Software Construction

RWTH AACHEN UNIVERSITY

The present work was submitted to
the Research Group
Software Construction

of the Faculty of Computer
Science

Bachelor Thesis

Structuring Risk and Mitigation Knowledge for LLM-Based Legacy System Modernization Analysis

Strukturierung von Risiko- und Gegenmaßnahmenwissen für die LLM-gestützte Analyse der Modernisierung von Legacy-Systemen

presented by

Christian Bick

Aachen, September 28, 2026

communicated by Prof. Dr. Horst Lichter

Examiner

Prof. Dr. rer. nat. Horst Lichter

Prof. Dr. rer. nat. Bernhard Rumpe

Supervisor

Ada Slupczynski, M.Sc. RWTH

Acknowledgment

I would like to sincerely thank my supervisor, Ada Slupczynski, M.Sc., for her guidance, helpful feedback, and continuous support throughout this thesis. I would also like to thank Prof. Dr. Lichter and Prof. Dr. Rumpe for reviewing this thesis. Furthermore, I am grateful to all experts who participated in the evaluation and contributed their time and valuable insights. Finally, I thank RWTH Aachen University for providing access to the academic literature and digital library resources essential to this research.

Christian Bick

Abstract

Legacy system modernization is a complex process involving technical, organizational, business and process uncertainties and risks. Knowledge about the associated risks and mitigation strategies is available across academic and grey literature, but is fragmented and often represented inconsistently. For the relationship focused analysis pursued in this thesis, a particular limitation is that relationships between risks and mitigations are frequently left implicit. This makes the knowledge difficult to systematically compare and reuse, limiting its use for LLM-based analysis.

To address the fragmented and weakly connected representation of modernization risk knowledge this thesis works by identifying risks and mitigation strategies of holistic legacy system modernization from academic and grey literature across all dimensions and by transforming the extracted knowledge into a structured, machine-readable and relationship-oriented representation that makes entities and their connections explicitly accessible and traceable for the analysis. This resulting knowledge base is implemented as an OKF representation and integrated into a prototype that provides LLM access and knowledge graph visualization.

The evaluation shows that external modernization knowledge improves LLM-based analysis compared with providing no additional context, as in the proof of concept evaluation, the mean Overall Score increased from 46.87 without additional context to 87.58 with Full Structured Context and 92.72 with Unstructured Full Context. Adding explicit relationships to the structured risk and mitigation context increased the mean Overall Score from 75.12 to 87.58. Therefore explicit relationships particularly improve the retrieval and reasoning over connected knowledge, including direct and indirect relationships between risks and mitigations. At the same time, the comparison shows that unstructured textual context can retain valuable information. The structured representation, however, provides stronger relational traceability and substantially lower runtime in the evaluated setting. An external evaluation with six experts further supports the usefulness of the complete structured context, particularly for relationship-based analysis.

The thesis therefore demonstrates how heterogeneous modernization risk knowledge can be transformed into a structured, traceable and executable representation for LLM analysis. The resulting knowledge model provides a foundation for more systematic and relationship-aware modernization risk analysis while retaining the contextual and evidential information required for human interpretation.

Contents

1. Introduction **1**
- 1.1. Motivation . . . 2
- 1.2. Problem . . . 3
- 1.3. Research Objective . . . 4
- 1.4. Research Questions . . . 5
- 1.5. Scientific and Practical Contribution . . . 5
- 1.6. Thesis Structure . . . 6

2. Background **7**
- 2.1. Legacy Systems . . . 7
- 2.2. Legacy System Modernization . . . 9
- 2.3. Modernization Strategies and Methods . . . 11
- 2.4. Risks, Challenges, and Causes . . . 12
- 2.5. Dimensions of Modernization Risks . . . 14
- 2.6. Risk Management in the Modernization Context . . . 15
- 2.7. Structured Knowledge Representation . . . 17
- 2.8. Large Language Models . . . 18
- 2.9. Context-Based LLM Systems and RAG . . . 19

3. Related Work and Research Gap **21**
- 3.1. Research on Legacy System Modernization Risks . . . 21
- 3.2. Research on Mitigation Strategies . . . 22
- 3.3. Existing Frameworks, Roadmaps and Strategy Selection . . . 23
- 3.4. Research Gap . . . 26

4. Research Method **31**
- 4.1. Research Design . . . 32
- 4.2. Planning and Conducting the Literature Search . . . 33
- 4.3. Knowledge Extraction and Modeling . . . 35
- 4.4. Software Artifact Development . . . 40
- 4.5. Validation and Evaluation Design . . . 40

5. Results of the Literature Analysis **47**
- 5.1. Overview of the Literature Corpus . . . 47
- 5.2. Overview of the Identified Risks . . . 48
- 5.3. Technical Risks . . . 48
- 5.4. Organizational Risks . . . 50

5.5. Business Risks . . . 51
5.6. Process Risks . . . 52
5.7. Cross-Dimensional Risks . . . 53
5.8. Typology of Mitigation Strategies . . . 55
5.9. Mapping Mitigation Strategies to Risks . . . 56
5.10. Risk Chains and Secondary Risks . . . 57
5.11. Evidence Base . . . 58
5.12. Gathered Knowledge . . . 59

6. Development of the LLM-Oriented Knowledge Model 61
6.1. Requirements for the Knowledge Representation . . . 61
6.2. Entity Model . . . 63
6.3. Risk Model . . . 64
6.4. Mitigation Model . . . 66
6.5. Generalized Relationship Model . . . 67
6.6. Relationship Types . . . 69
6.7. Evidence, Confidence, and Interpretation . . . 71
6.8. OKF-Based Serialization . . . 72
6.9. Prompt and Output Templates . . . 73
6.10. Derived Design Principles . . . 74

7. Prototype Implementation 75
7.1. System Architecture . . . 75
7.2. Conversion of the Extracted Data . . . 76
7.3. Python Data Models . . . 78
7.4. Parsing and Loading the Knowledge Base . . . 79
7.5. Construction of the LLM Context . . . 80
7.6. LLM API Integration . . . 81
7.7. Web Interface . . . 82
7.8. Knowledge Graph Visualization . . . 83
7.9. End-to-End Query Processing . . . 84
7.10. Reproducibility and Availability . . . 85

8. Validation and Evaluation 89
8.1. Validation and Evaluation Overview . . . 89
8.2. Technical and Knowledge Base Validation Results . . . 90
8.3. Knowledge Graph Validation Results . . . 91
8.4. Preliminary Validation . . . 91
8.5. Comparative Proof-of-Concept Evaluation . . . 92
8.6. Comparison of Relationship Dataset and Graph Variants . . . 96
8.7. External Expert Evaluation Results . . . 98
8.8. Summary of Evaluation Results . . . 103

9. Discussion **105**
9.1. Interpretation of the Results for RQ1 . . . 105
9.2. Interpretation of the Results for RQ2 . . . 106
9.3. Scientific Contribution . . . 109
9.4. Practical Implications . . . 110
9.5. Limitations of the Literature Base . . . 111
9.6. Limitations of the Knowledge Extraction . . . 112
9.7. Limitations of the Knowledge Model and Representation . . . 113
9.8. Limitations of the Technical Implementation . . . 114
9.9. Threats to Validity . . . 115

10. Conclusion and Future Work **119**
10.1. Summary . . . 119
10.2. Answers to the Research Questions . . . 120
10.3. Conclusion . . . 123
10.4. Future Work . . . 123

Declaration on the Use of AI Tools **127**

A. Literature Corpus and Search Documentation **129**
A.1. Complete Extraction Corpus . . . 129
A.2. Search Strategy and Search Strings . . . 137
A.3. Screening and Selection Information . . . 137

B. Extraction Templates **139**
B.1. Risk Template . . . 139
B.2. Mitigation Template . . . 140
B.3. Relationship Template . . . 142
B.4. Example of a Filled-Out Raw Extraction Entry . . . 143

C. OKF-Based Representation Example **145**
C.1. Risk Object: `R1.md` . . . 145
C.2. Mitigation Object: `M1.md` . . . 146
C.3. Risk–Mitigation Relationship: `Rel3.md` . . . 148
C.4. Interpretation of the Example . . . 148

D. Evaluation Materials and Detailed Results **151**
D.1. Functional Validation Questions . . . 151
D.2. Comparative Proof-of-Concept Evaluation Tasks . . . 156
D.3. External Expert Evaluation Tasks . . . 160
D.4. Expert Questionnaire and Rating Criteria . . . 161
D.5. Detailed Expert Ratings . . . 163

E. Additional Knowledge Graph Results **165**
E.1. High-Confidence Relationship Graph . . . 165

E.2. Extended High- and Medium-Confidence Relationship Graph 168

F. Repository and Prototype Usage **171**
F.1. Repository Structure 171
F.2. Installation 171
F.3. Configuration 172
F.4. Running the Prototype 172
F.5. Deployed Prototype 173
F.6. Reproducing Graphs and Evaluation 173

Bibliography **175**

List of Tables

3.1. Example of an explicit risk–mitigation mapping, adapted from Riihimäki [Rii23]. 24

4.1. Validation scoring scheme. 41

8.1. Overall Score results across the four context configurations. 93
8.2. Mean identifier retrieval metrics across the four context configurations. . . 94
8.3. Runtime of the complete comparative evaluation runs. 96
8.4. Aggregated expert ratings across the four context configurations. 99
8.5. Expert ratings by evaluation criterion across the four context configurations. 100

A.1. Scientific-literature extraction corpus 130
A.2. Grey-literature extraction corpus 134

D.1. Detailed results of the 20-question functional validation. 154
D.2. Summary of the final comparative proof-of-concept evaluation. 158
D.3. Question-level Overall Scores across the four context configurations. . . . 159
D.4. Individual expert ratings for Full Structured Context. 163
D.5. Individual expert ratings for Structured Context without Relationships. . 163
D.6. Individual expert ratings for No Context. 163
D.7. Individual expert ratings for Unstructured Context. 164
D.8. Individual optional general program ratings. 164

List of Figures

2.1. Evolution of system functionality over time through maintenance, modernization and system replacement relative to changing business needs by Comella-Dorda et al. [Com+00]. 8
2.2. Conceptual relationship between challenges or root causes, trigger events, risks, impacts, and mitigation. 13

3.1. CMMI Compliant Modernization Framework and its modernization roadmap, adapted from Khan et al. [Kha+21]. 25

4.1. Illustration of the method pipeline used throughout this thesis. 31

5.1. Example of a verified cross-dimensional Risk chain represented by two explicit Trigger Relationships. 57

6.1. Conceptual overview of the three main knowledge-object types and their core information. The complete field sets are described in the following sections and in the extraction templates. 64
6.2. Simplified examples of Relationships between the main knowledge entities. 68

7.1. Illustration of full pipeline architecture developed in this thesis. 77
7.2. Illustration of the complete end-to-end query processing pipeline. 85

8.1. Overview of the structural change between the conservative and extended Relationship datasets. The extended dataset integrates substantially more Risk and Mitigation entities into the dominant connected component. Full-scale clustered and unclustered visualizations are provided in Appendix E. 97
8.2. Selected quantitative differences between Structured Context without Relationships and Full Structured Context. Recall and F1 originate from the developer-conducted proof-of-concept evaluation, whereas Indirect Relationship Reasoning represents the corresponding mean expert rating. . . . 104

E.1. Knowledge graph of the conservative High-Confidence Relationship dataset without clustering. 166
E.2. Clustered knowledge graph of the conservative High-Confidence Relationship dataset. 167
E.3. Knowledge graph of the extended High- and Medium-Confidence Relationship dataset without clustering. 169

E.4. Clustered knowledge graph of the extended High- and Medium-Confidence Relationship dataset. 170

1. Introduction

Legacy systems remain essential for critical organizational operations [Kha+14]. They still play an important role in many organizations because they execute business-critical functionality that cannot simply be replaced or removed. At the same time, their maintenance and continued evolution become increasingly difficult and costly as technologies, business requirements and surrounding systems generally change. This creates a growing need for modernization. However, modernization itself is not a straightforward technical replacement process, but a complex and uncertain transformation that can and will affect several parts of an organization. The associated risks therefore extend beyond technology and also include organizational, business and process dimensions.

Dealing with these risks requires knowledge about both the risks themselves and suitable mitigation strategies. Such knowledge is available in academic literature, industry reports, case studies or other forms of grey literature. However it is fragmented and represented inconsistently across different sources. Risks and mitigation strategies are often discussed independently, all while their relationships are only described implicitly. The same applies to relevant contextual information such as root causes, triggers, consequences, applicability conditions, trade-offs and secondary risks introduced by mitigations. As a result, existing modernization knowledge is difficult to compare systematically and cannot easily be reused across different modernization scenarios in different organizations.

This becomes particularly relevant when the knowledge is intended to support data analysis with Large Language Models (LLMs). While LLMs provide a flexible interface for querying and reasoning over basic textual information, their answers should not rely solely on knowledge embedded in the model. Especially in risk analysis, it should remain possible to trace an assessment back to the knowledge on which it is based, ultimately, to the corresponding literature source. Simply collecting modernization literature as unstructured text therefore does not address the problem. The knowledge also needs to be structured in a way that preserves relevant entities, relationships, context and source information while remaining usable as input for an LLM.

This thesis addresses this issue in two main steps. First, existing knowledge on modernization risks and mitigation strategies is extracted from academic and grey literature and consolidated across technical, organizational, business and process dimensions. In addition to individual risks and mitigations, the extraction explicitly considers their relationships as well as information on causes, triggers, impacts, contextual constraints, trade-offs and supporting evidence. Second, the resulting knowledge is transformed into a structured, machine-readable and relationship-oriented representation that can then be used to evaluate its use with LLMs.

Based on this representation, the thesis further develops a prototype that provides

access to the extracted structured knowledge with the help of different models. The prototype is not intended to autonomously make modernization decisions, but to examine how structured modernization knowledge can support traceable analysis. Different context configurations are considered in order to compare how different context information affects generated answers. The extracted relationships are additionally represented as a knowledge graph, allowing the structure and connectivity of the resulting knowledge base to be analyzed separately from its use within the prototype, also used for further visualization.

This chapter in particular introduces the background and motivation of the thesis, outlines the underlying problem and research objectives and presents the main contributions. It covers research questions, defines the scope and concludes with an overview of the thesis structure. The subsequent chapters provide the theoretical foundations, examine related work, describe the literature-based knowledge extraction and resulting risk base, develop the relationship-oriented knowledge representation, present its prototypical implementation and evaluate both the representation and its use for LLM-based modernization analysis.

1.1. Motivation

One of the key aspects of legacy systems is their constant relevance. Legacy systems often rely on aging technologies and architectures while continuing to perform business-critical functions [Kha+14]. Because these systems implement and execute essential business logic, they cannot simply be discarded [Kha+13]. This can be applied for a lot of sectors such as banking, healthcare, higher education, insurance, and government services, where even in 2026 critical operations still depend on legacy mainframe environments and systems developed using the programming language COBOL [Rah26].

Referring back to a study from 2014, Khadka et al. [Kha+14] found out that technology consulting firms estimate around 180–200 billion lines of legacy code still being in active use. All of that despite the issues of legacy systems and acclaimed benefits of legacy modernization affecting daily business [Bat+14].

Those issues include various domains such as financial, security or process aspects. For example, the US government alone spent more than USD 100 billion in fiscal year 2023 on IT, of which about 80% was used to operate and maintain legacy systems [US 21; Ass+25]. Just looking at their personal federal systems, most parts were 8 to 51 years old and collectively cost approximately USD 337 million annually just to maintain and keep operational. Taking a broader spectrum into account ranging from big global players to small local businesses, every organization will allocate 40% of their IT budgets solely to maintaining technical debt if not modernized at a given point [SAP25]. Delaying only increases costs.

Beyond the financial burden associated with legacy systems, continued operation can also create substantial security risks. Hogan et al. [Hog+24], for example, report that legacy software used in medical devices may lack basic security features and operate on systems with publicly known vulnerabilities, hence revealing classified and critical

patient data to a potential large group of people. For example back in 2024, a legacy application without multi-factor authentication contributed to Microsoft's exposure to a password spray attack [SAP25].

Increasing maintenance costs, technical debt, security exposure, and limited adaptability create growing pressure on organizations to modernize legacy systems. According to companies that have modernized in the past, 95% of respondents believe that application modernization is essential for their organization's success [Red24]. Economic benefits have also been reported. IBM, for example, reports substantial returns on investment for different mainframe modernization approaches, ranging from 288% for app modernization on a mainframe to 362% for projects that move workloads from a mainframe onto other platforms [IBMe]. More than half of organizations that are currently modernizing have experienced benefits around security (58%), scalability (53%), and reliability (52%) [Red24].

However, those benefits do not come without challenges and risks. GAO added IT Acquisition and Operations to its High Risk List in 2015 because "federal IT investments too frequently fail to be completed or incur cost overruns and schedule slippages while contributing little to mission-related outcomes." [US 21]. Legacy modernization breaks business logic. Not sometimes – routinely [AG26] which creates a kind of double pressure. Continuing to operate legacy systems carries increasing technical and business exposure, while modernization itself introduces substantial transformation risk for companies [Hog+24; Lin25].

Furthermore, modernization is not something that only has to be done once. According to Bennett and Rajlich it is seductive to think that current technology developments, such as components, middleware, enterprise computing and so on will provide the ultimate answer to all problems, and once software applications are expressed in this form, there will be no more legacy software. Modernization is a recurring challenge. This shows that this field of research is never really finished and in desperate need for optimization.

One of those possibilities has just emerged recently: Large Language Models (LLMs). The surge of interest in those models has reached unprecedented levels of magnitude in recent years [De 23]. For this thesis, the relevant opportunity is not general code generation as a modernization tool, but using LLMs as an interface for accessing and reasoning over consolidated modernization knowledge.

1.2. Problem

However, this knowledge is distributed across heterogeneous academic and industrial sources and is represented inconsistently [Hog+24; FAA25]. Risks and mitigation strategies are often described independently while relationships involving root causes, triggers, consequences, dependencies, applicability conditions, trade-offs, and secondary risks remain implicit. The absence of a consistent relationship-oriented representation limits comparison across sources and complicates systematic reuse for subsequent analysis tools of any kind.

Furthermore, to understand what strategies and methods work you need to look at

failure cases, not only successful modernizations. Evidence from unsuccessful modernization initiatives is particularly relevant for identifying risks, consequences, and limitations of mitigation approaches. As Khadka et al. [Kha+15] said "An empirical study of failure cases is invaluable to fully understand modernization impacts". For this reason, the literature analysis in this thesis considers reported failure cases in addition to successful modernization initiatives.

In most cases there is a practical need for systematic modernization planning but sometimes it is even an obligation. In 2022 for example, Senator Maggie Hassan introduced the Legacy IT Reduction Act of 2022 that required agencies to develop an inventory of legacy IT systems as well as create a plan to modernize these systems [MIT24].

Large Language Models provide a potential interface for accessing and reasoning over consolidated modernization knowledge. However, relying solely on their internal knowledge is problematic for risk analysis because generated responses may contain unsupported or incorrect information [WT24]. Grounding and source traceability are therefore important requirements for LLM-based modernization analysis. At the same time, providing heterogeneous literature directly as textual context does not preserve the explicit structure and relationships required for systematic analysis. This creates a need for a structured and source-traceable representation of modernization risk knowledge.

1.3. Research Objective

The objective of this thesis is to support analysis of risks associated with holistic legacy system modernization by consolidating existing risk and mitigation knowledge and representing it in a form suitable for LLM-based use. It aims to provide IT decision makers with a tool that supports structured, transparent and traceable decisions in the legacy system modernization process in the future [Lan]. The process of going through literature therefore has to be done once only and not by every small company that does not possess the financial resources to do it themselves. Software modernization needs to be addressed as an interdisciplinary transition which is automatically included in the thesis.

To capture practical modernization risks and mitigation experiences, this thesis considers evidence from both academic literature and grey literature, including reported failure cases and contemporary industry cases. These sources are used to extract risks, mitigation strategies, and their explicit or implicit relationships across different modernization contexts. In general, the use of real-life cases with the help of grey literature adds to the holistic picture considering many different modernization scenarios. The research objective is deliberately two-stage: consolidating risks and mitigations from the literature, while transforming this knowledge into a structured, machine-readable and relationship-oriented representation that then can be used to evaluate its use with LLMs.

The resulting prototype is intended as a research-oriented decision-support artifact for knowledge access and exploratory analysis. It does not constitute a full operational risk management system, predict project-specific risk probabilities, replace professional risk assessment, or make autonomous modernization decisions.

As 63.64% of publications correspond to solution proposals, which suggests a lack of empirical studies trying to understand the benefits of using the existing approaches for software modernization, this thesis tries to implement a more practical step rather than developing just one general theoretical solution more.

1.4. Research Questions

To guide the research process and give a general aim for what has to be covered, two research questions were established at the very start:

RQ1 What are the key risks and their mitigation strategies associated with holistic legacy system modernization across technical, organizational, business, and process dimensions?

RQ2 How should modernization risk knowledge be structured and formatted to maximize its utility for Large Language Models?

RQ1 is addressed primarily through the structured literature analysis and presented in Chapter 5. It covers all the different aspects that concern themselves with knowledge gathering and what is even presented in current literature.

RQ2 on the other hand covers everything in the direction of knowledge structuring and is primarily addressed later on in Chapter 6 and Chapter 7 through the design of the knowledge model, serialization format, standardized entries, logic patterns, prompt templates, and prototype evaluation. Utility can subsequently be evaluated through correctness, completeness, groundedness, relationship coverage, ID referencing, format quality, hallucination rate and runtime.

1.5. Scientific and Practical Contribution

There is a general need for broadly applicable modernization knowledge. The complexity and diversity of legacy systems means that researchers struggle to find industry-wide applicable methodologies and solutions in an academic setting [Hog+24]. This means that every team of researchers has to go through the same extraction phase with similar results but wasted resources. This again is needed in academia as well as industry as legacy systems cannot simply be removed as they implement and execute critical business logic. At the same time there is a need for relationship-oriented risk analysis as just listing risks during modernization and their mitigation possibilities is insufficient. The dependencies between these risks must be identified and assessed. Those measurements, however, are highly labor intensive and too demanding for most organizations [Sof]. The thesis therefore does all of that to cover the academic part as well as the analysis part for modern industry and in general simplification.

This contribution aims to cover a couple of important aspects. It presents a consolidated overview of relevant modernization risks, a structured collection of associated

mitigation strategies and a relationship-oriented knowledge model. All of that is represented in a representation format suitable for LLMs so that the prototypical access via an LLM interface becomes possible. It also covers a visual representation of the knowledge relationships by using the underlying knowledge graph structure and evaluates all of those parts in a technical and expert-based evaluation. The contribution is therefore not only a catalogue of entities, but it also preserves relationships, context, evidence, confidence, and source traceability so that generated analyses can be inspected and challenged.

1.6. Thesis Structure

Firstly this Chapter 1 introduces the topic, motivation, problem, objectives, research questions, scope and contribution on a very general level. In the next chapter, Chapter 2, I will cover the theoretical foundations on legacy systems, modernization, risk management, knowledge representation, LLMs and RAG. Chapter 3 reviews related work and identifies the research gap followed up by Chapter 4 which describes the literature review, Citavi-based coding, knowledge extraction, artifact development, and evaluation design. After this, Chapter 5 presents the identified risks, mitigations, relationships, and risk chains and answers RQ1 as a checkpoint step which Chapter 6 then uses to develop the LLM-oriented knowledge model and answers RQ2. The complete implementation and testing of the framework is formalized in Chapter 7 which describes the technical implementation, including the parser, LLM interface, Streamlit application as well as knowledge graph representation and Chapter 8 which presents the technical validation, functional evaluation, context comparison and final expert evaluation. Before finishing the thesis with a summary, answers of the research questions and future work in Chapter 10, I will discuss the findings, contributions, limitations, and threats to validity in Chapter 9.

2. Background

This chapter establishes the theoretical background for the thesis. It first introduces legacy systems, their characteristics, and the motivation for modernization, followed by common modernization strategies and methods. Subsequently, relevant risk concepts, modernization risk dimensions, and principles of risk management are discussed. Finally, structured knowledge representation, Large Language Models, and context-based LLM systems are introduced to establish the conceptual context for knowledge-based LLM analysis.

2.1. Legacy Systems

A clear definition of legacy systems is required before discussing their modernization. According to Khadka et al. [Kha+14] a legacy system is any business-critical software system that significantly resists modification and whose failure can have a serious impact on the business. It is a highly complex socio-technical system made up of many different parts of which legacy software is merely one part, while people, expertise, hardware, data, business processes, business rules and approaches to software maintenance and development are additional parts. All these parts are subject to change [Lan]. Legacy systems are often characterized as long-established but also core systems, frequently supporting back-office functions while being rigid and inflexible [Kha+14]. In many cases, they no longer fully meet intended performance or adaptability requirements, which is associated with adaptation inertia [Hog+24; IBMb]. Khan et al. [Kha+21] additionally define an LS as not upgradeable. Batlajery et al. [Bat+14] characterize legacy systems as systems and technologies that no longer align with an organization's strategic technology goals. Legacy status should not be reduced to system age alone: business criticality, resistance to change, obsolete technology, and misalignment with current strategic needs jointly characterize the legacy problem.

Legacy systems typically emerge gradually throughout the lifecycle of a software system. Throughout the life of a software system, its architecture decays, its underlying technologies become obsolete, the user requirements change, or the company's business models evolve – ultimately, causing the software to morph into what is defined above as a legacy system [Ass+24]. One characteristic of such a system is its immense maintenance costs, where failure to take remedial changes gradually makes them costly to operate and maintain. These maintenance difficulties tend to increase over time [Kha+15]. Eventually, the existing information systems become too fragile to modify and too important to discard [Com+00]. An important finding is that systems represent their corresponding organisation when it comes to structure [Kha+21]. They often are large because the

organizations they support are large and complex, and because they evolve through accretion [GS14]. As time goes on, systems change with their organization and consequently, more software systems are turning into legacy systems. As illustrated in Figure 2.1, the

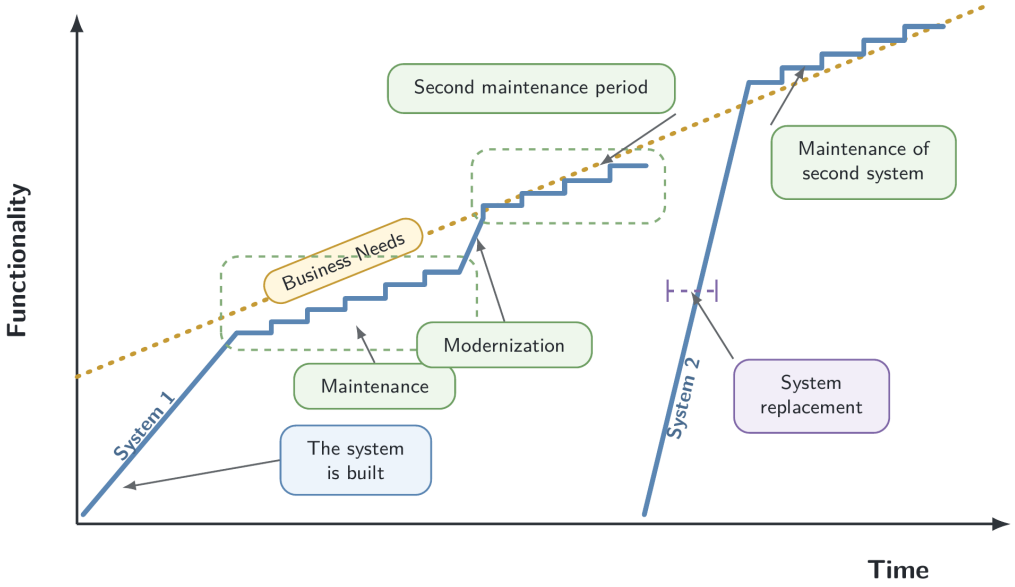


Figure 2.1.: Evolution of system functionality over time through maintenance, modernization and system replacement relative to changing business needs by Comella-Dorda et al. [Com+00].

functionality of a software system evolves through recurring maintenance activities and modernization efforts. Over time, increasing business needs may eventually require the replacement of the existing system by a new one. Finally, every IT system will sooner or later lose effectivity and efficiency when reflecting the business needs [Lan].

Considering the technical characteristics in more detail, legacy systems support core business processes of a business and their failure can have a significant impact on an organization [Kha+14]. They are perceived as reliable systems in industry because of the fact that they have been in production for years and possible bugs and errors have already been fixed [Bat+14]. Today's legacy applications tend to be either third- or fourth-generation language systems [Lan] which are often built on monolithic architectures [IBMb]. One concept that helps explain increasing maintenance difficulty is technical debt. Technical debt refers to the cumulative costs and long-term consequences of using practical, short-term solutions in software development rather than choosing the best overall solution according to Ponnusamy and Eswararaj [PE23]. Compared with systems based on microservices or service-oriented architectures, legacy systems typically lack clear boundaries between functional and technical areas [Sof]. This leads to them also being brittle and inflexible to change, combining this with incomplete or outdated documentation and difficulty to extend and integrate with other systems which can

substantially hinder maintenance, integration, and further evolution [Gho+17]. These limitations do not necessarily imply poor original design, rather, many systems were developed according to standards and technology assumptions that have since become outdated. Legacy systems like AS400 (now IBM i) were designed long before cloud, APIs, or SaaS (Software-as-a-Service) became the norm. As a result, they lack compatibility with modern design practices and technologies [Lan]. This inability to integrate modern development practices, such as continuous deployment and automated testing, then further exacerbates inefficiencies [Ogu+23]. A big part of those monolithic and old systems is that they operate in so called "silos", completely preventing seamless data sharing across departments and external partners needed for today's digital economy [Ogu+23].

