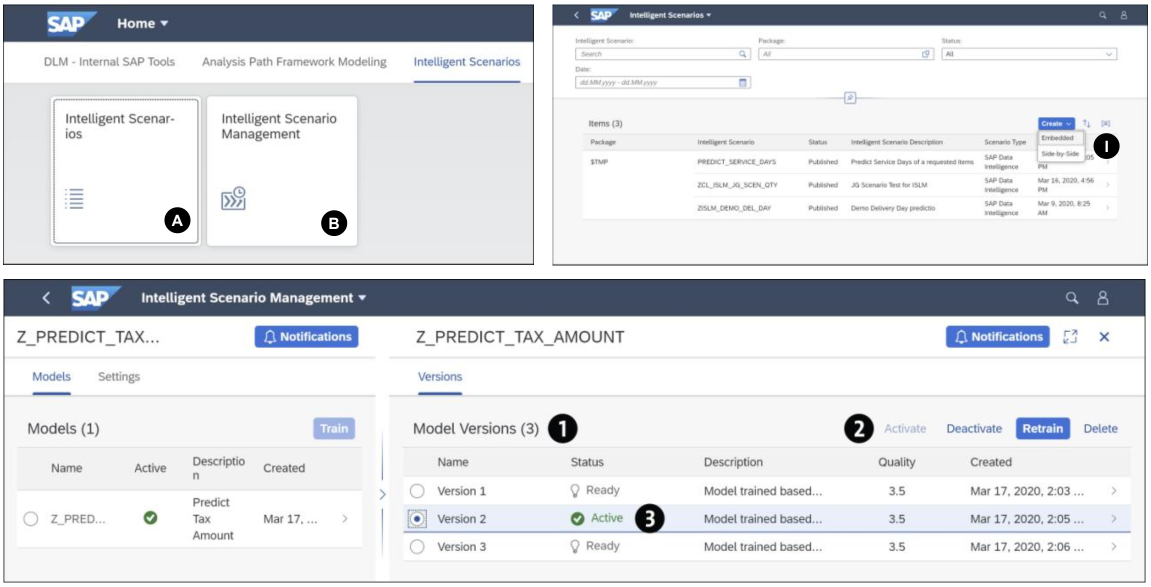

Figure 4. DevOps Framework – Realization

All our concepts from the previous section were effectively implemented into the DevOps framework shown in Figure 4, such that this component has become part of SAP's ERP product (SAP SE, 2025c), enabling customers and partners to also develop and operate their own AI use cases. This proves the transformability of our theoretical concepts into a practical solution. In the next section, we will depict Predictive Delivery Delay, Image-Based Buying, and Business Integrity Screening as exemplary AI use cases that were realized with the DevOps framework. We utilized the DevOps framework for numerous AI use cases (SAP SE, 2025a; SAP SE, 2025b) in the ERP lines of business, including sales and research, sourcing and procurement, inventory and supply chain, and finance. These AI scenarios were effectively and efficiently developed and operated with our suggested DevOps framework such that they have become part of SAP's ERP product (SAP SE, 2025a; SAP SE, 2025b), as has the DevOps framework itself (SAP SE, 2025c). Thus, the functional correctness, real-world applicability, effectiveness, and efficiency of the DevOps framework are demonstrated.

The Predictive Delivery Delay use case presents an innovative approach to enhancing supply chain management through the integration of artificial intelligence into ERP systems. This solution addresses a critical challenge in the lead-to-cash and order-to-fulfill processes by predicting potential delays in sales order deliveries. The architecture of this AI application is designed to seamlessly integrate with existing ERP platforms, employing an embedded AI implementation pattern as shown in Figure 5.

