Hochschule Neu-Ulm
University of Applied Sciences

Master Thesis

in the Master Program

Artificial Intelligence and Data Analytics

at University of Applied Sciences Neu-Ulm

AI-Driven Process Mining for Predictive and Explainable Error Management in Financial ERP Systems: A Case Study in Microsoft Dynamics 365

1st examiner: Prof. Dr. Jörg-Oliver Vogt

2nd examiner: Prof. Dr. Stefan Faußer

Author: Konstantin Kataman (Enrolment number: 265295)

Topic received: 24.11.2025

Date of submission: 28.02.2026

Abstract

Enterprise Resource Planning (ERP) systems such as Microsoft Dynamics 365 (D365 FO) support highly structured financial processes governed by accounting rules, configuration logic, and internal control requirements. While process mining and predictive process monitoring have advanced substantially, their application in financial ERP environments remains constrained by reconstructed event logs, configuration-dependent behaviour, and strict explainability requirements. This thesis investigates how forward-looking and explainable process analytics can support earlier identification and structured interpretation of review-relevant situations in ERP-based financial processes. Using the accounts receivable collection process in D365 FO as a case study, the research follows a Design Science Research approach to develop and evaluate a decision-support artifact.

The artifact integrates three components:

- structured consolidation of ERP lifecycle data using Microsoft Power Automate
- reconstruction of invoice-based event logs and descriptive process mining analysis
- a Copilot-based AI assistant for explanation and structured interaction over analytical outputs.

Process mining reconstructs invoice lifecycle paths from ERP event data, providing a structured representation of case execution. The AI assistant explains these reconstructed states in finance-domain language based strictly on the consolidated dataset, without generating independent analytical conclusions. Evaluation was conducted in a pre-production environment across multiple legal entities using scenario-based walkthroughs. Results indicate that consolidated lifecycle visibility and structured AI-assisted interaction reduce information search and cross-referencing effort compared to baseline ERP navigation. Explanation quality depends on event completeness, clarity of event definitions, and stability of process reconstruction. Observed limitations arise from ERP data representation constraints and variability in AI-generated formulations.

Table of Contents

1 Introduction

1.1 ERP Systems and Financial Process Automation

Enterprise Resource Planning (ERP) systems play a central role in managing and automating financial processes by integrating transactional data, business rules, and control mechanisms. The literature characterises ERP systems as the digital backbone of organisations, enabling standardised execution of end-to-end business processes and providing a unified source of financial information across organisational units (Yang et al., 2025). ERP systems support both operational transaction processing and higher-level financial control and reporting activities (Granlund & Malmi, 2002). ERP platforms such as Microsoft Dynamics 365 Finance & Operations (D365 FO) integrate core accounting, subledger, and control-relevant functionalities within a unified transactional environment (Microsoft, 2025). These systems embed accounting logic, validation rules, and configuration-dependent processing structures that govern how financial events are recorded and validated. By embedding accounting logic and validation rules into transactional processing, ERP systems constrain execution behaviour and promote consistency of financial postings across large transaction volumes. Empirical research examining ERP usage indicates that adoption within finance functions is associated with changes in information and coordination characteristics, including improved information timeliness, stronger process integration, and increased visibility across financial workflows (Nicolaou, 2004). These effects do not necessarily imply immediate performance gains; rather, they reflect how ERP systems restructure the informational environment and execution logic of accounting and finance activities. Within operational finance processes, ERP systems introduce standardised processing structures that influence how transactions are recorded, validated, and traced across activities such as invoicing, accounts payable, accounts receivable, and payment handling. These mechanisms promote consistency and process transparency by relying on shared data definitions and rule-based transaction logic. From a control and compliance perspective, ERP systems play a critical role in supporting internal controls and reporting. Centralised data models and standardised execution enable traceability of financial transactions from source documents to ledger postings, which is essential for statutory reporting, tax audits, and internal control assessments (Jans, Alles, & Vasarhelyi, 2014). ERP systems also serve as foundations for advanced financial analytics. The availability of detailed, time-stamped event data generated during execution enables systematic analysis of workflows, including performance measurement, deviation detection, and control evaluation using approaches such as process mining (Jans et al., 2014; van der Aalst, 2016).

1.2 Errors in ERP-Based Financial Processes

Despite high levels of automation and standardisation, financial processes executed within ERP systems remain susceptible to errors, exceptions, and process deviations. ERP implementation can reduce some classes of manual processing errors, but complex configurations still generate exceptions and control-relevant deviations. Errors may arise not only from manual input but also from configuration inconsistencies, master-data conditions, or deviations in execution behaviour. In financial ERP systems such as D365 FO, configuration defines posting logic, validation rules, and control behaviour. Consequently, recurring errors may reflect configuration-dependent processing conditions rather than isolated user actions. Typical error types include unmatched or partially matched transactions (e.g., invoices, payments, or bank statements), incorrect postings (e.g., account or tax determination issues), delayed settlements, and resulting inconsistencies in financial reporting. Such exceptions are particularly prevalent in high-volume transaction cycles such as purchase-to-pay, order-to-cash, and period-end closing, where small configuration issues or data inconsistencies can propagate across multiple postings (Granlund & Malmi, 2002). These errors interrupt automated processing and require manual investigation and correction.

The handling of ERP process errors has significant operational implications. Manual exception handling increases workload for accounting, controlling, and treasury teams, extends closing cycles, and reduces the efficiency gains expected from ERP implementation. Late-detected or unresolved errors reduce data reliability and may impair the quality of management reporting, limiting the usefulness of ERP systems for decision support (Nicolaou, 2004). In multi-entity environments with multiple currencies and regulatory regimes, error resolution becomes especially resource-intensive. ERP-related financial errors also represent a control and compliance risk. In regulated contexts such as statutory accounting, tax reporting, and financial auditing, erroneous postings or missing reconciliations may result in audit findings, control deficiencies, or regulatory non-compliance. Auditing research indicates that many material weaknesses and audit observations are linked not to fraud but to process breakdowns and control failures within ERP-supported workflows (Jans et al., 2014). Organisations are required to demonstrate not only correct financial results but also robust and traceable underlying processes. The literature emphasises the need for systematic detection and proactive management of errors in ERP systems. Traditional controls and post-hoc reconciliations are often insufficient to cope with growing transaction volumes and process complexity. Data-driven approaches such as process analytics and process mining can identify deviations, bottlenecks, and anomalous behaviour directly from event logs, shifting the focus from reactive correction toward earlier identification and prevention of issues (Jans et al., 2014).

1.3 Process Mining and Limitations

Process mining has emerged as a powerful analytical approach for understanding, monitoring, and improving business processes based on event data extracted from information systems such as ERP platforms. By reconstructing process executions from event logs, process mining enables transparency into how processes are performed in practice rather than how they are designed or documented (van der Aalst, 2016). In ERP-based financial processes, process mining has been widely applied to purchase-to-pay, order-to-cash, and audit-relevant workflows, supporting the identification of bottlenecks, deviations, and control weaknesses (Jans et al., 2014). Traditional process mining approaches are commonly categorised into process discovery, conformance checking, and performance analysis. Process discovery derives process models from event logs, conformance checking compares observed behaviour with predefined models or rules, and performance analysis evaluates temporal aspects such as throughput times and waiting periods (van der Aalst, 2016). These techniques primarily provide retrospective insights into execution and compliance. A key limitation is that many detected issues become visible only after financial impact has already occurred. Incorrect postings, delayed settlements, or reconciliation mismatches are often identified through ex-post analysis, and affected transactions may require manual correction, reversal, and reposting. Prior research highlights that late detection increases operational workload, prolongs closing cycles, and elevates control risk, particularly in high-volume environments (Jans et al., 2014). This limitation motivates analytical approaches that extend beyond descriptive and diagnostic insights toward more proactive support.

1.4 Forward-Looking Process Monitoring and Research Gap

Forward-looking process monitoring extends process mining by enabling the assessment of ongoing process instances based on partially observed event data and execution characteristics. Rather than restricting analysis to completed cases, forward-looking approaches aim to identify emerging deviations, risks, or undesired states during process execution (Maggi et al., 2014; Teinemaa et al., 2019). A substantial body of research implements forward-looking monitoring using supervised machine learning techniques. Predictive models are trained on historical event logs to estimate future outcomes such as delays, violations, or exceptional process states (Tax et al., 2017). Under controlled conditions and benchmark datasets, such models can capture temporal dependencies and recurring behavioural regularities. However, transferring these approaches to financial ERP environments is methodologically and practically complex. Much of the predictive process monitoring literature relies on benchmark datasets originating from domains such as customer service or IT service management, where event semantics, process flexibility, and governance constraints differ substantially from ERP-based finance processes (Teinemaa et al., 2019). Financial ERP processes are governed by accounting rules, validation mechanisms, and configuration-dependent execution logic, which systematically shape observable behaviour and feasible process outcomes. In ERP systems such as D365 FO, process-relevant information is not stored in a native process-centric representation. Instead, transactional records are distributed across relational tables and modules, requiring event logs to be reconstructed through extraction and transformation procedures (van der Aalst, 2016). Many deviations in ERP systems result from configuration settings, master data conditions, or control parameters rather than from random execution behaviour. An additional research gap concerns the interpretation and operational use of forward-looking analytical outputs in regulated financial environments. Even when analytical mechanisms identify deviations or risk patterns, finance professionals must be able to relate these outputs to accounting logic, configuration conditions, and control implications. Prior research on explainable AI emphasises that interpretability and transparency are critical prerequisites for analytical system adoption in high-stakes domains (Doshi-Velez & Kim, 2017; Rudin, 2019).

This study addresses these challenges by adopting a forward-looking monitoring perspective based on reconstructed invoice lifecycle and collection-status information extracted from D365 FO. Instead of supervised predictive model training, the proposed artifact consolidates process-relevant ERP extracts into a single structured dataset (‘super report’). In this study, predictive refers to forward-looking assessment of execution preconditions and emerging lifecycle states, not to supervised probabilistic model training. The consolidated dataset serves as input for process mining analysis and for a Copilot-based explanation layer that answers user queries based strictly on the provided fields. This design avoids probabilistic outcome prediction and restricts the AI component to structured explanation of observable ERP conditions, without executing actions in the ERP system. This distinction is particularly important in financial contexts, where governance and auditability requirements demand controlled and verifiable analytical behaviour.

1.5 Challenges of Financial ERP Systems: The Case of D365 FO

Financial ERP systems such as D365 FO pose challenges for forward-looking process analytics that differ from many benchmark settings. D365 FO supports tightly integrated finance-related modules, including general ledger, accounts payable and receivable, cash and bank management, fixed assets, tax, and intercompany accounting. Financial processes in such systems are cross-functional, highly structured, and governed by complex business rules and configuration settings, increasing both deviation risk and the difficulty of identifying root causes (Jans, Alles, & Vasarhelyi, 2014). A defining characteristic of financial ERP systems is the emphasis on compliance and regulatory correctness. Transactions are subject to accounting standards, tax regulations, and internal control requirements. Research on explainable AI highlights that black-box models can be problematic in such environments due to limited traceability and user trust, creating practical challenges for adoption (Rudin, 2019; Jans et al., 2014). D365 FO relies on a highly normalised data model in which event-relevant information is distributed across journals, vouchers, subledger tables, and master data entities. While this supports consistency, it increases the complexity of event-log extraction and interpretation. Process mining literature emphasises that transforming ERP data into event logs requires careful case definition, activity abstraction, and handling of concurrency and variant complexity, challenges that are amplified in large-scale ERP finance systems (van der Aalst, 2016; Berti et al., 2023). Organisational and governance factors further constrain analytics adoption. Finance users must interpret issues and justify accounting decisions, analytical outputs must be actionable and comprehensible. Prior work on XAI emphasises that transparency and local explanations are prerequisites for user acceptance in high-stakes domains such as finance (Guidotti et al., 2018; Doshi-Velez & Kim, 2017).

To address these challenges, this thesis investigates how process mining and an AI-assisted interpretation mechanism can support earlier identification and explanation of control-relevant situations in D365 FO. The study adopts a Design Science Research approach suitable for developing and evaluating artifacts that address practical problems while contributing to scientific knowledge (Hevner et al., 2004; Peffers et al., 2007). The artifact combines (1) consolidation of process-relevant D365 FO extracts into a structured dataset, (2) process mining analysis based on the dataset, and (3) AI-assisted interpretation for end users.

1.6 Research Objective and Contribution

This study addresses the following research questions:

RQ1: How can forward-looking and configuration-aware process analytics, combined with AI-assisted explanation, support investigation and interpretation of control-relevant situations in financial ERP processes?
RQ2: How can such an AI-assisted decision-support artifact be implemented in a D365 FO environment while preserving governance requirements, traceability, and human accountability?
D365 FO is used as a representative enterprise platform. In this study, AI components operate as interpretation and explanation mechanisms applied to a consolidated structured ERP dataset. The consolidated D365 FO dataset forms the basis for process mining analysis and AI-assisted explanations. AI mechanisms translate structured D365 FO fields and consolidated process information into domain-relevant explanations while preserving traceability of the underlying data. Recent research suggests that many AI initiatives fail not because of algorithmic limitations but due to misalignment with organisational realities. Adoption depends on compatibility with existing processes, controls, and technology landscapes; without such alignment, technically sound solutions may remain impractical or unused (Bouquet, Wright, & Nolan, 2026). This requirement is particularly important in financial contexts, where analytical solutions must coexist with established governance structures. In the proposed design, D365 FO remains the authoritative source of transactional data. Analytical computations are not executed within D365 FO. Instead, process-relevant data is extracted and consolidated using Microsoft Power Automate to produce a unified dataset representing process execution states. Forward-looking support is implemented through earlier and structured visibility of partially observed process states within a consolidated dataset, combined with process mining views and AI-assisted explanation to support investigative reasoning. Explainability and user interaction are supported through a Copilot-based interface that operates on the consolidated dataset and provides structured explanations based on invoice, customer, and collection attributes. This design reflects human-in-the-loop principles, where analytical systems support professional decision-making rather than automate it (Amershi et al., 2019; Shneiderman, 2020). The separation between D365 FO, automation, analytical processing, and AI-assisted interpretation also follows common enterprise analytics architecture principles, preserving system stability and supporting analytical flexibility (van der Aalst, 2016). The study provides contributions at the design, interpretability, and empirical observation levels. At the design level, it proposes an analytics framework for forward-looking process analytics within a configuration-driven D365 FO environment, incorporating event-log reconstruction and structured status-based

monitoring. At the interpretability and usability level, it introduces an operational approach for translating control-relevant signals into bounded, human-readable explanations aligned with finance users' decision contexts. At the empirical level, the study reports observations from applying the artifact to a real-world D365 FO finance process, highlighting applicability constraints, interpretation challenges, and practical limitations.