Beyond these technical characteristics, legacy systems also present maintenance problems, security issues, knowledge deprecation and operational limitations. Prior work has shown that the cost of operating and maintaining legacy systems increases over time and even worse, they may operate with known security vulnerabilities that are either technically difficult or prohibitively expensive to address [US 21]. Many such systems were built before modern cybersecurity threats became widespread, leaving them susceptible to data breaches, ransomware attacks, and compliance violations [Ogu+23]. Another problem is outdated programming languages, not because of the performance issues, but knowledge about those legacy systems is scarce as the original programmers leave the company or retire and up-to-date documentation is usually lacking [Bat+14]. Additionally, Ponnusamy and Eswararaj found that legacy systems tend to consume more power and physical space compared to their modern counterparts. Legacy systems may experience outages or slowdowns that disrupt business operations or customer experience, limiting an organization's ability to adapt to evolving market demands [Lin25; Ogu+23]. As customer expectations are evolving rapidly with the market this combination can lead to a disjointed customer experience and the business is limited in its ability to scale and may fail to meet those evolving needs [SAP25; SAPb]. To sum up, legacy systems are just composed of extensive maintenance and obsolete technology, costly to maintain, more exposed to cybersecurity risks, less effective in meeting their intended purpose, and simply push up costs [Ass+25; Gho+17]. Taken together, these characteristics explain why continued maintenance may eventually no longer be sufficient and why organizations consider more substantial modernization.

2.2. Legacy System Modernization

Having outlined the characteristics of legacy systems, the next step is to clarify what is meant by modernization. Modernization can be defined as the evolution of systems towards new business requirements of the organizations, involving new functionalities, error correction or technological updates [AAC16]. It is also a transformation process in which the software system is adapted or refactored to meet new user requirements and adapt to completely new market technologies [FAA25]. In general, modernization involves more extensive changes than maintenance, but conserves a significant portion of the existing

system [Com+00]. The software modernization part alone involves re-engineering or migrating entire legacy systems [Kha+14; Ass+25] while always considering the mainframe system or distributed platform where that code resides and runs, and the tech stack with which it interacts [IBMd]. Modernization of the complete system in its entirety is not only the technical part but it is also the adoption within an organization. It should therefore be distinguished from routine maintenance, involving a more substantial change to technology, architecture, platform, or operating context while in some strategies preserving parts of the existing system and its business knowledge [Com+00; FAA25].

Modernization is motivated both by pressures associated with continued legacy operation and by expected future benefits. One of these pressures is technical debt as managers do not want to accumulate even more debt and realize that modernization can become the more cost efficient step [PE23]. Modernizing legacy systems is critical for organizations striving to enhance operational efficiency, scalability, and security in an increasingly data-driven and digital landscape [Ogu+23; IBMa]. Many organizations eventually face the need to modernize legacy systems because of rapid technological advancements and evolving market demands [Fraa; SAPb]. Organizations must adapt to various changes, including intra-organizational changes, changes in laws and regulations, in business collaboration (mergers and acquisitions), and faster time-to-market strategies [Kha+14]. Beyond time-to-market pressures, one of the main drivers identified in literature is the very high maintenance costs of legacy systems [Ass+25]. The decision to migrate workloads to the cloud for example often starts with a desire to minimize not only the total cost of ownership, but also opportunity costs [AI23; Ins]. Connected to the previous old programming languages is a lack of knowledge, particularly scarcity of experts, unavailability of (up-to-date) documentation and limited number of suppliers as drivers including a "prone to failure" system [Kha+14]. At some point in the lifecycle of a software system, there are almost always situations that make a fundamental modernization of the system or of an entire system landscape necessary [Fraa]. In the case where companies delay modernization, they increase their exposure to risks that are expensive to fix, especially those that involve data breaches, non-compliance with regulations and customer churn [Lin25].

The objectives and intended benefits of modernization are closely related to these drivers. Common objectives include reducing operational costs and complexity, improving interoperability, performance, scalability, flexibility and data quality, and enabling faster innovation and time to market [Ass+25; Lan]. More generally speaking, the focus on software modernization is to preserve knowledge represented in legacy systems while employing disruptive and emerging technologies to the benefit of users, companies, and society. These objectives also include enhanced security and compliance [Hog+24]. All of this can contribute to a competitive advantage. In today's rapidly evolving digital environment, maintaining a competitive advantage is critical according to Hogan et al. [Hog+24]. Legacy modernization should enable the IT system to meet all business needs [Lan]. It is an opportunity to add skills and capabilities to employees, making them more competitive and able to contribute better to critical operations.

Modernization again has to be seen and tackled as a holistic decision. The suitability of an overall replacement strategy depends not only on technical factors but also on the entire corporate environment [Ass+25]. Hence a lot of factors have to be taken into account like inherent risks, the criticality of the system, the associated costs, and the system's operational performance as decisive parameters before modernization [US 21]. When finally deciding to modernize a legacy system, a team considers factors like the degree to which core mission functions of the agency or other agencies are dependent on the system. Affecting factors also include potential of legacy systems for migration, strategy of migration, SOA governance, business process and budgeting [GS14]. The combination of system criticality, migration potential, governance, business-process dependencies, costs and budgeting makes legacy modernization complex, risky and expensive [Lan]. Modernization happens at a point in a system's lifespan where risk is unavoidable, where leaving a system as-is presents several risks while embarking on a modernization effort also presents high levels of risk [Hog+24]. Modernization decisions therefore compare two risk positions rather than a risk-free and a risky option: the risks of continued legacy operation versus the risks introduced by transformation. The way this transformation is carried out can differ substantially, which motivates a closer look at common modernization approaches.

2.3. Modernization Strategies and Methods

Modernization can be implemented through different transformation and rollout approaches. Literature uses different taxonomies to describe these choices. R-models describe disposition or transformation options, while big-bang versus incremental describes roll-out sequencing and top-down versus bottom-up describes the direction of analysis or decomposition [Micb]. The 6Rs framework, for example, distinguishes Rehost, Replatform, Refactor, Repurchase, Retain, and Retire [Micc]. Rehost moves applications to the cloud as-is. Replatform updates the infrastructure using managed services without changing core code. Refactor rewrites the code for modern cloud-native features. Repurchase replaces custom software with a ready-made SaaS product. Retain keeps the application in its current state for now. Retire shuts down and deletes unused systems [CIOa]. It is important to note that these decision dimensions can be combined rather than treated as one mutually exclusive spectrum. Another definition of possible methods by Khadka et al. [Kha+14] compactly covers software modernization as the process of evolving existing software systems by replacing, re-developing, reusing, or migrating the software components and platforms. Microsoft, on the contrary, uses a more detailed version with Retired, Retained, Rehosted, Replatformed, Refactored, Rearchitected, Rebuilt, or Replaced. Because the literature uses different R-taxonomies, terminology therefore varies between sources rather than following one universally standardized R-model. There are many proposed possibilities in different papers but it always comes down to one decision for a company: To upkeep a software, companies have two choices: either create new software to replace the old or migrate the existing system into a new architecture [RS16]. This can be done in numerous sub-steps like cloudification of legacy

systems, architecture redesign, moving to a new programming language, targeting reuse optimization, software modernization for new hardware integration, practices to leverage automation, database modernization, and digital transformation [Ass+25].

Another distinction concerns the degree to which the internal structure of the legacy system is considered during modernization, specifically white- and black-box modernization. Legacy systems may be modernized at the functional (logic), data, or user interface level. White-box modernization requires an initial reverse engineering process to gain an understanding of the internal system operation while black-box modernization involves examining the inputs and outputs of a legacy system within an operating context to gain an understanding of the system interfaces [Com+00]. Therefore white-box approaches are based on the knowledge of the internal structure and program code and modernize on a deeper level with logic and underlying frameworks [Lan]. In general, the black-box approach only considers the behavior of the system or functionalities based on input and output. The internal structure is of no interest and the usability for the end-user is important. This scenario in particular introduced the need for adapters to provide a communication mechanism between the legacy system and the modernized system [Sea+01].

Beyond the technical transformation choice, modernization requires explicit planning and consideration of organizational implications. Modernization plans for legacy systems that, at a minimum, include three key elements [Mica]:

1. **Modernization Milestones** — Defined milestones for the successful completion of the modernization effort.

2. **Required Modernization Work** — A description of the activities and work necessary to modernize the legacy system.

3. **Legacy System Disposition** — Details on the intended disposition of the legacy system following modernization.

These planning elements also show that modernization has to deal with business processes, information flow, people, corporate structures and corporate culture and therefore is a holistic problem [Lan]. Because modernization approaches differ in scope, sequencing, and organizational impact, they are accompanied by different forms of uncertainty and risk.

2.4. Risks, Challenges, and Causes

A conceptual distinction is needed when talking about similar concepts as modernization literature frequently uses terms such as challenges, risks, problems, barriers, and concerns with partly overlapping meanings. For analytical clarity, challenge, risk, root cause, trigger event, impact, and mitigation are therefore distinguished in the following. A single source passage can contain several of these concepts at once, which makes clear definitions important.

ISO 31000 defines risk as the effect of uncertainty on objectives [Int]. Risk is commonly characterized with reference to potential events, their consequences and most importantly the associated uncertainty or likelihood [22]. In the modernization context, a risk can therefore be understood as an uncertain event or development associated with potentially negative consequences for a legacy-system modernization objective. Risks may also propagate when their consequences contribute to further uncertain events [22].

Challenges are related but conceptually different. Existing modernization literature frequently uses Challenge to describe existing difficulties or unfavorable conditions rather than uncertain future events [Kha+14; Ass+25; Red24]. Examples include system complexity, insufficient tooling support, lack of modernization experience, organizational resistance and competing priorities [Ame17]. A challenge is therefore not automatically a risk. Instead, it can be treated as an existing difficulty or condition that creates, contributes to, or increases exposure to a risk.

Closely related to challenges are root causes. A root cause denotes the underlying condition, weakness, dependency, or problem from which a modernization risk originates. Examples can include insufficient documentation, tightly coupled dependencies, missing expertise, or inadequate planning. This distinction is consistent with risk identification, which includes identifying risk sources, events, causes and potential consequences [22]. A risk source represents an element that can give rise to risk individually or in combination with other elements.

A trigger event is related to the root cause but describes a different point in the causal sequence. A trigger event represents the specific action, event, or change in circumstances that activates or materializes a previously identified risk. It is a more discrete event compared to a root cause that can persist over a longer period. This is consistent with ISO terminology, which distinguishes the underlying source of a risk from the event through which particular circumstances occur or change [22]. The distinction therefore separates why a risk exists from what causes it to materialize in a concrete modernization situation.

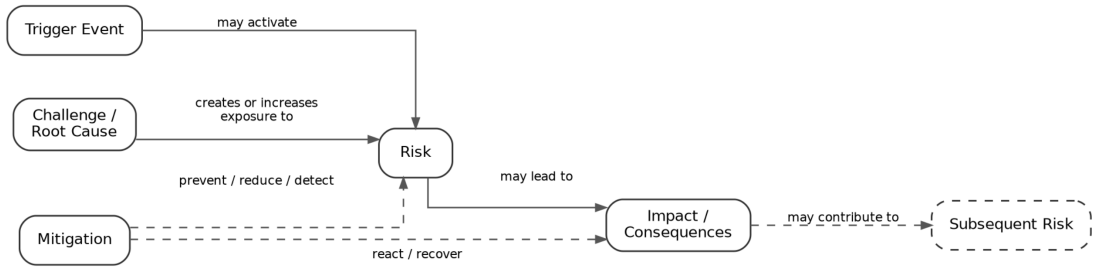


Figure 2.2.: Conceptual relationship between challenges or root causes, trigger events, risks, impacts, and mitigation.

As summarized in Figure 2.2, risks are associated with potential consequences resulting from events or changes in circumstances [22]. Impacts can affect technical system properties, organizational structures, business objectives or modernization processes and

may themselves contribute to subsequent risks. Consequently, the impact of one risk can become a condition or trigger for another risk, creating a risk chain.

Risk treatment refers to measures that modify risk, for example by avoiding a risk source or changing its likelihood or consequences [22]. Mitigations can therefore be understood as actions intended to prevent, reduce, detect, react to or recover from a modernization risk [Gho+17]. Risk treatment can also create new risks or modify existing risks [22]. This means that causes, trigger events, impacts, and mitigation effects need to be considered as related rather than isolated risk concepts.

These concepts can occur in different parts of a modernization initiative. A broader classification is therefore needed to distinguish the domains in which modernization risks originate and produce consequences.

2.5. Dimensions of Modernization Risks

The interdisciplinary nature of modernization is what substantially adds complexity to the process. Software evolution needs to be addressed as a business issue as well as a technology issue and is therefore fundamentally interdisciplinary [BR00]. Additionally, realizing that a large-scale migration is not only a technical and business endeavor but existing organizational and staff knowledge also needs to be considered as well as process aspects, you end up with a lot more than just new code [Kha+13]. Modernization has driving forces and impacts related to organizational, operational and business aspects that can equally hinder modernization and cause events described by risks [Ass+24]. Industrial practitioners similarly describe modernization challenges as technical, business, and organizational rather than exclusively technical [Bat+14].

Modernization risks can therefore be considered across technical, business, organizational, and process dimensions [CIOb]. These dimensions represent different perspectives from which modernization risks can originate and produce consequences. They should be understood as analytical categories rather than as mutually exclusive groups, because one risk can affect more than one dimension when its causes or consequences cross dimensional boundaries.

2.5.1. Technical Dimension

The technical dimension covers risks associated with software, source code, architecture, infrastructure, data, interfaces, dependencies, security, performance, and technical compatibility. Typical modernization concerns in this domain include legacy-system complexity, technical debt, tightly coupled dependencies, data migration, integration problems and limitations of existing technologies [Ass+25; Com+00; PE23; Red24; Sof].

2.5.2. Organizational Dimension

The organizational dimension covers risks related to people, knowledge, skills, responsibilities, communication, organizational structures, governance, culture and acceptance of modernization-related change [Bat+14]. Relevant concerns include knowledge loss,

skill shortages, organizational resistance, insufficient communication and dependency on individual experts [Ame17; CIOb].

2.5.3. Business Dimension

The business dimension covers risks affecting business objectives, costs, investment decisions, business value, operational continuity, strategic alignment, and other consequences for the organization using the modernized system. The importance of business implication dates back to early 2000 with Bennett and Rajlich [BR00] establishing that software evolution needs to be considered as both a business and a technology issue. Modernization decisions in general depend not only on technical factors but also on the wider corporate environment, costs, criticality and business requirements [US 21; Lan]. Relevant concerns include cost overruns, uncertain return on investment, disruption of business-critical functionality and insufficient business value [US 21; IBMc].

2.5.4. Process Dimension

The process dimension covers risks associated with the planning, coordination, execution, control, testing, monitoring, and sequencing of modernization activities [Ste]. Existing modernization research identifies process-related concerns such as insufficient tooling support, inadequate modernization processes and planning, inappropriate strategy selection, and insufficient evaluation mechanisms [Ass+25]. Practical modernization guidance similarly emphasizes structured and phased planning, monitoring, and change management, while also highlighting unclear requirements, insufficient testing, schedule problems, and incomplete migration or decommissioning processes as relevant concerns [CIOb; Ste].

2.5.5. Cross-Dimensional Risks

These four dimensions should not be interpreted as isolated risk groups as modernization challenges have been reported as interrelated [Red24]. Existing research identifies an intricate interdependence between technical, financial, organizational, and strategic modernization factors [Paz25]. A risk originating in one dimension can consequently produce impacts or trigger subsequent risks in another dimension. These cross-dimensional dependencies are therefore relevant for a holistic understanding of modernization risk.

Once risks are distinguished across these dimensions, the next question concerns how they can be systematically identified, assessed, and treated.

2.6. Risk Management in the Modernization Context

Identifying different risk dimensions alone does not address how such risks are handled. Risk management is a process in which potential risks and uncertainties that could affect a company's activities are identified, assessed, and controlled [acq]. It is fundamental for an organization that wants to become and remain successful to manage the risks they will encounter every day. Modern businesses tend to implement management systems

to govern one or more critical aspects of their operations while some companies dedicate complete positions to this job [DNV]. Generally speaking every organization needs a plan which provides a structured approach to identifying and assessing risks, supporting informed decision making and contributing to business risk management [DNV]. Effective risk management processes also provide valuable insights into the potential impacts of different business decisions [IBMc].

There are several practical implementations in the industry. International Organization for Standardization [Int] provides direction on how companies can integrate risk-based decision making into an organization's governance, planning, management, reporting, policies, values and culture. It is an open, principles-based system, meaning it enables organizations to apply the principles in the standard to the organizational context. Emphasis lies on both the involvement of senior management and the integration of risk management into the organization. This includes the recommendation to develop a statement or policy that confirms a commitment to risk management [Int]. One important rule found is the following: Be aware of your organization's key objectives – this will help you clarify the targets and requirements of your risk management system [22]. Risk management is not just useful for a company to establish, it is in some cases even mandatory. For example, the German Stock Corporation Act (AktG) requires management to carefully monitor risks and identify those types of risks that could threaten the company's continued existence [acq]. This stresses how important risk management really is.

General risk management literature distinguishes categories such as financial risk, operational risk, cybersecurity risk, strategic risk, compliance risk and reputational risk [IBMc]. Compliance risk is associated with the need to comply with laws, regulations and standards [DNV]. Risk avoidance involves taking steps to avoid a risk altogether even if this may prevent the business from gaining benefits. Risk treatment or risk avoidance does not necessarily remove uncertainty completely as residual risk, applicability conditions, and possible side effects remain relevant when judging whether a mitigation is suitable [Int].

Risk treatment can also alter the surrounding risk landscape and give rise to secondary risks. Power [Pow04], for example, discusses how an initial audit risk can be accompanied by financial and reputational risks for auditors themselves. More generally, this illustrates that controls and treatments can have consequences beyond the risk they were originally intended to address. At the level of risk management practice, this supports a view of risk management that also considers the effects of the controls themselves [Pow04].

All of this is relevant to legacy system modernization. All these characteristics of different risks and high dimensionality make the replacement of legacy systems complex, risky and expensive [Lan]. Risk management therefore depends on knowledge about risks, causes, consequences, treatments, and their inter-dependencies.

2.7. Structured Knowledge Representation

Structured knowledge representation concerns organizing information according to explicit structures that distinguish entities, attributes, and relationships. The central principle of schema-based extraction lies in its reliance on an explicit knowledge schema that provides both structural guidance and semantic constraints for the extraction process [Bia25]. Such structures support precision, logical consistency, and interpretability. Programming languages are inherently structured and are designed to represent complex data and relationships efficiently [WT24]. The main tasks of knowledge acquisition include relation extraction and other entity-oriented acquisition tasks such as entity recognition and entity alignment [Ji+22]. Knowledge-aware applications include natural language understanding (NLU), question answering, recommendation systems and other real-world tasks in which external knowledge can improve representation learning [Ji+22].

Beyond isolated entities, structured representations can explicitly encode relationships between concepts. To deal with complex multi-hop relations, more dedicated structures can support multi-hop reasoning by making relational information explicit [Ji+22]. A straightforward way to represent reasoning is through inference rules, composed of a body (if) and a head (then) [Hog+22].

There are several ways to serialize structured data. JSON is widely used for storing structured data, although it can be less convenient for representing complex multi-hop inference processes [WT24; Whi+23]. The choice of representation therefore depends not only on the serialization format itself, but also on whether entities, attributes, relationships, provenance, and uncertainty remain explicit and distinguishable for machine processing and human inspection.

One possible format for storing structured knowledge in a human- and machine-readable form is the Open Knowledge Format (OKF). OKF is an open and vendor-neutral format for representing knowledge as collections of Markdown files with YAML frontmatter [MH26]. Each knowledge concept is represented by an individual Markdown document, where the YAML frontmatter contains structured metadata while the Markdown body can contain more detailed and less strictly structured information [Goo26]. Relationships between concepts can additionally be expressed through links between the corresponding documents. The format is intentionally kept simple and does not require a central schema registry, database, or specific software framework, allowing the files to be read and modified with standard tools and stored in version-control systems such as Git [MH26]. This combination of structured metadata and readable textual content makes OKF one example of a representation that remains accessible to both software tools and human readers [MH26; Goo26].

A related relational approach is the use of knowledge graphs (KGs). Knowledge graphs have long served as an infrastructure for structured knowledge representation and reasoning. They are a form of structured data representation for relational information about real-world objects and the relationships among them [De 23]. They provide a semantic foundation for applications such as semantic search and question answering, while rela-

tionships can contain types and properties with a well-defined meaning [WT24; Ji+22]. KGs can be represented as directed graphs with nodes as entities and edges as relations [Ji+22]. A knowledge graph may additionally integrate information into an ontology and apply a reasoner to derive new knowledge [Ji+22]. Graphs provide a concise abstraction for domains in which edges and paths capture potentially complex relations between entities [Hog+22]. Mathematically a KG is represented as follows:

$$G = \{E, R, F\},$$

where E, R, and F are sets of entities, relations, and facts, respectively. A fact is represented as a triple

$$(h, r, t) \in F.$$

Knowledge-graph-based question answering answers natural language questions with facts from knowledge graphs [WT24]. Neural network-based approaches can use such structures to represent questions and answers in distributed semantic space and in some cases, combine them with symbolic knowledge for reasoning.

Structured external knowledge can therefore be represented in different forms, ranging from schema-based documents to graph-oriented structures. Such representations can also be supplied to systems that process natural language, including Large Language Models.

2.8. Large Language Models

Large Language Models (LLMs) have achieved state of art performance in many Natural Language Processing (NLP) tasks according to Wu and Tsioutsiouliklis [WT24]. The moving hypothesis in past years is that, by observing large volumes of heterogeneous textual data, LLMs are able to capture vast amounts of factual knowledge [De 23]. As LLMs understand natural languages very well, natural language representation is a straightforward way for either prompting or fine-tuning LLMs with structured knowledge [WT24].

However, LLMs may generate nonfactual, nonsensical, or incoherent output, commonly referred to as hallucination [WT24]. External structured context can improve grounding and traceability, although it does not guarantee correct reasoning. Explicit evidence and instructions for handling missing knowledge remain relevant when external information is supplied to a model [WT24]. It is worth noting a growing common concern about the lack of control over what these models learn, since it may lead to unpredictable results and toxic or biased content. Structured inputs and prompting can provide additional constraints, although they do not eliminate these limitations.

A prompt is a set of instructions provided to an LLM to guide its behavior for a specific task [Whi+23]. Prompt engineering refers to the design and refinement of these instructions in order to influence how the model processes the provided information and produces its output. Natural language prompts make LLMs flexible to use across different tasks and allow users to specify both the task itself and the expected form of the

response [Sch+24]. Prompts can also specify output structure and evidence requirements, which can improve consistency across repeated analytical tasks.

Because prompts can include external information in addition to task instructions, LLMs can be used in context-based systems that combine model capabilities with supplied knowledge.

2.9. Context-Based LLM Systems and RAG

A further distinction concerns how external knowledge is supplied to an LLM. LLM-based systems can be provided with external information in addition to the knowledge encoded in the model parameters. The information supplied as part of the prompt provides task-specific context that can be used by the model when generating an answer. Structured representations of external information can further support LLM reasoning and make relationships between entities more explicit [Zha+25; WT24].

One approach is to provide the relevant external knowledge directly as part of the LLM prompt context [Zha+25]. In a full-context approach, the available knowledge representation is serialized and included directly in the context supplied to the model. No separate retrieval component is required to select individual knowledge entries before the LLM request is generated. However, the amount of knowledge that can be supplied in this way is constrained by the available context window of the selected LLM model.

Retrieval-Augmented Generation (RAG) provides a different approach. Retrieval-Augmented Generation combines a generative language model with access to an external non-parametric knowledge source from which relevant information is retrieved for a specific input [Lew+20]. Instead of necessarily providing the complete external knowledge collection, a retriever identifies relevant passages or knowledge items and makes them available to the generator. The generated answer is consequently conditioned on both the model and the retrieved external information. Retrieval provides a mechanism for accessing external and updateable knowledge without encoding the complete knowledge source directly in the model parameters.

Direct full-context provision and retrieval-based context provision differ primarily in how external knowledge is selected before generation. Full-context provision supplies the available knowledge directly to the model without an additional retrieval step. Retrieval-based approaches instead select only the information considered relevant for a specific query before passing it to the LLM [Lew+20]. This reduces the amount of context supplied to the model, but the quality of the generated answer then also depends on whether the retrieval step selected the relevant information. The two approaches therefore differ in the trade-off between providing complete available context and limiting context size through query-dependent retrieval.

3. Related Work and Research Gap

The purpose of this chapter is to review existing research on legacy system modernization risks and mitigation strategies. It examines existing modernization frameworks, roadmaps and strategy-selection approaches and separately reviews approaches for structured knowledge representation and LLM-based reasoning. This comparison helps distinguish existing contributions from the literature-based synthesis pursued in this thesis.

Existing research provides relevant knowledge on modernization challenges, mitigation measures, decision frameworks, knowledge representation and LLM reasoning. However, these research streams remain only partially connected. The chapter therefore concludes by identifying the research gap at their intersection and clarifying what has already been studied and what remains missing.

3.1. Research on Legacy System Modernization Risks

Prior literature frequently discusses modernization problems using terms such as challenges, risks, barriers, and concerns without consistently distinguishing them. This terminological heterogeneity complicates the systematic comparison of risk-related findings across studies. For legacy-system modernization specifically, research has predominantly taken a technical perspective. Existing research has traditionally focused on technical challenges [Kha+14] or on adoption processes from an IT rather than a business perspective [GS14]. When only considering the technical dimension, software modernization is claimed to be successful when modernization is completed using predefined technical solutions [Kha+15]. Some studies use the term software modernization more narrowly than the holistic legacy-system modernization perspective adopted in this thesis. Their findings are therefore interpreted according to the terminology established in Chapter 2. In such studies, technical factors remain a determining factor [Sea+01]. However, industry studies also report business-oriented challenges that complement the technological challenges emphasized in academic research [Kha+14].

As established in Section 2.5, legacy-system modernization involves technical as well as business, organizational and process challenges. Prior work has therefore also extended the view beyond technical factors to technical, financial, organizational and strategic factors, as modernization involves adoption within an organization [Paz25; Kha+14]. This is consistent with the view that software evolution must be addressed as a business issue as well as a technology issue and is fundamentally interdisciplinary [BR00].

Within these dimensions, prior work identifies a broad range of modernization challenges. These range from data migration versus application migration dilemmas to user acceptance, integration hurdles and strategic decision-making [PE23]. Agencies report

multiple reasons for not modernizing legacy systems, including lack of resources, lack of experience, risk of disruption to mission programs, organizational resistance to change and risk of project failure [Ame17].

These challenges should not be interpreted in isolation, since prior work reports that they are interrelated [Red24].

3.2. Research on Mitigation Strategies

The technical emphasis identified for modernization risks is also visible in the mitigation strategies proposed in prior work [Ass+25]. One frequently discussed strategy is strong project planning from both technical and business perspectives, together with closely monitoring the project progress and outcome [GS14]. This also includes testing, as the technical process needs to be validated and verified [GS14]. Another commonly discussed approach is not modernizing everything at once, but instead using a gradual migration in which components are systematically modernized step by step [Ogu+23]. This can be combined with rollback planning if recovery becomes necessary [Gho+17; Van25]. Rigorous validation is also necessary to verify end-to-end functionality throughout development, preferably in combination with continuous monitoring and maintenance after implementation [PE23]. The testing framework for mainframe modernization should address both technical and business dimensions if modernization is to be effective for an organization [Van25]. However, mitigation strategies can also introduce new problems. Incremental development, for example, can cause the target architecture to resemble the legacy architecture [Sea+01], while such secondary risks are only discussed in a limited number of cases in the reviewed literature.

For organizational mitigation approaches, training users as well as staff for the new system through, for example, seminars appears to be a common strategy against resistance and adoption-related problems [Lan; CIOb]. Effective training programs can further alleviate some of these organizational concerns [PE23]. Another strategy discussed in literature is the complete documentation of requirements for the legacy system, which can support both organizational preparation and the technical process [Frab].

When it comes to mapping mitigations to risks, explicit mappings are still less common than isolated descriptions of risks or mitigation strategies. In many cases, relationships remain implicit because risks and mitigations are discussed in the same literature source without a direct connection. However, explicit mappings between modernization risks and corresponding mitigation strategies can be found in individual studies. For example, Riihimäki [Rii23] identified a set of risks for legacy-system cloud migration and directly associates each risk with one or more mitigation measures, as summarized in Table 3.1. Existing literature therefore includes mitigation measures and individual risk–mitigation mappings, but their applicability and side effects depend on the modernization context. Prior work provides many useful measures but less of a standardized many-to-many representation that links mitigations to risks while also preserving prerequisites, evidence and new secondary risks. Mitigation trade-offs are also discussed less frequently. One example is parallel operation, which can reduce operational risk while at the same time

increasing development risk, degrading performance and increasing maintenance costs [Sea+01].

3.3. Existing Frameworks, Roadmaps and Strategy Selection

Existing research provides several frameworks and roadmaps that support organizations throughout different stages of legacy system modernization. These approaches usually structure the modernization process into consecutive phases and provide guidance for planning and execution. They are relevant to modernization risk research because some approaches explicitly include risk analysis and mitigation as part of modernization planning and strategy selection [Rii23; Ste]. One example is the CMMI Compliant Modernization Framework proposed by Khan et al. [Kha+21], which combines modernization requirements, a modernization roadmap and the CMMI framework to guide the transformation of legacy systems into enterprise systems as illustrated in Figure 3.1. The modernization roadmap itself consists of several stages ranging from modernization need analysis and planning to architecture and design, implementation, data migration, verification and validation and finally system retirement. Other approaches such as SMART-MP focus more specifically on supporting decisions during modernization [Bal+08]. Similarly, modernization frameworks and agency roadmaps proposed in grey literature provide end-to-end guidance for assessing the current state, defining a desired future state and planning the transition between both [Ins; Ste]. Requirement approaches further support this process by systematically collecting and documenting information about the legacy environment before modernization decisions are made [Frab].

A major purpose of these frameworks is the assessment of the existing system and the selection of an appropriate modernization strategy. SMART-MP, for example, analyzes the gap between the current and desired state of a system and uses this information as input for modernization decisions [Bal+08]. Similar assessment steps can be found in practitioner frameworks, where organizations are advised to analyze their legacy application estate before setting modernization priorities [Ins; Mica]. The migration planning framework by Riihimäki [Rii23] follows a comparable approach by first creating an application profile, identifying goals and reasons for migration and subsequently selecting a migration strategy and cloud model before cost and risk analysis are performed. Strategy selection therefore depends on several characteristics of the system. This is also reflected by the weak concentration of influencing factors identified through the Gini coefficient [Lan], indicating that no single factor dominates modernization strategy selection.