The solution utilizes regression algorithms from the Automated Predictive Library, leveraging the DevOps framework within ERP sales order delivery business processes. The solution analyzes historical sales order data to predict delays and identify root causes, presenting this information through the Predicted Delivery Delay application. This interface allows sales managers to monitor delivery performance in real-time, offering insights into the ratio of delivered items to sales orders upon request. By doing so, it enables proactive management of potential delays, ultimately improving customer satisfaction and loyalty. The prediction model considers multiple factors to generate accurate forecasts. It examines the expected date of delivery creation, contrasting it with actual delivery dates from historical data. Additionally, it analyzes the planned goods issue date against the actual goods movement date for completed deliveries, using the maximum observed delay to predict future delivery processing delays. This sophisticated approach goes beyond traditional available to promise (ATP) solutions by accounting for potential future deviations. The system provides sales representatives with crucial information, including the expected date of delivery creation, requested delivery date, predicted production and processing delays, and the overall anticipated delay for each sales document item. The implementation of this AI service offers numerous benefits to businesses. It reduces the manual effort required to monitor and resolve delivery issues, improves overall delivery performance, and enhances customer satisfaction. The system provides sales representatives with a valuable tool to predict and prevent delays, enabling them to take proactive measures to ensure timely deliveries.

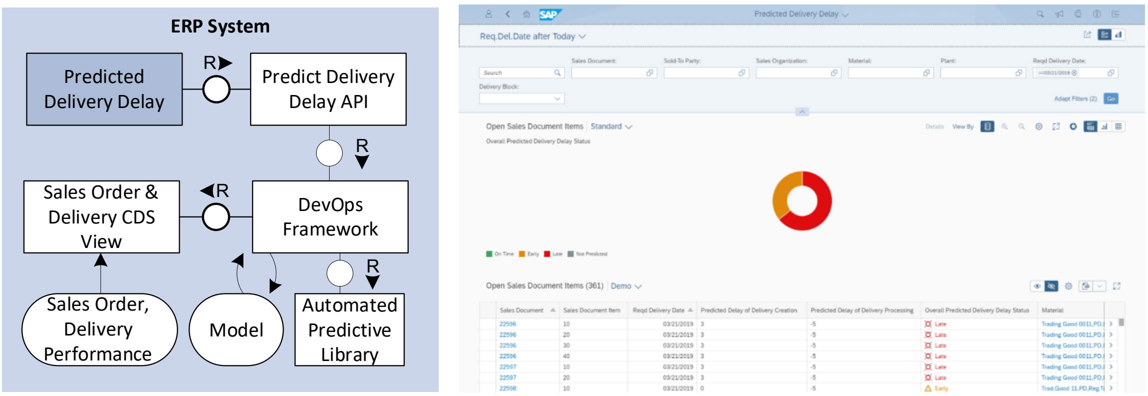

Figure 5. Predict Delivery Delay – Architecture and Application

The Image-Based Buying use case represents an innovative application of artificial intelligence in the source-to-pay and procure-to-receipt processes within ERP systems. This solution leverages advanced image recognition technology to streamline and enhance the purchasing requisition procedure, addressing common challenges in procurement processes. The architecture of this AI application is designed as a side-by-side implementation, utilizing SAP's AI Technology Platform to build the AI service as illustrated in Figure 6. The solution employs algorithms from the TensorFlow library, demonstrating the integration of cutting-edge AI technology with established ERP platforms. The AI application's functionality revolves around a product proposal service that uses historical catalog data for training. When a user uploads an image of a desired product, the AI service analyzes the input to locate matching or similar items within the catalog. This process involves sophisticated image processing techniques, including normalization procedures such as flipping, rotating, and adjusting brightness and colors to enhance similarity detection. The AI service then generates recommendations, providing product identification numbers from customer catalogs that it deems relevant based on the uploaded image. These recommendations are integrated into the ERP process, enabling a seamless user experience within the existing business process.