1.7 Structure of the Thesis

The remainder of this thesis is organised as follows. Chapter 2 reviews the theoretical foundations, including process mining, forward-looking process monitoring, artificial intelligence in business process management, and explainable AI. Chapter 3 presents the research methodology and outlines the Design Science Research approach adopted for artifact design and evaluation. Chapter 4 describes the data collection procedure and the construction of event logs derived from D365 FO, with particular emphasis on case definition, event abstraction, and configuration preparation. Chapter 5 introduces the monitoring and interpretation approach developed in this study. It describes how structured, status-based fields derived from D365 FO are prepared and used to support consistent case interpretation. The chapter also outlines the evaluation strategy, focusing on interpretability, reproducibility of the extracted status information, and operational relevance, and discusses threats to validity arising from event abstraction, configuration dependencies, and case-definition assumptions Chapter 6 presents the prototype implementation and interaction design of the analytical artifact. It explains how forward-looking analytical outputs and configuration-aware interpretations are implemented within a user-oriented interface. The chapter describes the integration of structured process attributes, consolidated D365 FO data ("super report"), and the Copilot-based explanation layer, illustrating how analytical signals are translated into structured, human-readable explanations that support finance users in interpreting control-relevant situations without introducing autonomous decision logic. Chapter 7 critically interprets observational findings, artifact behaviour, and methodological constraints affecting analytical validity, and practical applicability within credit and collection processes. The chapter also reflects on implications for forward-looking and explainable process mining solutions. Chapter 8 concludes the thesis by summarising the research objective, methodological approach, and key findings. It outlines directions for future research, particularly in relation to event-log design, analytical strategies, and explainability mechanisms in ERP analytics.

2 Literature Review

2.1 Business Process Management

Business Process Management (BPM) includes the analysis, design, execution, monitoring, and continuous improvement of organisational processes. Traditional BPM approaches primarily rely on manually constructed process models, expert assumptions, and predefined control mechanisms. While such models provide a conceptual representation of intended process flows, they frequently fail to capture how processes are executed in real operational environments. This gap between designed and actual process behaviour has been widely discussed in the BPM and information systems literature (van der Aalst, 2016).

As organisational activities increasingly rely on digital platforms and enterprise systems, BPM has progressively shifted toward data-driven perspectives. Instead of analysing processes solely through interviews or workshops, organisations can leverage system-recorded data to observe factual execution patterns. Within this development, process mining has emerged as a central analytical approach. Process mining provides a data-driven perspective on process execution by extracting and analysing event data recorded by information systems. Event logs typically contain case identifiers, activity names, and timestamps, which together enable the reconstruction of process flows based on empirical evidence rather than assumptions (van der Aalst, 2016). The relevance of process mining is particularly evident in complex and automated environments such as ERP systems. ERP platforms generate structured transactional data across interconnected modules, such as account receivable and project modules. Although ERP systems are not inherently process-centric logging systems, their transactional data structures provide a basis for reconstructing execution behaviour (Jans, Alles, & Vasarhelyi, 2014). The principal techniques of process mining are commonly categorised into process discovery, conformance checking, and performance analysis. Process discovery aims to derive process models directly from event logs without relying on predefined reference models. Conformance checking evaluates the alignment between observed process behaviour and expected models or rules. Performance analysis focuses on temporal and quantitative characteristics, including throughput times, waiting periods, and bottlenecks (van der Aalst, 2016). Together, these techniques enable organisations to obtain an evidence-based understanding of process execution, supporting both diagnostic analysis and continuous process improvement.

2.2 Process Mining

Process mining techniques fundamentally rely on event logs, where each event represents the occurrence of an activity within a specific process instance (case). Events associated with the same case form an ordered sequence commonly referred to as a trace. Event logs may contain additional attributes, including timestamps, resources, and data elements. These attributes may be associated either with individual events (event attributes) or with entire process instances (case attributes) (Di Francescomarino et al., 2018). Process mining has been extensively examined in ERP environments, where standardised transaction processing generates large volumes of structured, time-stamped data across interconnected business processes. In many enterprise systems, event logs are not available in a native process-centric form and must be reconstructed from transactional records, database tables, and system metadata. This reconstruction requires explicit decisions regarding case definition, activity abstraction, and event correlation. In the financial domain, prior research highlights the application of process mining to audit-relevant workflows, including purchase-to-pay, order-to-cash, and financial closing processes. These studies indicate that process mining can support analytical and monitoring tasks by identifying deviations from expected procedures, irregular execution patterns, and anomalous transaction behaviour within ERP data (Jans et al., 2014). Financial ERP processes nevertheless introduce specific analytical challenges. They are characterised by high transaction volumes, configuration-dependent execution logic, and regulatory constraints affecting data interpretation. Consequently, researchers must carefully define case notions, interpret activity semantics, and ensure that reconstructed event data accurately reflects process execution behaviour (van der Aalst, 2016). Recent advances, such as object-centric event logs (OCEL), address some of these challenges by allowing multiple interacting business objects to be represented within a unified modelling framework (Berti et al., 2023). Despite these methodological developments, traditional process mining techniques predominantly analyse completed process instances. The analysis focuses on retrospective interpretation of process behaviour and deviations observed after execution. While retrospective insights are valuable for compliance analysis and process diagnostics, they provide limited support for early operational intervention. This limitation has motivated research into forward-looking monitoring approaches that analyse partially observed process instances.

2.3 Predictive Process Monitoring

Predictive Process Monitoring (PPM) extends process mining by shifting the analytical focus from retrospective analysis of completed cases to forward-looking assessment of ongoing process instances. PPM investigates whether information contained in partially observed execution traces can support early assessment of potential process conditions or outcomes during runtime (Maggi et al., 2014). PPM research commonly distinguishes prediction targets such as numeric predictions (e.g., remaining time), categorical predictions (e.g., outcome classes), and sequence predictions (e.g., next activities). This distinction is methodologically relevant because the prediction target determines how event-log prefixes are represented, which features are engineered, and how performance is evaluated (Di Francescomarino et al., 2018). In financial ERP contexts, categorical assessments are particularly relevant when the analytical objective is to highlight process instances associated with specific control-relevant states or deviations. Most established PPM approaches implement supervised machine learning over event-log prefixes. These approaches typically include a training phase (model induction from historical logs) and a runtime phase (prediction for ongoing instances). Their validity depends on stable case definitions, consistent event semantics, and reliable feature representations. Recent systematic reviews confirm that machine learning applications in BPM are predominantly concentrated in predictive monitoring and process analysis phases, with limited attention to governance-sensitive ERP environments and lifecycle-wide integration (Weinzierl et al., 2024). Feature encoding strategies such as aggregation-based encodings, index-based encodings, and sequence-aware representations play a central role in model behaviour and performance (Di Francescomarino et al., 2018; Tax et al., 2017; Teinemaa et al., 2019). Applying PPM concepts in ERP environments introduces additional constraints. ERP platforms do not provide native process-centric logs; instead, event logs must be reconstructed from transactional records across relational structures (van der Aalst, 2016). In financial ERP processes, observed behaviour is further shaped by accounting logic, validation mechanisms, and configuration-dependent execution rules, which affect both event semantics and the interpretation of deviations. In this thesis, forward-looking monitoring concepts from the PPM literature are used as a conceptual reference rather than as a supervised prediction implementation. Forward-looking support is realised through consolidated and structured ERP status information that enables earlier visibility of open and overdue invoice situations during execution. An AI-assisted interaction layer supports user-oriented explanation and query-based interpretation of the consolidated dataset. Prompt-based interaction provides a mechanism for generating responses conditioned on structured inputs and instruction constraints (Liu et al., 2023). In the present design, this mechanism supports explanation without replacing process mining analysis or ERP-based evidence.

2.4 Explainable Artificial Intelligence in Process Mining

As analytical models increase in complexity, understanding how predictions are generated becomes more difficult. This lack of transparency raises concerns regarding trust and practical usability in finance. Explainable Artificial Intelligence (XAI) addresses this challenge by providing techniques that clarify how models produce their outputs and which factors influence prediction results (Guidotti et al., 2018). In the context of predictive process monitoring, explainability is not a technical enhancement but a functional requirement. Explainability is particularly important in domains where analytical outputs influence professional judgment and control activities. Interpretability becomes critical when decision contexts involve incomplete formalisation, regulatory constraints, or the need for human validation (Doshi-Velez & Kim, 2017). Finance users must be able to understand why specific cases are flagged as control-relevant or require follow-up investigation. Analytical outputs lacking interpretable reasoning often provide limited operational value, as users may struggle to relate analytical outputs to domain-specific decision contexts. Prior research emphasises that black-box predictive models may encounter significant adoption barriers. Even when predictive accuracy is high, the absence of transparent reasoning can reduce user confidence and complicate validation during internal control reviews (Rudin, 2019). In regulated environments, explainability supports both user trust and governance requirements. Process mining introduces an additional dimension to explainability. Unlike conventional prediction tasks based solely on independent variables, predictive process monitoring relies on structured event data and execution traces. Predictions are influenced not only by static attributes but also by process dynamics, including activity sequences, delays, and deviations. Consequently, explainability mechanisms must capture both attribute-level effects and process-level behaviour. Recent studies explore methods for integrating explainability into forward-looking process monitoring. These include feature importance analysis, which identifies globally influential variables, and local explanation techniques, which clarify predictions for individual cases (Guidotti et al., 2018). In financial process contexts, explainability serves a dual purpose. First, it improves the interpretability of predictions by linking risk assessments to recognisable process patterns, such as missing activities or unusual delays. Second, it reduces the risk of misleading conclusions by allowing users to distinguish between genuine execution anomalies and deterministic system behaviour. This distinction is particularly important in ERP systems, where configuration logic often shapes process outcomes. Overall, explainability mechanisms enhance the practical relevance and usability of forward-looking process analytics by linking outputs to domain-relevant process characteristics.

2.5 Design Science Research in Information Systems

Design Science Research (DSR) provides a methodological framework for the systematic design and evaluation of artifacts intended to address practical problems. In contrast to purely descriptive or explanatory research, DSR focuses on the creation of purposeful solutions grounded in scientific knowledge (Hevner et al., 2004). This orientation makes DSR particularly suitable for information systems research, where the objective is often to develop and assess novel technological or analytical artifacts. Peffers et al. (2007) propose a widely adopted DSR process model consisting of six activities: problem identification, definition of objectives, design and development, demonstration, evaluation, and communication. This process structure supports a rigorous yet practice-oriented research approach, ensuring that artifacts are both theoretically informed and operationally relevant.

DSR is especially relevant in domains such as Business Process Management and process mining, where research frequently involves the development of analytical frameworks, decision-support mechanisms, and prototype systems. Rather than evaluating isolated algorithms, DSR emphasises the utility and applicability of artifacts within their intended organisational context. In forward-looking analytical contexts, DSR enables the structured investigation of how analytical and AI-based artifacts can support operational decision-making. Financial ERP environments represent complex socio-technical systems characterised by configuration dependencies, control requirements, and established working routines. Within such settings, the effectiveness of analytical solutions depends not only on analytical correctness and data completeness but also on interpretability, usability, and compatibility with existing processes. By integrating technical evaluation with user-oriented and organisational considerations, DSR provides an appropriate foundation for assessing artifacts that combine process mining, structured data preparation, and explainable AI. This perspective aligns with the objectives of the present study, which seeks to demonstrate how forward-looking and explainable process analytics can be implemented as a decision-support artifact rather than as an autonomous system.

2.6 Research Gap

Prior research has established strong foundations in process mining, forward-looking process monitoring, and explainable artificial intelligence. Process mining primarily provides retrospective insights into process behaviour, whereas forward-looking monitoring approaches analyse partially observed executions (van der Aalst, 2016; Teinemaa et al., 2019). Recent studies further examine the use of large language models as interaction and interpretation mechanisms for process analytics. Recent conceptual work further proposes AI-powered ERP optimisation architectures integrating predictive monitoring, prescriptive automation, reinforcement learning, and digital-twin simulation across heterogeneous ERP landscapes (Nittala, 2025). These frameworks emphasise closed-loop optimisation and enterprise-wide transformation through automated intervention mechanisms. However, comparatively limited attention is given to configuration-dependent financial ERP contexts where accounting logic, validation rules, and governance requirements constrain analytical autonomy and require bounded, explainable decision-support mechanisms. Berti and Qafari (2023) show that LLMs can effectively interpret textual representations of process-mining artifacts such as event logs, directly-follows graphs, and declarative constraints. Rather than executing analytical computations, LLMs operate on structured analytical representations and support reasoning, summarisation, and explanation tasks. The authors demonstrate that LLMs are particularly suitable for answering ad-hoc questions, summarising process behaviour, and supporting sense-making tasks. This approach complements established analytical techniques by improving accessibility and interpretability for non-technical users. This study follows a similar design principle. The AI assistant operates on structured outputs derived from the consolidated dataset and process mining views. Its role is limited to explanation, summarisation, and user interaction. Explainable AI methods address transparency and interpretability requirements that are particularly important in high-stakes decision contexts (Doshi-Velez & Kim, 2017; Rudin, 2019). Despite these developments, important gaps remain, particularly regarding the application of forward-looking monitoring and explanation approaches in ERP environments. A substantial portion of predictive process monitoring research relies on benchmark datasets or comparatively stable process contexts. Process behaviour in ERP systems is shaped not only by user actions but also by configuration parameters, validation logic, and embedded accounting rules. As a result, event data must frequently be reconstructed from transactional records rather than captured in native process logs (Jans, Alles, & Vasarhelyi, 2014). Comparatively limited attention is given to ERP-specific analytical constraints, including configuration-dependent behaviour, reconstructed event logs, and the interpretability requirements of finance-governed processes. These characteristics introduce challenges that directly affect analytical validity, interpretation, and practical usability. Limited empirical evidence exists on how forward-looking monitoring and explanation mechanisms perform in ERP environments where outcomes are strongly influenced by configuration-dependent system behaviour. This thesis addresses this gap by investigating forward-looking and explainable process analytics within D365 FO. The study considers ERP-specific constraints, including reconstructed event logs, configuration dependencies, and the decision context of finance users. Rather than evaluating isolated predictive models, the research develops and assesses a prototype artifact that integrates consolidated customer data preparation, process mining analysis, and explanation mechanisms. The research positions explainable process mining not as a purely technical exercise but as a decision-support approach embedded in organisational and accounting realities.