Business value and available resources additionally influence modernization decisions. Modernization projects are dependent on sufficient funding and organizations therefore need to prioritize which legacy systems justify investments [US 21]. Systems providing high business value may still become candidates for modernization when costs and risks of continuing become high enough [Ste]. Consequently, modernization strategy selection cannot be considered solely from a technological perspective but requires balancing technical constraints, business value, costs, available expertise and organizational constraints.

Table 3.1.: Example of an explicit risk–mitigation mapping, adapted from Riihimäki [Rii23].

Identified risk	**Mitigation strategy**
Security	Confirm cloud provider's compliance offerings and security solutions. Refactor the application to utilize these security solutions if needed.
Underlying tech	Study cloud provider's documentation regarding the technology used. Study case studies or experience reports from clients of the cloud provider.
Connection	Confirm the SLAs of the cloud provider and examine how previous problems were handled.
Resource elasticity (in future)	Identify required modifications to the application to utilize cloud computing's elasticity.
Vendor lock-in (in future)	Rehosting does not result in vendor lock-in. If the application is refactored in the future, focus on application portability and establish an exit plan.
Documentation	Confirm information regarding VM resources and other relevant documentation.
Dependencies	Identify dependencies in the application profile and perform comprehensive testing before cloud deployment.
Limitations and problems	Existing limitations and problems may carry over to the cloud environment without refactoring. Refactor the application to address these issues.
Incorrect TCO calculations	Do not fully retire the on-premises application before the actual costs of operating the application in the cloud have been calculated.
Dependency on cloud provider	Confirm the cloud provider's SLAs and study experience reports from other clients.
Expertise for migration	Hire external expertise to support the migration process.
Time and cost of migration	Use external experts to estimate the total migration costs and required migration time.
Developers and IT Staff	Provide comprehensive training.

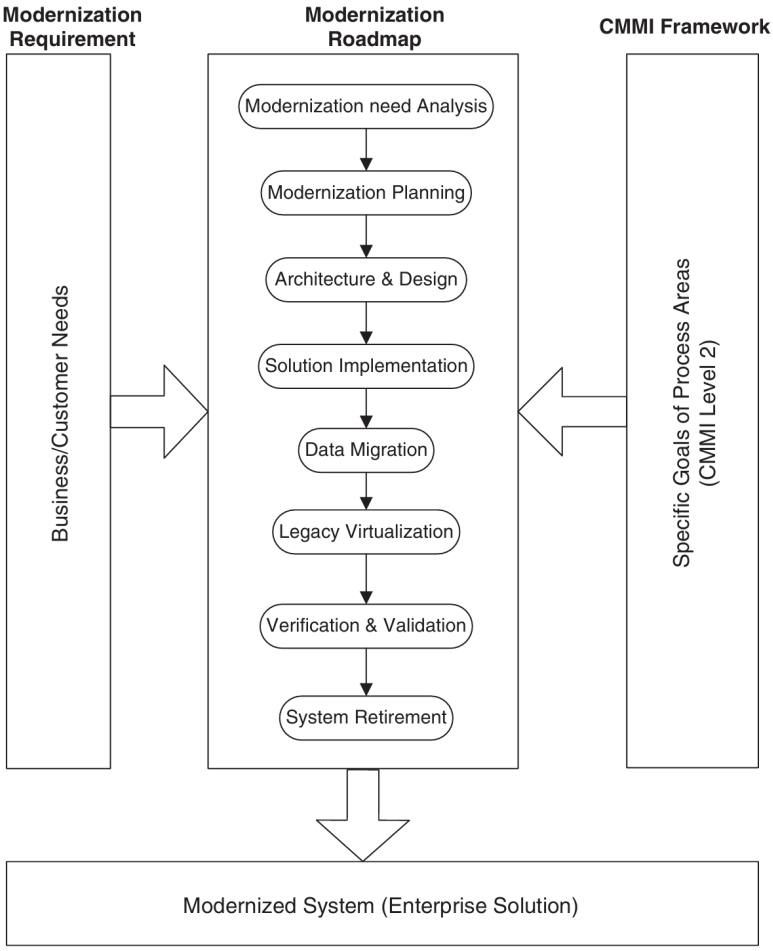


Figure 3.1.: CMMI Compliant Modernization Framework and its modernization roadmap, adapted from Khan et al. [Kha+21].

Risk considerations are also integrated into some of these frameworks. The migration planning framework by Riihimäki [Rii23], for example, explicitly contains a risk analysis phase and identifies common cloud migration risks alongside previously discussed mitigation strategies. Critical concerns identified during earlier phases can even result in cancellation or reconsideration of the migration plan. Other modernization guides similarly use risk and mitigation activities as part of modernization planning [Ste]. However, these approaches primarily use risks as one input into the modernization process or strategy decision. They are therefore not equivalent to a reusable risk knowledge representation in which risks, mitigation strategies and their relationships are represented as connected entities.

Existing modernization approaches are further limited by their scope and specialization. Within the corpus analyzed by Fávero, Almeida, and Affonso [FAA25], only one study (2.33%) focused on proposing a pipeline for supporting modernization. Other proposed approaches are tied to particular technologies or migration scenarios, therefore providing limited flexibility outside their normal context [Ass+24]. Additionally, differences in terminology and lack of standardization in modernization approaches make comparison and specifically strategy selection difficult [Hog+24; FAA25]. Existing frameworks consequently provide support for modernization planning and decision-making, but their primary goal is guiding the modernization process rather than providing a reusable representation of modernization risk knowledge across contexts.

3.4. Research Gap

3.4.1. Structured Knowledge Representation

When it comes to structuring modernization knowledge, current literature misses a holistic representation of it. There is no single work presenting a broader "big picture" of contemporary software modernization within the reviewed literature [Ass+25; Hog+24; Gho+17; Bat+14]. For modern microservice architectures, there is a lack in literature as well [Wol+21]. Consequently, there is still a lack of a comprehensive and contemporaneous body of modernization knowledge [Ass+24].

Knowledge graphs are used in different domains, however no application using them specifically for modernization risk and mitigation knowledge was found in the reviewed corpus. Knowledge graphs represent structural relations between entities and provide a concise and intuitive abstraction, with entities being represented as nodes and their relationships connecting them [Ji+22; Hog+22]. Existing knowledge graph research therefore provides general mechanisms for knowledge acquisition and relation extraction, but does not apply them specifically to legacy modernization risk knowledge [Ji+22].

Other knowledge representations of entities and relationships include schema-based forms such as coarse- and fine-grained typing [Ji+22]. The central principle of schema-based extraction lies in emphasizing precision, logical consistency and interpretability [Bia25]. Different literature sources describe risks and mitigations with different levels of depth and detail, which makes a consistent representation more difficult. A rigidly

defined schema can help to structure information despite these differences [Fen+20]. Rule-based approaches can additionally be used in knowledge graphs or separately to support reasoning over structured relationships [Hog+22]. However, these approaches remain general and do not define which entities, relationships or contextual information are specifically needed for modernization risk knowledge.

Entity-relationship representations for LLM reasoning have been researched more in recent years due to the rise of LLMs. JSON is well suited for storing structured data but showed some limitations for multi-hop reasoning settings as discussed in Chapter 2. Plain text is also an option for LLMs, but some additional structure can help in larger context scenarios [WT24]. Python representations of entity relationships outperformed both plain natural-language text and JSON representations in the experiments by Wu and Tsioutsiouliklis [WT24]. These representations store both structured data and parts of the inference process, providing a more controlled and unambiguous way of guiding LLM reasoning. Within the datasets evaluated by Wu and Tsioutsiouliklis [WT24], Python was the representation that performed most consistently, although this does not mean that it is automatically the best format for every domain-specific use case.

3.4.2. LLMs in Software Engineering and Modernization

LLMs have also been researched more in recent years, especially when it comes to reasoning and the use of structured context. Wu and Tsioutsiouliklis [WT24] point out that limitations in training data and the training process can lead to hallucinations, where generated text is nonfactual, nonsensical or incoherent. Because of that, grounding and traceability are important when LLMs are used with external domain knowledge. One possible approach is the use of structured context. Results with structured context significantly and consistently surpassed those of the original prompt without structuring in the evaluated setting [Zha+25]. Structured and unstructured context can also be combined, which was found feasible for basic one-hop reasoning processes but was not further elaborated for multi-hop criteria [Zha+25]. Another important finding was that performance generally improved as techniques became more complex, indicating that preprocessing data and prompts might help the LLM model in its tasks [Sch+24].

Graph-based representations are another approach that can help to make relationships explicit for LLM reasoning. Reasoning with Graphs (RwG) first constructs explicit graphs from the context and then uses these graphs to support reasoning. The experiments by Han et al. [Han+25] showed improvements in both logical reasoning and multi-hop question answering tasks. However, no application of this approach to legacy-modernization risk analysis was identified in the reviewed corpus.

Existing LLM-based modernization research already shows that LLMs can support specific modernization tasks. AgentModernize, for example, focuses on preserving business logic during code transformation through multi-agent LLMs and behavioral specification graphs [AG26]. This is more specialized toward code transformation and business-logic preservation than the cross-dimensional risk-analysis perspective considered in this thesis. It therefore shows that current LLM-based modernization research can already go very deep into specific technical tasks, while holistic risk and mitigation analysis remains

a different problem.

In general, existing LLM-oriented modernization research shows applications for modernization-specific technical tasks, while structured-representation research shows benefits of entity relationships and graph-based reasoning [Zha+25; WT24; Han+25]. However, these research streams do not directly provide a literature-grounded representation of modernization risks, mitigation strategies and their relationships for LLM-based risk analysis.

3.4.3. LLMs in Combination with Modernization

Prior modernization research already provides knowledge on modernization challenges and proposed solutions, but empirical coverage is still limited in parts of the literature. In the systematic mapping study by Agilar, Almeida, and Canedo [AAC16], 63.64% of the analyzed publications were classified as solution proposals, suggesting that proposed approaches received considerably more attention than their empirical evaluation in that study corpus. Similarly, Batlajery et al. [Bat+14] identified a relative absence of empirical research on industrial perceptions of legacy systems and modernization, while considerably fewer real-world case studies of legacy-to-modern-system migration are reported than theoretical proposals [Kha+13].

Existing literature does contain individual risks, mitigation strategies, roadmaps and also explicit risk–mitigation mappings [Rii23; OXD]. However, this knowledge is distributed across different frameworks, case studies, tables, surveys and textual descriptions. Within the reviewed literature corpus, no generalized representation was identified that consistently represents modernization risks and mitigation strategies across technical, organizational, business and process dimensions.

Another problem is the representation of dependencies and secondary effects. Risks and challenges are known to be interrelated [Red24] and the dependencies between them therefore need to be identified and quantified [Sof]. Literature recognizes these dependencies, but within the reviewed corpus no generalized relationship model was identified that consistently represents Risk–Risk, Risk–Mitigation and Mitigation–Risk relationships. Mitigation trade-offs and secondary risks can be found in individual cases, but they are not represented consistently.

Contextualization and traceability are also not covered consistently. Modernization risks and mitigation strategies depend on the modernization context and the selected approach, but existing representations do not consistently combine applicability conditions, evidence, uncertainty, source traceability and relationship reasoning within one reusable knowledge structure. This becomes even more difficult because terminology across the literature is not always clear or consistent [FAA25].

There is also a gap between modernization knowledge and LLM-oriented representation. Within the reviewed corpus, structured knowledge representation and LLM-based reasoning approaches have not been connected to holistic legacy-modernization risk analysis. Existing LLM-based legacy modernization research instead mainly focuses on more specific technical modernization tasks [AG26]. Within the reviewed literature, no approach was identified that combines holistic modernization risk knowledge, mitigation

strategies, explicit relationships, evidence, contextual conditions and source traceability in an LLM-oriented knowledge representation.

Another issue throughout the literature is the inconsistent use of terminology. Terms such as challenges, risks, barriers and concerns are often used without a clear distinction. Causes, triggers and consequences are also not always clearly separated from the actual risk. This makes findings from different sources harder to compare and also shows why consistent terminology becomes important when modernization risk knowledge from different literature sources is brought together.

Overall, the research gap lies at the intersection of legacy modernization risk research, mitigation research, structured knowledge representation and LLM-based reasoning. The individual research streams already provide important parts, but within the reviewed corpus they are not brought together into one traceable, relationship-oriented and LLM-ready modernization risk knowledge base. The gap is therefore not only about identifying more individual risks or mitigation strategies, but also about systematically bringing them together and representing their relationships, context, evidence and traceability.

4. Research Method

The research method combines literature-based knowledge identification at the start with qualitative knowledge extraction, consolidation and relationship modeling, as well as prototype development and finally validation followed by a two-step evaluation. As a basis, scientific and grey literature are used as the empirical foundation for identifying modernization risks and corresponding mitigation strategies. The extracted knowledge is subsequently transformed into a structured and machine-readable representation intended for LLM-based analysis and advisory support.

With this approach, RQ1 is addressed through literature research, literature management, extraction, consolidation and relationship identification. This contributes to a general knowledge base that can then be transformed into the desired output. RQ2 is therefore addressed through the development of the extraction templates, explicit relationship modeling, LLM-oriented fields and generally the OKF-based representation. Prototype implementation and comparative evaluation help to illustrate a working pipeline with this representation and to test the format.

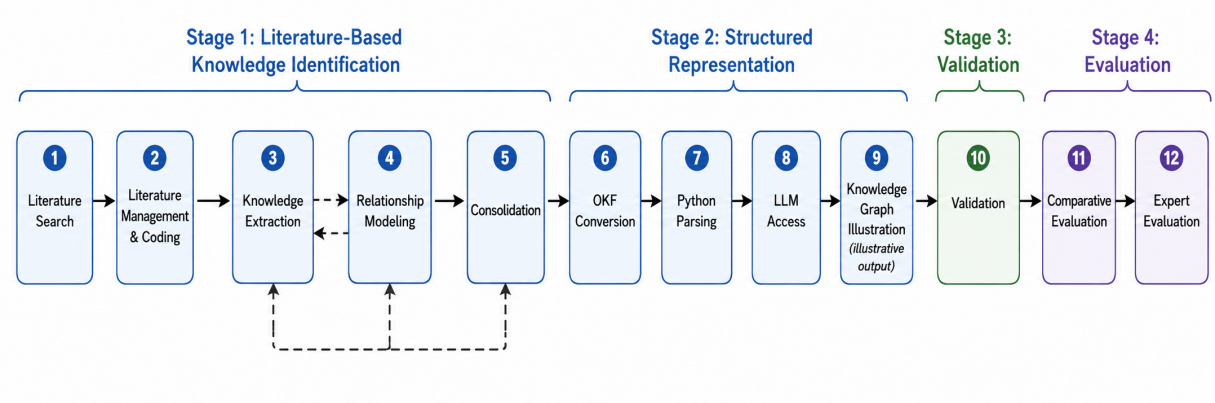


Figure 4.1.: Illustration of the method pipeline used throughout this thesis.

The overall research process is illustrated in Figure 4.1. The solid arrows indicate the main sequence of the method, while the dashed arrows indicate iterative feedback between knowledge extraction, relationship modeling and consolidation. After the literature search and literature management, the knowledge extraction focused mainly on identifying risks and corresponding mitigation strategies. In parallel, relationships between these elements were derived from the literature and modeled explicitly where sufficient evidence was available. The extracted knowledge was then standardized and converted into the OKF-based representation, which was parsed in Python and provided

as context to the LLM. The resulting knowledge graph serves as an additional illustration of the represented relationships. Validation was followed by a comparative evaluation and finally, an expert evaluation.

The knowledge representation was developed iteratively alongside the literature analysis. Risk and mitigation structures formed the initial basis, while relationship modeling was refined during the extraction process to capture risk–mitigation links, cascading risk chains and secondary risks. LLM-oriented fields and logic patterns were added subsequently, and cluster information was incorporated later based on the graph-based analysis.

4.1. Research Design

Starting with the literature foundation, the literature analysis forms the basis of the risk and mitigation knowledge base. Therefore, scientific publications and grey literature are used to identify risks, mitigation strategies, contextual information and relationships. This allows the analysis to consider information beyond reviewed literature papers alone. This primarily addresses RQ1 and therefore the knowledge-gathering part of this thesis.

Once this information has been gathered, the knowledge needs to be structured. For this, extracted information is documented using standardized templates for risks and mitigation strategies. Similarly, relationships follow a developed structure in the form of three templates that have slightly different fields but are based on a common main template. The relationship entities are used to explicitly connect risks, mitigations, risk cascades and secondary effects. The resulting structured representation is similar to a knowledge graph with relationships connecting risks and mitigations and provides the conceptual basis for answering RQ2.

The extracted and consolidated knowledge is then converted into an OKF-based Markdown representation. This creates a file system that makes every risk, mitigation and relationship easily accessible and also human-readable. The files are then parsed into Python data models and provided as context to an LLM. The knowledge is additionally represented through a knowledge graph and made accessible through a web-based prototype.

In order to evaluate this procedure, the resulting knowledge base and software artifact are first validated functionally for basic functionality. Afterwards, a comparative evaluation investigates different forms of providing the same modernization knowledge to an LLM and also examines how performance changes with less knowledge context. An additional expert evaluation examines the quality and practical usefulness of the generated answers.

The following sections describe the individual parts of this process illustrated in Figure 4.1.

4.2. Planning and Conducting the Literature Search

The scientific literature was searched using Google Scholar as a search platform. The literature research focused on legacy systems, legacy system modernization, modernization risks, modernization challenges and mitigation strategies. However, additional literature searches were used throughout the extraction process when further information on specific modernization risks or mitigation strategies was required. The scientific literature was primarily used for knowledge extraction as well as background information for this thesis.

The approach used was an iterative literature search, meaning the research was not limited to a single initial search but included smaller searches throughout the extraction process. Further searches were conducted during the extraction process when new risks, mitigation strategies, terminology or modernization concepts emerged that needed additional information. These additional searches focused specifically on modernization literature as well as risks and mitigation strategies because they form the main part of this contribution.

When it comes to in-depth searches and current information, grey literature sources were considered. Grey literature was searched separately to complement the scientific literature with practice-oriented information. This included industry reports and other publications from organizations such as IBM, Gartner, Microsoft, Amazon and SAP. Additional targeted web searches were used to identify relevant industry reports, practitioner material and other grey-literature sources.

All of these sources were evaluated and ranked on a scale representing their usefulness for the thesis and whether they were suitable information sources.

4.2.1. Inclusion and Exclusion Criteria

As described in the previous section, literature sources were rated on a 1–5 scale based on the defined extraction criteria. Sources were included if they met the following criteria:

- The source was directly related to legacy systems or legacy system modernization.
- It addressed relevant risks, challenges or mitigation strategies.
- It provided sufficient content to extract information relevant to the research questions.
- The source was traceable.

Sources were excluded if one or more of the following criteria applied:

- The publication was primarily promotional.
- The source did not meet the defined language criteria.
- The full text or sufficient information was not accessible.

In addition to scientific literature, sources from non-academic channels (grey literature) were considered when they provided relevant experience, modernization cases, identified risks or concrete mitigation approaches that were new or additional.

As no exact quantitative screening counts are available for the complete literature selection process, the selection of papers is documented through the final source collection in combination with the assigned paper IDs and the traceability of extracted entries to their respective sources.

4.2.2. Scientific and Grey Literature

This thesis used grey literature not as an alternative but as an additional source alongside scientific literature. Scientific and grey literature serve different but complementary roles within the literature analysis, and while scientific literature provides academically documented findings and evidence, grey literature provides additional practice-oriented experiences, modernization cases, industry reports and current observations from organizations actively involved in modernization projects. This is relevant because, to carry out a thorough search of the literature, it is necessary to look beyond conventionally published books and journals [How]. This is a major part of this thesis, so there is a need to define what grey literature is. The term 'grey literature' is often used to refer to reports published outside of traditional commercial publishing. Review authors should generally search sources such as dissertations and conference abstracts [Kin]. It is information produced outside of academic publishing and distribution channels [Sim]. Dissertations and conference abstracts are on the more scientific side of grey literature, because grey literature can include reports, policy literature, working papers, newsletters, toolkits, government documents, speeches, white papers and similar materials [Sim] that refer to factual and research-based material [Bri].

Grey literature is useful here because it can provide practice-oriented material from researchers and practitioners outside commercial academic publishing [How]. Possible producers include universities, government organizations, professional associations, businesses, consultancies, think tanks, non-profit organizations and international institutions according to Howell [How]. Relevant sources for this thesis therefore also include industry bodies and companies working directly in the field of modernization [Mon].

There are several reasons for including grey literature in research. For example, some information is only available as grey literature, including preliminary project findings and practice-oriented reports [How]. Grey literature can bypass longer traditional publication processes through which information may otherwise be delayed. This is also reflected by conference proceedings and study results appearing as grey literature before they are published through conventional academic channels [How]. Grey literature can therefore provide more current information on ongoing developments [Kin]. Furthermore, grey literature sources can also help identify additional relevant sources referenced by existing publications [Kin]. Another important aspect of scientific literature is publication bias. Publication bias refers to the tendency that studies reporting positive or statistically significant effects are more likely to be formally published than studies reporting weak or negative findings [Kin]. Including grey literature can therefore help reduce positive-

results publication bias and provide an additional source of evidence, especially useful for including negative modernization experiences in this thesis [Sim].

Grey literature can provide additional useful material, but sources require explicit assessment before their information is used. Relevant criteria include authority, currency, neutrality and traceability, which need to be checked for every source [How]. It is essential to evaluate and assess the validity of grey literature information before using it [Bri]. In this thesis, source type and origin are retained through the corresponding G IDs so that extracted knowledge remains traceable to the original grey literature source.

4.3. Knowledge Extraction and Modeling

4.3.1. Literature Management and Coding

Using both scientific and grey literature leads to more sources that need to be managed. Therefore, a literature management tool was needed. Citavi was used to manage the collected scientific and grey literature sources and each relevant source was assigned an ID for later traceability. Scientific publications use identifiers beginning with P and grey literature sources use identifiers beginning with G.

Relevant text passages were extracted while reading the sources. Categories, keywords and labels were assigned in Citavi to organize these markings for later reuse. The coding was used to organize the literature according to topics such as modernization, risks, mitigation strategies, relationships, LLMs and general implementation details.

However, this coding alone is not the actual extraction of risks, mitigations and relationships. Literature coding and template-based extraction were performed in parallel for every scientific or grey literature source. When, for example, a relevant risk, mitigation strategy or relationship was identified, the corresponding information was transferred into the extraction structure. At the same time, the relevant literature passage stayed linked to its source through the paper ID and the corresponding Citavi marking. The combination of paper IDs, Citavi coding, extracted quotations and standardized entries makes it possible to trace the final knowledge base back to the underlying literature. This traceability is retained throughout consolidation and knowledge representation. After the parallel extraction was completed, a final pass over the different keywords was conducted to improve completeness.

4.3.2. Development of the Extraction Templates

For consistent extraction, a general template approach is needed for standardization and a structured end result. Therefore, three main templates were developed iteratively throughout the literature analysis. The first versions focused on identifying risks and mitigation strategies. Initial fields included identifiers, names, descriptions, source references and additional descriptive information required to characterize an extracted risk or mitigation. Separate templates were developed for risk and mitigation strategy entries because, even though some fields overlap, there are conceptual differences between risks and mitigations, especially in applicability fields that go further than just describing.

The initial templates were expanded as additional recurring information was mentioned in different literature sources. The goal was to represent information from different publications in a consistent structure despite differences in terminology and presentation. This structure is needed for the efficient implementation of reformatting tools used in the prototype development.

Explicit relationships were added at a later stage of the extraction process because the initial templates captured risks and mitigations as separate entities, while recurring connections in the literature showed that these links also needed their own representation. This motivated a separate relationship template because representing the influence of mitigations on risks requires different attributes. Hence, a separate relationship template was introduced to connect risks and mitigation strategies. Later on, relationship modeling was extended apart from risk–mitigation links to include risk–risk cascades and risks introduced again by mitigation strategies.

After the initial descriptive fields, the need for an LLM-oriented extension became clear while extracting risks and mitigations. For this, additional fields, specifically for later LLM use, were introduced toward the end of the template development, including standardized entries and logic patterns. Logic patterns generally express extracted knowledge in conditional structures: IF [Condition] AND [Trigger] THEN [Risk] or IF [Condition] THEN APPLY [Mitigation]. These additions provide a more standardized representation of information that originally appeared in textual form extracted from literature sources.

At this point, the templates were mostly finished and consistently used throughout the extraction phase. However, one field was added later on: the cluster field. It was added to risk and mitigation entries after completion of the evaluation. Cluster membership was derived from a later-developed knowledge graph clustering and was therefore not available to the LLM in any context conditions used for the comparative evaluation. The reported evaluation results are consequently unaffected by cluster information and therefore, the cluster field represents a later refinement of the knowledge artifact rather than part of the originally evaluated knowledge representation.

Not all fields in the final knowledge representation have the same origin. To make this transparent, the information contained in the templates is divided into four categories:

1. **source-extracted:** Information directly obtained from the analyzed literature. This includes, for example, descriptions, reported impacts, prerequisites, observed outcomes, stakeholders, contextual conditions, tools and key quotes that were directly mentioned in a paper.

2. **analyst-coded:** Classifications assigned during the extraction process based on source material. Examples include dimension, granularity level, strategy type, evidence type and modernization strategy classifications. Relationship influence strength and confidence level also belong to this category as they represent analytical assessments.

3. **normalized or transformed:** Information created by standardizing terminology and descriptions in different sources. This includes standardized risk and

mitigation names, standardized risk entries, standardized mitigation entries, standardized relationship entries and in some cases logic patterns. These fields retain the interpreted meaning of the source material but are formulated as part of the knowledge modeling process.

4. **analytically derived:** Information calculated after the extraction process based on the structure of the knowledge base. The main example is the cluster field, which is derived from clustering within the knowledge graph.

Individual relationships may contain elements from more than one of these categories. A relationship may be stated directly in a source or modeled analytically by combining several source findings. Reasoning notes and confidence level are further used to document this distinction and make the interpretation traceable.

For most risks and mitigations, not every publication contains information for every template field and there are cases with little or no information, for example if a mitigation is only mentioned once. Missing information is retained as unspecified instead of being invented or set to a standard value. It is important to note that the distinction between source-extracted, analyst-coded, transformed and analytically derived information is maintained throughout the knowledge model development.

4.3.3. Risk Extraction

For the risk extraction, passages describing modernization problems, challenges, possible failures or negative consequences were examined for information. During extraction, risk, root cause, trigger event and impact were treated as separate concepts as explained in Chapter 2. A challenge or underlying condition was therefore not automatically represented as the risk itself but contributed to the completeness of the description.

Extracted risks were documented using the risk template. Risks were classified according to technical, organizational, business and process dimensions. Important is that they could also contain dependencies or effects extending beyond their primary assigned dimension. Root causes, trigger events, impacts, linked risks, affected stakeholders and contextual constraints were recorded where available.

Evidence type was used to record the type of support provided by the underlying source and detection source records where or how the risk may be identified when such information is available. Relevant terminology and tags were retained to support later categorization and retrieval. Standardized entries, relevant source quotations and logic patterns were additionally retained for standardization and later LLM usage.

4.3.4. Mitigation Strategy Extraction

Similar to the risk extraction, mitigation strategies were extracted from literature passages that describe actions oriented to prevent, reduce, detect, respond to or recover from the corresponding risks. Mitigations were first documented independently from risks because a mitigation can occur in literature without a directly mentioned risk or

can be relevant for multiple risks at the same time. Therefore, the actual connection between risks and mitigations was also modeled separately through the relationship entries explained in Section 4.3.5.

Extracted mitigations were documented using the mitigation strategy template and classified according to their strategy type, namely they are preventive, detective, reactive or recovery measures. Prerequisites, tools and technologies, effort, effectiveness and probability of success were included where corresponding information was available. As with the risk extraction, missing information was left again unspecified instead of adding values without literature source support.

Trade-offs and side effects were extracted where available because a mitigation can reduce one risk while introducing additional costs, complexity or even another risk. Stakeholders implementing or benefiting from the mitigation and contextual constraints were recorded where available and later also considered during relationship modeling.

The cluster field was added after the knowledge graph clustering and was not part of the original extraction as explained before.

4.3.5. Relationship Identification and Modeling

A central part of this contribution is the explicit relationship modeling, because risks and mitigation strategies were frequently described independently in the underlying literature. The relationship representation was introduced to make connections between individual knowledge base entries explicit. This makes these connections more traceable and machine-readable for later usage.

This thesis covers three different types of relationships: direct risk–mitigation relationships, risk–risk cascades and mitigation–risk relationships in which mitigation strategies introduce or amplify secondary risks. As with risks and mitigations, relationships were added when explicitly stated in the literature within one source and also when derivable by combining information from multiple relevant passages.

Risk–mitigation alignments connect extracted mitigation strategies with the risks they address. Risks were additionally examined for causal or cascading connections in which one modernization risk can trigger, contribute to or amplify another risk. Mitigations were also examined for documented trade-offs and secondary effects, and where applying a mitigation could introduce another risk, this effect was explicitly modeled as a relationship.

A further important part is the confidence assignment. Confidence level reflects the degree of support for the modeled relationship rather than the type of evidence alone. Therefore, evidence type and confidence level are represented separately because they describe different properties of a relationship. Evidence type describes the form in which the underlying support is provided, whereas confidence level represents the researcher's assessment of how strongly the available evidence supports the specific modeled connection.

Consequently, confidence assignment considers several factors in combination. These are mainly whether the relationship is stated explicitly or requires analytical inference, the form and strength of the supporting evidence in combination with the source ranking,

whether an observed effect is documented, whether multiple findings support the same connection and whether the available evidence is direct and contextually consistent with the modeled source, target and relationship type. Relationships supported by direct and comparatively strong evidence, for example statistical findings or concrete case evidence, were generally assigned a higher confidence. However, a relationship was not classified as high solely because a possible connection was mentioned in a source. Relationships based on weaker, preliminary, indirect or more strongly interpreted evidence were generally assigned medium confidence. Overall, confidence level provides a synthesis of evidential support and modeling certainty.