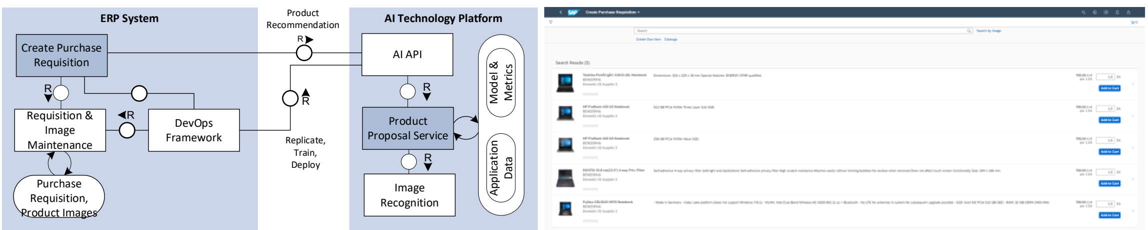

Figure 6. Image-Based Buying – Architecture and Application

One of the key features of this AI application is its ability to conduct cross-catalog searches based on image inputs. This functionality allows users to simply take a photo of a desired item using a mobile device, triggering an automatic search across catalogs and generating a corresponding purchase requisition. The system compares the uploaded image with catalog item images in the cross-catalog search index, identifying matching patterns and displaying results based on similarity scoring. This approach offers numerous benefits to both users and organizations. It significantly improves user efficiency by simplifying the procurement process, allowing buyers to quickly and accurately identify desired items without relying on potentially ambiguous text descriptions. This reduction in ambiguity leads to fewer discrepancies between user requests and actual purchases, minimizing delays and unnecessary costs associated with incorrect material procurement. Moreover, the Image-Based Buying solution enhances compliance and cost control in procurement. Ensuring that purchases align with catalog offerings prevents off-catalog buying and adheres to established procurement policies. This compliance-focused approach results in substantial financial savings, a crucial factor for procurement departments. The implementation of this AI service represents a significant advancement in the integration of intelligent technologies within ERP systems. By leveraging image recognition and machine learning, it transforms the traditional procurement process into a more intuitive, efficient, and cost-effective operation. This use case demonstrates the potential of artificial intelligence to not only streamline existing processes but also to introduce new paradigms in how users interact with ERP systems, paving the way for more innovative and user-friendly solutions in the future of ERP.