3 Research Methodology

3.1 Research Approach

This thesis adopts a Design Science Research (DSR) approach to design, implement, and evaluate a decision-support artifact for forward-looking monitoring and explainable interpretation in financial ERP processes. Design Science Research is appropriate because the study focuses on the systematic construction and evaluation of an artifact intended to address a practical problem (Hevner et al., 2004). This orientation aligns with the objective of developing an operational analytical mechanism rather than solely explaining observed phenomena. The artifact developed in this thesis is a decision-support prototype integrating consolidated ERP data preparation, process mining analysis, and AI-assisted interpretation mechanisms. The study focuses on technical feasibility, analytical traceability, and interpretability within a realistic financial ERP context. Consistent with Design Science principles, the study includes both artifact construction and evaluation activities. The evaluation considers monitoring consistency, explanation clarity, and relevance for finance users (Hevner et al., 2004). Consistent with the Design Science Research orientation, artifact evaluation requires a clearly defined reference condition to enable structured interpretation of observed effects. In this study, the baseline condition corresponds to system-supported monitoring and diagnostic routines natively available within D365 FO. Under baseline conditions, users investigate overdue invoices and collection behaviour through standard D365 FO interfaces and reports. Relevant information is retrieved from distributed system representations, including aging views, open transaction forms, settlement histories, and collection records. Diagnostic activities involve manual navigation, cross-referencing of document states, and reconstruction of execution context from heterogeneous system artifacts (Jans et al., 2014). This baseline workflow reflects typical ERP-supported financial control practices, where interpretation is grounded in observable transactional states. The baseline condition provides the comparative reference for assessing artifact-assisted interpretation and diagnostic interaction behaviour.

3.2 Design Science Research Framework

Design Science Research structures the research process into logically connected design and evaluation activities. This study follows the DSR framework proposed by Peffers et al.

(2007), adapted to the analytical and data-structural characteristics of financial ERP systems. The research process comprises five phases:

1. Problem identification and objective definition
2. Event-log extraction and process analysis
3. Process mining analysis and status-based monitoring
4. Explainability and interpretation design
5. Prototype implementation and evaluation

Each phase contributes directly to the design and assessment of the artifact. The first phase defines the research problem and objectives based on the literature and characteristics of financial ERP environments. Financial ERP processes frequently exhibit control-relevant deviations that are detected only after posting or settlement. Late detection leads to manual rework, extended closing cycles, and increased operational effort. Prior studies indicate that ERP-supported controls are predominantly retrospective (Jans et al., 2014). The objective of this research is to design a forward-looking monitoring approach that improves visibility and explanation of control-relevant process conditions during execution. In the third phase, consolidated ERP data is structured to provide consistent visibility of the invoice lifecycle and collection status information. Forward-looking monitoring refers to enabling earlier visibility of open, overdue, or incomplete collection situations based on currently observable invoice states, achieved through structured consolidation and presentation of customer execution information, complemented by process mining visualisation and AI-assisted explanation interfaces. This design aligns with process-mining literature that focuses on observable execution semantics and the reconstruction of execution behaviour (van der Aalst, 2016). Within the analysed accounts receivable collection process, structured invoice status information enabled observation of control-relevant execution patterns prior to completion, including delays, missing collection activities, and persistent overdue states. Analytical value is evaluated primarily in terms of interpretability and practical decision-support relevance for finance users. Explainability mechanisms are designed by linking consolidated invoice status information to recognisable process conditions and configuration-dependent logic. In the final phase, analytical results are delivered through a Microsoft Teams chat interface serving as the interaction layer between users and the artifact. The prototype is evaluated within the accounts receivable collection scenario, focusing on overdue invoices and collection-letter activities. The interface enables users to query invoices exceeding aging thresholds and review whether expected collection actions were performed. Invoices lacking corresponding collection activities are presented as cases requiring user review based on observable invoice status information. Cases exhibiting repeated reminders and persistent overdue status are similarly presented for structured interpretation. Because forward-looking monitoring is based on structured invoice status information, evaluation focuses on interpretability, consistency of data extraction, and usefulness for diagnostic interaction. The artifact is considered analytically consistent when repeated data extraction and consolidation produce stable and reproducible structured representations of invoice status information. Artifact effectiveness is assessed according to three criteria: visibility, defined as the ability to identify open, overdue, or incomplete collection states during execution; interpretability, defined as the correspondence between consolidated data fields, process mining views, and clearly defined collection and invoice lifecycle states; and extraction consistency, defined as the production of stable and reproducible structured representations across repeated extractions. These criteria reflect the analytical objective of supporting decision-relevant interpretation. The evaluation applies a scenario-based Design Science approach to assess how the artifact supports diagnostic and monitoring tasks in the ERP context. Artifact-assisted interpretation is examined relative to baseline monitoring routines, with observations focusing on diagnostic effort, explanation specificity, interaction stability, and interpretability. Measured variables are treated as descriptive characteristics of artifact-assisted diagnostic interaction. This interpretation logic is consistent with human-centred AI perspectives that prioritise interpretability and user-controlled decision processes (Shneiderman, 2020). This design aligns forward-looking monitoring and structured invoice status visibility with established accounts receivable control practices.

3.3 Summary of the Research Methodology

This thesis follows a Design Science Research approach to develop and evaluate a decision-support artifact for forward-looking monitoring and explainable interpretation in a credit and collection finance process. The methodology connects problem definition, D365 FO data extraction, structured dataset consolidation, process mining analysis, and prototype evaluation within a coherent framework. Process-relevant data is extracted from D365 FO and consolidated into a structured dataset containing invoice identifiers, lifecycle states, timestamps, and selected financial attributes. Process mining techniques are applied to visualise execution behaviour and deviations. Forward-looking monitoring is implemented by enabling earlier visibility of partially completed invoices and collection states based on currently observable ERP information. Interpretation of consolidated dataset outputs is supported by an AI-assisted interface based on Microsoft Copilot technologies. Copilot functions as an explanation and interaction layer operating on the structured dataset without executing predictive modelling or rule-based classification. Analytical results are implemented through a Microsoft Teams prototype interface and evaluated for interpretability and decision-support usefulness. Overall, the methodology integrates process mining, structured ERP data consolidation, and explainable AI within a Design Science Research artifact. The evaluation measurements describe how users investigate and understand invoice cases under baseline and artifact-assisted conditions. Observed differences between baseline and artifact-assisted conditions arise from consolidation of execution information and structured presentation of D365 FO status data. Under baseline conditions, users investigate the status and history of an invoice by navigating across customer records, transaction lists, collection history, and settlement screens. Artifact-assisted interpretation consolidates this information into a structured representation, reducing manual search and cross-referencing effort. Measurements describe structural characteristics of analytical interaction. This interpretation is consistent with ERP system design principles that prioritise correctness, traceability, and validation integrity (Nicolaou, 2004).

4 Data Collection and Event-Log Design

Figure 1 summarizes the analytical data pipeline used in this study: extraction of process-relevant data from D365 FO, normalisation and structuring into a reconstructed event-log representation, process mining analysis, and AI-assisted interpretation of the resulting analytical outputs.

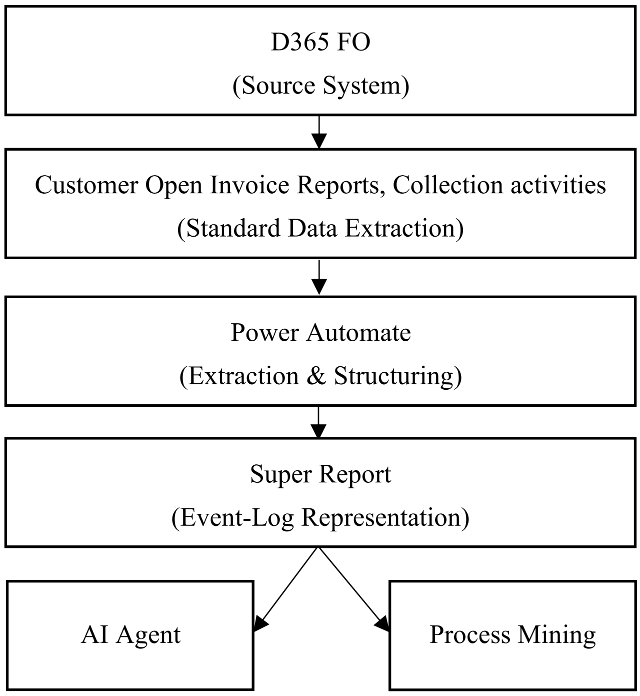

Figure 1: Analytical data pipeline of the proposed artifact

4.1 Process Selection and Scope

D365 FO supports a broad spectrum of business processes across finance, procurement, supply chain and operations. From a process mining perspective, not all ERP processes are equally suitable for event-log–based analysis. The applicability of process mining depends on the availability of structured event data, process repeatability and a clearly defined sequence of observable state transitions that allows individual process instances to be reconstructed. These criteria are commonly used in process mining research to assess the empirical suitability of business processes for event-log-based analysis (van der Aalst, 2016).

Within the finance domain, several candidate processes were evaluated, including accounts payable, general ledger postings and accounts receivable collections. Vendor payment processes are often batch-driven and strongly standardised, which reduces observable variation at case level and limits the value of variant-based analysis. General ledger processes primarily aggregate upstream activities and provide limited traceable case-level execution sequences suitable for event-log reconstruction. The accounts receivable (AR) collection process was selected as the focal process of this study because it combines characteristics that are particularly suitable for process mining and AI-based analysis. First, the process is inherently case-centric, as each customer invoice represents a distinct and traceable process instance. Second, the invoice lifecycle from posting and due-date calculation to reminders, escalation steps and settlement produces multiple timestamped events that are systematically recorded in the ERP system. Third, the process is highly repetitive and governed by D365 FO configuration settings, while still exhibiting deviations caused by master data quality and operational delays. Beyond its analytical suitability, the AR collection process is of high practical relevance for financial management. Deviations such as missing collection letters, delayed reminders, or inconsistent escalation steps can have a direct impact on cash flow while remaining difficult to detect through traditional financial reporting. This makes the process well suited for demonstrating how process mining can be extended with AI-assisted interpretation to support earlier identification and informed decision-making.

4.2 Process Scope and Data Sources

This study reconstructs and analyses the AR collection process using Power Automate Process Mining with event data extracted from D365 FO. The process scope covers the lifecycle of customer invoices from posting to settlement, including intermediate collection activities such as reminders and escalation steps. The AR collection process was selected not only due to its operational relevance, but also because of its strong dependency on ERP configuration and control logic. From a process mining perspective, AR collections provide a clearly defined case structure, a mix of system-driven and user-driven events and frequent deviations from the nominal process flow. Event data is extracted from D365 FO using standard reporting and data export mechanisms. Using standard reporting interfaces for event-log extraction is consistent with prior process mining studies in ERP environments, where direct access to fully normalised logs is often not available (Berti et al., 2023). Power Automate cloud flows are employed to automate data retrieval, transformation and consolidation across multiple D365 FO data source tables. Extracts from multiple D365 FO data sources are consolidated into a reconstructed event-log representation suitable for process mining analysis. The use of standard extraction and automation mechanisms keeps the analysis aligned with realistic D365 FO transaction structures.

4.3 Case Definition and Analytical Unit of Observation

In process mining, a case represents a single instance of a process and serves as the primary unit of analysis for control-flow and performance evaluation (van der Aalst, 2016). In this study, each customer invoice is defined as one case. This case-centric approach enables the complete lifecycle of an invoice to be traced across both operational and collection-related activities. The lifecycle of a case includes, but is not limited to, the following events:

- Invoice posting
- Due date reached
- Invoice becoming overdue
- Collection letter created
- Collection letter sent
- Collection letter posted
- Invoice settled or written off

Event labels represent analytically defined activity abstractions derived from observable D365 FO state transitions rather than direct system event names. Each event is associated with a precise timestamp and enriched with relevant financial details, such as invoice amount, currency, customer group, aging bucket and collection status. This event-log design supports control-flow analysis including the measurement of overdue duration and delays between collection steps. The event log uses invoice ID as Case ID and represents each lifecycle milestone as an activity with an associated timestamp. Case-level attributes that remain invariant across the invoice lifecycle (invoice amount, currency, customer group) are treated as case attributes, while state-dependent attributes such as aging bucket and collection status are recorded as event attributes. Each row in the event log corresponds to one event within an invoice lifecycle. Table 1 presents the event-log schema applied in this study. The schema follows established process mining practice and defines the core attributes required for trace reconstruction and lifecycle analysis.

Table 1. Event-log schema

Attribute	**Description**
Case ID	Unique identifier of the customer invoice
Activity	Invoice status or collection-related event
Timestamp	Date and time of the status change or action
Invoice Amount	Monetary value of the invoice
Currency	Invoice currency
Customer Group	Classification of the customer
Aging Bucket	Overdue category at event time
Collection Status	Current collection stage

This schema supports measurement of delays between events, overdue duration, and comparison of execution variants across customer and invoice categories. The event log is produced by consolidating multiple process-relevant extracts into one structured event table, ordering events chronologically per invoice case. This consolidation is necessary because ERP systems typically do not provide a single, process-centric log and event data must be derived from transactional and reporting sources (Berti et al., 2023).