Relationships complete the extracted data part of the knowledge base. A conservative relationship dataset was maintained using the more strongly supported relationships and was used as the basis for the prototype. An additional extended relationship dataset was created that also includes medium-confidence relationships. This extended dataset results in a more strongly connected graph structure but also increases the amount of context that must be provided to the LLM. The two variants form the basis for a separate graph-based comparison.

4.3.6. Consolidation and Normalization

After the extraction phase, there is a normalization or consolidation phase. It is needed because different publications frequently describe the same or highly similar modernization concepts using different terminology and as many literature sources contribute, overlapping descriptions were found. The same issue can therefore occur multiple times during independent extraction from different papers.

To identify semantically equivalent entries, risks and mitigations were compared based on their names and descriptions. Entries were consolidated when different sources described essentially the same underlying risk or mitigation but used different wording. Similarity of terminology alone was not sufficient if the underlying meaning of the risks differed. In cases where different fields were filled out in the templates, entities were merged into a single entry that combined the relevant information from both entries.

Importantly, consolidation does not remove the fact that an entry can be supported by information from multiple sources. When fields from several entries were merged, source references remained traceable through the associated paper IDs and literature notes by listing all contributing paper IDs in the paper ID field.

After consolidation, names and terminology were standardized if not already done. Duplicate entries and duplicate IDs were checked to ensure a consistent assignment of IDs, and dimensions and relationship types were standardized to the defined values. Missing values were kept unspecified instead of being completed without source support to avoid adding unsupported claims. Furthermore, information directly stated in the literature is distinguished from interpretation performed afterwards in standardization. This distinction is particularly important for relationships that are derived from multiple pieces of evidence.

4.4. Software Artifact Development

After completing the cleaned knowledge base, the transition from extracted knowledge to a software artifact starts. The knowledge catalogue was then transformed into a machine-readable representation because plain text was not sufficiently structured for the intended processing as discussed in Chapter 3. For this, risk, mitigation and relationship information was converted into separate structured entries, all following the same principle but remaining separate because the templates contain different fields.

The presented OKF format introduced in Chapter 2 was used in this thesis, as its Markdown and YAML structures are sufficiently abstract and structured while still readable by a human. Each knowledge entry is stored as a Markdown file containing structured metadata and additional textual descriptions.

The structured knowledge is subsequently processed through the developed software artifact and provided as context to the LLM. Different knowledge-context configurations can be selected for validation and evaluation. The functionality was additionally integrated into a web-based interface, while the extracted risks, mitigations and modeled relationships are represented in a knowledge graph for further visualization and analysis. The technical implementation of these components is described in detail in Chapter 7.

4.5. Validation and Evaluation Design

4.5.1. Validation Design

After the prototype implementation was completed, the framework was first validated in a developer-conducted functional validation. The purpose of this validation is to examine whether the implemented knowledge base and LLM-based access function as intended before conducting the comparative evaluation. The objective is to verify whether risks, mitigation strategies, relationships, risk chains and knowledge boundaries can be correctly retrieved and processed by the prototype and to confirm that the basic pipeline works as intended. Particular attention is given to whether generated answers remain grounded in the available knowledge base.

For this, the functional validation uses 20 predefined test questions which cover basic retrieval, relationship retrieval, trade-offs and secondary risks, risk chains, simple consulting scenarios and importantly, knowledge-boundary tests.

Starting with basic retrieval, these questions examine whether individual risk and mitigation entries can be retrieved correctly from the knowledge base. This includes direct information such as risk descriptions, causes and consequences but also mitigation-specific information such as prerequisites, effort, effectiveness, stakeholders and possible trade-offs. The purpose here is mainly to check if the basic knowledge entities are accessible through the prototype and whether the LLM returns the information that is actually stored in the corresponding entries.

Relationship retrieval goes one step further because here, not only single entities but also the explicit connections between risks and mitigations need to be identified. These questions therefore check for the correct relationship IDs, relationship types, influence

strengths, applicability conditions and supporting evidence. This is relevant because the explicit relationships are one of the main extensions of the knowledge base compared to just storing risks and mitigations.

Another group of questions focuses on trade-offs and secondary risks. The idea behind these questions is that a mitigation does not necessarily only have a positive effect. A mitigation can reduce one risk while at the same time introducing another risk, side effect or complexity. Therefore, these questions test whether the prototype can distinguish the positive and negative consequences of a mitigation.

Risk chain questions examine whether multiple risk–risk relationships can be followed in the correct order. This includes direct cascades between two risks but also longer chains. Hence, the generated answers are checked especially for unsupported intermediate steps, wrong directions between risks and inconsistencies compared to the relationships, including hallucinated intermediate steps.

The simple consulting scenarios are broader than the previous question types and require the system to combine several knowledge entries at once. A scenario can contain different modernization conditions, and the prototype has to identify relevant risks, possible mitigation strategies and their relationships. These questions are used to check whether the knowledge base can handle a broader analysis. At the same time, it is important that the answer distinguishes between relationships that are explicitly documented and conclusions that are derived by the model.

Lastly, knowledge-boundary tests are included because correct retrieval alone is not sufficient for an LLM-based system. The model also needs to recognize when information is not present in the knowledge base. For this purpose, the tests include requests for a non-existing risk ID, a numerical risk probability that is not documented and a question that explicitly asks only for documented risk–mitigation relationships. These questions therefore test whether the model avoids inventing risks, probabilities, relationships or mitigation recommendations that are not supported by the provided knowledge base.

To ensure comparability, each response is rated on a given scale based on correctness, completeness and grounding in the knowledge base.

Score	**Interpretation**
5	Fully correct and grounded
4	Mostly correct but minor omissions or unsupported claims
3	Partially correct but some inconsistency or unsupported interpretation
2	Major inaccuracies, omissions or insufficient grounding
1	Unsupported, incorrect or fails to address the question

Table 4.1.: Validation scoring scheme.

The validation is performed before the comparative evaluation. This is important because problems in the underlying knowledge base or in the modeled relationships could directly influence later LLM responses and therefore affect the evaluation results.

4.5.2. Evaluation Design

Proof-of-Concept Evaluation

The evaluation of the framework consists of a developer-conducted comparative evaluation as a proof of concept and an expert evaluation. Both evaluations follow similar basic principles and investigate how different representations of modernization knowledge influence LLM answer quality. Therefore, the primary objective is to evaluate the usefulness of structured risk, mitigation and relationship information for LLM-based modernization analysis across different context configurations and structures. Particular attention is given to the added value of explicitly modeled relationships.

Four different context configurations are compared in both evaluations:

- **No Context:** The LLM answers without access to the modernization knowledge base.
- **Unstructured Full Context:** The available modernization knowledge is provided as plain text.
- **Structured Context without Relationships:** Structured risk and mitigation entries are provided without explicit relationship entries.
- **Full Structured Context:** Risks, mitigations and explicit relationships are provided together.

Because this thesis has a focus on the influence of adding relationships, the most important comparison is between structured context without relationships and full structured context. Both configurations contain the structured risk and mitigation information and differ only in whether explicit relationship entries are included, providing a direct comparison. This comparison therefore evaluates the added value of the relationship template.

The interpretation of the representation comparison needs to consider what happens internally in the prototype. The structured conditions are generated from the OKF-based Markdown knowledge base, but the Markdown files themselves are not supplied directly to the LLM because the files are parsed into structured Python objects before prompt construction. The evaluation therefore compares different runtime knowledge-context representations rather than directly comparing OKF Markdown with unstructured text. The evaluation of RQ2 should consequently focus on the underlying schema, explicit entity separation and relationship representation rather than attributing observed effects solely to the Markdown serialization format, as the focus of this thesis lies in evaluating relationships and providing a structured knowledge base that can easily be processed.

With these shared conditions defined, the comparative evaluation was conducted first by the developer as a proof of concept. For this, the comparative evaluation uses 35 predefined questions which are divided into direct knowledge questions, relationship questions, risk chain and secondary risk questions, scenario-based questions and as in the validation part, hallucination and structured output tests.

The direct knowledge questions examine whether individual information stored in risk and mitigation entries can be retrieved correctly by the framework, similar to the validation questions but with greater depth and complexity. Therefore, they include root causes, trigger events, impacts, stakeholders, causes, consequences and documented mitigation strategies in different combinations.

The relationship questions then examine whether the model correctly reconstructs explicit connections between risks and mitigations. For example, they test relationship types, applicability conditions and the way in which a mitigation influences a risk, further adding complexity to the simple risk and mitigation retrieval.

Next, risk chain and secondary risk questions examine multi-step relationships within the knowledge base to assess the prototype's ability to perform multi-hop reasoning. They test whether risk–risk cascades can be reconstructed, and they also examine whether risks introduced by mitigation strategies can be identified. These questions are used to look at the performance of non-standard relationships that differ from the direct risk–mitigation structure, therefore testing the different relationship templates.

The scenario questions provide the closest approximation to practical use and have greater complexity. Scenario-based questions describe realistic modernization situations involving several simultaneous conditions. The model must identify relevant risks, mitigations and relationships across multiple entries in a single answer. Hence, these questions examine the ability to combine stored knowledge for a broader modernization analysis.

Lastly, hallucination tests examine whether the model rejects unsupported probabilities, guarantees, relationships and other information that is not documented, directly aiming to expose answers that are hallucinated and not present in or based on the knowledge base. Structured-output tests, on the other hand require machine-readable responses such as JSON. These tests examine the correct representation of risk IDs, mitigation IDs, relationship IDs, relationship types, applicability conditions, evidence information and confidence levels.

To keep the comparison between the four context configurations as controlled as possible, the same model and general setup were used for every evaluation run. GPT-5.4 mini was used throughout the comparative evaluation with the same system prompt, questions and model parameters. Therefore, the only part that changed between the individual runs was the context that was supplied to the LLM. Questions, generated answers, selected context configuration and runtime were stored after each run so that the results could later be compared and evaluated under the same conditions.

For evaluating the answers, multiple metrics are used because a single score would not be enough to describe the different types of errors that can occur. The main metrics are F1 score, gold fact accuracy, gold fact coverage, groundedness and format quality. Additionally, precision and recall are calculated for the retrieved risk, mitigation and relationship IDs. Hallucination rate and runtime are also recorded separately. For the final overall score, the individual metrics are weighted with 30% F1 score, 25% gold fact accuracy, 20% gold fact coverage, 15% groundedness and 10% format quality.

The F1 score is mainly used for evaluating the retrieval of the expected IDs. For

every question, a gold standard defines core IDs which are required for the answer and additional IDs which can also be valid depending on how the answer is constructed. Precision describes how many of the returned IDs are relevant, while recall describes how many of the required core IDs were successfully found. The F1 score then combines both values. This is especially relevant for questions where multiple risks, mitigations and relationships need to be retrieved at once.

IDs alone, however, do not show whether the actual explanation of the model is correct. Therefore, gold fact accuracy and gold fact coverage are evaluated separately. Gold fact accuracy describes whether the factual statements made in the response are correct compared to the predefined gold standard. Gold fact coverage, on the other hand measures how much of the expected information is actually included. This distinction is useful because an answer can be correct in everything it says while still missing an important part and therefore not being complete.

Groundedness is used to check whether statements made by the model are supported by the provided knowledge context. This also connects directly to the hallucination tests. Examples of hallucinations are fabricated identifiers, unsupported relationships, invented numerical probabilities or guarantees that are not present in the knowledge base. The no context configuration is included to show how the model behaves without access to the developed modernization knowledge base.

Format quality forms the last component of the overall score and checks whether the requested answer format is followed. For normal text questions, this mainly concerns whether the requested structure is respected. It becomes more important for the structured-output questions that explicitly require JSON or another machine-readable representation. A correct answer can still be difficult to process further if it does not follow the requested output format.

Finally, runtime is recorded for every answer. Runtime is not directly included in the weighted overall quality score because a faster answer is not automatically a better answer. Still, it provides another comparison point between the different context configurations. Especially the full structured context contains considerably more information than the no context or reduced context variants, and therefore answer quality has to be considered together with the amount of context and computational effort.

Relationship retrieval receives additional attention in the analysis. Precision and recall for relationship IDs make it possible to compare the full structured context directly against the structured context without relationships. This comparison is intended to show whether explicit relationship entries improve performance on risk–mitigation questions, secondary-risk questions and longer risk–risk chains. Therefore, these questions are not only evaluated through the general overall score but also considered separately when answering the relationship part of the research question.

Expert Evaluation

The comparative evaluation is complemented by the expert evaluation. The expert evaluation uses the same four context conditions but reduces the number of questions to six predefined tasks representing different types of interaction with the knowledge base.

The six tasks were selected to cover the main interaction types that are also relevant for the comparative evaluation, but in a much smaller setup that is feasible for an expert evaluation. These tasks consist of risk information retrieval, direct risk–mitigation relationships, an indirect risk chain, scenario-based analysis, evaluation of unsupported or incorrect statements and finally a machine-readable risk chain generation. Therefore, the tasks range from rather simple information retrieval up to more complex combinations of multiple entities and relationships.

The expert evaluation is conducted through the developed web application in combination with a separate questionnaire. Experts first access the application and select GPT-5.4-mini as the model. Afterwards, one of the predefined tasks and the corresponding context condition is selected. The generated answer can then be reviewed directly in the application. In particular, the mentioned risk, mitigation and relationship IDs are intended to help the experts trace the answer and evaluate whether the returned information and connections are correct.

After reviewing an answer, the corresponding rating questions are answered in the questionnaire. The same procedure is then repeated for the other context conditions. Experts are not required to complete every single task and only have to answer the rating questions. This was mainly done to keep the expert evaluation manageable and allow participants to focus on the parts they can assess based on their experience. The estimated completion time communicated to the experts was around 30 minutes, depending on the number of evaluated tasks.

For the individual responses, experts rate answer quality on a scale from 0% to 100%. The main reference points of this scale were given with:

- **0%:** Completely insufficient response.
- **25%:** Mostly insufficient response.
- **50%:** Partially sufficient response.
- **75%:** Mostly sufficient response.
- **100%:** Fully sufficient response.

Intermediate values are possible as well if one of the predefined steps does not properly represent the expert's assessment. This allows a more detailed rating without forcing every answer into only five fixed categories while still providing common reference points for later comparison.

The exact assessment criteria depend on the corresponding task. For direct risk information, the main focus lies on correctness and completeness of the returned risk information. Risk–mitigation questions focus more on whether the correct mitigation is connected to the corresponding risk and whether the relationship direction and type are represented correctly. For the indirect risk chain, the logical order of the risk–risk relationships is relevant, while the scenario task focuses on identifying the relevant risks, mitigations and relationships from a broader modernization situation. Another relevant

point is whether documented relationships and conclusions derived by the model are clearly distinguished.

The unsupported statement task has a slightly different purpose because here a good answer should reject information that is not supported by the knowledge base instead of trying to provide an answer regardless. Therefore, unsupported probabilities, guarantees or relationships are especially considered. Finally, the machine-readable risk chain task evaluates both the actual correctness of the represented chain and whether the requested structured output is valid and usable for further processing.

Besides the task-specific ratings, the questionnaire also contains general questions about the prototype itself. These questions look at whether the program is understandable and easy to use, whether it could be useful for analyzing risks during legacy system modernization and whether the expert could generally imagine using such a system in a professional context. These general feedback questions are optional and are mainly used as an additional practical perspective besides the direct response ratings. All questions and tasks can be found in the appendices of this thesis.

One limitation in both evaluations comes from the amount of available relationship information. For the LLM-based comparative evaluation, the conservative relationship dataset containing only high-confidence relationships is used. The extended relationship dataset additionally contains medium-confidence relationships and creates a considerably larger and more connected representation. However, using this complete dataset as another LLM context condition would cause the serialized context to approach or exceed the available context window. Therefore, it is not included in the comparative or expert evaluation. Instead, the conservative and extended relationship datasets are compared separately through the structural knowledge graph analysis.

5. Results of the Literature Analysis

The purpose of this chapter is to present the results of the structured extraction and consolidation of modernization risks and mitigation strategies. In contrast to Chapter 3, the focus is on the findings resulting from the analysis of the complete literature corpus rather than on individual contributions of previous studies. Therefore, the chapter focuses primarily on answering RQ1 (Section 1.4).

The results are structured as follows: Risks are analyzed across the technical, organizational, business and process dimensions defined in the extraction framework. Cross-dimensional risks and relationships are analyzed separately in a short section to highlight their importance. Lastly, mitigation strategies, Risk–Mitigation relationships, Risk–Risk relationships, secondary risks and the underlying evidence are considered.

The scope of the chapter is to present representative and recurring patterns in the literature. The complete list of individual extracted Risk, Mitigation and Relationship entries remains available in the structured knowledge base.

5.1. Overview of the Literature Corpus

A general overview over the literature corpus is needed to grasp the dimensionality and depth of this analysis. In total 79 sources were covered for extraction purposes alone of which 37 were classified as scientific literature papers (P01–P37) and 42 as grey literature (G01–G42). Corpus metadata was processed manually in Citavi and the literature data base. A full list of every extraction paper can be found in the appendix.

Diving deeper into the observed literature, scientific literature includes empirical studies, case studies, systematic and mapping studies, modernization frameworks, migration approaches and related conceptual research. Examples include empirical modernization studies Khadka et al. [Kha+14; Kha+15], Gholami et al. [Gho+17], Batlajery et al. [Bat+14], and Khadka et al. [Kha+13] and systematic/mapping studies Agilar, Almeida, and Canedo [AAC16] and Fávero, Almeida, and Affonso [FAA25]. Grey literature includes governmental reports, industry studies, modernization guidance, practitioner frameworks and vendor documentation for example U.S. Government Accountability Office [US 21], American Council for Technology and Industry Advisory Council [Ame17], Punihani [Pun26], Red Hat [Red24], Stevens [Ste], and Software Improvement Group [Sof]. As mentioned in Chapter 4, academic studies provide empirical and conceptual perspectives, while grey literature contributes operational modernization experiences and practitioner-oriented risks and mitigation measures. This combination is particularly relevant because several modernization challenges reported in industrial contexts extend beyond technical concerns that are only seen in application.

When all of these sources were considered and extraction with help of the templates was done, the resulting structured knowledge base ended up containing 154 risk entries and 166 mitigation entries. Relationships can be observed in two different datasets, medium and high confidence, where the conservative high confidence relationship dataset contains 357 Relationship entries. The extended relationship dataset contains 697 Relationship entries, adding 340 relationships compared with the conservative one. The demonstrated method makes sure every Risk and Mitigation entry remains connected to its originating literature source. Relationships are connected indirectly when extracting source and target entities and additionally contain evidence and confidence information.

5.2. Overview of the Identified Risks

Risks can occur across different modernization dimensions, namely technical, organizational, business and process dimensions. As already established, modernization cannot be understood solely as a technical problem because organizational, business, and strategic factors also influence modernization outcomes [PE23; BR00]. Pazova [Paz25] also identifies a lack of a holistic view when it comes to technical, financial, organizational and strategic factors. As modernization is not only the technical part but especially the adoption in an organization, every dimension has to be considered [Kha+14].

Covering these different dimensions, the next sections deal with recurring risks identified in the literature. The following sections group recurring risks by dimension and show how they can affect technical correctness, cost, schedule and adoption. Risks in general occur at different levels of granularity and may concern individual code elements, components, applications, architectures, projects or broader organizational modernization programs. The reported granularity also differs across sources, especially in grey literature, depending on the modernized system.

5.3. Technical Risks

Even though the technical dimension is not the only one in consideration, it remains a prominent one in the reviewed corpus. One common concern is the complexity of legacy systems and their dependencies. In the Red Hat survey [Red24], legacy-system complexity was reported by 48% of respondents. A common issue in legacy systems is "spaghetti code," where dependencies are too intertwined to analyze [Sof]. Combined with limited modularity of older systems, decomposition approaches may struggle to separate the system effectively into distinct services [Com+00]. This complexity makes system understanding and analysis more difficult, especially because legacy systems have often been in use for a long time [Com+00], so structural complexity and hidden dependencies complicate system understanding and decomposition during modernization [Sof]. This creates technical risks ranging from incorrect decomposition to modernization in an unsuitable technical direction. Limited modularity, for example, increases the risk that decomposition approaches incorrectly identify service or component boundaries. If dependencies or behavioral relationships are overlooked, modernization decisions may be

based on an incomplete representation of the existing system and may lead to missing functionality.

Closely connected to this complexity are the size and documentation of a legacy system. Other factors such as the scale of changes required, support software required, service abstraction, service discoverability and code quality were also identified as technical challenges [GS14; Bat+14; MIT24]. Technical risks can be triggered by these challenges or amplified when several of them occur together. For example, insufficient documentation creates the risk that assumptions about existing functionality, dependencies or required changes cannot be reliably verified [GS14; Bat+14]. Large system size increases the risk of underestimating the technical scope and complexity of the modernization effort, which also has process-related consequences but is considered here as a technical risk [Com+00]. As a consequence, modernization teams may select inappropriate transformations, overlook affected components, or require additional analysis and rework during implementation [Bat+14].

An important factor in today's organizations is the amount of data, introducing the risk of migration and interoperability failures. Data migration, application migration and system integration are interdependent activities rather than isolated technical challenges [PE23]. They influence each other and can increase the overall migration risk. Modernization creates the risk that migrated applications, data structures or interfaces are no longer technically compatible with surrounding systems [Gho+17]. Decisions concerning application migration without sufficient consideration of corresponding data and interface dependencies therefore increase the risk of functional disruption in the modernized environment. On the other hand migrated data may become corrupted and unusable [Kha+14].

Risks can occur on concrete instances or affect the whole new legacy system as behavioral incompatibility between legacy and target technologies can come up. Substantial differences may exist between legacy languages or technologies and the selected target environment [Hog+24]. These technological gaps create the risk that existing functionality and behavior cannot be reproduced during transformation or, worst case, even after finishing modernization [Sim]. This happens because legacy-specific language constructs, platform behavior or technological assumptions have no direct equivalent in the new system. The resulting risk in such a case is unintended behavioral change or additional redevelopment effort required, creating a little risk chain to business risks.

Every risk can be further exaggerated by technical debt, which poses a risk in itself, because accumulated technical debt increases the complexity that must be understood and addressed during modernization [PE23; Sof]. Technical debt and highly intertwined code create the risk that problematic structures are either overlooked or transferred into the modernized system [IBMb]. Modernization without sufficiently addressing existing technical debt may therefore preserve legacy weaknesses. As mentioned, technical debt can additionally increase the required modernization effort and amplify risks associated with system understanding, decomposition and/or implementation [PE23; Sof; IBMb].

The technical risk dimension includes all tool related risks and limitations, for example that insufficient tooling support and inadequate evaluation mechanisms reduce the reli-

ability with which legacy systems and modernization outcomes can be analyzed upfront [Ass+25], which can then cascade into further risks like overlooking dependencies. Tools are important for systematic analysis and automation, therefore, inadequate evaluation metrics create the additional risk that the technical quality or success of modernization results cannot be assessed systematically at the end [Ass+25]. Tooling limitations therefore increase uncertainty in both modernization decisions and subsequent validation.

One last part that is important covers the accumulation of technical risks over time and different systems. Technical risks may accumulate when high system complexity is combined with historically grown architectures and distributed deployment structures that can be split up geographically [Sea+01]. Such environments create the risk that modernization must address several technical problems simultaneously rather than one isolated technical concern at once, leading to exponential complexity development [Kha+15].

5.4. Organizational Risks

Continuing with Organizational Risks, a major area concerns modernization errors caused by insufficient expertise. Modernization requires knowledge of both the existing legacy environment and the intended target environment [Hog+24]. Shortages of legacy and target-system expertise due to a lack of sufficiently experienced personnel therefore create the risk that important legacy behavior, dependencies or target-technology requirements are misunderstood and thus lead to failures at different levels [Rah26]. A lack of experience can also lead to incorrect technical or organizational decisions and can slow down modernization activities because additional knowledge must first be acquired [Ame17]. The organizational risk is therefore not the skill shortage itself, but its potential effect on the correctness, efficiency and continuity of decisions in modernization.

Closely connected to this knowledge shortage are risks of critical knowledge loss and key-person dependencies, because legacy-system knowledge may be concentrated on only a small number of long-term employees that have specialised in this field [Kha+15]. Case evidence demonstrates modernization environments with only a few remaining specialists for legacy technologies such as COBOL, reported by Khadka et al. [Kha+15]. This concentration creates a key-person risk because modernization activities become dependent on individuals [Rah26]. If these individuals are unavailable, leave the organization or become bottlenecks, critical system knowledge may become inaccessible, which can further increase planning uncertainty [Rah26]. Potential consequences include delays, incorrect assumptions about the legacy system, reduced knowledge transfer and increased modernization uncertainty.

Another main risk apart from expertise-related risks is the risk of failed organizational adoption and insufficient user acceptance. Organizational resistance, cultural resistance and insufficient user acceptance can create the risk that a technically successful modernization is not successfully adopted by the organization or end-users [Kha+14]. In more detail, modernization changes established technologies, workflows, responsibilities and/or working practices, which may trigger resistance among affected stakeholders

[Ame17]. Insufficient stakeholder acceptance may result in low adoption, resistance to changed processes or even continued reliance on legacy practices, again showing how interconnected risks of the same dimension can be [PE23]. Modernization success is therefore at risk even when the technical implementation itself functions correctly.

There is also a risk of delays and quality reductions caused by resource competition. Because ongoing operations usually cannot simply be stopped, modernization initiatives compete with ongoing operations and other organizational initiatives for personnel, time and organizational resources [Ass+25]. This leads to limited resources and competing priorities creating a risk that modernization activities cannot be executed as planned, adding more risks to other dimensions [Ame17]. The resulting risk concerns schedule deviations, incomplete execution and reduced modernization quality rather than resource scarcity as an isolated challenge [Red24].

Another organizational risk is modernization failure due to insufficient organizational governance, which concerns the absence of repeatable routines for planning and executing the modernization in itself [Pun26]. This means that unclear governance creates the risk that responsibilities, decision rights and handoffs are inconsistently handled during modernization. This gap between policy modernization intentions and operational implementation may further prevent modernization requirements from being translated into concrete execution [Pun26]. The resulting organizational risk is inconsistent or failed modernization execution even when the required technical capabilities are theoretically available.

5.5. Business Risks

The business risks are on a higher decision level than the other dimensions and often appear as consequences of risks from other dimensions. One general risk is insufficient return on modernization investment, as uncertainty regarding modernization costs and the value ultimately realized creates the risk that investments do not generate the desired business return [GS14]. Additionally, predicting ROI before modernization is difficult because costs, technical complexity and project duration remain partly uncertain before the project starts [Lan]. Therefore enterprises may invest substantial financial resources without realizing corresponding operational or strategic benefits [CIOb]. Inadequate technical and business planning can further lead to cost overruns and growing IT costs [Kha+14]. The central business risk is therefore financial underperformance rather than ROI estimation difficulty itself [Lan].

Connected to the parallel operations mentioned in the process section above, modernization of business-critical legacy systems creates the risk that migration, replacement or transition activities interrupt established operations [US 21]. Such a disruption is particularly critical where legacy systems directly support operational business processes [Ame17] as failures during transition affect not only the modernized application but also the services and business activities that depend on it [Ste]. The resulting business risk is temporary or sustained loss of operational capability during modernization [Ste].

There is also the Risk of escalating legacy system costs when modernization is post-

poned, directly introducing a cost overrun. This emerges because organizations must balance the cost and risk of modernization against the continuing cost and risk of operating legacy systems [Kha+15]. High-value legacy systems can become increasingly expensive and risky to maintain as they age and postponing modernization therefore creates the business risk that maintenance expenditure continues to grow [US 21; CIOa]. Consequently, the decision not to modernize can produce financial problems in addition to the risks associated with conducting modernization itself [Ste].

But modernization also depends on direct financial support, which can in the worst case introduce the Risk of delayed or incomplete modernization due to funding uncertainty [Gar]. Funding uncertainty can be particularly consequential for critical legacy systems where modernization is necessary but competes with other organizational investments [US 21]. As a result, the business risk is prolonged exposure to existing legacy system problems or an incomplete modernization that fails completely to deliver its promised value [Ame17].

Lastly, business risks can also arise from the continued use of legacy systems themselves. Business risk is not just limited to unsuccessful modernization projects, continued dependence on unsuitable legacy systems can itself create negative strategic exposure by restricting an organization's ability to adapt [Paz25; CIOa]. Postponing or avoiding necessary modernization therefore creates the risk of reduced organizational adaptability and lost strategic opportunities [Kha+13; Lin25]. Modernization decisions must consequently consider both the risks of changing the existing system and the risks of maintaining the existing one [Paz25].

5.6. Process Risks

After looking at technical, organizational and business risks, the process-related dimension concerns how modernization is actually planned and coordinated. One of the first important risks is the risk of selecting an unsuitable modernization strategy. Determining the right modernization approach is already a major decision point by itself, because the selected strategy has to fit the technical aspects of the legacy system as well as business and organizational requirements [GS14; Red24]. If this fit is not given, an unsuitable strategy could be selected and the project can develop in a direction that does not match the actual system situation [Lan; Sof]. Since the selected strategy influences many subsequent activities, such a wrong decision can propagate into later technical, organizational or business problems. Therefore, the process risk is not the difficulty of choosing a strategy itself, but the consequences that arise once an unsuitable path is chosen.

Closely connected to strategy selection is the risk of scope, schedule and resource deviations due to insufficient planning. If planning is incomplete, the project scope, sequencing of activities, needed resources, transition activities or expected outcomes can be estimated incorrectly [Rii23; Bal+08; Ste; IBMb]. A problem here is that wrong assumptions during planning may only become visible during execution, when correcting them becomes more expensive and difficult. This can then lead to delays, additional

costs, resource conflicts, incomplete modernization activities or disruptions during the transition itself [GS14; Ste].