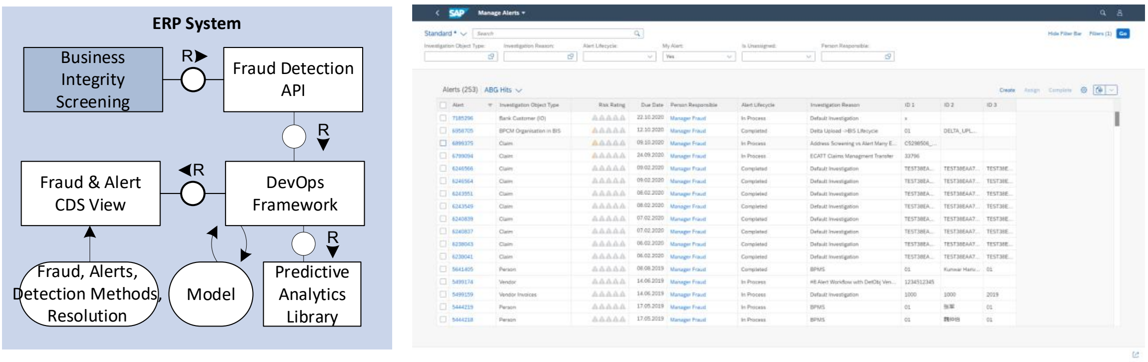

Figure 7. Business Integrity Screening – Architecture and Application

The Business Integrity Screening use case presents a sophisticated application of artificial intelligence within the governance, risk, and compliance (GRC) domain of ERP systems. This solution addresses the critical need for efficient fraud identification and prevention in modern organizations. As depicted in Figure 7, the architecture of this AI application is designed as an embedded solution within the ERP platform, integrating specifically with ERP and other third-party sources. It leverages in-memory database technology to process vast amounts of data in real-time, enabling swift and effective fraud detection and prevention. The solution employs classification and regression algorithms from the Predictive Analytics Library, working in conjunction with the DevOps framework. This combination allows for the implementation of predictive detection techniques that learn from previous investigative decisions, constantly adapting to evolving fraud strategies. The solution addresses a critical pain point for fraud investigators and screening experts who often struggle with managing an ever-increasing number of suspicious cases. By implementing predictive detection methods, the system ranks new cases based on their potential business impact, allowing investigators to focus on high-risk, high-impact cases more efficiently. The AI service in this use case serves multiple functions within the fraud detection process. It automatically detects and ranks attributes within classified data that correlate positively with anomalous cases. These insights are then integrated with existing detection methods to form new strategies. This approach not only enhances the ERP system's ability to detect new suspicious patterns but also significantly reduces the occurrence of false positives. One of the key strengths of this solution is its accessibility to investigators who may not have extensive knowledge of data science. The AI application's threshold parameter for predictive detection methods allows these users to effectively manage their workload and prioritize tasks, leading to substantial improvements in efficiency and profitability. The solution demonstrates the potential of artificial intelligence in transforming traditional governance, risk, and compliance processes. By providing real-time insights and predictive capabilities, it enables organizations to respond more quickly to threats and establish proactive fraud prevention procedures. This not only helps in reducing the risk of fraud but also enhances overall analysis capabilities and visibility into potentially fraudulent activities. Furthermore, the solution's ability to support various fraud scenarios, such as employee theft, corruption, and warranty fraud, makes it a versatile tool for diverse organizational needs. Its integration with ERP ensures that the AI-driven insights are seamlessly incorporated into existing business processes, allowing for immediate action based on the generated information. This use case exemplifies how artificial intelligence can be effectively implemented within ERP systems to address complex challenges in governance, risk, and compliance. By combining advanced predictive techniques with the established business processes, it offers a powerful solution that not only detects and prevents fraud but also continuously evolves to meet the changing landscape of fraud tactics. This adaptive and intelligent approach to governance, risk, and compliance represents a significant advancement in how organizations can protect their core values and enhance overall performance in an increasingly complex business environment.

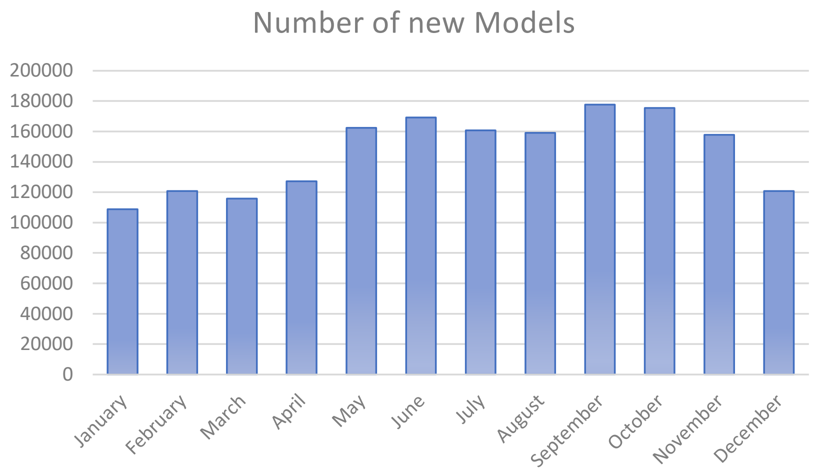

Figure 8. Models managed by DevOps Framework in 2023 for 10 AI Scenarios

We asked SAP to provide us with aggregated usage numbers of new models for 10 AI scenarios in year 2023, which are depicted in Figure 8. The statistics show that for just 10 AI scenarios within one year, 1.8 million new AI models were handled by the DevOps framework. It is evident that such a digitalized solution, compared to the previously "manual" approach, saves significant efforts and increases efficiency and effectiveness. For those 10 AI scenarios, we also performed a quantitative business value assessment by measuring the processing time, costs, and outcome quality before and after the incorporation of AI. As illustrated in Figure 9, the AI features clearly add value as they reduced the processing time by 27%, increased cost savings by 17% and improved the quality of the outcomes by 15%.