4.4 Event-Log Construction Using Power Automate

In D365 FO, financial processes are not stored as explicit, process-centric execution logs. Process-relevant information is distributed across transactional records, document lifecycle states, and status attributes associated with financial objects such as customer invoices. For process mining, this requires an explicit transformation step in which operational data is reorganised into an event-log structure suitable for analysis. Consistent with process mining research in ERP environments, this study applies a state-based event-log construction strategy. Instead of relying on user interface actions or low-level technical logs, process events are derived from observable state transitions recorded in the system. In D365 FO, invoice lifecycles are governed by defined accounting and posting logic, and changes in economically meaningful states (e.g., posting, due-date status, reminder stages, settlement) are systematically persisted with timestamps. These state transitions provide a practical basis for reconstructing invoice lifecycle progression at a level of abstraction aligned with finance operations. This abstraction is consistent with established practices in ERP-oriented process mining, where state transitions are used as suitable event representations for process mining analysis (van der Aalst, 2016). Power Automate is used as the extraction and transformation mechanism. Cloud flows retrieve invoice-related attributes and timestamps from D365 FO reporting and data interfaces, normalize them to a consistent schema, and generate event records. Each event record contains:

- Case identifier (invoice ID)
- Activity (interpreted state or collection action)
- Timestamp (time of the observed state change)
- Context attributes required for later analysis (amount, currency, customer group, aging bucket, collection status)

The resulting event log provides a temporally ordered representation of invoice-related execution. It is not intended to reproduce the complete internal logic of D365 FO. Instead, it captures the minimum set of analytically relevant signals needed to reconstruct variants, measure waiting times, detect deviations and support forward-looking monitoring through structured visibility of D365 FO status information. Finally, event-log construction in ERP contexts necessarily involves modelling assumptions. State transitions reflect system design choices, configuration parameters, and document-management rules. The reconstructed log represents an analytical view of process behaviour rather than a direct recording of all human activities. This limitation is inherent to ERP-based process mining and is addressed through transparent case and event definitions.

4.5 Source Data and Status-Based Event Definition

The accounts receivable (AR) collection process in D365 FO is reflected in a sequence of invoice lifecycle states and collection-related actions. Each customer invoice progresses from posting to settlement, while the system records timestamped states and collection steps. In this study, these timestamped lifecycle and collection states are defined as events.

Examples of event types include:

- Invoice posted
- Due date reached
- Invoice becomes overdue
- Collection reminder created
- Collection reminder sent
- Invoice settled

Each occurrence is treated as a discrete event with an associated timestamp. This definition aligns with process mining principles, where events represent observable, time-stamped changes recorded by an information system (van der Aalst, 2016). Defining events in this way supports reconstruction of logs based on consistently persisted system evidence rather than subjective interpretation of user actions.

4.6 Configuration-Aware Event-Log Preparation

In financial ERP systems, execution behaviour is influenced not only by user actions but also by system configuration, which defines validation, posting logic, payment terms, and collection parameters (Granlund & Malmi, 2002). In D365 FO, elements such as posting profiles, account determination rules, validation settings, payment terms, and collection policies can directly affect whether collection steps are triggered, when reminders become eligible, and how exceptions arise. Consequently, observed deviations in reconstructed traces may reflect configuration-driven system behaviour. To address this characteristic, the study incorporates a configuration-aware preparation step in which customer-group-specific system configuration parameters are integrated into the consolidated Super Report dataset prior to deviation interpretation. The objective is not to optimize configuration, but to document configuration conditions that materially shape event occurrence and timing. Configuration context is used to interpret detected gaps and avoid misclassifying configuration-driven system outcomes as execution deviations. Configuration parameters are not used as computational inputs; they are referenced as explanatory context when interpreting process mining results and consolidated ERP status information (Jans et al., 2014).

4.7 Process Discovery

The consolidated event log is imported into Power Automate Process Mining via a dedicated Microsoft Teams folder. During the import procedure, core event-log attributes are explicitly mapped to ensure the correct reconstruction of process traces.

Table 2: Event-log attribute mapping

Event-log attributes	**Event file**
Case ID	Invoice identifier
Activity	Invoice status or collection event
Timestamp	Event execution time

Following ingestion, Power Automate Process Mining derives a process model from the reconstructed event-log representation using automated process discovery techniques. The generated analytical report provides an initial representation of observed execution behaviour within the accounts receivable collection process.

The reconstructed model supports several analytical perspectives, including:

- Process maps illustrating dominant execution paths
- Variant analysis identifying alternative behavioural patterns
- Performance metrics such as activity frequencies and waiting times

These outputs provide a descriptive visualisation of reconstructed process execution patterns derived from ERP event data.

4.8 AI-Assisted Interpretation of Process Mining Results

The event-log design and process discovery procedures enable reconstruction of the accounts receivable collection process based on system-recorded events. Process mining techniques reveal execution paths, deviations, and temporal characteristics of invoice lifecycles. Prior research establishes process mining as an effective mechanism for analysing factual system behaviour in ERP environments (van der Aalst, 2016).

Process mining models require technical interpretation and are not directly consumable by finance users. Discovered models and deviations show behaviour but do not inherently explain causal or operational relevance. In financial ERP contexts, deviations may originate from configuration settings, master-data conditions, or policy-driven exceptions rather than execution failures. Consequently, analytical transparency alone may not directly support finance users' decision processes. To bridge this interpretative gap, the study introduces an AI-assisted explanation layer implemented as a declarative Copilot agent. The agent operates exclusively on structured ERP status information and process mining outputs generated through data consolidation. The declarative agent architecture supports controlled and transparent behaviour. Responses are governed by predefined instructions and controlled prompt design, ensuring alignment with governance and auditability requirements typical of financial environments. The agent's function is restricted to (1) summarisation of findings, (2) contextual explanation of detected deviations, (3) translation of technical process mining outputs and ERP status information into domain-relevant descriptions. This separation maintains methodological clarity by keeping analytical outputs traceable to structured ERP data, while the AI component supports human interpretation. For example, when an overdue invoice lacks a corresponding collection activity recorded in the event log, the agent can interpret this condition by referencing known process states, configuration dependencies, or missing intermediate events. Such explanations reduce investigative effort. The AI component provides information to users, while evaluation and decision-making remain fully under human control. This design aligns with explainable AI principles that prioritise transparency and human oversight in regulated financial ERP environments (Doshi-Velez & Kim, 2017; Rudin, 2019). Importantly, explanation quality is dependent on prior event-log design choices. Case definitions, activity abstraction, and attribute selection directly shape which analytical patterns can be interpreted. Integrating explainability considerations at the data preparation stage improves downstream interpretability. Overall, the AI-assisted layer extends process mining from descriptive analytics toward operationally meaningful decision-support by translating analytical outputs into finance-oriented explanations.

4.9 Summary

This chapter described the data collection strategy and event-log design adopted for analysing the accounts receivable collection process in D365 FO. A central methodological challenge of ERP-based process mining lies in the absence of native process-centric logs. To address this constraint, the study applied a structured consolidation approach. Multiple process-relevant extracts capturing invoice status, aging conditions, and collection activities were retrieved and normalised using Power Automate. These datasets were merged into a unified analytical representation serving as the event log for process mining. The consolidated event log enables reconstruction of invoice lifecycles, identification of execution variants, and measurement of temporal process characteristics. Process discovery techniques generate visualisations of factual system behaviour, revealing dominant execution paths and deviations. Recognizing that analytical outputs alone may not provide sufficient operational clarity, the chapter introduced an AI-assisted interpretation layer. The declarative Copilot agent consumes structured analytical results and generates constrained natural-language explanations. By integrating event-log reconstruction, process mining analysis, and explainable interpretation, the chapter established the analytical foundation required for subsequent forward-looking monitoring and artifact evaluation. The reconstructed event log and analytical outputs represent an abstraction of operational behaviour intended for analytical interpretation.

5 Forward-Looking Monitoring and Evaluation

5.1 Forward-Looking Assessment Objective and Error Definition

The objective of the artifact is to support forward-looking monitoring within the accounts receivable collection process by enabling structured visibility of partially completed invoice lifecycles. Forward-looking monitoring refers to the use of currently observable process information to identify operationally relevant situations before process completion. Conceptually, this perspective relates to research on predictive process monitoring (PPM), which examines how partially observed process information can support earlier visibility of relevant conditions during execution (van der Aalst et al., 2011; Teinemaa et al., 2019). In financial ERP environments, many control-relevant situations become apparent only after transaction finalisation, settlement, or periodic review cycles. Delayed identification increases manual effort and operational disruption. The monitoring objective of this study is to enhance earlier visibility and structured interpretation of review-relevant situations within the invoice lifecycle. Consistent with the event-log design described in Chapter 4, each process instance corresponds to a single customer invoice. Process behaviour is reconstructed from persisted ERP status transitions and document lifecycle states. This case definition enables direct alignment between reconstructed lifecycle progression, process mining outputs, and operational review activities. For monitoring purposes, a process instance is considered review-relevant when consolidated ERP status information indicates conditions such as:

- Prolonged overdue states beyond defined aging thresholds
- Absence of collection activity during visible overdue periods
- Repeated reminder cycles without settlement
- Unusual ordering of lifecycle events

The monitoring mechanism incorporates prerequisite validation logic. Selected master-data attributes (customer communication settings) are evaluated against configured collection mechanisms. Missing prerequisite attributes are treated as forward-looking risk indicators, as they may prevent execution of expected collection steps once the invoice becomes overdue. This extension enables the identification of structurally vulnerable cases at the earliest observable lifecycle stage. By evaluating prerequisite master-data attributes against configured collection mechanisms (presence of customer email for email-based reminders), the artifact exposes configuration–data misalignment before a missing reminder becomes visible in the reconstructed trace. This anticipatory validation does not rely on statistical prediction but on deterministic assessment of execution preconditions. Across analysed instances, corrective interventions frequently occurred after approximately two to three business days, reflecting operational constraints such as validation procedures and workload prioritisation. Measuring elapsed time between the first observable review-relevant state and the corrective action provides contextual understanding of monitoring responsiveness. The artifact supports earlier structured visibility and interpretation of such states.

5.2 Configuration-Aware Interpretation Strategy

Financial ERP processes are strongly influenced by configuration logic, validation rules, and master-data dependencies. Prior research emphasises that observable ERP behaviour reflects embedded system design choices rather than purely user actions (van der Aalst, 2016; Jans et al., 2014). For example, the timing and eligibility of collection activities are determined by payment terms, aging parameters, and collection policies. Consequently, the absence of a collection action does not inherently indicate process failure. To address this characteristic, the study applies a configuration-aware interpretation strategy. Configuration parameters serve exclusively as interpretive context and are not encoded as computational rules within the monitoring mechanism. This separation reflects established distinctions between configuration-induced system behaviour and execution-level deviations in ERP process analysis (Jans et al., 2014). Financial behaviour in D365 FO is governed by parameterized accounting rules, validation mechanisms, and document lifecycle logic, which systematically shape observable execution states (Microsoft, 2025).

5.3 Structured Monitoring Mechanism

The monitoring mechanism integrates three components:

1. Consolidated ERP status representation
2. Event-log reconstruction and process mining visualisation
3. AI-assisted explanation of structured outputs

The consolidated dataset aggregates invoice identifiers, timestamps, lifecycle states, aging categories, collection stages, and financial attributes into a unified representation.

Event-log reconstruction enables descriptive process mining analysis consistent with established event-log abstractions in process-oriented analytics (Di Francescomarino et al., 2018). Process discovery provides visualisation of:

- Execution paths
- Variant differences across invoice lifecycles

These outputs remain directly traceable to reconstructed D365 FO event data. The AI-assisted explanation layer operates on consolidated ERP status information and process mining outputs. It translates structured lifecycle representations into domain-relevant descriptions, clarifies relationships between events, and summarizes observable conditions. This interaction layer supports interpretability while maintaining full traceability to underlying ERP data. In this study, forward-looking monitoring is defined as structured interpretation of partially evolving execution states, conceptually aligned with monitoring perspectives discussed in process mining research (van der Aalst et al., 2011).

5.4 Evaluation Strategy and Assessment Criteria

The evaluation follows a Design Science perspective and considers artifact utility, transparency, and interpretability (Hevner et al., 2004). Because the artifact provides structured monitoring and explanation, evaluation focuses on interpretive quality and diagnostic interaction. The baseline condition corresponds to standard ERP-supported monitoring practices observed prior to artifact introduction. Under baseline conditions, review of overdue invoices required:

- Retrieval of aging reports
- Inspection of open transactions
- Navigation to collection history and reminder records
- Manual reconciliation of lifecycle states across interfaces
- Spreadsheet-based filtering and cross-referencing

Such workflows are consistent with prior observations of ERP-supported diagnostic processes (Jans et al., 2014). The artifact presents invoice statuses through consolidated status representation and process mining visualisation.

Evaluation examines the interpretability of consolidated invoice and collection lifecycle representations, the consistency of event-log reconstruction across repeated extractions, the stability of monitoring outputs, and the usefulness for diagnostic review. The observed differences arise from changes in how information is accessed and organised, which reduce users' need to manually reconstruct execution context, rather than from automated decision logic or computational acceleration. This evaluation perspective aligns with human-centred AI principles that prioritise interpretability, transparency, and user control in high-stakes decision environments (Rudin, 2019; Shneiderman, 2020).

5.5 Interpretation of Analytical Results

Interpretation of the results must consider that observable invoice lifecycle behaviour in D365 FO is shaped both by user execution and by configured system rules such as payment terms, collection policies, and blocking indicators. Consequently, detected deviations (e.g., overdue invoices without reminders) may reflect configuration-defined behaviour rather than operational failures. Analytical emphasis is placed on deviations that remain observable under configuration-aware analysis, including delayed follow-up activities, missing intermediate collection steps, extended overdue states, and deviations in execution order. Several validity dimensions define the boundaries of interpretation. Internal validity may be affected by implicit configuration rules in D365 FO that influence observable lifecycle behaviour. Construct validity depends on how review-relevant conditions and operational errors are defined within the collections process. External validity is limited because the study examines a single ERP platform and one financial process. Evaluation validity is constrained by the qualitative, prototype-oriented assessment design and the absence of statistical testing. These constraints limit the generalizability of the findings while remaining consistent with the objectives and methodological principles of Design Science Research.

5.6 Summary

This chapter described the monitoring objective and evaluation framework of the proposed artifact within a reconstructed ERP process context. Forward-looking monitoring is implemented through structured consolidation of ERP lifecycle information, event-log reconstruction, and process mining analysis. Configuration awareness and operational relevance are treated as primary design objectives. By incorporating configuration-driven behaviour into the consolidated dataset and focusing on observable lifecycle progression, the artifact supports structured identification of review-relevant situations without automating corrective action. The monitoring mechanism provides the analytical foundation for AI-assisted explanation while maintaining transparency and human oversight in financial decision contexts.