Planning alone is also not sufficient when the actual execution is not governed consistently. Modernization needs repeatable routines [Pun26]. Insufficient governance creates the process risk that planning and execution are performed differently across teams and without clearly defined responsibilities. This can lead to coordination failures, unclear ownership or incomplete handoffs between different parts of the whole project. In such a case, even a technically reasonable modernization plan can deviate during execution because there is no consistent process that ensures the plan is followed [Pun26]. Governance deficiencies therefore connect the organizational and process dimension, but from the process perspective the main risk is inconsistent modernization execution.

Another central process-related risk is the risk of undetected defects due to insufficient testing and validation. Testing is necessary to verify that the modernized system still preserves the expected functionality and behavior after changes are introduced [GS14; PE23]. If test coverage is incomplete or validation is not done sufficiently, defects introduced during migration can remain undetected [Van25]. This becomes especially important in already modernized components and parts of the legacy system that still remain untouched [PE23; Van25] as undetected defects can become visible later in following modernization stages, where they create additional rework and disrupt daily business again.

Similar to testing, monitoring is needed to identify problems early enough to still react to them. Insufficient project monitoring creates the risk that deviations, implementation problems or differences between expected and actual outcomes remain undetected [GS14]. This does not stop once the new system is implemented. Insufficient post-implementation monitoring can also lead to operational problems only being detected after they have already created larger consequences [PE23]. The process risk is therefore mainly the late recognition of problems, which reduces the ability to take corrective action before consequences occur.

Lastly, several of the previous process risks can be further increased by having no defined process at all. Literature identifies the definition of an appropriate modernization process as an important concern together with tooling and evaluation support [Ass+25]. Complex activities can be performed inconsistently for that case. When analysis, transformation, testing and validation are not connected with a repeatable procedure, individual activities can be executed in an uncalled manner [Ass+25; Pun26]. This increases the dependence on spontaneous decisions and reduces the reliability of the modernization process in its entirety.

5.7. Cross-Dimensional Risks

Even though the risks were separated into technical, organizational, business and process dimensions before, they do not only occur inside of these individual dimensions. Literature also shows that modernization risks are interrelated and can influence each other [PE23; Paz25; Red24]. This means that a risk from one dimension can increase

either the probability or the impact of another risk in a different dimension. Technical migration and integration problems for example can affect business operations, while organizational problems such as missing user acceptance can still endanger a modernization even when the technical part works [PE23]. Therefore, some risks cannot really be looked at as completely isolated problems.

A good example of this is the connection between organizational knowledge problems and technical risks. As mentioned in the technical dimension, complex legacy systems already make it difficult to fully understand dependencies and system behavior. When the required legacy knowledge is additionally only available through a few employees, this problem is further increased [Kha+15; Com+00; Rah26]. Missing expertise can therefore lead to dependencies being overlooked, wrong assumptions about the system or unsuitable modernization decisions [Ame17]. In this case, an organizational problem increases an already existing technical risk. This can also work in the other direction, as large technical changes can create organizational problems when users are not willing or able to adapt to the modernized system [Kha+14; PE23].

The same can be observed between technical and business risks. A technical defect or integration problem does not stay only technical when the affected system is important for daily business operations [Kha+15; Ame17]. If such problems delay the project, require more development effort or interrupt business-critical functionality, they directly create consequences on the business side as well [GS14; CIOa]. High technical complexity can therefore increase modernization costs or reduce the expected financial benefit of the project. So depending on the affected system, the consequence of one technical risk can become much larger than the original technical problem itself.

Governance is another case where the separation between dimensions becomes difficult. Governance deficiencies affect organizational responsibilities, but at the same time also how modernization activities are planned and executed [Pun26]. When responsibilities or decision rights are unclear, decisions made on an organizational level may not be transferred correctly into the actual modernization process. These problems can then continue into technical execution, schedules, costs and finally business outcomes. Governance can therefore create risks across several dimensions instead of belonging only to one of them.

For the risk knowledge base, the four dimensions are still useful to structure the identified risks, but they should not be understood as completely separate groups. Individual Risk entries describe one specific risk, while Linked Risks and the explicit Risk–Risk and Risk–Mitigation relationships can keep the connections between them. This is important because one condition can create several risks at once and one risk can also trigger consequences in other dimensions. Cross-dimensional relationships therefore show how risks can propagate through a modernization project instead of only showing that different risks exist next to each other.

5.8. Typology of Mitigation Strategies

The following section looks in more detail at the actual mitigation strategies. These can be classified into four different categories, namely Preventive, Detective, Reactive, or Recovery strategies. This classification originates from the Mitigation extraction template used in this thesis. Looking at the complete data, Preventive-oriented strategies form the dominant group while Detective, Reactive, Recovery and combined strategy types occur less frequently. As with the risks, the categories are not mutually exclusive and can be combined.

As the dominant category, preventive strategies also cover many Risk–Mitigation connections because many of the extracted measures aim to act before a risk materializes. Preventive mitigations seek to reduce the likelihood of a modernization risk before it materializes [Frab; Lan]. A prominent strategy here is strong project planning from both technical and business perspectives [GS14]. Because many of the risks discussed above are connected to missing knowledge or insufficient planning, a good modernization plan is necessary. Another mitigation strategy that focuses on controlling the process of modernization is a gradual migration involving systematically modernizing components [Ogu+23]. Risks with organization-wide consequences can be mitigated by taking smaller migration steps and limiting the impact if one of them fails [Ass+25]. One preventive strategy that focuses on organizational risks rather than technical interventions is effective training, which can alleviate some resistance concerns [PE23]. Training can focus on building workforce skills for dealing with modernization and also prepare users through, for example, tutorials or seminars for the modernization stages [Lan; Ass+24]. Another strategy that is frequently used and recommended is documenting the requirements and relevant legacy-system information needed for the modernization, training and possible rollback. This approach is useful not only for the current modernization but also for future modernization activities [Frab].

Detection is the next category. A common strategy for detecting deviations before they escalate is rigorous testing and validation [GS14]. This goes together with monitoring the modernization at every stage and keeping track of deviations [GS14]. A related challenge is providing tooling support, together with defining a modernization process connected to planning and considering better evaluation metrics [Ass+25]. Testing, monitoring, analysis and evaluation mechanisms can reveal defects or deviations during modernization and signal when the modernization moves away from the intended outcome [PE23].

Once problems are detected, reactive measures are needed. For example, monitoring can continue after the actual modernization and focus on the implemented system [PE23]. Controlling the performance of a new system can also provide valuable insights in terms of business, technical and organizational acceptance. One other strategy type is Continuous improvement. As a business grows and market conditions change, a system may need to be refined and enhanced to accommodate new requirements or trading partners. This reaction to changes could arguably also be a mitigation strategy of a preventive type, showing the overlap between different types [IBMa]. However, reactive strategies

modify or improve the modernization solution after problems or changing requirements become visible, including continuous improvement [PE23].

The last category is Recovery. A common example is the rollback mechanism that needs to be incorporated into a modernization plan [Gho+17]. It is important if a modernization step fails and a company has to recover from this mistake [Ogu+23]. Rollback and recovery mechanisms provide fallback options when a migration step or target solution does not operate as intended and are therefore important for big-bang as well as iterative approaches [Van25]. In most cases, these rollback mechanisms are implemented by saving backup files or using a version control system [Van25].

As with risks, mitigation strategies can also span dimensions and strategy types. Mitigation strategies are not limited to technical interventions and include planning, training, documentation, governance, testing, monitoring, incremental migration and recovery mechanisms [GS14; CIOb]. A strategy is not bound explicitly to only one type, as it can act, for example, as a detective strategy for the current modernization but at the same time as a preventive strategy for future migrations.

5.9. Mapping Mitigation Strategies to Risks

What characterizes this contribution is the explicit mapping between risks and mitigation strategies instead of only listing both separately. Such direct mappings already appear in prior work, for example in the adapted risk–mitigation table by Riihimäki [Rii23] shown in Table 3.1. In this thesis, this idea is applied systematically to the complete extracted knowledge base by storing Risk and Mitigation connections as individual Relationship entries. This makes it possible to see which mitigation addresses which risk and whether it prevents, reduces, detects or recovers from it. Hence, the resulting structure keeps the connection between both sides instead of losing it when risks and mitigations are stored as separate entries.

These relationships follow a many-to-many structure, meaning individual risks can be connected to multiple mitigation strategies and vice versa, mitigation strategies can address more than one risk. This can be observed in the knowledge base: within the conservative High-confidence relationship dataset, 294 Risk–Mitigation relationship entries represent 263 unique Risk–Mitigation pairs. 78 Risks are connected to more than one Mitigation, while 74 Mitigations address more than one Risk.

A couple of examples of these relationships can be found among the planning mitigations. Planning measures are used in response to risks that are related to modernization preparation and execution as seen in Section 5.8, therefore demonstrating one mitigation that is linked to many potential risks [GS14; Rii23]. Another example is testing and validation measures that address risks related to functional correctness and migration outcomes, controlling the desired outcome [PE23; Van25]. These risks are linked to multiple mitigation strategies using different forms of testing, showing the other direction of the many-to-many mapping. Other risk dimensions are also covered. Training measures address organizational knowledge and skill-related concerns and counter organizational resistance in the organizational dimension [Lan], while Recovery-oriented mitigations

such as incremental migration and rollback mechanisms address risks associated with disruptive or unsuccessful migration steps [Sea+01].

However, not every relationship is equally clear in terms of effectiveness or confidence because the mappings depend on context. Mitigation entries retain prerequisites and contextual constraints, and applicability conditions can be associated with individual relationships. The literature indicates that mitigation suitability depends on the selected modernization approach and surrounding project context across the complete modernization case [GS14; Gho+17; Ogu+23; Van25].

Furthermore, mitigations come with trade-offs and can introduce new risks, which is important to cover in a holistic analysis. For example, parallel operation can reduce operational risk, but it also increases cost and resource requirements for the organization [Sea+01]. Mitigation strategies can therefore reduce one risk while introducing additional effort or transition complexity [Thoa]. As mentioned before, these trade-offs can create further risks.

5.10. Risk Chains and Secondary Risks

This leads to risk–risk relationships that can create longer risk chains. As before, these risk chains can be across different dimensions because the base challenges were interrelated and there is an intricate interdependence between them [Red24; Paz25]. The reviewed literature provides evidence that risks originating in one dimension can produce consequences in another dimension [Kha+14; GS14; PE23; Paz25; Ame17] and explicit relationships are used to represent such connections where they were identified during extraction. Again the dependencies between these risks must be identified and quantified with the help of relationships which enforces risk–risk connections directly between risks, represented separately from Risk–Mitigation relationships.

Individual Risk–Risk relationships can form chains when the consequence of one risk is related to another identified risk. For example, time overruns can lead to cost overruns and therefore represent a direct relationship of this type [Kha+14]. A verified multi-step example from the extracted Relationship dataset is shown in Figure 5.1.

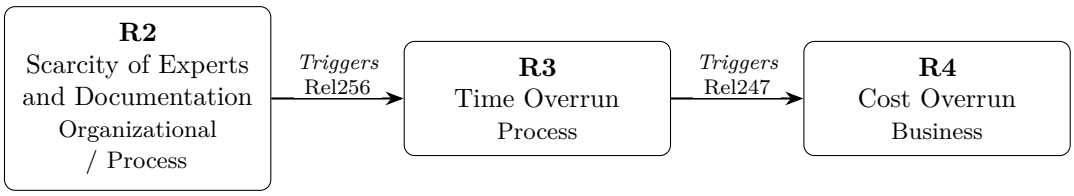


Figure 5.1.: Example of a verified cross-dimensional Risk chain represented by two explicit Trigger Relationships.

The example connects R2 (*Scarcity of Experts and Documentation*) to R3 (*Time Overrun*) through Rel256 and subsequently connects R3 to R4 (*Cost Overrun*) through Rel247. Both Relationship entries are classified with High Confidence.

When looking at risk chains, they can also be started by a mitigation that then produces a secondary risk as Mitigation–Risk relationships allow secondary risks introduced by mitigation strategies to be represented. Mitigation can therefore involve trade-offs rather than purely reducing overall risk which contrasts the simple positive annotation with mitigations [Sea+01; Thoa].

Quickly looking back at the relationship confidence, the initial relationship dataset contains the relationships retained with High confidence while the extended dataset additionally contains Medium-confidence relationships and was not used due to later development and context size limitations. The conservative relationship dataset contains 357 relationships, whereas the extended dataset contains 697 relationships, adding 340 additional relationships. Detailed structural comparison of both relationship dataset with the help of a graph comparison belongs to the evaluation chapter rather than the descriptive RQ1 results.

5.11. Evidence Base

Another important part of the resulting knowledge base is the evidence behind the extracted risks, mitigations and especially their relationships. During extraction, Evidence Type is retained for both Risk and Mitigation entries, while relationships additionally contain an Evidence Type and a Confidence Level. The main evidence categories used in the extraction framework are Statistical, Case Study, Expert Opinion and Anecdotal evidence. These categories cover most of the extracted entries, although some sources use more specific evidence types that differed such as Survey, Literature Review, Systematic Review / Literature Analysis, Audit or Research Opportunity. Nevertheless, keeping these different evidence types is important because not every identified risk or mitigation originates from the same kind of observation and therefore should not automatically be interpreted equally.

One example of quantitative evidence can be found in the industry survey by Red Hat [Red24]. In this survey, the top modernization challenges were complexity of legacy systems with 48%, competing priorities with 42% and determining the right modernization approach with 41%. Such statistical evidence gives an indication how frequently certain modernization challenges occur across different organizations instead of only reporting one individual modernization project. Especially for common risks like legacy complexity or resource competition, this type of evidence can therefore support that the identified problem is not only limited to one specific case.

However, a large part of the extracted evidence also comes from case studies. For example, Khadka et al. [Kha+15] provide observations from five software modernization cases, while Galinium and Shahbaz [GS14] look at legacy system migration from both business and technical perspectives. Other examples are Gholami et al. [Gho+17], investigating challenges related to migrating legacy systems to the cloud and Khadka et al. [Kha+13] which report challenges and lessons from migrating a large legacy application towards SOA (Service oriented architecture). Compared to statistical evidence, these studies do not necessarily show how frequent a risk is, but they give more contextual

information about how and why risks emerge in actual projects.

Grey literature further adds important evidence perspectives. Governmental sources such as U.S. Government Accountability Office [US 21] provide evidence about critical legacy systems and the difficulties connected to modernization planning, while American Council for Technology and Industry Advisory Council [Ame17] covers operational barriers. Industry sources such as Red Hat [Red24] contribute survey results, whereas practitioner guidance like Stevens [Ste] and Software Improvement Group [Sof] describes modernization risks and possible mitigation measures from a more operational point of view. This is especially relevant for this thesis because modernization does not only happen in a scientific context and many practical problems are reported directly by practitioners actively working with legacy system modernization. Grey literature therefore complements academic evidence with more current experiences.

The evidence becomes especially important when looking at the extracted relationships because not every Risk–Risk or Risk–Mitigation relationship is supported with the same strength. For this reason, High- and Medium-confidence relationships are separated instead of treating every extracted connection as equally certain. A relationship that is directly stated by a source should for example not automatically have the same interpretation as one that required more inference during the extraction. Keeping the Confidence Level together with the Evidence Type makes this distinction visible inside the knowledge base and prevents weaker connections from looking equally certain as relationships that are supported more directly.

Overall, the resulting catalogue should therefore not be interpreted as a set of universal rules that always apply to every modernization project. The reviewed literature shows that modernization strongly depends on the system, organization, strategy and surrounding project context [Kha+14; Kha+15; GS14; Gho+17; PE23; Bat+14; Paz25; Ame17; Red24]. A risk that is critical in one modernization does not necessarily have the same importance in another one and the same applies to mitigation strategies and their effectiveness. Therefore, Contextual Constraints, Evidence Type and relationship Confidence Level are retained to keep this uncertainty instead of reducing the extracted knowledge to simple universally valid Risk–Mitigation statements.

5.12. Gathered Knowledge

After covering the individual results from the literature analysis, this section now directly summarizes the findings relevant for RQ1. The previous sections showed that modernization risks can not only be found on one technical level, but instead occur across Technical, Organizational, Business and Process dimensions and influence each other.

Looking first at the risk side, the literature analysis identified recurring modernization risks across all four dimensions. For mitigating these risks, the extracted strategies were separated into Preventive, Detective, Reactive and Recovery strategies. Common measures include planning, incremental migration, documentation, training and knowledge transfer, testing, validation, monitoring and rollback or recovery mechanisms.

Another finding relevant for RQ1 is that risks and mitigations do not follow a simple one-to-one mapping. One Risk can be addressed through multiple Mitigations while one Mitigation can at the same time address different Risks. The usefulness of a Mitigation additionally depends on its surrounding modernization context, prerequisites and the specific Risk it should address. Risks themselves are also interdependent and can form Risk Chains, while Mitigations can introduce additional secondary Risks.

Evidence and uncertainty also have to be considered because not every finding has the same empirical strength and not every Relationship is equally directly supported. The resulting modernization knowledge should therefore not be understood as a fixed collection of universal rules but as literature-grounded findings depending on the system and modernization context.

Overall, the literature analysis provides the empirical basis of Risks, Mitigations and their corresponding Relationships used for the structured knowledge representation developed in the following chapter.

6. Development of the LLM-Oriented Knowledge Model

After finding the most important risks, mitigations and relationships in the previous chapter, this chapter develops the conceptual knowledge representation used to structure the modernization knowledge identified through the literature analysis. While Chapter 5 primarily addresses which risks, mitigation strategies and relationships were identified, this chapter addresses how this knowledge should be represented for LLM-based analysis. It therefore addresses RQ2 (Section 1.4) by developing the knowledge model and serialization, using the risks, mitigations and relationships identified for RQ1 as its basis.

Moving from the extracted knowledge to an actual knowledge model requires some additional structure. The literature extraction showed that modernization knowledge consists not only of isolated risks and mitigation strategies but also of causes, triggers, impacts, contextual conditions, evidence, trade-offs and relationships between individual knowledge elements. A suitable representation must therefore preserve both information about the individual entities and the connections between them for LLM analysis. As already used during the extraction phase, Risks, Mitigations and Relationships are represented as explicit knowledge objects in the knowledge model.

Two important aspects for the knowledge model are machine readability and that it remains manually inspectable and editable. The model should retain source information as well as contextual conditions. Relationships and uncertainty are represented explicitly rather than remaining implicit in free text. Additionally, standardized fields are included to make heterogeneous literature findings easier to process consistently by an LLM.

The following sections derive the representation requirements, describe the Risk, Mitigation and generalized Relationship models, define the relationship semantics, explain evidence and confidence handling and describe the LLM-oriented representation and serialization.

6.1. Requirements for the Knowledge Representation

Some requirements are necessary to involve before developing the actual models, for example the need for an explicit scheme. Because knowledge extracted from the literature originally occurs in heterogeneous textual forms and uses inconsistent terminology and levels of detail, a common scheme is required to transform these heterogeneous findings into comparable knowledge objects for later processing. Consequently, the representation should hold up to two standards: Which information is stored and what each field semantically represents.

As mentioned in the introduction, machine readability and structural consistency form the base for processing the knowledge in any type of application. The representation must therefore use recurring fields and predictable structures so that individual knowledge entries can be parsed and processed automatically. As this thesis covers three different knowledge types, Risks, mitigation strategies and relationships should be represented through clearly distinguishable entity types. Some fields hold multi-valued information, such as stakeholders, tags, related risks, strategies or evidence references, so they should be represented consistently. Missing information should remain explicitly absent or unspecified rather than being completed without evidence from a literature source.

The other aspect mentioned is human readability which is closely connected to maintainability. The knowledge base should remain manually inspectable because literature-derived knowledge may require correction, consolidation or extension at some point. Domain-specific descriptions should therefore remain readable without requiring a dedicated database interface that first transforms the data in a new way. Hence most importantly the representation should support later manual extension when additional literature or expert knowledge becomes available.

This contribution values traceability and provenance when it comes to data representation, meaning every Risk and Mitigation entry should remain traceable to the literature from which it was extracted. For that, stable IDs allow individual knowledge objects to be referenced independently from their descriptive names because paper IDs and supporting quotations preserve the connection between standardized knowledge entries and their original evidence in literature. Relationship-level provenance should also be retained wherever possible because modeled connections can involve additional interpretation beyond the individual entities.

What is new in this thesis is the explicit relationship representation. Modernization risks cannot be represented adequately as an independent list because risks may trigger other risks and mitigations may affect multiple risks, so the representation must support explicit and directed relationships. Those Relationships should themselves carry attributes because applicability, evidence, confidence and observed effects belong to the connection rather than exclusively to either endpoint, independent if it is a risk or a mitigation due to their many-to-many relation.

Another requirement on the knowledge representation involves context and uncertainty. Modernization knowledge is context-dependent and should not be represented as universally applicable to every specific scenario. Applicability conditions, prerequisites, modernization strategies used, dimensions and other contextual information therefore need to be retained in the model. Evidence and confidence should be represented separately so that strongly documented knowledge can be distinguished from weaker relationships.

Regarding the requirements for LLM usage, the representation should reduce ambiguity when supplied to an LLM. Individual entities and relationships should be identifiable through stable IDs and important information should also be available in short standardized natural-language representations. Conditional knowledge should be represented in a

consistent form where possible. Very importantly, the model should make unsupported or missing information identifiable instead of encouraging the generation of plausible but undocumented values.

6.2. Entity Model

The knowledge base consists of several hundred entities and therefore is an entity-oriented representation. Structured knowledge representations commonly organize information around identifiable entities and relationships and this approach was applied in this thesis too. This entity-oriented perspective provides the conceptual basis for separating modernization knowledge into distinct objects.

The main knowledge objects are categorized according to the three distinctions that were used throughout this thesis: Risks, Mitigations and Relationships. Risks represent potential negative modernization developments or events, while Mitigations represent actions that are intended to prevent, reduce, detect, respond to or recover from such risks. Relationships represent explicit directed connections between the other knowledge objects and are therefore used to describe how Risks and Mitigations are connected.

The sources from which the different objects were extracted are kept as provenance information. Scientific publications and grey Literature sources are referenced through Paper IDs as they are in this case not modeled as a separate entity, which would be conceptually possible. In the current representation, source information is primarily retained as provenance metadata rather than as an independent knowledge object. This is done as it keeps the core model focused on modernization risk knowledge while maintaining traceability to the literature.

Information describing an individual Risk or Mitigation is stored in the corresponding entity. In those entities, a template structure was used that will be covered in the next sections. In the risk and mitigation template are some fields that were used in early developing stages, namely The fields Linked Risks and Related Risks which are retained as lightweight cross-references originating from the extraction and consolidation process. These fields indicate that another entity is associated with the current one, but they do not define the semantics, evidence or confidence of the whole connection. However, typed relationship objects provide the representation for explicit relationship claims within the knowledge model. Where such a typed relationship exists, Source, Target, Relationship Type, Applicability Conditions, Evidence Type, Confidence Level and Reasoning Notes help determine the interpretation of the given connection. Nevertheless, such a lightweight reference without a corresponding Relationship can indicate an association, but should not be interpreted as evidence for a concrete relationship type.

One more general part of the representation are the stable identifiers assigned to every knowledge object. Every Risk, Mitigation and Relationship gets its own unique identifier. Risk identifiers use the R prefix, while Mitigation identifiers use M and Relationship identifiers use Rel as a prefix. This makes it possible to explicitly reference individual knowledge objects, especially when they are later used in LLM context.

The resulting representation therefore consists of three main knowledge-object types

that share a common design philosophy while capturing different kinds of modernization knowledge. Figure 6.1 summarizes this conceptual structure before the individual object models are described in the following sections.

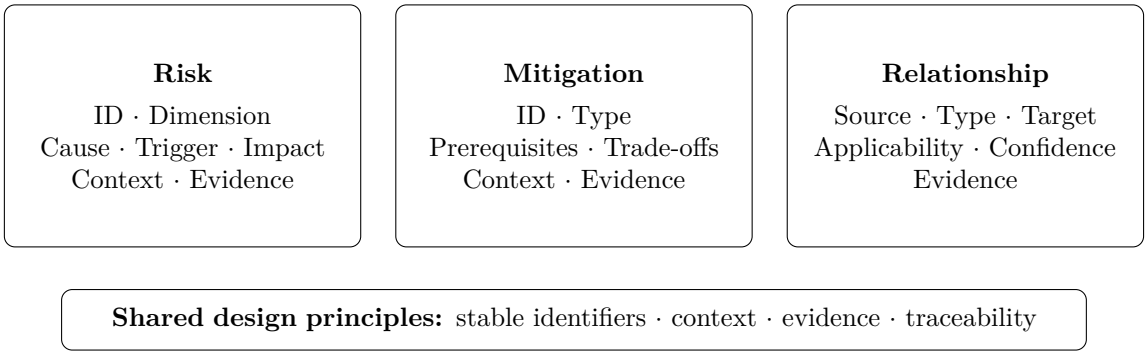


Figure 6.1.: Conceptual overview of the three main knowledge-object types and their core information. The complete field sets are described in the following sections and in the extraction templates.

6.3. Risk Model

The first template is developed to represent Risk entities. The Risk model represents an identified modernization risk as a structured knowledge object instead of only keeping it as an isolated statement or literature quotation. Even though the information is derived from literature, the different fields are used to give the extracted Risk a more standardized structure. The model therefore captures the Risk itself, its context, possible causes, triggers, consequences and also the available evidence.

Starting with the general identification fields, every Risk receives a unique *Risk ID*. The *Paper ID* keeps the connection between the Risk and the literature source it was extracted from. The *Risk Name* gives the Risk a short and standardized name, while the *Risk Description* explains what the Risk actually represents and why it can become problematic during modernization. This is especially important because different papers do not always use the same terminology for similar modernization problems and these fields help distinguish the standardized knowledge base entry from individual publications.

The next fields give more information about the context of the Risk. *Modernization Strategy* describes in which modernization approaches the Risk is relevant. The *Dimension* categorizes it into Technical, Organizational, Business or Process, based on the four dimensions used throughout this thesis. Additionally, the *Granularity Level* describes whether the Risk mainly occurs on enterprise, system, component or code level. Together, these fields give additional context instead of treating every extracted Risk as generally applicable. They make it possible to distinguish risks that may have similar consequences but arise in different modernization contexts.

Another important part of the Risk model is the distinction between *Root Cause /*

Challenge, *Trigger Event* and *Impact / Consequences*, as described in Chapter 2. The Root Cause represents an underlying problem or condition that increases the possibility of a Risk. The Trigger Event describes an action or situation through which the Risk can become active, while the Impact describes what happens if the Risk actually materializes. This gives the Risk a simple causal structure:

$$\text{Root Cause / Condition} \rightarrow \text{Trigger} \rightarrow \text{Risk} \rightarrow \text{Impact}. \tag{6.1}$$

This distinction is useful because causes, actual Risks and consequences can otherwise easily get mixed when extracting information from different literature sources. For later LLM use it also gives a more explicit structure, instead of requiring the model to derive those parts again from one longer description.

Some additional fields describe how the Risk is connected to its surrounding context. *Linked Risks* contains other Risks which were identified as being related to the current one. However, this should only be seen as a lightweight reference. It does not describe the exact relationship between two Risks. Explicit Relationship entities provide a more detailed and typed representation of these connections through the Relationship model described later. *Stakeholders* identify actors that are affected by or responsible for the Risk, while *Contextual Constraints* describe under which conditions the Risk is especially relevant in an organization.

Furthermore, the Risk entity also contains information related to evidence and possible detection. *Evidence Type* describes what kind of evidence supports the Risk, for example Statistical Evidence, a Case Study, Expert Opinion or Anecdotal Evidence. *Detection Source* stores where information for detecting or assessing the Risk can be obtained in practice, for example source code, architecture documentation, interviews, project plans or operational data. *Key Terminology / Tags* are additionally used to retain important terms for filtering and later retrieval/guidance.

Lastly, the template contains some fields which are especially included for later LLM-based usage. The *Standardized Risk Entry* summarizes the Risk in one or two sentences and therefore gives a shorter and more consistent version of the extracted knowledge. The *Key Quote for Prompting* keeps an important quotation from the underlying literature and the *Logic Pattern* represents the Risk in the standardized form

$$\text{IF [Condition] AND [Trigger] THEN [Risk]}. \tag{6.2}$$

These fields do not replace the more detailed description but provide another representation that can be easier to process when constructing LLM input.

A later extension of the Risk model is the addition of cluster information. This field does not come directly from literature but is derived later from the Knowledge Graph and its clustering. Because this extension was added after the comparative LLM evaluation was already completed, the cluster information was not part of the Risk objects used in the reported evaluation. It therefore represents a later version of the developed artifact.

6.4. Mitigation Model

The second template represents Mitigation entities. A mitigation is modeled independently from a particular risk because the same strategy may address multiple risks and may have additional trade-offs or secondary effects. One Mitigation can address several different Risks and at the same time, one Risk can also have multiple possible Mitigations.

Closely related to Risks, every Mitigation receives a unique *Mitigation ID*. The *Paper ID* again keeps the reference to the literature source, while *Mitigation Name* provides a short standardized name. The *Mitigation Description* then explains the actual action or strategy which is proposed to deal with modernization Risks.

The *Related Risks* field provides a lightweight indication of risks associated with the mitigation and supports direct navigation and initial alignment between the two entity types from earlier development stages. However, Related Risks alone does not describe how the mitigation affects a risk. For example, the same field is not able to distinguish if a Mitigation prevents a Risk, reduces or helps detecting it. Those more detailed semantics are then represented through explicit Relationship objects.

The *Strategy Type* categorizes the Mitigation as Preventive, Detective, Reactive or Recovery. This allows a distinction between Mitigations which are applied before a Risk happens and strategies which are more relevant during or after the Risk already materialized.

Some Mitigations also require certain conditions before they can actually be applied. The *Prerequisites* field therefore describes what needs to exist, similar to the contextual constraints, before implementing the Mitigation. *Tools / Technologies* contains technical tools, frameworks or other technologies that are explicitly mentioned in connection with the strategy. Keeping these fields in the model adds practical context and avoids representing a Mitigation completely independent from the environment in which it is applied.

There are also fields used to describe the expected effect and required effort of a Mitigation. *Effectiveness* records information about the reported or expected effect of the strategy. *Probability of Success* stores a qualitative value of Low, Medium, High or Not specified if such information is available. *Effort of Mitigation* similarly describes the expected effort with Low, Medium or High that will come up if the strategy is applied. These values are only used when they can be supported by the underlying literature. If the literature on the other hand does not provide this information, the value stays unspecified instead of estimated in some way.