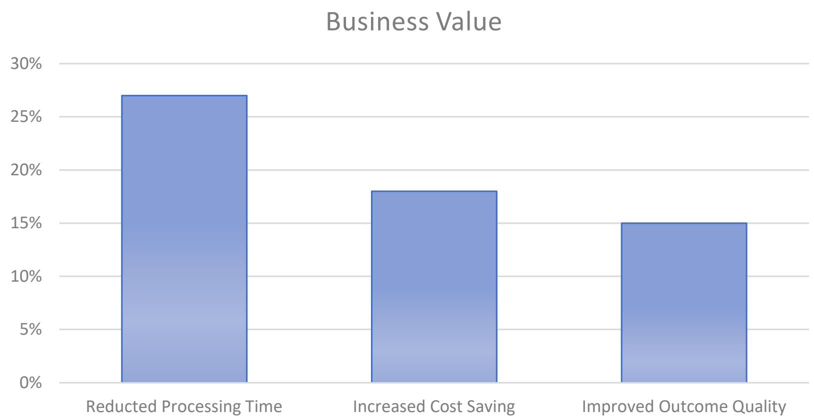

Figure 9. Business Value Assessment for the 10 AI Scenarios

The concrete DevOps framework implementation is evidence of the concepts working correctly and completely in real-world circumstances, but can the concept be applied to arbitrary ERP solutions? All ERP products are developed with individual programming languages. From theoretical computer science, we know that all those programming languages are Turing-complete (Yanofsky, 2022), meaning they have equivalent expressive power. Thus, having proven that the DevOps framework was successfully implemented using the ABAP programming language of SAP, we can conclude that the DevOps framework can also be realized with arbitrary programming languages of other ERP platforms. Consequently, the DevOps framework can be utilized by other ERP products and is therefore generally valid.

The evaluation of our DevOps framework involved both technical assessment of use cases and systematic feedback from domain experts and customer pilots. In addition to documenting reductions in implementation effort and improvements in operational transparency, we collected qualitative feedback to assess the framework's usability and organizational impact. For example, experts from ERP development emphasized the value of standardization across use cases: "Previously, each AI feature required custom scripts; with the DevOps framework, we can reuse lifecycle components across scenarios." Customer pilots confirmed these benefits from an operations perspective: "The automated prerequisite checks gave us confidence to proceed with training, something that used to take days of manual validation." Another customer highlighted the governance implications: "Having an auditable log of model deployments and deactivations helps us meet compliance requirements that were previously unmanageable." The evaluation also revealed areas for refinement. For instance, experts noted that monitoring should include not only technical KPIs (e.g., inference latency) but also AI-related indicators (e.g., forecast accuracy against demand plans). Customers requested clearer guidance on interpreting model performance metrics, underscoring the need for domain-specific explanations in non-technical language. These insights informed adjustments to [REQ 6] Monitor and [REQ 2] Check in later framework iterations. Overall, the evaluation supports the feasibility and value of the proposed DevOps framework in real-world ERP contexts. It reduced the total cost of ownership by streamlining configuration and retraining processes, improved transparency through self-service lifecycle management, and strengthened governance by providing traceability of model histories. Importantly, the stakeholder feedback demonstrates that the framework not only addresses technical gaps but also supports socio-organizational concerns of adoption, trust, and accountability. This aligns with sociomaterial perspectives on AI-in-the-making (Orlikowski & Scott, 2025), where AI's value emerges through the interplay of technical mechanisms and organizational practices.

7 Conclusion

ERP solutions, with their extensive data foundation and wide range of supported business processes, offer great potential for integrating artificial intelligence. AI's ability to derive valuable insights from data significantly optimizes business processes, although incorporating AI into ERP systems poses challenges due to their complexity. The research question we focused on was: *how to systematically develop and operate (DevOps) AI business applications in ERP systems?* The paper aimed to address these challenges by suggesting a DevOps framework for AI applications in ERP systems. For this, we considered the implementation of real-life AI use cases for ERP and iteratively identified gaps, derived business requirements, defined solution concepts, and evolved them to a DevOps framework. In this context, we applied well-known software engineering methods from requirement analysis to software development and deployment. The added value of the DevOps framework is to standardize the development and operations of embedded and side-by-side artificial intelligence patterns. Lifecycle management tasks like training models, analyzing model KPIs, and deploying models are handled by the DevOps framework uniformly. The DevOps framework consists of two applications: one for the implementation of intelligent business applications by developers and one for operating them by administrators. The practical use and evaluation with SAP ERP, along with its broad applicability to other ERP systems due to its Turing-completeness, confirms the framework's viability and efficiency. Successful application across various scenarios highlights the real-world advantages of embedding AI into ERP systems. Our research developed a comprehensive, flexible, and validated framework for integrating AI into ERP software. This framework not only addresses current shortcomings but also lays the groundwork for future developments in intelligent enterprise solutions. As AI technology progresses, the principles and methods detailed in this paper will act as a foundation for ongoing innovation, ensuring that ERP systems remain at the leading edge of digital transformation.