6 Prototype Implementation and Interaction Design

6.1 Role of Explainability in Financial ERP Processes

Explainability is a central requirement in financial ERP environments because analytical outputs must be understandable, traceable, and compatible with accounting logic and internal control expectations. In such contexts, analytical mechanisms are evaluated not only by their ability to identify deviations, but also by their ability to support informed decision-making by finance users. The artifact developed in this study evaluates reconstructed execution states through structured consolidation of ERP status information and process mining analysis. The usefulness of these assessments depends on whether users can trace them back to observable process conditions and domain-relevant logic. In D365 FO, many exceptions are not resolved at the level of a single transaction. Instead, recurring issues often originate from configuration settings, master-data inconsistencies, or control parameters that influence process execution across many cases. As a result, explainability in this study is not limited to describing observed patterns. It focuses on review-relevant states to observable process conditions that are familiar to finance and collections specialists. Building on the configuration-aware approach introduced in Chapters 4 and 5, the explanation layer links control-relevant conditions to interpretable statuses such as missing or delayed collection steps, atypical execution sequences, and combinations of overdue status with communication gaps. This supports validation against domain knowledge and reduces the effort required to interpret observable execution patterns. Explainability is also relevant from a governance perspective. In the collections process, highlighted lifecycle conditions may influence prioritisation of overdue invoices, follow-up scheduling, or escalation decisions. These actions require transparency so that users can verify consistency with customer agreements, internal policies, and configured collection procedures. Explainability functions as the bridge between analytical assessment and operational decision-making in a regulated ERP context.

6.2 Prototype Objective and Design Principles

The prototype is designed as a conceptual decision-support solution rather than a production-ready system. Its purpose is to demonstrate feasibility: how earlier visibility derived from consolidated ERP lifecycle data can support earlier identification, interpretation, and structured follow-up in accounts receivable collections.

Three design principles guided the prototype:

- Alignment with ERP controls. D365 FO provides validation rules, posting checks, and workflow-based approvals. The prototype complements these mechanisms by improving visibility and interpretability.
- Structured decision support. The prototype provides structured information and explanations that support user judgment.
- Explanation and prevention over automation. The prototype prioritizes interpretability and root-cause understanding. It helps users understand why a case is flagged and which conditions contributed to the indicator outcome.

Within the prototype, structured monitoring outputs allow users to assess not only which invoices are highlighted, but also how the highlighted conditions relate to observed execution patterns and overdue behaviour. The prototype functions as a controlled demonstration environment to examine the interaction between structured status consolidation, process mining outputs, and explainable interpretation.

6.2.1 Prototype Architecture Overview

The prototype is implemented as a layered analytical architecture that separates transactional ERP processing, structured data consolidation, process mining analysis, and AI-based interpretation into distinct components. This separation keeps results traceable and limits AI interpretation to the prepared dataset. At the source layer, D365 FO functions as the system of record providing transactional and status-oriented data structures, including invoice states, settlement indicators, due-date attributes, and collection artifacts. At the orchestration and consolidation layer, Power Automate cloud flows coordinate data extraction, schema normalisation, and dataset merging. The resulting "Super Report dataset" acts as a unified analytical representation that supports structured monitoring and process mining analysis. This dual function keeps analytical interpretation and reconstructed process traces based on the same underlying evidence. At the process analytics layer, Power Automate Process Mining visualises reconstructed execution behaviour, including dominant variants, activity sequences, and observable deviations. These outputs serve as contextual evidence supporting the interpretation of highlighted states. At the interaction layer, Microsoft Teams provides the user interface and operational access point. The AI assistant operates on structured analytical outputs and process summaries. Its functional role is restricted to explanation, summarisation, and navigation of analytical states. This architecture reflects a design decision: accounting transactions remain within D365 FO, while analytical processing and AI-based explanations operate only on extracted and consolidated datasets.

6.3 D365 FO Configuration

In financial ERP systems, process outcomes are influenced by configuration as well as by execution behaviour. In D365 FO, collection policies, payment terms, posting rules, and master-data settings directly affect whether collection steps are triggered, when reminders become eligible, and how exceptions arise. For example, an overdue invoice without a reminder event may reflect delayed operational execution, but it may also be consistent with configured exclusion rules. Configuration-aware interpretation helps users distinguish between cases requiring operational intervention and cases indicating configuration or master-data issues that should be escalated to system owners. This improves the practical usefulness of the artifact while maintaining traceability and accountability.

6.4 Microsoft Platform Integration Logic

The prototype leverages existing Microsoft platform components to minimise architectural intrusion into the ERP environment while maintaining compatibility with enterprise IT governance structures. Rather than introducing external analytical infrastructure, the design relies on native extensibility mechanisms within the Power Platform and collaboration ecosystem. Power Automate functions as the central orchestration mechanism for scheduling, and transformation of D365 FO datasets. The use of low-code orchestration tools aligns with enterprise integration practices, where automation logic is frequently implemented outside core ERP systems to reduce operational risk. Process analytics are implemented using Power Automate Process Mining, which enables event-log reconstruction and variant discovery without requiring modification of transactional subsystems. This integration model keeps analytical experimentation separate from the production D365 FO environment. Microsoft Teams acts as the interaction surface, reflecting realistic organisational usage patterns in finance departments. Embedding analytical access within Teams aligns interpretation with existing communication channels. The AI assistant is implemented using Copilot interaction logic, constrained to structured analytical outputs. This positioning treats the assistant not as a reasoning engine but as a translating structured analytical state into domain-relevant language. Collectively, these integration choices illustrate a platform-conformant design strategy: analytical capability is introduced through composition of existing enterprise components rather than through architectural replacement or deep system modification.

6.5 AI Assistant Integration

The prototype includes an AI assistant to support the interpretation of analytical results. Its role is to make consolidated monitoring outputs and process mining evidence accessible and interpretable for finance users. Recent research indicates that large language models can interpret textual abstractions of process mining artifacts and support user interaction without executing analytical logic themselves (Berti & Qafari, 2023). In this study, the assistant is implemented conceptually as a Microsoft Copilot agent that operates on structured outputs (e.g., the consolidated Super Report and derived deviation indicators). The assistant translates these structured outputs into domain-oriented explanations, using terminology aligned with accounting and collections work. In a monitoring scenario, the assistant draws attention to invoices highlighted based on consolidated status conditions and explains why a case requires review based on observable conditions such as overdue duration, missing activities, and repeated reminders. The assistant supports informed decision-making by summarising relevant evidence, explaining detected gaps, and highlighting follow-up areas that should be validated by users.

6.5.1 Functional Role of the AI Assistant

Within the prototype, the AI assistant performs a strictly bounded interpretive function. It does not generate predictions, derive indicators, or execute process mining procedures. Instead, it operates as a semantic mediation layer between analytical outputs and user cognition. This role can be characterised as analytical translation rather than analytical computation.

The assistant supports three primary interpretive functions:

- State clarification – explaining why specific invoices or customers are highlighted based on observable lifecycle conditions in the consolidated dataset.
- Navigation support – enabling users to query analytical datasets using domain-oriented language rather than technical query constructs.

By constraining the assistant to precomputed evidence, the prototype avoids a common failure mode of AI-augmented systems: conflation of language-model reasoning with analytical inference.

6.6 Cognitive Role of Copilot in Analytical Systems

The AI assistant implemented through Copilot Studio fulfils a cognitive rather than computational role. Its primary function is representation transformation: converting structured analytical states into human-interpretable narratives aligned with domain terminology. Users frequently navigate multiple reports, screens, and status representations to reconstruct case histories. Even when analytical results are technically available, interpretation effort can limit practical usefulness. The assistant reduces this friction by reorganising outputs into task-relevant explanations. This function is consistent with research on human-centred AI systems, which emphasises that AI components may create value by reducing interpretive complexity rather than by increasing analytical sophistication. By constraining Copilot to datasets and explicit prompts, the artifact maintains analytical traceability while improving interaction efficiency.

6.7 Platform-Native Integration within the Microsoft Ecosystem

A distinguishing characteristic of the prototype is that all analytical and interaction components are implemented using Microsoft platform services. This design choice is methodologically relevant because it demonstrates how forward-looking and explainable process analytics can be embedded into existing enterprise system landscapes without architectural disruption.

The artifact integrates three Microsoft capabilities:

- Power Automate serves as the structured data pipeline and normalisation mechanism. It retrieves ERP data using supported connectors and reporting interfaces. Transformation logic is realised through configured flows.
- Process Mining provides process discovery, variant analysis, and performance visualisation over the reconstructed event log.
- Copilot Studio provides a declarative mechanism for defining assistant behaviour through constrained prompts.

Combining these Microsoft platform services offers several practical architectural advantages. First, data flows remain observable and support auditability expectations typical in financial environments. Second, the AI component operates without privileged system access, reducing governance and compliance risks. From a Design Science Research perspective, this platform alignment demonstrates that explainable, forward-looking analytics can be implemented as an extension of enterprise infrastructure rather than as an external analytical overlay. The artifact illustrates not only analytical feasibility but also deployment realism under typical enterprise constraints. The selected architecture intentionally limits functional scope in order to support traceability and governance compatibility. The artifact does not implement real-time monitoring, adaptive learning mechanisms. Instead, it focuses on lifecycle visibility and structured explanation within a controlled analytical boundary.

6.8 Alignment with Compliance and Control Requirements

The prototype is designed for use in finance-governed processes where decisions affecting customer invoices and accounting records must remain under user responsibility. Analytical outputs are presented in a structured form that users can review and validate. All outputs are grounded in recorded transactional data and reconstructed process states. This supports transparency, auditability, and defensible decision-making, consistent with explainable AI principles for high-stakes domains (Doshi-Velez & Kim, 2017; Rudin, 2019). Users retain responsibility for evaluating results and deciding whether follow-up actions are required.

6.8.1 Interaction Design Considerations

Interaction design within the prototype prioritizes compatibility with established finance workflows rather than open-ended conversational behaviour. Financial operations typically rely on structured exception handling, verification routines, and audit-oriented reasoning patterns. Accordingly, assistant responses are designed to include explicit references to analytical objects such as customers, invoices, and reporting periods, to avoid speculative or advisory language, and to use consistent terminology aligned with established collections processes. The response structure is defined through explicit prompt configuration in Copilot Studio. The assistant is instructed to refer to identifiable analytical objects such as customer, invoice, and reporting period, to avoid speculative or advisory language, and to present findings in a consistent and ordered format. This design limits explanations to the available dataset fields and prevents speculative or generalised responses.

6.9 Prototype Evaluation and User Feedback

The prototype was evaluated using a qualitative, scenario-based assessment approach consistent with Design Science Research principles. The evaluation objective was to assess utility in the target context, focusing on interpretability and fit with daily collections work. Evaluation was conducted in a pre-production environment across three representative legal entities (one per operating cluster) over a 30-day period. Participants were collections specialists from the finance and collections function. Scenarios reflected recurrent operational situations such as missing, delayed reminders or collection letters, repeated reminders, and incomplete customer master data affecting communication. Participants were asked to interpret the presented case behaviour, assess whether activities appeared missing or delayed, and evaluate whether the provided information was sufficient to guide follow-up investigation. Feedback was collected via questionnaires (clarity, usefulness, perceived correctness) and follow-up discussions. Observed effects and comparative findings are reported in Chapter 7.

6.10 Use Cases and Interaction Scenarios

The prototype supports interaction patterns aligned with typical accounts receivable monitoring tasks:

1. Daily collection review. Users request an overview of overdue exposure by aging category, with highlighted exceptions such as missing collection steps, prolonged inactivity, or repeated reminders without settlement.

2. Exception-focused analysis. Users ask targeted questions (customers with overdue invoices lacking communication events). The assistant returns a structured explanation grounded in event evidence and indicator outcomes.
3. Trend and workload assessment. Users request aggregated summaries such as counts of missing reminders, overdue distribution by aging bucket, or recurring deviation categories, derived from the consolidated dataset.

The assistant's role is to organise and explain analytical outputs in a format aligned with finance information needs.

6.11 Observational Usage During Evaluation Period

To obtain initial insights into practical applicability, users were given access to the prototype alongside their established monitoring routines. The prototype provided AI-assisted summaries of overdue invoices, highlighted cases with missing or delayed collection steps, and supported ad-hoc queries related to aging status and potential control gaps. Users continued to rely on existing Excel-based controls for final checks and actions. This reflects established working routines and high reliability requirements typical in finance operations. The prototype was primarily used for early orientation and prioritisation: identifying customers or invoices that required additional verification prior to final validation. Users reported that consolidating aging information and collection status into a single overview reduced the effort required to cross-check multiple reports. They also indicated that additional contextual fields such as collector assignment, customer pool classification, and recent contact channel would increase usefulness. These observations are used in Chapter 7 to explain differences in outcomes across entities and to derive practical implications.

6.11.1 Adoption Behaviour Observations

Observed usage during the evaluation period indicates a characteristic adoption pattern consistent with enterprise decision-support technologies. Users initially treat AI-assisted analytical tools as orientation mechanisms. The prototype was primarily used to reduce search and consolidation effort, while final validation remained anchored in existing ERP screens and Excel controls. This behaviour does not indicate rejection of the artifact but reflects risk-aware operational norms within finance functions.

6.12 Prompt Engineering and Iterative Refinement of the AI Assistant

The AI assistant's behaviour is controlled through a declarative prompt that defines scope, response structure, and constraints. In this setting, the main design task is not model training but shaping how deterministic outputs are interpreted and communicated to finance users. Prompt design is treated as an artifact design activity within the DSR framework (Hevner et al., 2004; Peffers et al., 2007). The prompt defines response structure, terminology, and behavioural constraints to ensure consistency and user responsibility. Early interactions indicated that responses could become overly generic when user queries lacked explicit identifiers. The prompt was refined iteratively: queries were tested, response clarity was assessed, and instructions were adjusted. Key refinements included explicit inclusion of aging buckets, differentiation of letter statuses (created, sent, posted), a stronger focus on exceptions rather than full listings, and constraints preventing the assistant from suggesting or implying operational actions such as posting, blocking, or customer communication. Later prompt versions also enforced structured response templates to improve readability and reduce cognitive effort. The prompt and interaction layer were implemented and tested using Microsoft Copilot Studio, Power Automate for data orchestration, and Microsoft Teams as the user-facing environment. This combination of platform components enabled rapid iteration of the interaction layer without requiring changes to ERP workflows or process mining configurations.