A very important field is *Trade-offs / Side Effects*. Applying a Mitigation does not automatically mean that the overall modernization situation improves. A strategy can also increase complexity, require additional resources or introduce new dependencies or risks. If such a side effect itself represents another modernization Risk, the connection can later be modeled explicitly through a Mitigation–Risk Relationship.

The remaining contextual fields follow the same idea as in the Risk model. *Stakeholders* describe who implements or benefits from the Mitigation, while *Contextual Constraints*

specify in which situation the strategy can actually be applied. *Evidence Type* describes the type of evidence supporting it and *Key Terminology / Tags* keeps important concepts for filtering and later retrieval.

For LLM-oriented use, the Mitigation model again contains a *Standardized Mitigation Entry*, a *Key Quote for Prompting* and a *Logic Pattern*. The standardized entry provides a shorter version of the Mitigation while the Key Quote keeps the connection to the original literature wording. The Logic Pattern represents the application in the form

$$\text{IF [Condition] THEN APPLY [Mitigation].} \tag{6.3}$$

As with the Risk Logic Pattern, this should not be understood as a formally verified rule. It is mainly a standard representation of the knowledge that was previously extracted from literature sources.

Cluster information was also added to Mitigation entities in a later version of the artifact. As with Risks, the information is derived from Knowledge Graph analysis rather than directly from literature and was not included in the Mitigation objects used for the comparative LLM evaluation.

6.5. Generalized Relationship Model

The third knowledge object is the Relationship. Relationships are especially relevant for the developed knowledge model because modernization knowledge does not only consist of individual Risks and Mitigations. The connections between those entities also contain information which is necessary to understand how one problem can lead to another or how a Mitigation can influence a Risk. Because of this, Relationships are represented as their own knowledge objects instead of only being stored as references inside Risk or Mitigation entries.

Modeling Relationships separately also makes it possible to store information which belongs specifically to the connection between two entities. This includes, for example, the Relationship Type, the strength of the influence, conditions under which the connection applies, the available evidence and the confidence in the modeled connection. Storing this information only on a Risk or Mitigation would be difficult because those attributes do not necessarily describe either entity individually. They describe how two entities are connected. Therefore, Relationship modeling is not only used for visualization in a Knowledge Graph but already forms part of the actual knowledge representation.

The general idea follows the common representation of structured relational knowledge through entities and relations. In a Knowledge Graph, entities can be represented as nodes and their relations as directed edges between them. Such relational facts can also be represented in the form of a triple consisting of a source entity, a relation and a target entity. The generalized Relationship used in this thesis follows the same basic structure:

$$\text{Source Entity} \rightarrow \text{Relationship Type} \rightarrow \text{Target Entity.} \tag{6.4}$$

The original Relationship template was mainly designed around Risk–Mitigation con-

nections. During the development of the knowledge model, however, this was found to be too limited. Modernization Risks can also lead to other Risks, especially when multiple problems form a chain-like structure. Furthermore, Mitigations can also create secondary Risks instead of only having a positive effect. A secondary Risk in this model refers to a new Risk that can arise as a consequence or side effect of applying a Mitigation. Because of this, the Relationship model was generalized so that Source and Target can represent either a Risk or a Mitigation for the supported combinations, apart from Mitigation–Mitigation relationships.

For example, a Mitigation can reduce the probability or impact of one Risk while at the same time introducing another secondary Risk. Similarly, one Risk can trigger another Risk and thereby form a longer Risk chain, as illustrated by the verified example in Figure 5.1. These cases require the connection itself to be represented explicitly instead of only storing the involved entities separately.

A simplified view of the supported connections is shown in Figure 6.2.

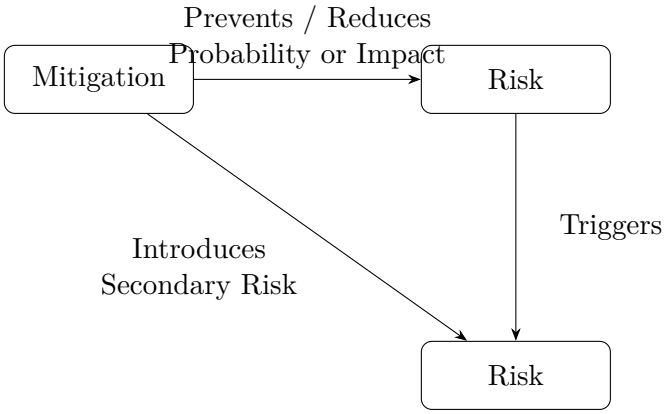


Figure 6.2.: Simplified examples of Relationships between the main knowledge entities.

Every Relationship receives its own *Relationship ID* again to identify it uniquely. Instead of only using Risk ID and Mitigation ID as in the first version of the template, the generalized model uses *Source ID* and *Target ID* combined with the additional fields *Source Type* and *Target Type*, which describe whether the connected object is a Risk or a Mitigation. This allows the same structure to be used for different combinations without creating a separate Relationship model for every possible case.

The *Relationship Type* describes what the connection between Source and Target actually means. For relationships between Risks and Mitigations, the stored relationship types include Prevents, Reduces, Detects, and Recovers. In the implemented representation, the Risk is stored as the Source and the Mitigation as the Target. The relationship type therefore describes the effect of the target Mitigation on the source Risk. For example, Reduces means that the target Mitigation lowers the probability, severity, or impact of the source Risk rather than necessarily removing the Risk completely. Mitigation–Risk relationships are used when a Mitigation creates a secondary Risk. For these relationships, the Mitigation is the Source, the Risk is the Target, and the relationship

type Introduces is used. For Risk–Risk connections, types such as Triggers, Amplifies or Contributes can be used. Keeping a clear direction is important here because it should always be possible to read the Relationship directly from Source to Target.

Additional information is stored in form of *Influence Strength*, *Applicability Conditions*, *Observed Outcome* and *Secondary Risk Created*. Influence Strength provides a qualitative indication of how strong the connection is considered, Applicability Conditions describe under which circumstances the Relationship holds, Observed Outcome stores a reported effect where this information is available in the used literature and Secondary Risk Created is especially useful when a Mitigation introduces another modernization problem.

The model also needs to represent that not every identified Relationship has the same level of support or trust. As in the other templates, *Evidence Type* therefore describes the type of evidence for the connection, while *Confidence Level* represents how confident the modeled Relationship is based on the available information. Especially when a Relationship is not directly stated in one source but is created through interpretation or by combining multiple findings, *Reasoning Notes* are used to make this process more transparent. The *Standardized Relationship Entry* finally gives a short summary of the complete connection for later LLM usage.

One current limitation is the provenance of Relationships. Risks and Mitigations have their own Paper IDs, while the Relationship model currently does not contain an independent Paper ID field. The evidence therefore partly depends on the connected entities and the information documented in the Reasoning Notes. Adding Supporting Paper IDs or direct Evidence References to Relationships could improve this in a later version of the knowledge model.

The generalized structure also allows Relationships to be connected into longer paths. For example, one Risk can trigger another one, which again contributes to a third Risk. Mitigations can then be connected to different parts of such a chain. This provides the structural basis for multi-step and relationship-oriented queries with the LLM.

6.6. Relationship Types

The general Relationship model already gives a suitable description for the template. However, there are some additional requirements to ensure an unambiguous relationship direction. The basic structure follows a directed relationship with a consistent Source–Predicate–Target interpretation. For all three categories of relationships, the relationship type should semantically describe how the Source affects or relates to the Target. This convention is necessary so that relationship direction remains interpretable both by humans and by automated processing.

When distinguishing different categories of relationships, the dominant category in this thesis is the risk–mitigation category. Here, the Risk represents the Source and the Mitigation represents the Target. The stored relationship type describes how the target Mitigation affects the source Risk. The following standard relationship types are used:

- Prevents: The target Mitigation can prevent the source Risk.
- Reduces: The target Mitigation can reduce the probability, severity, or impact of the source Risk.
- Detects: The target Mitigation can support the detection of the source Risk.
- Recovers: The target Mitigation supports reactive behavior after the source Risk has materialized.

The second category is particularly important for representing risk chains, namely the risk–risk relationship category. In this case, both Source and Target are risks, and the relationship type describes how the source risk influences the target risk. The following types are used:

- **Triggers:** Materialization of the source risk can activate the target risk – not a guarantee.
- **Amplifies:** The source risk increases the probability or, more often, severity of the target risk.
- **Contributes To:** The source risk is one of multiple factors contributing to the target risk, which is of higher granularity.
- **Depends On:** The existence or materialization of the source risk depends on the target risk or a condition represented by it.

Thirdly, there is the mitigation–risk relationship category, which concerns risks that can be introduced or amplified by applying a mitigation strategy. Here, the mitigation represents the Source and the affected risk the Target. It uses the following relationship types:

- **Introduces:** Applying the source mitigation creates or contributes to a new target risk.
- **Amplifies:** The source mitigation increases the probability or, more often, severity of the target risk.

The relationship types are intentionally limited to meanings that can be consistently distinguished from the available evidence. The directional structure is not only easier to read for the user. It also reduces ambiguity for graph visualization and simplifies the construction of risk chains and multi-hop paths. Furthermore, those directional Relationships allow an LLM to interpret a relationship without inferring which entity is performing the described action to which entity, which also improves the interpretability of structured machine-readable outputs.

6.7. Evidence, Confidence, and Interpretation

In Relationships, it is especially important to preserve uncertainty. Because modernization knowledge originates from sources with different forms and strengths of evidence, there are different weights when looking at what to consider. Some relationships are directly reported by empirical studies or case studies, whereas others are inferred through the combination of multiple finding sources including grey literature. Modernization knowledge therefore needs to preserve context and uncertainty rather than representing every extracted statement as equally applicable.

One of those fields that help represent importance is the Evidence Type. Evidence Type records the form of evidence underlying for an entity or Relationship, with examples including Statistical Evidence, Case Study, Expert Opinion or simple Anecdotal Evidence. It is a rather descriptive value than a scalar value which is why a Case Study is not automatically assigned a fixed confidence value, nor is Statistical Evidence automatically treated as sufficient for High Confidence.

However speaking about confidence, the evidence type plays an important role for assessing the confidence score, which applies particularly to modeled Relationships. It represents the degree of confidence that the modeled connection accurately reflects the available literature evidence and stands for the importance of said entity. The assignment, additionally to evidence type, considers directness of the reported connection, empirical support, consistency with the modeled Source and Target, observed outcomes and the amount of analytical interpretation required. Confidence Level is therefore a qualitative assessment of the support, detailed in Chapter 4.

Furthermore, information directly reported by the literature should be distinguished from relationships derived through analytical interpretation, which is why Reasoning Notes provide an explicit location for documenting the rationale behind an interpreted connection. This is used as it improves transparency when a relationship combines evidence from multiple passages or entities. The knowledge model therefore also distinguishes between source-extracted information, analyst-coded classifications, normalized or transformed representations and analytically derived metadata as defined in Chapter 4, important because a standardized field or derived attribute should not be interpreted as a verbatim finding.

Similar to risks and mitigations, the relationship template also guarantees for no added artificial information as missing probabilities, effectiveness values or applicability information remain unspecified if not presented in literature. Again, this is particularly relevant for LLM use because explicit absence provides a basis for distinguishing undocumented knowledge from available knowledge.

However in some cases different sources may provide different or partially conflicting observations. Such differences should not automatically be collapsed into one apparently certain statement and should just be merged if adding to each other. Additionally, separate evidence references, contextual constraints, confidence assessments or multiple Relationship entries can be retained where necessary.

6.8. OKF-Based Serialization

Some of the fields described are included specifically to support later LLM use. Standardized Entries and Logic Patterns provide recurring representations of the extracted knowledge while stable IDs allow Risks, Mitigations and Relationships to be referenced easily. Contextual information, evidence and confidence additionally help make applicability and uncertainty visible instead of leaving them implicit in free text. These fields complement the more detailed source information rather than replacing it.

The next step is to store this conceptual knowledge model in a representation that remains machine-readable and manually inspectable. For this, an OKF-based Markdown representation was selected for this purpose as it combines structured metadata with natural-language content, described in Chapter 2. The purpose of this section is therefore not to introduce OKF again, but to describe how the developed knowledge model is represented literally.

Each Risk, Mitigation and Relationship is stored as an individual Markdown file, which follows the entity-oriented structure of the knowledge model rather than storing the complete base in one monolithic file. The file names are based on the stable object IDs, for example `R1.md`, `M1.md` and `Rel1.md`, creating a direct connection between the conceptual knowledge objects and their persistent representations. Individual objects can consequently be inspected, modified or extended in the future independently, without altering big files.

Each one of those Markdown files combines YAML frontmatter with named Markdown sections, with compact and recurring metadata stored in the YAML frontmatter. Depending on the object type, this includes information such as ID, entity type, dimension, modernization strategy, Evidence Type, Confidence Level, tags, and Relationship Source and Target references. Multi-valued attributes such as Paper IDs, tags, strategies, or related entities can be represented as lists when written in the frontmatter. Longer descriptive fields remain in named Markdown sections, for example Risk Description, Root Cause, Trigger Event, Impact, Mitigation Description, Trade-offs, Applicability Conditions, Reasoning Notes, Standardized Entries and Logic Patterns. The exact fields depend on whether the file represents a Risk, Mitigation or Relationship, with examples found in the appendix of this thesis.

The combination of structured metadata and textual sections supports the requirements defined at the beginning of this chapter. YAML gives the files recurring field boundaries that can be parsed automatically, while Markdown keeps the longer literature-derived descriptions readable and editable. Stable IDs and Paper IDs keep the references between the different objects and their underlying sources. At the same time, the file-based structure allows the knowledge base to be maintained and extended when new literature, relationships or later derived information is added. The representation therefore serves as the persistent form of the knowledge model and not only as an export format.

An important distinction is required between the persistent OKF-based representation and the representation actually supplied to the LLM. Moreover, the Markdown files

are not passed unchanged to the LLM in the structured-context configurations, but during execution, the files are read and parsed into typed Python Risk, Mitigation and Relationship objects, which are then fed into further-processing LLMs. OKF-based Markdown therefore represents the storage layer that can be used for many applications and representations for example into graph or retrieval representation, while the parsed objects form the basis for the runtime context. The term "OKF-based" is used because the representation follows the general OKF approach of individual Markdown objects combined with structured YAML metadata.

6.9. Prompt and Output Templates

The knowledge model is not only supposed to work for completely open questions but also for some recurring types of modernization analysis. For this, prompt templates can be used which match the information stored inside the Risks, Mitigations and Relationships. The exact technical prompts are implemented later in the prototype and are therefore covered in Chapter 7. Here, the focus is more on what kind of questions the developed knowledge model should support.

A simple use case is questions about one specific Risk or Mitigation. The user can for example ask for the Root Cause, Trigger Event, Impact, Stakeholders or Contextual Constraints of a Risk. Similar questions can be asked for Mitigations, including Prerequisites, Effectiveness or possible Trade-offs. The IDs are useful here because they give a direct way to reference one specific object without depending only on the name of the Risk or Mitigation.

More relevant for this thesis are questions connecting Risks with their Mitigations. In this case, the answer should not only return which Mitigation is related to a Risk, but also what this connection actually means. The Relationship Type can therefore show whether the Risk is *Prevented By*, *Reduced By*, *Detected By* or *Recovered By* the corresponding Mitigation. If available, the Relationship ID and Confidence Level can additionally be included to make the answer more traceable.

The generalized Relationships also make queries about risk chains possible. One Risk can trigger or contribute to another Risk and this can again continue with another one. A query can therefore ask for a chain such as R2 $\rightarrow$ R3 $\rightarrow$ R4. In this case, the individual Relationships between the Risks are also relevant because just listing the Risk IDs would not explain why the chain exists.

There are also scenario-based questions where the user does not know the IDs beforehand. A modernization situation can first be described and the LLM can then use the supplied knowledge to identify fitting Risks, Mitigations and Relationships. This is closer to how such a system could later be used in practice. However, the result should stay inside the available knowledge base and should not add additional Risks only because they would sound reasonable for the described scenario.

Another possible question concerns secondary Risks created by Mitigations. As described before, applying a Mitigation does not automatically have only a positive effect. Trade-offs or new Risks can be introduced as well. The combination of the *Trade-offs*

/ *Side Effects* field and the Mitigation–Risk Relationships allows those effects to be included in such a query.

Missing information also has to be handled in the prompts. If the knowledge base does not contain a probability, an effectiveness value or a certain Relationship, the LLM should return that this information is not documented instead of estimating a value on its own. This is especially relevant because otherwise missing information could easily lead to an unsupported answer.

Depending on the use case, the output can also be requested in a more structured form such as JSON. Risk IDs, Mitigation IDs, Relationship IDs, Relationship Types or Confidence Levels can then be returned as explicit fields. Unknown information should again stay unknown or not documented instead of being replaced with generated values.

These different query types form the basis for using the developed knowledge model with an LLM. The concrete questions used for validation and evaluation, together with their scoring and metrics, are handled separately in Chapter 4.

6.10. Derived Design Principles

Based on the developed knowledge model, modernization knowledge should combine clearly separated knowledge objects such as Risks and Mitigations with explicit Relationships, contextual information and standardized representations. The goal is not only to make the knowledge machine-readable. Enough information around every object should also be kept so that the original meaning from literature does not get lost.

One central principle is the separation between Risks, Mitigations and Relationships, including that Risks and Mitigations retain their own descriptive, contextual, evidence and provenance information, while Relationships are represented independently instead of only being mentioned inside another object. This is especially important because the connection between two objects can have its own Relationship Type, Confidence Level, Applicability Conditions or Reasoning Notes which would otherwise be difficult to assign if simply modeled implicitly.

Another important point is the direction of Relationships. Every Relationship follows the Source–Predicate–Target structure introduced before. Risk–Risk Relationships can represent Risk Chains, Risk–Mitigation Relationships show how a Risk can be addressed by a Mitigation and Mitigation–Risk Relationships can represent secondary Risks that are introduced or amplified through a Mitigation.

The representation also needs to keep context and uncertainty instead of treating every extracted statement as a proven fact from literature. Contextual Constraints, Applicability Conditions, Evidence Type and Confidence Level are used for this. If information is not available in the literature, it stays undocumented in the model. Standardized Entries, Logic Patterns and stable IDs add some further structure for later LLM usage.

For storing the complete model, the developed schema is represented through the OKF-based structure. It combines structured metadata with longer textual fields. This makes the knowledge machine-readable while the individual files can still be inspected and changed manually.

7. Prototype Implementation

After establishing the knowledge base in the previous chapter, this chapter describes the technical realization of the prototype for analysis. Whereas Chapter 6 defines how modernization knowledge should be represented on a conceptual basis, this chapter explains how this representation was converted into an executable software artifact. Therefore, the implementation connects the literature-derived knowledge base with Python-based processing, an LLM interface, a web application and a graph visualization of the knowledge base.

The implementation follows the processing chain established during the artifact development. The filled-out textual templates first needed to be converted into OKF Markdown files and then parsed into Python representations, which are used by the LLM together with user input. The same parsed knowledge objects additionally provide the basis for Knowledge Graph generation and evaluation-related functionality. What exactly the model receives as context from the parsed knowledge base is controlled by the user through an interface that also provides an ID inspection window directly showing the relevant knowledge base entries.

For context selection, the prototype implements four different context configurations. These configurations make it possible to use the same application architecture while varying the knowledge representation supplied to the LLM, which is especially useful for testing the performance of different contexts. The configurations also form the technical basis of the comparative evaluation described in Chapter 8.

The prototype is also accessible through a command-line implementation, which provides the basic interaction and processing workflow. The Streamlit-based web interface provides a more accessible interface for selecting context configurations, models and evaluation tasks. The following sections describe the system architecture, data conversion, Python data models, parsing, context construction, API integration, graph generation, web interface, end-to-end query processing and reproducibility of the prototype.

7.1. System Architecture

From a general perspective, the prototype can be understood as a sequence of four loosely separated processing layers:

1. Source and knowledge data
2. Conversion and parsing
3. Context construction and LLM

4. User-facing and visualization components

This separation allows the literature-derived knowledge base to remain independent from the model used for question answering and keeps the knowledge representation reusable across models.

The data layer's main concern is representing both the original textual data and a usable structured representation. The repository retains both the original extracted text representation and the converted structured knowledge base after conversion into the OKF format. Original extracted data is stored under data/raw while structured Risk entries are stored under data/risks, structured Mitigation entries under data/mitigations and corresponding structured Relationship entries under data/relationships. Each knowledge object is uniquely represented by a separate Markdown file identifiable through the proposed IDs in Chapter 6.

The processing layer is first used to transform the original line-oriented template data into the OKF-based Markdown representation. After that, parser modules reconstruct Risk, Mitigation and Relationship objects from the Markdown files into an internal structure. Python dataclasses provide this common internal representation that can be reused by the LLM component, Knowledge Graph component and evaluation functionality.

The application layer of the prototype consists mainly of three parts. A context-selection component first determines which knowledge representation is supplied to the model. The second part is an LLM component that constructs the request and communicates with the external chat-completions API. The selected knowledge representation is supplied to the LLM as complete prompt context. Finally, the generated response is presented through either the command-line interface or the web interface.

Visualization components then form the last part of the layers, with the graph representation being the most dominant one. The graph uses the same Risk, Mitigation and Relationship objects to construct a directed network, while a separate clustering step can derive cluster assignments from the same objects. The web interface additionally contains functionality for inspecting IDs referenced by generated answers, originally designed to aid the expert evaluation. The full pipeline can be found in the following Figure 7.1.

7.2. Conversion of the Extracted Data

The conversion process starts with the structured text input produced during the manual extraction process. Risks, Mitigations and Relationships follow the extraction templates defined in Chapter 4; an example of a filled-out literature template in raw data can be found in the appendix, as well as the template itself and a corresponding OKF example. The raw files retain the original field labels such as Risk ID, Risk Description, Mitigation ID, Relationship Type and Confidence Level. Keeping the original text files provides an intermediate representation between manual extraction and the final machine-readable knowledge base and also provides a simple starting point for extensions of the knowledge base.

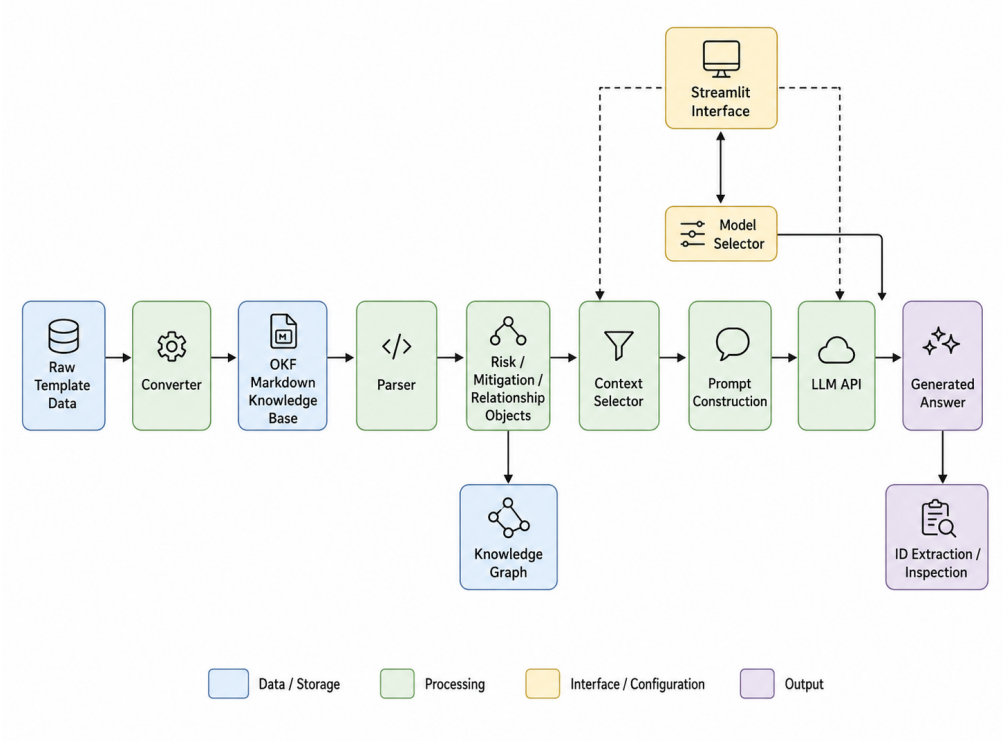


Figure 7.1.: Illustration of full pipeline architecture developed in this thesis.

These raw text files are converted with the help of separate converter scripts implemented for Risks and Mitigations. Relationships are converted using separate scripts for the different endpoint combinations Risk–Mitigation, Risk–Risk and Mitigation–Risk. Although separate converters are used at this stage, their output follows the generalized Relationship representation. Looking more closely at the converters, they divide the raw text into individual knowledge blocks using the corresponding ID field as a delimiter. Risk entries are identified by the occurrence of "Risk ID:", for example, and Mitigations and Relationships follow the same principle. Each block is subsequently interpreted as one knowledge object. Having identified those blocks, individual template fields are parsed as key-value pairs, as each line is separated at the first colon while preserving additional colons occurring within the field value. The implementation therefore assumes that the extracted literature template follows the expected line-oriented structure.

At this point, there is a normalization of multi-valued fields. Selected values are then converted into lists before serialization, for example Paper IDs, Modernization Strategies and Evidence Types. Commas and for selected fields, slashes are used to separate multiple values. This normalization reduces syntactic variation in the resulting knowledge files.

One important step is mapping fields to YAML frontmatter and Markdown sections. One general rule applied is that compact metadata is written into YAML frontmatter while longer descriptive fields are stored in named Markdown sections.

Risk, Mitigation and Relationship files are stored in separate directories for organizational reasons.

Even though there are in total five different converters, a master converter was introduced so that only the raw text files are needed and the complete conversion process runs automatically by activating the master converter.

7.3. Python Data Models

After conversion into the OKF format, the data is stored in a transferable form but is not yet directly used by the application. The knowledge entries are then represented internally through Python dataclasses. The three central classes correspond directly to the conceptual entity model of Risk, Mitigation and Relationship to maintain a clear correspondence between the knowledge model, stored Markdown files and actual runtime representation.

The Risk dataclass contains fields corresponding to the Risk model introduced in Chapter 6, including identifiers and provenance, descriptions and classification fields, causal information, stakeholder and context information, evidence information, standardized entries, quotations and of course Logic Patterns. Multi-valued fields are simply represented as Python lists. The Mitigation dataclass analogously represents the fields of the Mitigation model corresponding to the mitigation template. Relationship is implemented independently from Risk and Mitigation. Source Type and Target Type distinguish Risk and Mitigation endpoints through their corresponding IDs without requiring different

runtime classes for each relationship combination. Therefore, the thesis uses just one relationship class instead of three for every relationship category (direction).

Following the principle that all fields have to be based on literature findings, string fields use empty strings as default values while multi-valued fields use empty lists generated through default factories. This allows entries with undocumented fields to remain loadable without automatically inventing missing information. Missing information therefore remains syntactically representable within the software artifact.

Risk and Mitigation objects additionally contain a cluster attribute because cluster values are used by the Knowledge Graph visualization. However, because clustering is performed after knowledge extraction, this attribute should be clearly distinguished from the template fields.

7.4. Parsing and Loading the Knowledge Base

Establishing the knowledge base as well as a suitable Python representation, the parser that combines these two can be developed. The parser layer afterwards transforms the structured Markdown representation into the Python data models that are used at runtime. Separate parsers exist again for Risk, Mitigation and Relationship files as well as a master parser. These make sure that every mitigation, risk and relationship gets encoded as one instance per entity.

When first looking at the knowledge base each Markdown file is read as UTF-8 text. The parser then separates YAML frontmatter from the remaining Markdown content and is using PyYAML to parse the YAML part. Files without valid YAML frontmatter cause a dedicated parsing error rather than being wrongly interpreted as valid entries. When being done with the YAML part, the corresponding markdown body is divided according to first-level headings. Each heading becomes a key in an internal section mapping and the corresponding text is used as the value. The entity-specific parser can therefore retrieve fields using predictable section names such as Description, Root Cause, Impact or Reasoning Notes. For handling lists and quotations sections are simply converted into Python lists and quotation markers used in Key Quote for Prompting are removed before storing the quote in the Python object.

Risk and Mitigation parsing works very similarly. Risk and Mitigation parsers combine information from YAML metadata and Markdown sections, with YAML primarily being used for compact metadata and classifications and Markdown sections providing longer descriptions and context-dependent information. Missing fields generally result in empty strings or lists.

Relationship parsing requires one additional step, as the parser determines Source Type and Target Type from the endpoint metadata contained in the Markdown file. A source_risk_id results in source_type = risk and source_mitigation_id results in source_type = mitigation. The same logic is applied to target entities. The parser therefore converts the serialized endpoint representation into the generalized runtime model.

As mentioned at the beginning of this section, the master parser loads everything into

Python instances: Risks, Mitigations and Relationships. The load_all component is that master and scans the Risk, Mitigation and Relationship directories for Markdown files, passing each file to the corresponding parser. The resulting objects are accumulated into three collections: risks, mitigations, relationships. The complete knowledge base is returned as these three collections, finishing the loading process.

7.5. Construction of the LLM Context

With the knowledge base loaded into Python objects, the next part is about how this data is actually used as context for the LLM. The context construction is kept separate from the stored knowledge files so that different representations can be supplied while keeping the actual question and model interaction the same. For that, the prototype implements the four already mentioned context configurations: Full Structured Context, Structured Context without Relationships, No Context and Unstructured Full Context. The same context options are available in both the command-line implementation and the Streamlit interface.

Starting with the Full Structured Context, all available Risk, Mitigation and Relationship files are loaded and parsed into their corresponding Python objects. All three collections are then supplied to the prompt, meaning that the model receives both the information about individual Risks and Mitigations as well as the Relationships between them. The Structured Context without Relationships works almost the same, with the main difference that the Relationship collection is replaced by an empty list before the prompt is constructed, therefore completely leaving out relationship information. Risks and Mitigations are therefore still provided in their structured form, but without the additional Relationship information.