By explicitly grounding our work in the six stages of the design science research process, we ensure that the proposed framework is both academically rigorous and practically relevant. Beginning with a systematic analysis of existing AI use cases in different ERP modules, we identified concrete challenges that go beyond those typically discussed in prior ERP and IS literature. These challenges formed the definition of our six requirements, which in turn guided the design and iterative refinement of the framework. The demonstration and evaluation in collaboration with ERP customers confirm its applicability, particularly in contexts where organizations need to extend vendor-delivered AI scenarios or develop highly specific AI use cases not covered in the ERP core. By situating our work within the design science research methodology, we contribute not only a novel extension of DevOps tailored to AI-enabled

ERP systems but also a methodological blueprint for future research at the intersection of enterprise software and emerging technologies. Our study makes several theoretical contributions:

1. **Extending Sociomateriality to Continuous Algorithmic Evolution:** While sociomaterial studies have examined how technologies become embedded in organizational practices, less attention has been paid to technologies that continuously evolve their performance through learning. Our framework reveals how organizations must develop practices for managing ongoing sociomaterial reconfiguration rather than one-time implementation.
2. **Lifecycle Perspective on Enterprise AI:** Whereas much AI research focuses on model development or deployment in isolation, our framework demonstrates that managing AI in enterprise contexts requires coordinated practices across the entire lifecycle. Each stage represents different sociomaterial configurations requiring different forms of expertise, authority, and technical capability.
3. **Design Artifacts as Mediators of Sociomaterial Practice:** The DevOps framework itself becomes part of the sociomaterial assemblage, shaping how organizations can engage with AI capabilities. By standardizing lifecycle management, it materially constrains some possibilities while enabling others, influencing how AI agency can be enacted within ERP environments.

These theoretical contributions complement the practical design science contribution of the framework itself, positioning this research at the intersection of sociomateriality theory, ERP implementation research, and AI operations practices - an intersection that prior literature has not systematically addressed. Our study is not without limitations. The evaluation is grounded in SAP use cases, which, while representative due to SAP's market leadership, may not capture the full diversity of ERP implementations. Future research could validate the proposed framework across different ERP vendors, organizational contexts, and longitudinal deployments. Further studies could also investigate the organizational and cultural implications of AI-ERP integration, such as shifts in user roles, trust in AI-driven decision-making, and the balance between automation and human judgment.

References

Aktürk, C. (2021). Artificial intelligence in enterprise resource planning systems: A bibliometric study. *Journal of International Logistics and Trade, 19*(2), 69-82.

Arachchi, S. M., Chong, S. C., & Lakshanthi, A. D. S. M. (2015). Literature-based review: Risks in ERP systems including Asian countries. *European Journal of Computer Science and Information Technology, 3*(1), 1–14.

Bandla, J. (2023). *No enterprise resource planning (ERP) left behind: A comprehensive analysis.* Deloitte. https://www2.deloitte.com/content/dam/Deloitte/us/Documents/deloitte-analytics/us-ai-institute-ai-in-erp.pdf

Baskerville, R. L., & Wood-Harper, A. T. (1996). A critical perspective on action research as a method for information systems research. *Journal of Information Technology, 11*(3), 235–246.

Batra, G., & Santhanam, N. (2017). *Artificial intelligence: The time to act is now.* McKinsey & Company White Paper. https://www.mckinsey.com/industries/industrials-and-electronics/our-insights/artificial-intelligence-the-time-to-act-is-now

Bergdahl, J. (2022). *The AI revolution: A study on the present and future application and value of AI in the context of ERP systems.* https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1220408&dswid=-2417

Broy, M., & Kuhrmann, M. (2021). *Einführung in die Softwaretechnik.* Springer.

Cheng, H. (2022). Research on the application of artificial intelligence machine learning technology in enterprise information systems. In *Proceedings of the 3rd International Conference on Artificial Intelligence and Advanced Manufacture* (pp. 2446–2450). Association for Computing Machinery.

Costin, A., Tanasie, A., & Cojocaru, D. (2020). Enterprise resource planning for robotic process automation in big companies: A case study. In *2020 24th International Conference on System Theory, Control and Computing (ICSTCC).*

Dechow, N., & Mouritsen, J. (2005). Enterprise resource planning systems, management control and the quest for integration. *Accounting, Organizations and Society, 30*(7–8), 691–733.