6.12.1 Final Prompt Design and Configuration

The final prompt defines the assistant as a decision-support agent for accounts receivable monitoring. It prioritises exception situations such as overdue invoices without recorded collection actions and repeated reminders without settlement. It enforces a consistent response structure, including a case overview and explicitly identified gaps. The prompt specifies that explanations must be derived from the consolidated dataset generated during data preparation. The assistant may only reference information contained in this structured file and is not permitted to introduce external assumptions or initiate operational actions. The prompt further makes explicit that all decisions and follow-up activities remain the responsibility of the user. This design reflects Microsoft Copilot principles that emphasise responsible AI use, traceability of outputs, and human oversight.

6.12.2 Alternative Design Options Considered and Rejected

Several alternative designs were considered and rejected due to governance, interpretability, and operational constraints:

- Automated operational decisions related to customer blocking, automatic collection triggers were rejected due to accountability and compliance risks in finance-governed processes (Rudin, 2019; Shneiderman, 2020).
- Introducing statistical prediction models was rejected because the objective is a transparent interpretation of observable lifecycle states.

- Real-time event streaming was rejected because collection work is typically periodic and data stabilises after validation and batch updates; batch-oriented consolidation is more consistent with ERP reporting structures (van der Aalst, 2016).

The selected architecture uses Copilot as an interpretation and interaction layer operating on structured outputs. This maintains traceability and supports human-in-the-loop decision support.

6.13 Summary

This chapter described prototype implementation and interaction design for a decision-support artifact combining structured lifecycle monitoring, reconstructed event evidence, process mining outputs, and a constrained AI explanation layer. The prototype is designed to complement existing ERP controls, maintain user responsibility, and improve interpretability in a finance-governed process context. Chapter 7 reports empirical observations and discusses the results, implications, and limitations.

7 Evaluation Results and Discussion

7.1 Discussion of Results

This study demonstrates that earlier visibility through consolidated lifecycle evidence can be applied to financial ERP processes, provided that monitoring indicators are derived from reconstructed lifecycle states and interpreted with configuration awareness. The AI assistant was implemented as an explanation and interaction layer operating on structured process-mining outputs and consolidated dataset fields. This design is consistent with recent work showing that large language models can support interpretation and user interaction for process mining artifacts without executing analytical logic (Berti & Qafari, 2023). In D365 FO, where late detection of issues often causes manual rework, interpretability-oriented consolidation provides practical value by accelerating investigation and supporting earlier understanding of potential control gaps. The design adopted in this study addresses this issue by grounding highlighted lifecycle conditions in observable dataset states. Overall, the results support the design goal of providing decision support that is traceable, interpretable, and compatible with financial control practices (Hevner et al., 2004; Peffers et al., 2007).

7.1.1 Process Mining as an Enabler of Explainable Error Management

The process mining fulfils two functions within the artifact. First, it reconstructs actual ERP process execution from event data. In line with Microsoft documentation, process mining uses reconstructed event logs to visualise real process flows and visualises real process flows, including activity sequences, variants, throughput times, and bottlenecks. This provides a structured representation of how cases are executed in practice rather than how they are formally designed. Second, process mining supplies the behavioural context required for explainable error management. Traditional ERP monitoring tools, such as transaction lists, aging reports, or exception flags are typically rule-based and transaction-focused. By contrast, process mining operates at the trace level. It captures complete case lifecycles and variant differences. This enables identification of patterns that may not constitute technical errors but indicate operational instability, such as inactivity, repeated reminder cycles, or atypical state transitions. Within the artifact, lifecycle evidence is consolidated into a structured dataset representing control-relevant case conditions. The AI assistant then translates this evidence into domain-specific explanations. The result is explainable error management rather than isolated exception detection.

Explainable error management differs from conventional exception reporting. Instead of merely flagging anomalies, the artifact explains them using traceable execution data. For finance users, this distinction is critical. Not every deviation represents an accounting error; some reflect configuration choices or defined business rules. Without contextual interpretation, monitoring systems may generate alerts that are technically correct but operationally misleading. The approach does not attempt to predict future outcomes. Instead, it enhances visibility by consolidating evolving lifecycle conditions associated with increased operational risk. In this way, Microsoft's process mining capability provides execution-based evidence that supports structured and explainable decision support.

7.1.2 Interpretation of Diagnostic Effort Reduction

The reduced diagnostic effort may contribute to improved operational efficiency, however, the study does not formally measure the overall productivity impact. The measured effects arise from structural properties of information presentation and analytical structuring. D365 FO investigations require navigation across multiple screens, reports, and object views. Information relevant to overdue assessment, reminder history, settlement status, and customer attributes is distributed across functional modules such as customers, projects, credit and collections. The delay in diagnosis is driven primarily by manual search and context reconstruction effort, not by computational complexity. The artifact reduces diagnostic effort by consolidating execution evidence and presenting pre-structured analytical states. Users no longer reconstruct execution context manually. Instead, they evaluate already organised case representations. The observed reduction in diagnostic effort reflects decreased navigation time and reduced manual cross-referencing. Variations across entities further reinforce this interpretation. Where reconstructed datasets included complete events and visible communication steps, explanations were more specific and case resolution time decreased.

7.1.3 Effects on Diagnostic Interpretation

D365 FO finance data is state-driven and distributed across many fields and screens. When users only see isolated status values or report rows, results can look like disconnected facts rather than a coherent case history. This increases ambiguity during investigation, because users must manually reconstruct what happened and in which order. Process mining provides a structural view that connects extracted events into an execution trajectory. Variants and state transitions show typical paths from invoice created to reminder created to reminder sent to settlement and make it easier to see when a case follows an unusual sequence (overdue state with missing communication steps). This reduces interpretive ambiguity because users can interpret a case as a process pattern, not just as a set of separate fields. In the artifact, the dataset provides the observable invoice and reminder statuses, process mining organises these observations into an interpretable behavioural structure, and the AI assistant explains the case in domain language using only this provided evidence. Explanation quality depends both on the underlying data and on how the prompt and process structure organise that data into a readable explanation.

7.2 Artifact Configuration and Behaviour

Table 4 shows the artifact pipeline and processing layers. It illustrates how data flows from D365 FO into a consolidated analytical dataset, how this dataset is used for descriptive process mining, and how the AI assistant operates only on these prepared outputs.

Table 4: Artifact pipeline and processing layers

Layer	Component / Tool	Input	Transformation	Output	Frequency
Source system	D365 FO	Invoices, reminders, payments	Standard reporting	Extracts	Daily
Structured consolidation	Power Automate	Extracts	Normalize + consolidate	Super Report Dataset	Daily
Process analytics	Process Mining Tool	Super Report Dataset (event-log representation)	Discovery and analysis	Variants, execution paths	Daily or weekly
Explanation	Copilot and Teams	Super Report Dataset	Prompt-driven interpretation	Structured explanations	Daily or weekly

ERP processing remains separate from analysis. Dataset preparation is traceable, process mining is descriptive, and the AI assistant is limited to the explanation of the prepared outputs.

7.3 Evaluation Context and Entity Selection

The empirical investigation began with a structural analysis of 25 legal entities operating within a shared D365 FO environment. The objective was to identify differences in collections operating models that influence observable execution behaviour. Three recurring operating clusters were identified. Table 5 shows the identified operating clusters and their distinguishing operational characteristics.

Table 5: Operating clusters

Cluster Type	Representative Entity	Operational Characteristics
Direct-interaction	France entity	Finance teams communicate directly with customers
Partner-mediated	German entity	Customer interaction occurs via external collection partners
Small-volume	Belgium entity	Low invoice volumes and simplified collections workflows

ERP and process mining literature indicates that such structural differences strongly influence observable execution behaviour and trace characteristics (van der Aalst, 2016).
Following clustering, one representative entity was selected per cluster for artifact deployment. Selection prioritized structural diversity rather than statistical representativeness. The inclusion of the Belgium entity serves a distinct analytical purpose. Rather than supporting cross-entity performance comparison, the entity was intentionally selected to observe artifact behaviour under low-volume operational conditions. Small-volume environments represent a relevant boundary condition for ERP analytics, where limited case diversity and reduced interaction complexity may influence indicator triggering and explanation dynamics. Consequently, observations derived from this entity are interpreted as structural robustness validation rather than statistically representative evaluation evidence.

7.3.1 Structural Sensitivity to Event Semantics

The evaluation shows that the artifact's usefulness depends more on what events are recorded than on how much data is available. For example, whether sending a reminder, generating a collection letter, or recording a payment is captured as a distinct time-stamped event. If important lifecycle steps are not recorded (for example, communication events are missing), even a large dataset will not produce clear explanations. The system can only organise and explain what is visible. In such cases, explanations become less specific because key transitions in the invoice lifecycle are not observable. On the other hand, even a moderate-sized dataset can provide strong interpretability when event definitions clearly reflect economically meaningful transitions, such as invoice creation, reminder generation, settlement, dispute status, or blocking indicators. When these states are explicitly recorded, process mining can reconstruct understandable execution paths, and the AI assistant can provide grounded explanations. This confirms an important principle from process mining: analytical quality depends on how events are defined and structured, not simply on data volume.

7.4 Dataset Characteristics

Table 6 shows the dataset characteristics of the selected entities, highlighting substantial differences in customer base size, open invoice volumes, and overdue exposure across France, German, and Belgium.

Table 6: Dataset characteristics

Metric	France	Germany	Belgium
Active customers	8,825	2,490	416
Open invoices	3,642	5,802	927
Overdue invoices	1,092	580	185

France has a large customer base and a high number of overdue invoices. This creates many different case situations and more behavioural variation. The German entity has fewer customers but a high number of open invoices, which increases case complexity per customer. Belgium represents a small-scale environment with limited case diversity and simpler workflows. If important lifecycle events are missing, even a large dataset will not produce clear explanations. In contrast, a smaller dataset can still support stable and understandable results if key invoice and communication states are clearly recorded. Dataset scale affects visualisation complexity and case diversity, but interpretability depends more on the clarity and completeness of recorded lifecycle events than on volume alone (van der Aalst, 2016).

7.5 Evaluation Procedure and Observed Information Demands

The artifact was evaluated in a pre-production environment over a 30-day observation period. Dataset definitions and consolidation rules remained unchanged during the evaluation period. The baseline condition corresponds to the standard diagnostic workflow applied prior to artifact introduction. Under baseline conditions, identification and interpretation of overdue and collection-related situations required sequential navigation across multiple D365 FO interfaces and spreadsheet-based representations, involving manual reconstruction of execution context from distributed transactional artifacts. Table 7 shows the volume and type of information demands observed during the 30-day evaluation period.

Table 7: Observed information demands

Variable	France	Germany	Belgium
Customer requests (30 days)	24	12	4
Management requests (30 days)	7	5	1
Events reviewed during testing*	76	48	12

*Events reviewed refers to the number of scenario walkthrough instances evaluated using the artifact outputs, not raw ERP event-log rows. Most observed requests focused on explaining why invoices were overdue, verifying whether reminders were issued correctly, confirming payment or settlement status, and checking whether expected collection steps had been performed. The evaluation did not aim to produce statistically generalizable results. The feedback was collected in a structured way using predefined evaluation dimensions. This ensured that user perceptions could be compared consistently across scenarios while keeping the focus on interpretability and practical relevance rather than statistical testing. Participants were finance and collections specialists involved in routine overdue monitoring and exception handling. A total of 5 participants across the three entities completed the questionnaire. The questionnaire used a fixed Likert-type response scale for each evaluation dimension (1 = strongly disagree, 5 = strongly agree). Given the limited number of participants (n = 5), results are interpreted as indicative perception. Responses were aggregated across scenarios per entity and reviewed descriptively across entities. Scaled responses generally indicated higher agreement on “Understanding Support” and “Ambiguity Reduction” dimensions, particularly in scenarios with complete lifecycle visibility. Lower ratings on “Explanation Specificity” were observed in cases where communication events were not fully represented in the dataset. Table 8 shows the qualitative evaluation dimensions applied in the structured user feedback questionnaire.

Table 8: Qualitative User Feedback Dimensions

Evaluation Dimension	Assessment Focus	Interpretation Objective
Understanding Support	Whether explanations clarified overdue status and reminder logic	Identify perceived analytical usefulness
Ambiguity Reduction	Whether outputs reduced uncertainty during investigation	Assess decision-support contribution
Explanation Specificity	Whether responses were sufficiently case-specific and precise	Evaluate interpretability quality
Interpretation Friction	Whether outputs caused confusion or required additional verification	Detect usability or reasoning limitations

Ratings were analysed descriptively using average tendencies and response distributions to identify consistent perception patterns across entities. Diagnostic effort observations were conducted using a structured scenario-based walkthrough protocol designed to maintain comparability across baseline and artifact-assisted conditions. Each walkthrough instance represented a bounded diagnostic task defined by a stable analytical objective (explanation of a missing collection letter). For each scenario category (customer-level and management-level), identical analytical objectives and investigation boundaries were applied. Time measurement commenced upon presentation of the diagnostic context and concluded when a stable diagnostic conclusion or required follow-up action was identified. Baseline and artifact-assisted walkthroughs were performed by the same users under comparable task definitions to reduce variation arising from user experience or task interpretation. Because dataset preparation and process mining outputs were fixed during the evaluation period, observed duration differences primarily reflect differences in interaction structure rather than variation in analytical computation. The feedback mechanism does not constitute a statistically representative usability evaluation. It provides structured perception-based validation consistent with Design Science Research practices that focus on interpretability (Hevner et al., 2004).

7.5.1 Evaluation Measurement

Diagnostic effort was assessed using a structured scenario-based walkthrough procedure to ensure consistency of analytical tasks across baseline and artifact-assisted conditions. The unit of measurement was a diagnostic walkthrough instance, defined as a complete investigation of one accounts receivable case (e.g., an overdue customer account) in response to a defined inquiry context. User requests themselves were not treated as measurement units, since requests may vary in scope, referenced artifacts, and objectives. Instead, effort was measured based on the resolution of a bounded analytical objective (for example, explaining a missing collection letter). A single request could generate multiple walkthrough instances when separate analytical objectives were examined. For each scenario type (customer-level and management-level), identical analytical objectives were applied across both conditions. Observations recorded the elapsed time required to interpret and resolve the diagnostic task. Measurement began when the inquiry context was presented and ended when a stable diagnostic conclusion was reached or required follow-up verification was identified. Repeated walkthroughs were conducted to ensure stability of interaction durations. In the baseline condition, users typically navigated across multiple D365 FO screens and relied on clarification via Teams to reconstruct case context. The consolidated dataset reduced this coordination and search effort. Since dataset preparation and configuration remained constant, observed variation across repetitions reflects differences in user interaction behaviour rather than changes in analytical processing.