The No Context configuration goes one step further, as Risks, Mitigations and Relationships are all passed as empty collections. The model still receives the general instructions and the user question, but none of the knowledge extracted in this thesis. In contrast, the Unstructured Full Context uses the original textual representation. For this, the Risks.txt and Mitigations.txt files are read directly and the different Relationship text files are combined into one Relationship context. This means that this configuration does not use the Markdown parsing and Python object representation that is used for the two structured configurations.

One important detail here is that the structured Markdown files themselves are not directly inserted. As described in Section 7.4, they are first parsed into Risk, Mitigation and Relationship objects. These Python objects are then used in the prompt. The OKF-based Markdown files therefore mainly represent the persistent structured form of the knowledge base, while the Python objects are the runtime representation. Because of that, the structured context does not directly test Markdown as a format, but the structure of the knowledge model after it has been loaded into Python.

Inside the prompt, Risks, Mitigations and Relationships are separated into the three regions `<RISKS>`, `<MITIGATIONS>` and `<RELATIONSHIPS>`. The user question is then added after these knowledge regions. Additionally, the prompt contains instructions

that the model should only answer based on the supplied knowledge base and should reference relevant Risk, Mitigation and Relationship IDs. If information needed for answering the question is not available in the supplied data, the model is instructed to answer with "This is not in the knowledge base". This is used to keep the answers inside the information that was actually provided to the model.

The current implementation does not use an additional retrieval step before constructing the context. There are no embeddings, vector databases, semantic search or ranking of entries. Instead, every knowledge object belonging to the selected context configuration is added to the same request. As a result, the amount of available knowledge also directly influences the prompt size. This becomes especially relevant for Relationships, since adding further Relationship entries also adds additional information to the context.

Because the Relationship files are loaded directly from the corresponding directory, the used Relationship dataset also has to be treated as part of the current knowledge-base version. Replacing the Relationship dataset therefore also changes the context given to the model.

A similar version difference exists regarding the later introduced cluster information. The comparative evaluation was performed with the version of the knowledge base before cluster values were added to Risks and Mitigations. The cluster field was only introduced afterwards during Graph analysis. Therefore, the Full Structured Context used for the recorded evaluation does not contain this additional cluster information, even though it can be part of the current version.

7.6. LLM API Integration

When context selection and construction are done, the framework uses API access to generate the answer. The prototype communicates with an external LLM through the KI:connect chat-completions interface. Communication is implemented directly in Python using HTTP requests, as this separates the knowledge processing from a single locally hosted language model.

For communication, authentication is needed using an API key through the KICONNECT_API_KEY environment variable. The key is loaded from the execution environment rather than being hardcoded into the source code, and in the web application corresponding secrets can be mapped to the same environment variable. This allows the existing LLM module to be reused in both local and web-based execution.

When it comes to model selection, available models are obtained dynamically from the API's model endpoint. However, not all of them are usable with the chat-completions endpoint and embedding models are removed from the returned list. The remaining models can be selected by the user.

After successfully selecting every parameter, the request can finally be constructed. Each request contains a system message and a user message, where the system message defines the domain as legacy system modernization and instructs the model to use only the supplied knowledge base while the user message contains the selected Risk, Mitigation and Relationship context together with the actual question. The selected model

identifier is passed as part of the request body.

The API response is then checked for unsuccessful HTTP status codes and the generated assistant message is extracted from the first returned choice. The resulting text is returned to the calling interface directly, without an additional LLM post-processing stage.

If something goes wrong during the procedure, the Streamlit interface catches these errors and presents an error message rather than terminating silently. For example, a missing API key produces an explicit runtime error and network or API errors are propagated through the request layer.

7.7. Web Interface

The purpose of the web interface is relatively simple: usability, originally for the expert evaluation. Therefore, a Streamlit interface was developed to make the prototype accessible without requiring direct interaction with the Python command-line interface.

The sidebar contains the main configuration controls, namely context selection out of the four contexts as well as model selection of one of the chat-compatible models. The active context condition and model are displayed before a request is submitted. Users can then enter a modernization question of their choice in a text area. For the expert evaluation, six predefined tasks can be quick-selected through a drop-down window that automatically pastes the task into the question box. This reduces manual copy-paste during the expert evaluation runs.

After submission, the selected knowledge context is loaded and passed together with the question and selected model to the LLM. The corresponding answer is then stored in the Streamlit session history, which is reset once the application is reloaded. Previous answers are still visible during the active session in combination with their context condition and used model. For every answer, the interface searches the response for referenced Risk, Mitigation and Relationship IDs to compare those IDs to the knowledge base. Existing IDs can then be extracted, displayed and finally expanded to display the corresponding documented knowledge entry, allowing evaluators or users in general to compare a model statement directly with the underlying knowledge entry. If the answer contains an ID that does not exist, the interface explicitly flags the reference and displays the mismatch to the user. This does not fully prove that the complete answer is hallucinated, but it provides a direct indication that a supposedly documented Risk, Mitigation or Relationship identifier was generated without a corresponding entry in the literature-based knowledge base.

It is important to note that the complete ID lookup used for displaying documented entries is generated independently from the context condition supplied to the model. Consequently, information that is displayed to the evaluator does not provide additional hidden knowledge to the LLM, but rather allows quick fact-checking of the given answer.

Looking briefly at performance, the structured knowledge loading process is cached so that the underlying files do not need to be re-parsed and re-loaded for every request. The Unstructured Full Context is cached separately, as is the list of available models for

a certain amount of time. This reduces repeated file operations and unnecessary API requests during interactive use.

7.8. Knowledge Graph Visualization

In addition to the user-facing components, the generalized Relationship model allows the knowledge base to be represented as a directed graph. The graph serves primarily as a visualization and analysis artifact and is, in this prototype, not used as the retrieval mechanism for the LLM component.

The first step to a graph is the node and edge construction. For that, Risks and Mitigations are represented as graph nodes and Relationships are represented as directed edges from the stored Source ID to the stored Target ID. The node identifier corresponds to the Risk or Mitigation ID and for further visualization, Risk nodes and Mitigation nodes use different shapes to preserve the distinction between entity types. Hover information additionally contains the corresponding entity name and assigned cluster. For edges, the Relationship Type is displayed as the edge label and Reasoning Notes are provided as additional hover information. The resulting graph therefore exposes both the existence and the documented interpretation of a modeled connection.

Certain checks have to be made to construct a valid graph. For example, before adding an edge, the graph generator verifies that both endpoint IDs correspond to existing Risk or Mitigation nodes. Therefore, relationships containing invalid or unresolved endpoints are not added as graph edges. The set of connected nodes is recorded to later identify entities without modeled relationships.

Looking at clustering, a separate clustering script constructs an undirected network from the available Risk, Mitigation and Relationship objects. With this network, Louvain community detection is used to identify groups of strongly connected entities. A fixed random seed is used to increase reproducibility, and the resulting communities are ordered by size and assigned identifiers such as C01 and C02. Lastly, the calculated cluster value is written back into the corresponding Risk and Mitigation OKF-based Markdown files.

Importantly, the graph visualization itself does not recalculate communities, but it uses the calculated clusters to visualize them. For that, it reads the previously stored cluster attribute from the Risk and Mitigation objects and assigns different colors to the clusters. Full graph representations of both relationship datasets, high and medium confidence, can be found in the appendix.

The directed network in general is generated using PyVis, with the graph being exported as an interactive HTML file. Here, users can inspect nodes, relationships, entity names, Relationship Types and Reasoning Notes interactively. Basic statistics such as the number of clusters, largest cluster sizes and the number of nodes without relationships are additionally produced during graph generation; however, they are only visible in the Python console, not in the actual HTML file.

The same graph-generation logic can be applied to different Relationship datasets, which in this thesis is done for the high- and medium-confidence ones. However, the

extended non-standard dataset first needs to be parsed and loaded again. This bigger dataset changes connectivity without requiring a different graph implementation.

7.9. End-to-End Query Processing

After describing the individual implementation components, this section now combines them into one complete end-to-end use case of the proposed framework. A user query always passes through the same central processing pipeline, independent of the selected context configuration. The main difference between the different executions is therefore which knowledge representation is selected and supplied before the actual prompt is sent to the LLM. The complete process is illustrated in Figure 7.2.

Step 1 starts directly with the user, who selects one of the available context configurations and submits a modernization question. In the web interface, this question can either be manually entered or selected from one of the predefined evaluation tasks. The model itself can additionally be selected through the application interface before starting the request. Empty questions are rejected before continuing with the query process.

In Step 2, the Context Selector receives the selected configuration and loads the corresponding knowledge context through the Knowledge Loader as described in Section 7.5.

Step 3 then provides the selected knowledge to the Prompt Builder.

In Step 4, the Prompt Builder combines this knowledge representation with the general grounding instructions and the submitted user question. The final prompt request is then sent to the KI:connect API. The selected model identifier is included as part of the request.

Step 5 forwards this request from the KI:connect API to the selected LLM. This means that the communication with the actual model is handled through the external API instead of directly through the prototype itself.

In Step 6, the LLM generates an answer based on the supplied prompt and returns the generated result through the API. The model is instructed to stay inside the supplied knowledge context and where possible, reference the corresponding Risk, Mitigation and Relationship IDs.

Step 7 then passes the returned answer to the response component of the application. At this point, the generated answer itself is already complete and no additional LLM-based post-processing is performed.

Step 8 applies the ID extraction functionality described in Section 7.7. Risk, Mitigation and Relationship IDs occurring inside the generated response are extracted and compared against the complete ID lookup of the structured knowledge base. Existing IDs can therefore later be connected to their documented entries, while IDs that do not exist can be flagged as potentially unsupported references.

Finally, Step 9 displays the generated answer together with the extracted ID information to the user. The ID information only supports inspection of the generated response and does not change the answer itself or provide additional knowledge to the LLM.

The command-line implementation follows the same general query-processing logic, although without the additional visual ID inspection provided through the Streamlit interface. The web application therefore mainly adds another user-facing layer around the same context loading, prompt construction and LLM communication components.

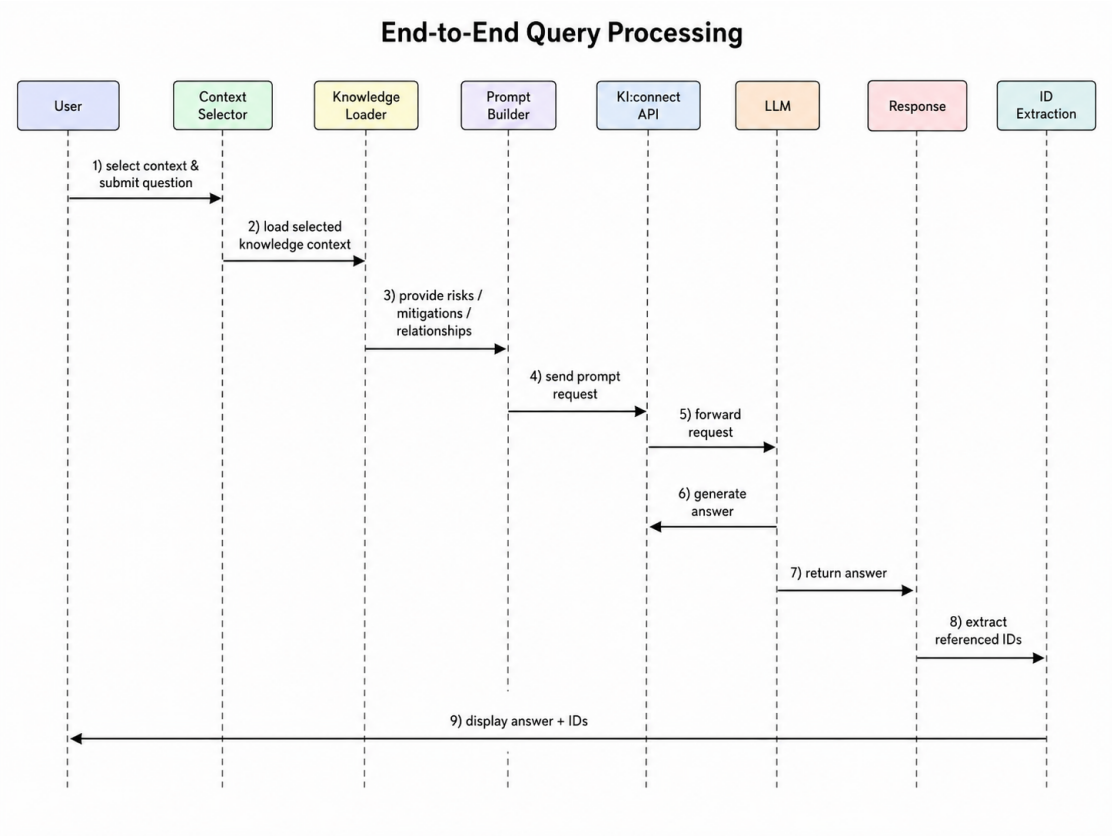


Figure 7.2.: Illustration of the complete end-to-end query processing pipeline.

7.10. Reproducibility and Availability

Lastly, the implementation should also be reproducible and accessible outside of the development environment used during this thesis. For that, the software artifact together with the corresponding knowledge representation is maintained in a public GitHub repository. Git is therefore not only used for version control of the actual source code, but also for tracking the text-based Risk, Mitigation and Relationship knowledge files used by the prototype.

The repository itself is separated into different directories based on their responsibilities. The converter directory contains the scripts used for transforming the original extracted data into the OKF-based Markdown representation, while the data directory contains both the raw and structured knowledge representations. Python dataclasses are

located inside the models directory and the different parsers for loading the knowledge base are contained in the parser directory. LLM communication is handled separately inside the llm component, while graph generation and evaluation-related functionality are again separated into their corresponding directories. The main.py file provides the basic command-line execution and streamlit_app.py starts the web-based interface described in Section 7.7.

The implementation additionally depends on a number of Python libraries. Core dependencies include Streamlit for the web interface, Requests for communicating with the external API, python-dotenv for loading local environment variables and PyYAML for processing the YAML frontmatter of the knowledge files. The Knowledge Graph additionally uses PyVis for creating the interactive HTML representation, while NetworkX is used during graph construction and cluster calculation. The dependencies needed for executing the final artifact are documented in the repository requirements file.

Configuration information that should not be included inside the source code is handled separately. Most importantly, the API key required for KI:connect is supplied through the KICONNECT_API_KEY environment variable and is therefore not hardcoded into the implementation or committed into the repository. Local environment files are excluded from version control. For the deployed Streamlit application, the same information can instead be provided through Streamlit secrets. This makes it possible to use the same LLM communication component locally as well as inside the deployed application.

The prototype can be executed in two main ways. The command-line implementation can be started through main.py and provides the basic context selection and question-answering workflow. Alternatively, the Streamlit application provides the more user-friendly interface used especially during evaluation. Conversion of the original extracted data and calculation of the clusters are separate preprocessing operations. Once the corresponding Markdown knowledge files and cluster values exist, these steps do not have to be repeated for every new user query.

For reproducibility, however, not only the implementation itself but also the used version of the knowledge base is important. This becomes especially relevant for the Relationship data because this thesis contains different Relationship datasets. The original dataset mainly contains the more conservative High-Confidence Relationships, while the extended version additionally contains Medium-Confidence Relationships to increase connectivity of the Knowledge Graph. Using another Relationship dataset therefore changes both the graph structure and when used as LLM context, the actual information supplied to the model. The exact dataset used for an evaluation therefore has to be treated as part of the experimental configuration.

A similar limitation exists regarding the used language models. The selected model identifier is stored together with the evaluation configuration, but the models themselves are provided through an external service and are not part of the repository. Since the list of available models is dynamically retrieved through KI:connect, available models can change over time. Additionally, changes to a model on the provider side can result in different generated answers even when using the same source code, prompt and knowl-

edge base. Complete reproducibility of the generated LLM output can therefore not be guaranteed only through versioning the prototype.

Finally, the implementation also has some technical boundaries that are relevant when reproducing or extending the artifact. The complete selected knowledge context is currently supplied to the LLM without an additional retrieval mechanism, meaning that the usable knowledge-base size is limited by the available context window of the selected model. The implemented Markdown parsers are designed for the thesis-specific OKF structure and are therefore not intended as general Markdown parsers. Additionally, the Python dataclasses provide a structured runtime representation but do not represent a complete semantic validation system. The generated Knowledge Graph is also an interactive HTML visualization rather than an actual graph database. These limitations are discussed further together with possible extensions in Chapter 9.

8. Validation and Evaluation

This next chapter looks at the observed results of the validation and evaluation activities defined in Chapter 4. It examines whether the implemented knowledge representation and prototype operate as originally intended and whether different representations of modernization knowledge affect the LLM-based analysis in terms of performance. The fully documented method can be found in the corresponding chapter (Chapter 4).

In total, this thesis uses three different evaluation layers.

The dataset that was used in all of these stages is the conservative Relationship dataset used by the prototype with high-confidence relationships. It contains 357 entries in total, compared to the larger extended dataset with almost 700. This extended dataset was not included in the same LLM comparison because the resulting serialized knowledge context exceeds the practically available context limit. Consequently, conclusions from the comparative LLM evaluation refer back to the 357-Relationship dataset. This chapter focuses on reporting scores, observed differences, technical findings, graph characteristics and expert assessments. Broader interpretation regarding RQ1, RQ2, scientific contribution, practical usefulness and threats to validity is addressed in Chapter 9.

8.1. Validation and Evaluation Overview

First, technical and knowledge base validation examines if the integrity of the implementation, structured entries, references and graph representation function as intended in basic tests. It establishes whether the artifact provides a sufficiently consistent basis for subsequent LLM-based testing. In more detail, this includes the conversion pipeline, parser, Python data models, ID references, knowledge-base loading, Relationship representation, and Knowledge Graph generation.

Afterwards, a developer-conducted proof-of-concept comparative evaluation examines whether the prototype can correctly retrieve and combine the represented knowledge. It is important to note that the functional validation and comparative context evaluation were conducted by the developer of the artifact. Their purpose is to demonstrate technical feasibility, reveal weaknesses in the knowledge representation, and provide an initial controlled comparison before looking at a practical view as these results are explicitly treated as proof-of-concept evidence rather than as an independent assessment of professional effectiveness. This second step also compares different context scenarios.

Lastly, the external expert evaluation provides the independent evaluation layer as it involves real industry practitioners who assess the generated answers and the practical usefulness of the prototype from this industrial view. This part also demonstrates the practical relevance of the represented knowledge.

A difference to other literature is the relationship specific evaluation perspective as explicit Relationships represent one of the central design contributions of the thesis. Their contribution is examined when looking at the comparison between Structured Context without Relationships and Full Structured Context specifically. A separate graph-based comparison additionally examines the structural consequences of extending the 357-Relationship dataset to the 697-Relationship dataset.

8.2. Technical and Knowledge Base Validation Results

Starting with the general validation of the prototype, the loaded knowledge base needs to be checked. The validated knowledge base contains 154 Risk entries and 166 Mitigation entries and the Relationship dataset used for the developer-conducted LLM proof of concept contains an additional 357 Relationship entries. All of these are represented in OKF markdown files in the framework and the parser successfully loads these three entity collections for context generation. The resulting entity model therefore contains 320 Risk and Mitigation entities before Relationships are represented as graph edges. Furthermore, all 154 Risk IDs are unique as well as all 166 Mitigation IDs and all 357 Relationship IDs. Therefore no duplicate identifiers were found in the three entity collections used for the proof-of-concept evaluation.

Next up is the relationship endpoint integrity. All 357 Relationship entries reference Risk and/or Mitigation entities that exist in the corresponding entity collections, which leads to all of them being represented as edges in the corresponding knowledge graph. Based on relationship semantics, the dataset contains 294 Risk–Mitigation Relationships, 51 Risk–Risk Relationships and 12 Mitigation–Risk Relationships, demonstrating that Risk–Mitigation Relationships are the most prominent in the observed literature. Risk–Risk Relationships primarily represent cascading effects, while Mitigation–Risk Relationships represent risks introduced by mitigation decisions.

Looking into the Relationship Type distribution, the most frequent Relationship Types in the dataset are Prevents (159), Reduces (96), Triggers (50), Detects (23) and Introduces (12). Additional entries use Recovers or combined labels, for example Detects/Reduces. Only one Risk–Risk relationship uses the Type Amplifies. Other types were mentioned rarely.

Another relevant aspect in this thesis is completeness and traceability. All Risk, Mitigation and Relationship entries contain a standardized LLM-oriented entry, important for the use of LLM analysis. Furthermore, all 166 Mitigation entries contain a Paper ID and of the 154 Risks, only one does not have a Paper ID, with that one being contributed by an expert of the field.

However, there were some relationship inconsistencies identified during functional validation in individual Relationship entries. For example, Rel250 contains an internal mismatch between the referenced target and the described introduced risk or effect and Rel255 contains slightly contradictory source and target information that can lead to an incorrect interpretation. Those were fixed before continuing with the evaluation part, demonstrating the importance of a validation.

The technical validation shows that the implemented dataset is structurally usable for LLM-based access and graph generation, with unique IDs and valid Relationship endpoints.

8.3. Knowledge Graph Validation Results

Following up on the general validation of the knowledge base, the resulting Knowledge Graph itself was validated as well. The graph generated from the conservative dataset contains all 154 Risk nodes and 166 Mitigation nodes, resulting in 320 nodes in total. All 357 Relationships are represented as directed edges between those entities. Since there can be more than one Relationship between the same two entities, collapsing those parallel Relationships for the connectivity analysis results in 315 unique connected node pairs.

Looking further into the actual connectivity, the directed graph was projected to an undirected graph for this part of the analysis. The reason for this is that for the component analysis it is mainly relevant whether two entities belong to the same connected structure and not in which direction the Relationship is pointing. This results in 89 connected components in total. Furthermore, 57 of the 320 nodes are completely isolated in the validated dataset and therefore have no Relationship edge connected to them. Consequently, 263 entities participate in at least one modeled Relationship.

There is also a clear difference in the size of those connected components. The largest connected component contains 103 nodes while the second-largest already contains only 46 nodes. All remaining components are substantially smaller. This demonstrates that the conservative graph does not form one single connected modernization knowledge structure, but instead consists of some larger connected areas together with many smaller components and isolated entities.

However, those graph values only describe the structural connectivity of the represented knowledge. A highly connected entity or component does not automatically mean that the represented Relationships are more correct, useful or supported by stronger evidence. The results therefore mainly validate that the graph can be generated from the represented Relationships and that its structure can be analyzed. A broader interpretation of the connectivity and especially the trade-off between Relationship coverage and confidence is discussed later in Chapter 9.

8.4. Preliminary Validation

After securing a valid knowledge base, the functional validation was conducted by the developer of the knowledge base and prototype and it therefore serves as an internal proof-of-concept validation of basic functionality. The actual method for evaluation can be found in Chapter 4 and all questions of the validation can be found in the appendix.

The 20 questions used in the validation achieved a score of 94 out of 100 possible points, based on the used 5-point scale. This corresponds to an average rating of 4.7 out of 5. The result indicates that the prototype was generally capable of retrieving and

combining the information represented in the structured knowledge base at least on a basic level.

Taking a more detailed look at the validation, the strongest results were observed for direct retrieval questions and questions based on clearly defined Relationship IDs. Additionally, individual Risk and Mitigation information could generally be retrieved accurately from the structured knowledge base. Furthermore, the model also handled several knowledge boundary questions appropriately and did not automatically invent Risk or a numerical probability when the requested information was absent in the knowledge base. It performed therefore well on the basic key functionalities.

Extending on relationship retrieval, Explicit Risk–Mitigation and Risk–Risk Relationships could generally be retrieved when the corresponding Relationship entries were available. Again, Relationship IDs, relationship types and related entities could be included fairly well in the generated answers. However, relationship questions were more sensitive to inconsistencies in the underlying Relationship entries than direct entity questions for risks and mitigations.

The main factors for not scoring perfectly in every category were Risk Chains and in general, complex questions, where weaknesses were observed for multi-step relationship queries, risk chains and broader scenario questions. These tasks were hard as they require information from several Risk, Mitigation and Relationship entries to be combined into one answer. Therefore, errors or ambiguities in individual Relationship entries that were misinterpreted can propagate through such chains and therefore have a larger effect on complex answers than on simple, direct retrieval questions. A recurring weakness in broader scenario questions was that the generated answer sometimes combined several relevant entries without clearly indicating whether the final recommendation was explicitly represented or inferred from multiple knowledge base entries. The problem is that this distinction is relevant because one objective of the knowledge model is to preserve traceability between documented relationships and LLM-generated synthesis.

Nevertheless, this was by far not the case in every answer and most of them were correct and easily traceable. Therefore the 94/100 result provides evidence that the implemented workflow is suitable for the subsequent comparative proof-of-concept evaluation. Reminder, that it should not be interpreted as independent confirmation of professional usefulness because both execution and assessment were performed by the artifact developer. External assessment is provided separately through the expert evaluation later in this chapter.

8.5. Comparative Proof-of-Concept Evaluation

Following the functional validation, the next step was the comparative proof-of-concept evaluation. This evaluation was again conducted by the developer and therefore has to be understood as a controlled proof-of-concept experiment and not as an independent evaluation of the prototype. As defined in Chapter 4, 35 predefined questions were evaluated under four different context configurations while keeping the model, system prompt, questions and model parameters the same. GPT-5.4 mini was used for all recorded runs.

Consequently, the main changing factor between the individual runs was the knowledge context supplied to the model. The Full Structured Context in this evaluation uses the conservative dataset containing 357 Relationships. It is also important to mention that this evaluation was conducted before the later Cluster metadata was added to Risk and Mitigation entries. Therefore, none of the four context configurations evaluated here contain Cluster information and the later cluster-enhanced knowledge base has not been evaluated against the original Full Structured Context yet.

The four compared configurations are No Context, Unstructured Full Context, Structured Context without Relationships and Full Structured Context as described in Section 7.5.

For comparing the generated answers, the Overall Score combines several of the metrics introduced in Chapter 4. The Overall Score therefore does not represent a universal measure of LLM answer quality but instead combines the dimensions that were selected for this proof-of-concept evaluation. The Maximum score caps at 100.

Looking first at this Overall Score, clear differences can be seen between all four configurations. No Context produced the lowest mean score with 46.87 and a median score of 48.00. Adding the structured Risk and Mitigation entries without Relationships already increased the mean score to 75.12 with a median of 80.00. Full Structured Context then reached a mean Overall Score of 87.58 and a median of 90.50. The highest result was achieved by Unstructured Full Context with a mean score of 92.72 and a median of 95.00. Therefore, Full Structured Context improves the mean Overall Score by 40.71 points compared to No Context. More importantly for the Relationship representation, adding the explicit Relationships to the otherwise structured context improves the mean score by 12.46 points compared to Structured Context without Relationships.

Table 8.1.: Overall Score results across the four context configurations.

Context Configuration	**Mean**	**Median**
No Context	46.87	48.00
Structured Context without Relationships	75.12	80.00
Full Structured Context	87.58	90.50
Unstructured Full Context	92.72	95.00

Looking more specifically at the retrieval of Risk, Mitigation and Relationship IDs provides another perspective on these differences. No Context achieved a mean Precision of 0.608, Recall of 0.444 and F1 Score of 0.497. Additionally, it produced the highest mean ID Hallucination Rate with 0.249. This is the weakest identifier retrieval result among the four configurations and is expectedly the condition where the model has to rely on its own knowledge and the information provided directly in the question.

Providing Structured Context without Relationships already improves these values substantially. Mean Precision increases to 0.801, Recall to 0.614 and F1 to 0.667. The ID Hallucination Rate also decreases to 0.170. This shows that the structured Risk and Mitigation entries themselves already provide useful grounding for the model even if no explicit Relationship entries are supplied.

The Full Structured Context achieves a mean Precision of 0.778, Recall of 0.885 and F1 Score of 0.813, while the mean ID Hallucination Rate is 0.165. Compared to the structured configuration without Relationships, Precision is therefore slightly lower, but Recall and F1 increase considerably. The highest identifier metrics overall are again reached by Unstructured Full Context with a Precision of 0.877, Recall of 0.952 and F1 of 0.893. This configuration also has the lowest mean ID Hallucination Rate with 0.123.

Table 8.2.: Mean identifier retrieval metrics across the four context configurations.

Context Configuration	**Precision**	**Recall**	**F1**	**ID Halluc.**
No Context	0.608	0.444	0.497	0.249
Structured Context without Relationships	0.801	0.614	0.667	0.170
Full Structured Context	0.778	0.885	0.813	0.165
Unstructured Full Context	0.877	0.952	0.893	0.123

Especially interesting is again the direct comparison between the two structured configurations. Adding explicit Relationships increases mean Recall from 0.614 to 0.885. This is an absolute increase of approximately 0.271 or around 44.2% relative to the configuration without Relationships. Mean F1 increases from 0.667 to 0.813, corresponding to an absolute increase of approximately 0.146 and roughly 21.8%. Precision on the other hand changes from 0.801 to 0.778 and is therefore slightly lower with Relationships. The ID Hallucination Rate only changes slightly from 0.170 to 0.165. Based on these values, the largest observed effect of adding the Relationship representation is therefore visible in Recall and the retrieval of expected connected entities rather than in Precision.

The differences also depend on the type of question being asked. Direct Risk and Mitigation questions show smaller differences between Structured Context without Relationships and Full Structured Context. This makes sense in the context of the experiment because both configurations contain the same structured Risk and Mitigation entries. Information such as descriptions, causes, impacts or individual mitigation information can therefore be retrieved in both cases without requiring an explicit Relationship entry.

A larger difference becomes visible for Relationship-oriented questions, secondary risks and Risk Chain questions. Here, Full Structured Context has direct access to Relationship IDs, Relationship Types and explicit source-target connections. The configuration without Relationships instead has to reconstruct possible connections from the descriptions contained in the individual Risk and Mitigation entries. This can still lead to a plausible connection in several cases, but expected Relationship IDs or additional connected entities are more frequently left out, which is also reflected in the lower Recall.