Dick, J., Hull, E., & Jackson, K. (2017). *Requirements engineering.* Springer.

Dooley, J., & Kazakova, V. (2024). *Software development, design, and coding.* Springer.

El Hairech, O., & Lyhyaoui, A. (2022). The new generation of ERP in the era of artificial intelligence and industry 4.0. In *Advanced Intelligent Systems for Sustainable Development (AI2SD'2020)* (pp. 1086–1094).

Fox, M. (1986). Industrial applications of artificial intelligence. *Robotics, 2*(4), 301-311.

Godbole, M. V. (2023). Revolutionizing enterprise resource planning (ERP) systems through artificial intelligence. *International Numeric Journal of Machine Learning and Robots, 7*(7).

Gregor, S. (2006). The nature of theory in information systems. *MIS Quarterly, 30*(3), 611–642.

Hau, E., & Aparício, M. (2008). Software internationalization and localization in web-based ERP. In *Proceedings of the 26th Annual ACM International Conference on Design of Communication.*

Heinzelmann, R. (2017). Accounting logics as a challenge for ERP system implementation: A field study of SAP. *Journal of Accounting & Organizational Change, 13*(2), 162–187.

Ivanović, T., & Marić, M. (2021). Application of modern enterprise resource planning (ERP) systems in the era of digital transformation. *Strategic Management, 26*(4), 28–36.

Jawad, Z., & Balázs, V. (2024). Machine learning-driven optimization of enterprise resource planning (ERP) systems: A comprehensive review. *Beni-Suef University Journal of Basic and Applied Sciences, 13*(1), Article 4.

Kaniyar, S., Srivastava, K., & Tisnovsky, R. (2019). *Driving impact at scale from automation and AI.* McKinsey White Paper. https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20digital/our%20in sights/driving%20impact%20at%20scale%20from%20automation%20and%20ai/driving-impact-at-scale-from-automation-and-ai.pdf

Karamitsos, I., Albarhami, I., & Apostolopoulos, C. (2020). Applying DevOps practices of continuous automation for machine learning. *Information, 11*(7), Article 363.

Kitchenham, B., & Charters, S. M. (2007). *Guidelines for performing systematic literature reviews in software engineering*. EBSE Technical Report EBSE-2007-01, Version 2.3. https://legacyfileshare.elsevier.com/promis_misc/525444systematicreviewsguide.pdf

Kreuzer, T., Papapetrou, P., & Zdravkovic, J. (2024). Artificial intelligence in digital twins—A systematic literature review. *Data & Knowledge Engineering, 151,* 102304.

Kumar, S. (2018). Smarter artificial intelligence with deep learning. *SSRG International Journal of Computer Science and Engineering, 5*(6), 10–16.

Lin, R. (2021). Analysis on the application of artificial intelligence in the global value chain upgrade of manufacturing enterprises. In *2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture* (pp. 1400–1404).

Lwakatare, L. E., Crnkovic, I., & Bosch, J. (2020). DevOps for AI—Challenges in development of AI-enabled applications. In *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)* (pp. 1–6).

Mediavilla, M., Dietrich, F., & Palm, D. (2021). Review and analysis of artificial intelligence methods for demand forecasting in supply chain management. *Procedia CIRP, 99,* 604–609.

Mhaskey, S. (2024). Integration of artificial intelligence (AI) in enterprise resource planning (ERP) systems: Opportunities, challenges, and implications. *International Journal of Computer Engineering in Research Trends, 11*(12), 1–9.

Muthusamy, V., Slominski, A., & Ishakian, V. (2018). Towards enterprise-ready AI deployments: Minimizing the risk of consuming AI models in business applications. In *2018 First International Conference on Artificial Intelligence for Industries.*

Omar, H., & Rachid, S. (2023). Challenges and opportunities of integrating AI with IFRS in accounting systems: Insights from Morocco. In *Proceedings of the 6th International Conference on Networking, Intelligent Systems.*

Orlikowski, W. J. (2007). Sociomaterial practices: Exploring technology at work. *Organization Studies, 28*(9), 1435-1448.