7.6 Evaluation Protocol

The evaluation covered 47 invoice cases across three legal entities over a 30-day observation period. Cases were selected based on predefined review-relevant conditions, including overdue status beyond configured aging thresholds, missing collection activities, repeated reminder cycles, and prerequisite master-data gaps affecting communication. Case selection followed two mechanisms. First, invoices flagged by the structured monitoring indicators in the consolidated dataset were included. Second, a subset of non-flagged overdue invoices was reviewed to provide contextual comparison. This approach ensured that both indicator-positive and standard cases were examined. The review was performed by collections specialists within the finance function who routinely conduct overdue monitoring and follow-up activities in D365 FO. Participants evaluated each case using both the artifact interface and baseline D365 FO navigation routines. Baseline comparison relied on standard D365 FO aging views, open transaction forms, collection history screens, and Excel-based reconciliation reports. Users reconstructed lifecycle information manually under baseline conditions and compared this to the consolidated representation provided by the artifact. Each scenario was reviewed twice to reduce learning effects and to assess consistency of interpretation across repeated interactions. Observations focused on interpretability, diagnostic effort, information search behavior, and explanation clarity rather than statistical performance metrics. This evaluation protocol emphasises procedural transparency and reproducibility within a realistic ERP monitoring context.

7.7 Comparative Diagnostic Effort

Descriptive observations of diagnostic effort were obtained under two conditions: (1) manual investigation using ERP screens and Excel reports (baseline), and (2) artifact-assisted interpretation combining process mining outputs and AI-generated explanations. Table 9 presents the indicative average diagnostic durations observed under baseline and artifact-assisted conditions.

Table 9: Indicative diagnostic effort (minutes)

Review Activity	France	Germany	Belgium
Time to review customer request manually (min)	8	8	4
Time to review management request manually (min)	14	14	12
Time with artifact assistance – customer (min)	2	10	4
Time with artifact assistance – management (min)	10	14	9

The reported values represent descriptive averages derived from repeated scenario-based assessments conducted under comparable task conditions. Each scenario type was evaluated multiple times by the same users under consistent task definitions to reduce variability arising from user familiarity or interpretation differences. Tables 9a–9c present dispersion measures for each entity, illustrating how the mean values reported in Table 9 were derived.

Table 9a: Diagnostic duration dispersion in France (minutes)

Condition	Entity	Mean	Min	Max	Std. Dev.
Baseline – Customer	France	8	7	9	0.8
Artifact – Customer	France	2	2	3	0.5
Baseline – Management	France	14	12	15	1.2
Artifact – Management	France	10	9	11	0.9

Table 9b: Diagnostic duration dispersion in Germany (minutes)

Condition	Entity	Mean	Min	Max	Std. Dev.
Baseline – Customer	Germany	8	7	9	0.9
Artifact – Customer	Germany	10	9	11	0.8
Baseline – Management	Germany	14	13	15	0.7
Artifact – Management	Germany	14	13	16	1.1

Table 9c: Diagnostic duration dispersion in Belgium (minutes)

Condition	Entity	Mean	Min	Max	Std. Dev.
Baseline – Customer	Belgium	4	3	5	0.8
Artifact – Customer	Belgium	4	3	5	0.7
Baseline – Management	Belgium	12	11	13	0.9
Artifact – Management	Belgium	9	8	10	0.8

Across entities, dispersion measures indicate stable diagnostic durations under repeated walkthrough conditions (n = 5 per scenario type and condition). The limited standard deviation values and narrow minimum–maximum ranges suggest that observed duration differences were not driven by isolated extreme values. Where baseline and artifact-assisted ranges overlap substantially (Germany), no structural diagnostic acceleration can be inferred. Where ranges separate consistently across repetitions (France customer-level scenarios), a stable interaction restructuring effect is observable at a descriptive level. Cross-entity variation reflects structural differences in dataset completeness and communication-event visibility rather than instability of the measurement protocol. France demonstrates a consistent reduction under artifact-assisted conditions. German entity exhibits overlapping duration ranges attributable to incomplete communication-event semantics. Belgium shows moderate reduction in management-level scenarios within a low-volume operational context. Observed diagnostic durations were shortest in the direct-interaction entity (France). In France, customer-level interpretation decreased from approximately 8 minutes to 2 minutes per case, while management-level interpretation decreased from 14 minutes to 10 minutes. In Belgium, improvements were moderate and limited primarily to management-level tasks, reflecting lower baseline complexity. In the German entity, diagnostic duration did not improve because missing communication-event visibility limited reconstruction of the reminder execution context. This is consistent with previously identified limitations in communication-event visibility, which constrained reconstruction of execution context. These differences indicate that observed effectiveness is primarily conditioned by data completeness and event semantics rather than by modifications to the artifact itself. Under baseline conditions, investigation required navigation across multiple ERP modules and reconstruction of distributed transactional information. Diagnostic execution did not occur within a unified analytical interface but instead required manual reconstruction of context across system artifacts. When investigating collection-letter-related inquiries under baseline conditions, users followed a recurrent navigation sequence within D365 FO. The procedure required manual reconstruction of the execution context across distributed ERP components. Table 10 summarizes the typical manual navigation and verification steps required under baseline conditions.

Table 10: Manual Steps to review a request

Number	Step in D365 FO	Activities
1	Customer Identification	Determine the customer account by searching for the customer name within the Accounts Receivable module. Where a unique identifier was unavailable, users accessed the All Customers view and manually resolved the corresponding customer ID.
2	Transaction Context Reconstruction	Open the customer account and review All Transactions to identify relevant open invoices and document identifiers associated with the inquiry.
3	Overdue State Verification	Navigate to the Credit and Collections workspace to inspect overdue invoices, aging status, and collection eligibility conditions.
4	Collection Activity Validation	Review reminder and collection-letter history to determine whether expected collection actions were generated for the identified invoices.
5	Status Consistency Checks	Cross-reference invoice lifecycle states (e.g., dispute status, settlement conditions, blocking indicators) with collection artifacts to detect potential exclusion conditions.
6	Delivery Verification	Where a collection letter was generated, access batch-processing or execution logs to confirm transmission status and identify delivery failures.

7	Exception Diagnosis	Investigate configuration or master-data-related causes of non-delivery, including missing customer email address, invoice dispute states or timing effects.

This multi-step navigation structure explains why baseline diagnostic effort reflects structural information fragmentation rather than analytical complexity. Diagnostic duration under baseline conditions is primarily driven by information distribution and context reconstruction requirements rather than ambiguity in process states. Diagnostic effort varied according to the informational scope of incoming requests. Customer-level inquiries were typically narrow and state-oriented, focusing on verification of overdue status, reminder presence, or payment confirmation. Resolution was usually achieved once the relevant transactional state or collection artifact was identified. Management-level inquiries required broader interpretive reconstruction, including explanation of execution history, clarification of causal factors, validation of communication events, and justification of follow-up actions. Diagnostic interaction extended beyond state verification toward structured reasoning and contextual explanation. Under artifact-assisted conditions, AI-generated explanations provided structured orientation and clarification of observable execution states. This support proved particularly effective in customer-level scenarios, where bounded explanations aligned with limited informational scope. In management-level scenarios, AI assistance did not eliminate the need for verification, rather, explanations functioned as interpretive scaffolding that supported subsequent human validation. Diagnostic duration remained influenced by contextual justification requirements, even when the artifact provided early orientation. These measurements describe interaction characteristics under controlled prototype conditions and should not be interpreted as inferential evidence of organizational productivity effects.

7.8 Prompt Constraints and Interaction Behaviour

Because AI interpretation relies on prompt-governed output generation rather than predictive inference, prompt construction is a critical design variable. Early interaction tests showed that responses became generalised when user queries lacked explicit anchors (invoice ID, customer ID, or date period). Prompt constraints were treated as interaction controls to enforce bounded interpretation and alignment with dataset semantics. Table 11 shows how query anchoring influenced response specificity before and after prompt refinement.

Table 11: Query anchoring and response stability

Query Type	Example User Question	Required Parameters	Result Before Refinement	Result After Refinement
Customer clarification	Why did I get a reminder?	Customer ID	Generalised explanation with low contextual grounding	Case-specific breakdown of overdue invoices
Payment mismatch	I paid already	Payment reference / date	Ambiguous interpretation dependent on missing context	Explanation linked to settlement and reconciliation status
Process gap	Which overdue invoices have not been sent?	Customer ID + temporal reference	Broad response with reduced precision	Structured listing of invoices and observable gaps in the dataset

To stabilise interaction behaviour, prompt design was iteratively refined. The refinement strategy introduced fixed reference parameters, response constraints, and structured explanation templates. Table 12 shows the prompt stabilisation mechanisms introduced to improve interaction reliability.

Table 12: Stabilisation mechanisms

Stabilisation Mechanism	Functional Objective
Fixed reference parameters	Enforce explicit identification of analytical objects
Response constraints	Prevent speculative outputs
Structured explanation templates	Standardize reasoning structure

Queries containing explicit identifiers consistently produced precise and operationally interpretable explanations. Queries lacking anchors produced generalised answers with reduced decision-support specificity. Output quality depended on the completeness of the provided dataset inputs.

7.9 Cross-Entity Differences and Explanatory Factors

The artifact showed stronger effects in entities where the reconstructed dataset contained complete lifecycle events and visible communication steps. Under these conditions, structured explanations reduced manual investigation effort and supported case prioritisation. Where communication events were incomplete or not visible in the extracted data, explanations became less specific and interaction benefits were reduced. Table 13 shows the cross-entity observation summary across key analytical and interaction aspects.

Table 13: Cross-entity observation summary

Observation Aspect	France	Germany	Belgium
Data completeness	High	Partial	High
Condition consistency	High	Limited by communication visibility	High
Explanation reliability	High	Variable	High
Operational usefulness	High	Limited by communication visibility	Moderate

Entities exhibited substantial variation in request volumes and observable event structures. Entities with very low request counts, such as Belgium, do not provide a reliable basis for quantitative comparison and are interpreted as illustrative boundary-condition cases. Observed constraints in the German entity dataset were attributable to incomplete visibility of communication-related events. This limitation affected explanation specificity but did not change dataset preparation or process mining outputs, indicating a dependency on reconstructed event semantics. Differences observed across entities primarily reflect variations in event completeness, communication visibility, and information architecture rather than differences in the artifact's preparation steps.

7.10 Implications for Research

This study contributes to earlier visibility and explainable process analytics in D365 FO contexts by demonstrating an approach to early identification and interpretation of control-relevant conditions in a credit and collection module. While predictive process monitoring literature often assumes stable process semantics and homogeneous event logs, financial ERP processes require explicit separation of configuration-driven outcomes from execution-driven deviations to avoid misleading interpretations. The results suggest that future ERP-focused monitoring research should incorporate a systematic treatment of configuration effects, including formal approaches to documenting which configuration elements shape event occurrence and timing. Methodologically, the study also demonstrates the usefulness of Design Science Research for developing and evaluating composite artifacts that combine traceable dataset preparation, process mining, and AI interpretation.

7.11 Practical Implications

The artifact supports collections monitoring by consolidating process evidence and supporting identification of missing activities, delays, and atypical execution patterns based on consolidated lifecycle evidence. This reduces reliance on retrospective validation routines and supports prioritisation of cases requiring investigation. A practical contribution is the integration of an AI-assisted interpretation layer into a realistic Microsoft ecosystem workflow. By restricting the AI assistant to structured outputs and applying explicit prompt constraints, the design maintains traceability and reduces governance risk. The assistant acts as an interpretation layer rather than a knowledge-generating mechanism, which is particularly important in finance-governed environments. Automation of data extraction and consolidation via Power Automate reduces manual preparation effort and improves consistency compared to fragmented Excel-based controls. Users reported the artifact as most valuable for early orientation and exception identification, while final verification and actions remained human-controlled.

7.11.1 Organisational Implications of AI Analytics

The introduction of an AI-assisted interpretation layer alters how users engage with analytical outputs. Instead of interacting with reports or dashboards, users engage through question-driven exploration. This interaction mode shifts analytical consumption from passive inspection toward active inquiry. Users formulate operational questions “why”, “which”, “missing” rather than navigating predefined visualisations. Such interaction patterns may reduce training barriers and improve accessibility for non-technical stakeholders. Importantly, the assistant does not replace analytical reasoning but restructures access to analytical states. Organisational adoption depends on perceived reliability of explanations and alignment with established verification practices.

7.11.2 Potential Extension to Customer-Facing Interaction

Post-evaluation discussions identified that potential high-value application scenario for the artifact could be integration of the AI-assisted explanation mechanism into a customer portal environment. In this configuration, customers could query invoice status, reminders, and overdue conditions using natural language, while the system would provide structured explanations derived from analytical states. Responses would need to be strictly grounded in explicit dataset fields and restricted to controlled templates to avoid ambiguous or speculative phrasing. From an operational perspective, this extension appears attractive because a substantial portion of the collections workload originates from repetitive customer inquiries concerning payment status, reminder validity, and overdue calculations. An explanation-oriented interface could reduce manual clarification effort and improve consistency of responses. However, introducing AI-generated explanations into a customer-facing channel raises significant governance and risk considerations. Financial communications require strict correctness, controlled language, and avoidance of speculative interpretation. Even when the analytical logic is clearly defined, AI-generated explanations may sound correct but lack precision. In regulated financial contexts, this can lead to communication and compliance risks. For this reason, deployment of the explanation component in a customer-accessible interface was intentionally deferred. The evaluation indicated that additional safeguards are required, including strictly constrained response templates and formal validation of permitted explanation scopes. Without such controls, ambiguous responses could negatively affect customer relationships. A second unresolved consideration concerns economic feasibility. Although the artifact demonstrated interpretive benefits during prototype evaluation, no formal cost–benefit assessment was conducted. The economic viability of a customer-facing AI explanation service depends on variables not analysed within this study, including expected inquiry volumes, interaction frequency and infrastructure costs. Consequently, financial efficiency implications remain indeterminate. These observations highlight that extension of AI-assisted analytical artifacts into customer-facing environments is primarily constrained by governance, reliability, and economic evaluation requirements rather than analytical capability alone.

7.12 Limitations

This study has several limitations. First, the investigation evaluates a single financial process within one ERP platform, which constrains external validity. Observed artifact behaviour and analytical interaction characteristics may differ across alternative ERP systems, process configurations, or organisational control structures. Second, the dataset definitions and consolidation rules were designed with limited structural complexity to maintain interpretability and implementation feasibility. The results consequently illustrate the applicability of earlier visibility through consolidated lifecycle evidence. Future research may evaluate more complex lifecycle-condition definitions, alternative detection mechanisms, or adaptive analytical strategies. Third, treatment of configuration effects relies on domain knowledge applied during data preparation and interpretation. Although the approach improves analytical relevance in the examined ERP context, there is currently no standardized and broadly applicable method for systematically identifying behavioural effects caused by system configuration. Developing such generalizable methods remains an open research challenge. Fourth, empirical observations are conditioned by confidentiality and data protection constraints inherent to ERP environments. Although legal entities, invoice volumes, and aggregated evaluation metrics are derived from real operational data, direct disclosure of transactional records including invoice-level details, customer identities, and financial amounts is restricted. Evaluation and illustration rely on abstracted analytical representations rather than raw transactional instances. This restriction reflects standard enterprise governance requirements and does not affect the validity of traceable dataset preparation and consolidation. Fifth, cross-entity comparisons are inherently sensitive to variations in event completeness and reconstruction scope. Differences in observable execution states may arise from logging structures, reporting artifacts, or communication visibility. Such variations may reflect differences in event definitions or extraction scope rather than limitations of the monitoring approach. Finally, the artifact was implemented and evaluated as a research prototype in a pre-production environment. No claims are made regarding production scalability, runtime characteristics, or system engineering performance. Further work is required to assess long-term organisational adoption, integration patterns, and scalability under production conditions.

7.13 Summary

The evaluation shows that earlier visibility in financial ERP collections can be achieved by consolidating lifecycle evidence and reconstructing execution behaviour at the case level. Process mining reconstructs and visualises invoice, reminder, and settlement events as execution paths, and the AI assistant explains these observable states in domain language within defined prompt constraints. Reduced diagnostic effort results primarily from improved information organization and decreased need for manual navigation across fragmented D365 FO screens. Users can more quickly understand why an invoice is overdue, whether reminders were issued correctly, and which lifecycle conditions require follow-up. Effectiveness depends on event completeness and communication visibility. Where lifecycle data is clearly recorded, explanations are more specific and interaction effort decreases. Prompt constraints further stabilise explanation quality by enforcing structured and evidence-based responses. Overall, the study validates a traceable and configuration-aware decision-support design that improves clarity and interpretability in accounts receivable collections while preserving human responsibility for final decisions.

8 Conclusion and Future Work

8.1 Conclusion

This thesis examined how forward-looking and explainable process mining can support earlier identification and structured interpretation of control-relevant situations in financial ERP systems. Using D365 FO as a representative enterprise platform, the study addressed the challenge of delayed visibility of control-relevant conditions in finance-related processes. Within the analysed accounts receivable collections process, the artifact highlighted control-relevant states that are observable before process completion. The AI assistant explained these states using the prepared dataset. The study further shows that analytical detection capability alone does not ensure reliable interpretation in ERP environments. Financial ERP processes exhibit outcomes shaped by configuration logic and embedded control rules. Without configuration awareness, indicator results may reflect configuration outcomes and be misinterpreted as execution-related issues. The results underline that configuration awareness is a necessary condition for meaningful analytical interpretation in ERP contexts. The AI-assisted approach complements established control practices by providing consolidated and explanation-oriented analytical representations. Rather than replacing existing procedures, the prototype supports prioritisation and structured investigation of control-relevant situations, reducing interpretive effort while preserving human validation responsibilities. Explainability mechanisms provide interpretable representations of analytical outcomes, enabling users to understand the process conditions associated with indicator states. This transparency supports traceability and interpretability, which are essential properties for analytical systems operating in ERP environments. From a methodological perspective, the thesis demonstrates how earlier visibility through consolidated evidence, explainable interpretation mechanisms, and ERP-specific constraints can be integrated within an analytical framework using a Design Science Research approach. From a practical perspective, the prototype illustrates how insights from the consolidated report can support finance users' diagnostic reasoning and investigation activities. Overall, the study confirms that consolidated lifecycle evidence and explanation-oriented interpretation can support earlier identification and structured investigation of control-relevant conditions.

8.2 Answers to the Research Questions

This section addresses the research questions formulated in Section 1.6 by synthesising the analytical, architectural, and evaluation findings of the study. RQ1 examined how forward-looking and configuration-aware process mining analysis, supported by AI-assisted explanation, can enhance the investigation and interpretation of review-relevant financial process states in ERP environments. The study shows that earlier visibility can be achieved through structured consolidation of invoice lifecycle data combined with configuration-aware prerequisite validation. Reconstructed D365 FO event logs provide transparency into partially observed execution states, including overdue conditions and missing collection activities. Evaluation of master-data prerequisites against configured collection mechanisms enables identification of invoices that cannot trigger the expected collection steps before the resulting error becomes visible in the invoice lifecycle. Proper interpretation requires distinguishing between issues caused by process execution and outcomes determined by system configuration. The AI-assisted explanation layer translates structured outputs into finance-domain language, supporting case assessment while maintaining direct linkage to ERP evidence. RQ1 is addressed by demonstrating that lifecycle visibility, configuration-aware validation, and AI-supported explanation together enhance interpretability without replacing human judgment. RQ2 examined how such a decision-support artifact can be implemented in D365 FO while maintaining governance, traceability, and accountability. The artifact applies a layered architecture separating transactional processing, event-log structuring and process mining analysis, and AI-based interpretation. All analytical outputs are based on system-recorded ERP data and clearly defined event-log structuring steps, while users retain responsibility for validation and corrective action. RQ2 is addressed by demonstrating that AI-assisted analytical interaction can be embedded into D365 FO through architectural separation and human-in-the-loop control, without compromising compliance or auditability. This study addressed the research questions by designing and evaluating a configuration-aware monitoring artifact grounded in reconstructed D365 FO lifecycle data. The results demonstrate that structured process mining visibility combined with AI-supported explanation can strengthen investigative clarity in financial ERP processes while remaining consistent with internal control and compliance requirements. Forward-looking support is achieved without predictive modelling, relying instead on structured lifecycle transparency and prerequisite assessment.

8.3 Contributions

This thesis contributes at the theoretical, methodological, and practical levels. From a theoretical perspective, the study refines understanding of forward-looking analytical assessment in configuration-driven D365 FO environment. The thesis shows that forward-looking assessment in D365 FO can be achieved through structured lifecycle consolidation and early identification of control-relevant conditions, without relying on probabilistic prediction. Early insight does not require statistical prediction. It can be achieved through structured analysis of current process states when event definitions and configuration rules are properly understood. The thesis contributes to explainable analytics in process mining contexts by demonstrating how consolidated lifecycle representations and bounded AI explanations can enhance interpretability and traceability in financial ERP collections. From a methodological perspective, the study demonstrates how Design Science Research can be applied to develop and evaluate a forward-looking analytical artifact. This includes structured event-log design, construction of execution-based indicators, and prototype-oriented evaluation. From a practical perspective, the thesis demonstrates how structured forward-looking indicators and explanation mechanisms can support finance users in identifying and interpreting control-relevant execution conditions within ERP-supported processes.

8.4 Future Work

Several limitations define directions for future research. First, the approach was evaluated only within the accounts receivable collections process in D365 FO. Future research could apply the same lifecycle consolidation and explainable analytics design to other finance processes, such as accounts payable, bank reconciliation, or intercompany settlements, to assess whether the approach extends beyond collections. Second, while system configuration effects were considered during data preparation and interpretation, future research may investigate more systematic and partially automated methods for handling configuration-driven outcomes. Developing structured techniques for distinguishing execution-related deviations from system behaviour remains an important challenge in ERP analytics. Third, future research may examine supervised predictive modelling approaches as an alternative analytical paradigm, enabling structured comparison with indicator-based assessment mechanisms. Fourth, future research may extend the architectural design of the analytical artifact. The present study employs a single AI-assisted interpretation component. However, enterprise environments may benefit from modular agent architectures in which distinct interpretation functions are separated according to analytical responsibility domains (e.g., deviation interpretation, configuration diagnostics, control validation). Recent research on AI-native enterprise systems suggests that ERP environments may increasingly incorporate specialised AI agents operating on structured business-process representations rather than monolithic analytical models (Yang et al., 2025). This architectural direction aligns with the separation of analytical computation, interpretation, and control validation mechanisms observed in this study. Future extensions could introduce modular AI agents with clearly separated roles, such as process-state explanation, deviation analysis, control validation, and user guidance. Separating these functions could improve scalability by distributing analytical responsibilities, reduce cognitive load by narrowing the scope of each interaction, and establish clearer governance boundaries between analytical interpretation and user support. Finally, additional empirical research is required to examine long-term organisational adoption and user interaction patterns. Studies involving finance professionals may analyse how AI-assisted process analytics influence decision behaviour, workload distribution, and integration with existing control practices.

8.5 Final Remarks

The study demonstrates that D365 FO data can be consolidated and transformed into an event-log structure suitable for process mining and analytical evaluation. A central outcome is the finding that analytical value depends not only on detection capability, but also on alignment with established control practices. In ERP environments, practical usefulness requires explanations grounded in process context and system states to ensure user trust and operational relevance. The prototype illustrates how structured data extraction, lifecycle consolidation, process reconstruction through process mining, and AI-assisted interpretation can be integrated into a user-oriented interaction model. This approach enables finance users to understand overdue situations, identify missing collection activities, and interpret process deviations without navigating fragmented technical tools. In control-oriented financial settings, structured explanations and consolidated insights reduce manual investigation effort while preserving human oversight. The findings further indicate that limitations of AI-assisted ERP analytics arise both from constraints in data representation and from variability in AI-generated interpretation. Explanation quality depends on event completeness, clarity of event definitions, and accurate reconstruction of process history. Future research may extend the approach to additional finance processes, refine event-log construction strategies, and explore deeper integration between process mining and AI-assisted interpretation.

References

Amershi, S., Weld, D. S., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S. T., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R., & Horvitz, E. (2019). Guidelines for human-AI interaction. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3290605.3300233

Berti, A., Park, G., Rafiei, M., & van der Aalst, W. M. P. (2023). Advancements and challenges in object-centric process mining. *arXiv*. https://arxiv.org/abs/2311.08795

Berti, A., & Qafari, M. (2023). Leveraging large language models for process mining. *arXiv*. https://arxiv.org/abs/2307.12701

Bouquet, C., Wright, C. J., & Nolan, J. (2026). *Match your AI strategy to your organization's reality*. Harvard Business Review, January–February 2026.

Di Francescomarino, C., Ghidini, C., Maggi, F. M., & Milani, F. (2018). *Predictive process monitoring methods: A survey.* Information Systems, 78, 6–15. https://doi.org/10.1016/j.is.2018.01.003

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv*. https://arxiv.org/abs/1702.08608

Granlund, M., & Malmi, T. (2002). Moderate impact of ERPS on management accounting: A lag or permanent outcome? *Management Accounting Research, 13*(3), 299–321. https://doi.org/10.1006/mare.2002.0189

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys, 51*(5), 1–42. https://doi.org/10.1145/3236009

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75–105.

Jans, M., Alles, M., & Vasarhelyi, M. A. (2014).
Process mining of event logs in auditing: Opportunities and challenges. *Accounting Horizons, 28*(1), 1–18. https://doi.org/10.2308/acch-50613

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023).
Pre-train, prompt, and predict: A survey of prompting methods in natural language processing. *ACM Computing Surveys, 55*(9), 1–35. https://doi.org/10.1145/3560815

Maggi, F. M., Di Francescomarino, C., Dumas, M., & Ghidini, C. (2014).
Predictive monitoring of business processes. In *Advanced Information Systems Engineering* (pp. 457–472). Springer. https://doi.org/10.1007/978-3-319-07881-6_31

Microsoft. (2024). *Power Automate Process Mining documentation*. Microsoft Learn. https://learn.microsoft.com

Microsoft. (2025). *Dynamics 365 Finance documentation.* Microsoft Learn. https://learn.microsoft.com/en-us/dynamics365/finance/

Nicolaou, A. I. (2004).
Firm performance effects in relation to the implementation and use of enterprise resource planning systems.
Journal of Information Systems, 18(2), 79–105.
https://doi.org/10.2308/jis.2004.18.2.79

Nittala, E. P. (2025). *AI-powered ERP process mining and optimization techniques for agile enterprise transformation*. American International Journal of Computer Science and Technology, 6(6), 15–24. https://doi.org/10.63282/3117-5481/AIJCST-V7I6P102

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007).
A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Rudin, C. (2019).
Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206–215. https://doi.org/10.1038/s42256-019-0048-x

Shneiderman, B. (2020).
Human-centered artificial intelligence: Reliable, safe & trustworthy. Oxford University Press.

Tax, N., Verenich, I., La Rosa, M., & Dumas, M. (2017).
Predictive business process monitoring with LSTM neural networks. In *Advanced*

Information Systems Engineering (pp. 477–492). Springer. https://doi.org/10.1007/978-3-319-59536-8_30
Teinemaa, I., Dumas, M., La Rosa, M., & Maggi, F. M. (2019).
Outcome-oriented predictive process monitoring. *ACM Transactions on Knowledge Discovery from Data, 13*(2), 1–31. https://doi.org/10.1145/3301300
van der Aalst, W. M. P. (2016).
Process mining: Data science in action (2nd ed.). Springer.
van der Aalst, W. M. P., Schonenberg, M. H., & Song, M. (2011).
Time prediction based on process mining. *Information Systems, 36*(2), 450–475.
https://doi.org/10.1016/j.is.2010.09.001
Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., & Zhou, D. (2022).
Chain-of-thought prompting elicits reasoning in large language models. *arXiv*.
https://arxiv.org/abs/2201.11903
Weinzierl, S., Zilker, S., Dunzer, S., & Matzner, M. (2024). *Machine learning in business process management: A systematic literature review*. Expert Systems with Applications, 253, 124181. https://doi.org/10.1016/j.eswa.2024.124181
Yang, H., Lin, L., She, Y., Liao, X., Wang, J., Zhang, R., Mo, Y., & Wang, C. (2025).
FinRobot: Generative business process AI agents for enterprise resource planning in finance [Preprint]. arXiv. https://arxiv.org/abs/2506.01423