Orlikowski, W. J., & Scott, S. V. (2008). Sociomateriality: Challenging the separation of technology, work and organization. *The Academy of Management Annals, 2*(1), 433–474.

Orlikowski, W. J., & Scott, S. V. (2025). Exploring AI-in-the-making: Sociomaterial genealogies of AI performativity. *Information and Organization, 35*(1), 100558.

Patel, J. K. (2021). Critical success factors for implementation of enterprise resource planning software. *SSRG International Journal of Computer Science and Engineering, 8*(5), 1–5.

Paul, S. K., Riaz, S., & Das, S. (2022). A conceptual architecture for AI in supply chain risk management. In *TENCON 2022 – IEEE Region 10 Conference.*

Peffers, K., Tuunanen, T., & Rothenberger, M. A. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77.

Pokala, P. (2024). Artificial intelligence in enterprise resource planning: A systematic review of innovations, applications, and future directions. *International Journal of Research in Computer Applications and Information Technology, 7*(2), 1276–1289.

Rashid, M. A., Hossain, L., & Patrick, J. D. (2002). The evolution of ERP systems: A historical perspective. In *Enterprise resource planning: Solutions and management* (pp. 35–50). IGI Global.

Ribeiro, J., Lima, R., Eckhardt, T., & Paiva, S. (2021). Robotic process automation and artificial intelligence in industry 4.0: A literature review. *Procedia Computer Science, 181,* 51–58.

Rojek, I., & Jagodziński, M. (2022). Hybrid artificial intelligence system in constraint-based scheduling of integrated manufacturing ERP systems. In *Hybrid Artificial Intelligent Systems* (pp. 229–240).

Sadrzadehrafiei, S., Chofreh, A., & Hosseini, N. (2013). The benefits of enterprise resource planning (ERP) system implementation in dry food packaging industry. *Procedia Technology, 9,* 518–526.

SAP SE. (2025a). *Product roadmap.* https://roadmaps.sap.com/

SAP SE. (2025b). *Discovery center: SAP business AI features.* https://discovery-center.cloud.sap/ai-catalog/

SAP SE. (2025c). *Product documentation.* https://help.sap.com/

Srinivasan, D., Cheu, R. L., & Tan, C. W. (2004). Development of an improved ERP system using GPS and AI techniques. In *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems.*

Sustrova, T. (2016). A suitable artificial intelligence model for inventory level optimization. *Trendy Ekonomiky a Managementu, 10*(25), 45–56.

Tallón-Ballesteros, A. J. (2020). The design of ERP intelligent sales management system. In *Fuzzy systems and data mining VI.* Sage.

Themistocleous, M., Irani, Z., & O'Keefe, R. M. (2001). ERP and application integration: Exploratory survey. *Business Process Management Journal, 7*(3), 195–204.

Wang, A., Guo, S., & Li, R. (2022). Artificial intelligence technology enables the development of management accounting: The generation of intelligent accounting. In *3rd Asia Service Sciences and Software Engineering Conference.*

Yanofsky, N. S. (2022). *Theoretical computer science for the working category theorist.* Cambridge University Press.

Zdravković, M., Panetto, H., & Weichhart, G. (2024). AI-enabled enterprise information systems for manufacturing. *Enterprise Information Systems.*

Zheng, N., Zhao, G., Chen, A., Wang, L., Liu, Y., & Huang, H. (2022). The adoption of enterprise resource planning and business intelligence systems in small and medium enterprises: A conceptual framework. *Mathematical Problems in Engineering, 2022(2), 1-15.*

About the Authors

Siar Sarferaz is a chief software architect for the enterprise resource planning (ERP) solution of SAP, working in the research and development department of the headquarter in Walldorf, Germany. In this role he drives the digital transformation by defining the solution architecture for the product and by providing concepts scaling for mission critical business processes. He is the lead architect for artificial intelligence implementations for SAP's ERP product and is responsible for all conceptions on how to add intelligence to business processes. In the context of ERP software, he owns 30+ patents and published numerous books. Siar had studied computer science and philosophy and holds a Ph.D. in computer science. He began his career as a method researcher at Siemens, before moving to SAP, where he has now worked for more than 20 years.