This becomes particularly relevant for risk chains and secondary risks because these questions require multiple Relationships to be combined. Risk–Risk Relationships can represent cascading effects and Mitigation–Risk Relationships can represent additional risks introduced through a mitigation decision. With Full Structured Context these connections are explicitly represented and can therefore be followed across multiple entries. At the same time, those questions are also more sensitive to individual Relationship inconsistencies. If one Relationship in the middle of such a chain is interpreted incorrectly,

the error can influence several following parts of the generated answer rather than only one direct retrieval.

Scenario-based questions showed more heterogeneous results than the direct retrieval questions. These questions combine several contextual conditions and require the model to identify multiple relevant Risks, Mitigations and Relationships in one answer. Both Full Structured Context and Unstructured Full Context produced strong answers for several of these scenarios. There were also individual examples where Structured Context without Relationships was able to infer a fitting connection between entities even though no explicit Relationship was included in its context. Therefore, not having an explicit Relationship does not automatically prevent the model from producing a plausible answer. The difference in this case is that an inferred association does not offer the same explicit Relationship ID and source-target representation as a stored Relationship entry.

Another relevant aspect is hallucination and the general boundedness of the generated answers. No Context shows the highest ID Hallucination Rate with 0.249. All configurations that provide the modernization knowledge base reduce this value, with Unstructured Full Context reaching the lowest mean rate of 0.123 and Full Structured Context reaching 0.165. However, the ID Hallucination Rate only considers unsupported identifiers and should therefore not be confused with broader factual hallucination. A response can use only valid IDs while still making an unsupported statement about them. This is why Groundedness and the other factual metrics remain separate parts of the evaluation.

The structured-output questions add another view on the usefulness of the representation. Full Structured Context provides direct access to Risk IDs, Mitigation IDs and Relationship IDs as well as Relationship Types, applicability information, evidence and confidence information. This allows the requested information to be transferred into machine-readable outputs without first reconstructing all of these elements from plain text. Still, the Full Structured Context did not perform consistently better for every single structured-output question. Individual large retrieval failures were observed, particularly in some hallucination-verification and machine-readable Risk Chain questions. These outliers also contribute to the lower mean Overall Score of Full Structured Context compared to Unstructured Full Context even though its general performance was strong.

A particularly large difference between the representations can be seen when looking at the runtime of the complete evaluation runs. No Context required 1 minute and 3 seconds for all questions. Structured Context without Relationships required 2 minutes and 26 seconds, while Full Structured Context required 2 minutes and 45 seconds. In comparison, Unstructured Full Context required 23 minutes and 16 seconds. The unstructured configuration was therefore approximately 8.46 times slower than Full Structured Context in the recorded experiment. Its mean Overall Score is around 5.14 points higher, but this advantage comes together with a substantially longer runtime.

Taken together, the proof-of-concept results do not show one context representation being better in every evaluated dimension. Unstructured Full Context achieves the

Table 8.3.: Runtime of the complete comparative evaluation runs.

Context Configuration	Runtime
No Context	1 min 03 s
Structured Context without Relationships	2 min 26 s
Full Structured Context	2 min 45 s
Unstructured Full Context	23 min 16 s

strongest average quality and identifier retrieval metrics in this experiment, but at the same time has by far the longest runtime. Full Structured Context reaches a lower mean Overall Score but provides the explicit Risk, Mitigation and Relationship structure while remaining substantially faster. Most importantly for the Relationship representation, adding the explicit Relationships to the structured Risk and Mitigation context considerably increases Recall and F1, while Precision and ID Hallucination Rate remain relatively similar.

These results have to remain within the scope of the experiment. They were produced in a developer-conducted proof of concept using one fixed model and one controlled experimental configuration. They therefore demonstrate measurable differences between the context representations, but do not independently demonstrate their professional usefulness in real modernization projects. The broader implications of the observed quality, structure and runtime differences are discussed in Chapter 9, while the external practitioner perspective is evaluated separately in the expert evaluation later in this chapter.

8.6. Comparison of Relationship Dataset and Graph Variants

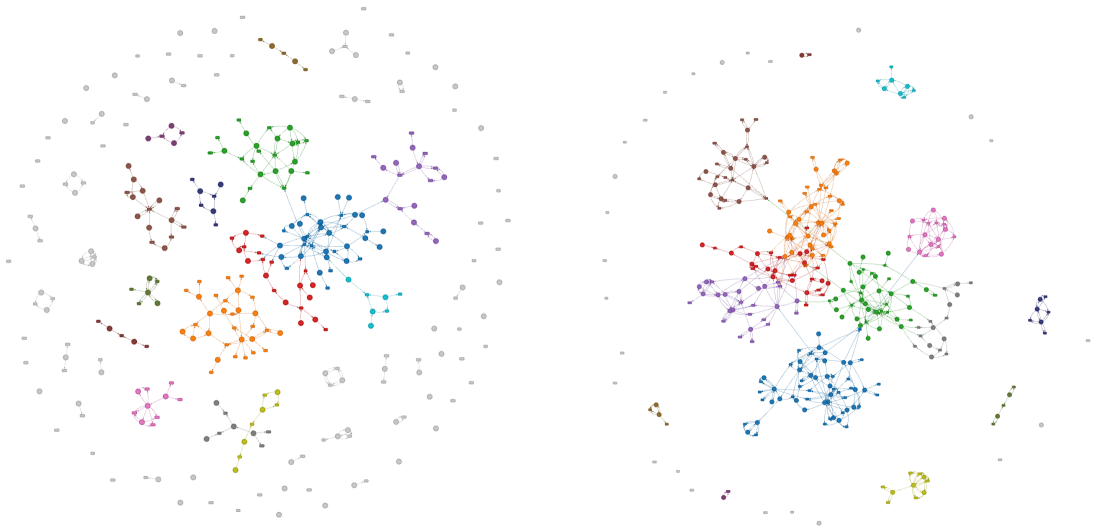
A short digression is needed here to look at the different possible datasets. This separate comparison examines the structural consequences of extending the Relationship dataset beyond the 357-entry dataset used in the proof-of-concept parts. The comparison therefore concerns graph connectivity, density, component structure and the distribution of relationship types rather than LLM answer quality directly. It focuses on general graph objectives when changing relationship edges. Both graph variants contain the same Risk and Mitigation entities.

Compared with the conservative graph validated in Section 8.3, the extended graph contains 697 Relationship edges and 574 unique connected node pairs when parallel connections are collapsed. With more relationships, the undirected connectivity projection contains 33 connected components and 25 isolated nodes. The largest connected component contains 257 nodes, corresponding to approximately 80.3% of all Risk and Mitigation entities, while the second-largest component contains only 10 nodes. The average unique node degree increases to approximately 3.59 and the graph density increases to approximately 0.0112.

These changes show that previously separate areas become integrated into a considerably larger connected structure and that the extended dataset is less fragmented than

the conservative graph.

The structural difference between both datasets is illustrated in Figure 8.1. For readability, the main-text figure uses reduced clustered views, while the corresponding full-scale clustered and unclustered visualizations are provided in the appendix (Chapter E).



(a) High-Confidence dataset: 357 Relationships

(b) Extended High- and Medium-Confidence dataset: 697 Relationships

Figure 8.1.: Overview of the structural change between the conservative and extended Relationship datasets. The extended dataset integrates substantially more Risk and Mitigation entities into the dominant connected component. Full-scale clustered and unclustered visualizations are provided in Appendix E.

Now focusing on how this extension takes place, the largest increase is observed for Prevents Relationships, which increase by 178 entries for risk–mitigation relationships. There is also an increase in risk–risk relationships, because Triggers Relationships increase by 132 entries and therefore account for a second major share of the additional connectivity. Some smaller contributors are Detects which increases by 17 entries, Introduces by 8, Recovers by 3, Reduces by 1 and also the combined label Detects / Reduces by 1. These changes provide a structural explanation for the stronger integration of risk chains and mitigation paths in the extended graph by adding more variety specifically between risks with risks.

The centrality of individual entities changes substantially when the additional Relationships are introduced, for example in the conservative graph, M6 (Proof of Concept) has the highest unique degree with 10 neighbors. In the extended graph, R152 (Architecture overcomplexity from too many integrations and components) and R4 (Cost Overrun) each reach 14 unique neighbors. Furthermore, R151 (Amalgamation issues among replaced applications) reaches 13 unique neighbors, while R124, R149, and R58

each reach 11. Several entities that were previously isolated or weakly connected become substantially more central and the added risk–risk relationships boost risks more than mitigations. The extended dataset therefore changes not only the total number of edges but also the structural role of individual Risk and Mitigation entities.

Looking at the complete graph again, the community analysis indicates that the extended graph does not simply create larger versions of the conservative clusters. Several previously distinct areas become integrated into broader modernization complexes that combine planning, data migration, transition safety, architecture, cloud readiness, security, organizational factors and portfolio-level concerns, covering a lot of dimensions in one cluster. However, these are more general risks and mitigations. It is important that several specialized domains remain comparatively stable, including AI governance, microservice granularity, LLM semantic preservation and legacy–modern coexistence. The observed community changes therefore indicate increased cross-domain connectivity while exactly preserving the specialized aspects.

However, adding more information does not always need to be positive. Serializing the extended Relationship dataset substantially increases the amount of context required for direct LLM access. Therefore in the current implementation, the complete extended dataset exceeds the available context limit when combined with all Risk and Mitigation entries and was not included as an additional Full Structured Context condition.

8.7. External Expert Evaluation Results

The expert evaluation now represents the external evaluation component of the thesis, arguably the most important part. Unlike the developer-conducted proof-of-concept evaluation, the generated answers are assessed by practitioners with relevant modernization or IT expertise. The objective here is to obtain an independent assessment of answer quality and of the practical usefulness of the prototype.

The final expert questionnaire, tasks, four context conditions, assessment criteria, and 0–100% rating scale are defined in Section 4.5.2. Important to stress here is that earlier questionnaire drafts or setup variants are not part of the reported evaluation and that this section reports only responses obtained using the final version distributed to the experts.

The experts access the implemented Web interface and interact with the evaluated context configurations, helped by the predefined tasks for concrete modernization-analysis situations that support consistent assessment of the generated answers. The tasks support the corresponding rating questions and general evaluation rather than constituting a separate experimental dataset and free testing could be performed by the experts with the framework.

The sample consists of six experts and the number of submitted criterion ratings differs between participants, so the results are interpreted descriptively rather than as population-level statistical estimates. As there were different amounts of answers, Expert-level averaging prevents participants with more submitted ratings from receiving greater weight in the overall context comparison. Nevertheless, all actually submitted

ratings are retained in the analysis and can be found in detail in the appendix.

8.7.1. Expert Ratings Across Context Configurations

Analyzing the experts' answers, this thesis first states how many samples were collected and what preparations were needed. In total, six completed expert evaluations were included in the analysis with participants covering different professions with different perspectives including IT consulting, IT coordination and security, solution architecture, transformation program management, application migration, enterprise architecture and IT risk. Their reported professional experience ranges from more than five years to approximately twenty years. The areas of expertise include legacy system modernization, cloud and application migration, system integration, governance, delivery risk, data migration, cut-over planning, enterprise architecture as well as general technology risk. Important to note is that the experts were not required to complete all six tasks and therefore selected between two and five tasks depending on their expertise and available time. The tasks served as orientation for interacting with the prototype and are not used to filter the submitted ratings. Each expert used the same selected task subset across all four context configurations, allowing a within-expert comparison without changing the evaluated task mix between contexts. However, not every rating question was answered by every expert. Every non-empty rating entered by an expert is included in the analysis regardless of whether the corresponding task was marked as completed, leading to a total of 107 answers collected. These split into 27 for Full Structured Context, 27 for Structured Context without Relationships, 27 for No Context, and 26 for Unstructured Full Context.

Looking into the aggregation procedure and overall context ratings, different numbers of answers per expert and context could give higher weight to experts with more answers. Therefore, to give every expert equal weight despite different numbers of submitted criterion ratings, a mean context rating is first calculated for each expert across all valid ratings submitted for that context. Afterwards, the six expert context means are aggregated for each context configuration as seen in the following Table 8.4: Highlighting

Table 8.4.: Aggregated expert ratings across the four context configurations.

Context Configuration	**Mean**	**Median**	**SD**	**Range**
Full Structured Context	91.83%	91.00%	5.19	85.00–100.00%
Structured Context without Relationships	80.17%	80.50%	10.23	64.00–95.00%
Unstructured Full Context	74.66%	72.88%	14.77	55.00–95.00%
No Context	19.00%	21.00%	13.62	0.00–39.00%

the most important findings of Table 8.4, Full Structured Context achieves the highest mean expert rating at 91.83%, with a median of 91.00% and a standard deviation of approximately 5.19 percentage points. No Context receives by far the lowest mean expert rating at 19.00%, a standard deviation of approximately 13.62 percentage points

and a range from 0.00% to 39.00%, showing some variation in the No Context ratings.

Compared with the proof-of-concept evaluation, Full Structured Context is rated higher than Unstructured Full Context by all six experts. Full Structured Context exceeds Unstructured Full Context by approximately 17.18 percentage points. More importantly for this thesis, Full Structured Context is rated higher than Structured Context without Relationships by five of the six experts, while one expert assigns the same mean rating to both structured variants, resulting in an average difference of approximately 11.67 percentage points. Criterion-level results show that the difference is comparatively small for Accuracy but substantially larger for Direct Relationship Reasoning and especially Indirect Relationship Reasoning. Full Structured Context also exceeds No Context by approximately 72.83 percentage points, with No Context producing the lowest mean context rating for every expert.

8.7.2. Expert Results by Task Type

Covering the general context-level, this thesis established a dominance by the Full Structured Context condition. Now the focus will be more on the different questions and corresponding tasks. All used values can be seen in the underlying figure and will be used in the upcoming analysis in more detail.

Table 8.5.: Expert ratings by evaluation criterion across the four context configurations.

Evaluation Criterion	Full Structured	Struct. w/o Rel.	Unstructured	No Context	Δ Full–No Rel.
Accuracy	89.17% ($n = 6$)	85.00% ($n = 6$)	76.67% ($n = 6$)	2.50% ($n = 6$)	+4.17 pp
Direct Relationship Reasoning	88.33% ($n = 6$)	73.33% ($n = 6$)	69.17% ($n = 6$)	0.00% ($n = 6$)	+15.00 pp
Indirect Relationship Reasoning	92.00% ($n = 5$)	61.00% ($n = 5$)	71.25% ($n = 4$)	0.00% ($n = 5$)	+31.00 pp
Scenario Mapping	83.75% ($n = 4$)	76.25% ($n = 4$)	62.50% ($n = 4$)	11.25% ($n = 4$)	+7.50 pp
Hallucination Resistance	99.00% ($n = 5$)	92.00% ($n = 5$)	76.00% ($n = 5$)	83.00% ($n = 5$)	+7.00 pp
JSON Validity	100.00% ($n = 1$)	100.00% ($n = 1$)	96.00% ($n = 1$)	95.00% ($n = 1$)	0.00 pp

Accuracy is rated by all six experts under all four context configurations. Full Structured Context here achieves the highest mean Accuracy rating at 89.17% closely followed by the Structured Context without Relationships at 85.00%, while Unstructured Full Context achieves 76.67%. No Context as a failure test reaches only 2.50% due to constant answer rejection as intended in the system prompt. Full Structured Context therefore exceeds Structured Context without Relationships by approximately 4.17 percentage points, concluding that Direct Risk retrieval works strongly with both structured knowledge representations, while the explicit Relationship layer provides only a comparatively small additional benefit for this criterion.

This changes when looking at Direct Risk–Mitigation Relationship reasoning. Again, all experts answered this question. Full Structured Context achieves a mean rating of 88.33% while the Structured Context without Relationships drops to 73.33% and Unstructured Full Context also goes down to 69.17%. No Context again receives 0.00%. The 15.00 percentage-point difference between the two structured configurations represents a substantially larger difference than for Accuracy. However, qualitative feedback

shows that plausible Relationships can sometimes be inferred from Risk and Mitigation descriptions even when explicit Relationship entries are absent.

Indirect Relationships and risk chains are the first criterion where ratings are missing. Five ratings are available for Full Structured Context, Structured Context without Relationships, and No Context, while four ratings are available for Unstructured Full Context. Full Structured Context again achieves 92.00% but Structured Context without Relationships achieves 61.00% and gets surpassed by Unstructured Full Context with 71.25%. No Context receives 0.00%. The difference of 31.00 percentage points between the two structured configurations is the strongest effect associated with the explicit Relationship representation, which is plausible as this task extends the previous Relationship task to longer chains. As the unstructured representation contains relationships, it performs better than the no relationship dataset. Furthermore, qualitative comments describe the configuration without explicit Relationships as relying more strongly on inferred links and as providing a weaker audit trail or less reliable risk chains.

Switching to scenario applications, this task has four submitted ratings for each context configuration. Full Structured Context achieves the best mean rating at 83.75%, Structured Context without Relationships achieves 76.25%, Unstructured Full Context falls behind on 62.50%, while No Context reaches 11.25%. Full Structured Context again exceeds Structured Context without Relationships by 7.50 percentage points, however the difference is smaller than for direct and indirect Relationship reasoning, indicating that broader scenario analysis can also be supported by individual Risk and Mitigation descriptions without every connection being explicitly represented.

As traceability and hallucination resistance are central objectives of the prototype, this task is of significant importance with five submitted ratings for each context configuration. Full Structured Context achieves a mean rating of 99.00% followed by Structured Context without Relationships at 92.00% showcasing the strength of structured inputs. No Context receives a comparatively high mean rating of 83.00%, while Unstructured Full Context falls back at 76.00%. Qualitative comments explain the comparatively high No Context value through safe refusal behavior, including responses indicating that requested information is not contained in the knowledge base. Consequently, No Context can avoid unsupported knowledge base claims while simultaneously receiving extremely low Accuracy and Relationship Reasoning ratings.

The last part is the machine readable output to test a different output format. Sadly, JSON Validity has only one submitted expert rating for each context and therefore provides insufficient evidence for a broader comparison. The single expert rating is 100% for Full Structured Context, 100% for Structured Context without Relationships, 96% for Unstructured Full Context, and 95% for No Context. These values show that valid machine-readable output was achievable in the tested observation but should not be generalized beyond this individual assessment.

8.7.3. General Expert Assessment of the Prototype

At the end of the questionnaire, there was a section for general questions which all six experts completed. These general ratings are independent of the task selections and are

therefore unaffected by the correction to the context-rating analysis. The Overall Quality of the Program received in these questions a mean rating of approximately 82.17%, with a median of 82.50% and individual ratings ranging from 75% to 88%. Usability and Ease of Interaction received a slightly higher mean rating of approximately 83.17%, with a median of 84.00% and ratings ranging from 75% to 90% while Usefulness for analyzing Risks in Legacy System Modernization achieved the highest of the three general mean ratings at 83.50%, with a median of 83.00% and a range from 75% to 90%. Therefore no general program rating was below 75%, indicating consistently positive assessments of the prototype across the six experts.

Looking at the professional use cases stated in the evaluation questionnaire, five of the six experts described concrete professional situations in which they could imagine using the prototype. Mentioned use cases include pre-evaluation and process drafting, architecture reviews, migration planning, dependency workshops, risk workshops, project reviews, steering preparation, migration and cutover preparation, data-readiness sessions, risk assurance, architecture boards and early transformation reviews. Most of these focus on early stages of modernization and preparation or review activities. However, one expert considered the current use case too limited or specific for direct professional adoption.

Relationship-oriented knowledge is one of the most consistently mentioned strengths in the qualitative feedback. For this, experts highlight Risk–Mitigation links, explicit Relationships, mitigation mapping, traceable reasoning, a shared Risk view, and the ability to connect information across the knowledge base. Additional positively mentioned aspects include structured knowledge, confidence information, concrete Risk entries as well as quick lookup and the possibility to use the system as a discussion starter. The qualitative feedback therefore aligns strongly with the relationship and traceability objectives used in the design of the prototype.

Main Areas for improvement were seen in original presentation and traceability of evidence. Experts request a clearer source or evidence view, improved readability of the fact checking help and better visibility of source freshness. A second recurring theme concerns greater actionability of the generated results, where requested extensions include explicit priorities, responsible owners, recommended next actions, business impact and more detailed business-level tasks. These comments indicate that the represented knowledge itself is generally viewed positively while the translation from knowledge retrieval to operational decision support remains an area for further development.

The role the prototype fills was described by the experts as useful for preparation, reviews, workshops, or checklist creation rather than as a replacement for professional assessment. One expert states that a technical review would still be required, another describes the prototype as not yet being a project-control tool, and another emphasizes the need for governance controls. The feedback therefore positions the current implementation primarily as a decision support. This practitioner view is again nicely consistent with the intended scope of the prototype, which supports risk analysis without autonomously making modernization decisions.

Taken together, the quantitative general ratings and qualitative comments indicate a

positive external assessment of the prototype's basic usability and practical relevance. Usefulness receives the highest general average rating at 83.50%, closely followed by usability at 83.17% and overall quality at 82.17%. Important is the main finding that the strongest recurring qualitative theme is the value of explicit Relationships. However, at the same time, the experts identify evidence presentation, prioritization, actionability, domain-specific detail, and governance as relevant areas for further development.

8.8. Summary of Evaluation Results

Across the different evaluation steps, one general result is that modernization-specific knowledge improves knowledge-dependent LLM tasks compared with providing no additional knowledge. The evaluations also show that explicit Relationships are especially useful for questions that require connected Risk and Mitigation information or longer Risk Chains. However, there is a difference between the developer-conducted and expert evaluation. In the automated proof-of-concept evaluation, Unstructured Full Context achieved the highest mean Overall Score, while the experts rated Full Structured Context highest overall. Since both evaluations use different evaluation approaches, those results cannot be directly compared. The trade-offs between answer quality, structure, traceability, runtime, Relationship coverage and context size are discussed further in Chapter 9.

The contribution of explicit Relationships is particularly visible when the two structured context configurations are compared. Figure 8.2 summarizes selected quantitative differences observed across the developer-conducted proof of concept and the external expert evaluation.

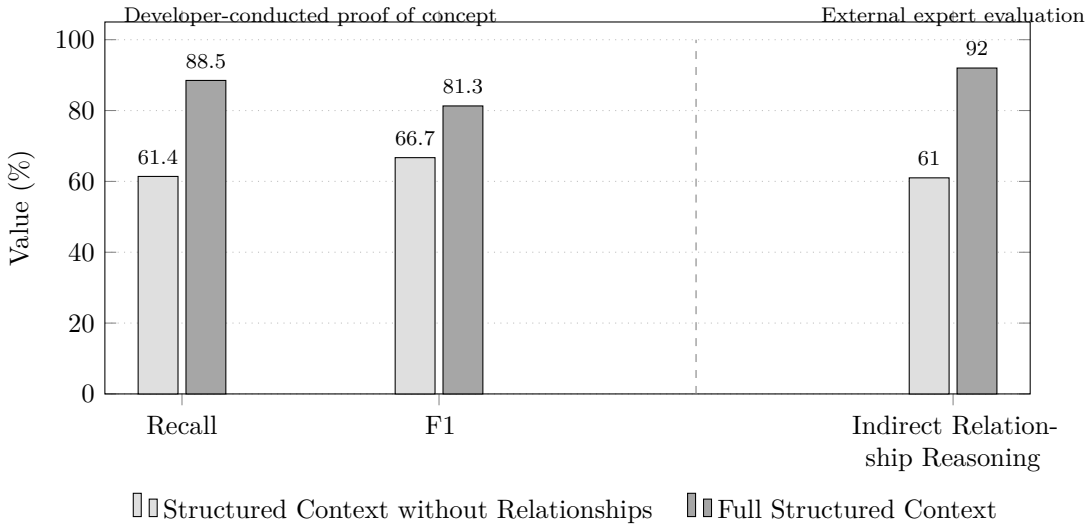


Figure 8.2.: Selected quantitative differences between Structured Context without Relationships and Full Structured Context. Recall and F1 originate from the developer-conducted proof-of-concept evaluation, whereas Indirect Relationship Reasoning represents the corresponding mean expert rating.

9. Discussion

After careful evaluation of the proposed framework, this chapter interprets the findings of the literature analysis, knowledge-model development, prototype implementation and evaluation in relation to the two research questions. The objective is not to repeat the results reported in Chapters 5–8, but to explain their implications for legacy-system modernization risk analysis and LLM-oriented knowledge representation.

First of all, this chapter starts by interpreting the findings with regard to RQ1 and RQ2. Afterwards, the scientific contribution and practical implications of the thesis are discussed, followed by the limitations of the literature base, knowledge extraction, knowledge representation, technical implementation, and threats to validity.

9.1. Interpretation of the Results for RQ1

RQ1 focuses on knowledge gathering and for that, the first finding has to be that Modernization risk is a multidimensional problem. The literature analysis supports the view that legacy-system modernization risk cannot be reduced to an exclusively technical problem, because Risks were identified across technical, organizational, business and process dimensions. Risks as well as general challenges in modernization were interrelated [Red24]. Another key finding in the reviewed literature is the interdependence between different modernization challenges and corresponding risks [Paz25]. This supports the holistic perspective adopted by the thesis as the four dimensions should therefore primarily be understood as an analytical structure for organizing the knowledge rather than as independent risk domains.

Another important part is the cross-dimensional propagation of risks. Risks originating in one dimension can and will in most cases generate consequences in another dimension. For example, insufficient legacy knowledge is primarily an organizational concern but can increase technical uncertainty, project duration and ultimately modernization cost. Similarly, technical migration failures may produce operational disruption and therefore business consequences. The results therefore suggest that isolated risk assessment can overlook relevant consequences if assuming independence.

When it comes to connecting risks and mitigations with each other, Mitigation strategies are many-to-many and context-dependent. The extracted knowledge does not support a simple one-risk–one-mitigation interpretation, due to the fact that individual risks can be addressed by several mitigations, while individual mitigations may address several risks. Furthermore, the suitability of a certain mitigation additionally depends on prerequisites, modernization strategy, organizational conditions and other contextual conditions. Mitigation strategies should therefore not be interpreted as universally

applicable best practices because their value depends on the conditions under which a modernization project is conducted.

This thesis also considers the importance of mitigation trade-offs as mitigation strategies do not necessarily reduce total modernization risk without consequences. Mitigations such as parallel operation, incremental migration or rollback mechanisms that seem positively annotated can reduce transition risk but simultaneously increase cost, duration, operational complexity or coexistence dependencies in a modernization [Sea+01]. This supports explicitly representing Trade-offs / Side Effects and Mitigation–Risk relationships where a risk can be introduced by a mitigation. Risk treatment should therefore consider both the intended reduction of a primary risk as well as possible secondary risks introduced by the mitigation itself.

One further relationship between risk and mitigations however does not even include mitigations and uses a risk–risk dependency. The identified Risk–Risk relationships show that modernization risks can form contributing risk chains. Explicit Risk–Risk relationships therefore provide information that would be lost if risks were represented only as independent entries. This is important for holistic modernization analysis because the consequences of a risk may emerge indirectly through several intermediate development steps.

All of these findings increase the complexity of the represented knowledge and therefore make a unified knowledge base important. The extracted knowledge base should be interpreted as literature-grounded and context-dependent modernization knowledge rather than as a set of universal causal rules [Kha+15; Gho+17; PE23]. The evidence varies between statistical evidence, case studies, expert opinion, practitioner experience and other forms of evidence, and the existence of a modeled relationship therefore does not imply that the same effect will necessarily occur in every modernization project. Evidence Type, Confidence Level, Contextual Constraints, and Applicability Conditions become important for preserving this distinction.

Reverting back to the specific RQ1, this question cannot be answered adequately through a flat list of risks and mitigation strategies alone. The results indicate that holistic modernization risk knowledge consists of risk entities, mitigation strategies, contextual conditions, trade-offs and many interdependencies between them. The most important conceptual finding is therefore not only which risks and mitigations exist, but that their applicability and consequences depend on a network of contextual and cross-dimensional relationships.

9.2. Interpretation of the Results for RQ2

RQ2, on the other hand focuses on how the previously gathered modernization knowledge should be structured and formatted for LLM-based use. Looking at the evaluation first, it becomes clear that the provided knowledge itself already has a large effect on the generated answers. This shows first of all that giving the model an external modernization knowledge base resulted in considerably better scores than expecting it to answer based only on its internal knowledge.

More important for the developed knowledge model, however, is the comparison between Structured Context without Relationships and Full Structured Context. Those two configurations contain the same structured Risk and Mitigation entries, meaning that the main additional information is the explicit Relationships. This means that the Relationships mainly helped the LLM to find more of the expected connected knowledge, even though Precision did not increase. A similar result can be seen in the expert evaluation. Five of the six experts rated the complete structured context higher and the remaining expert rated them equally.

The contribution of explicit Relationships across both evaluation layers is summarized in Figure 8.2.The difference becomes even more visible when not only looking at general Accuracy. For Accuracy, the expert ratings only differed by 4.17 percentage points between the two structured configurations. Direct Relationship Reasoning, however, showed a difference of 15.00 percentage points and Indirect Relationship Reasoning a difference of 31.00 percentage points. This is interesting because these are exactly the tasks for which the Relationship model was introduced. If only a direct Risk or Mitigation has to be retrieved, both configurations already contain this information. Once the answer depends on how several entries are connected, the explicit Relationships become much more relevant. The LLM can sometimes also infer a reasonable connection from the descriptions without having a stored Relationship. The problem is that such a connection is only generated by the model and does not come with the corresponding Relationship ID, type, direction, Confidence Level or Applicability Conditions. Therefore, an answer can look reasonable in both cases while only one of them is directly traceable to a represented Relationship.

However, the comparison with Unstructured Full Context makes the answer to RQ2 less straightforward. In the developer evaluation, the unstructured context reached the highest Overall Score of 92.72 and also the strongest mean identifier retrieval values. Therefore, it would be wrong to conclude from this thesis that more structure automatically results in a better LLM answer. At the same time, the expert evaluation resulted in almost the opposite ranking. All six experts rated Full Structured Context higher than Unstructured Context and the expert-level means were 91.83% and 74.66%. Both evaluations looked at the generated answers differently, so those percentages cannot be compared directly. Still, the different result is important because it shows that the utility of the representation depends on which aspects are considered. The unstructured knowledge keeps a large amount of textual information together, while the structured version separates the information into individual fields and knowledge objects. This can remove some textual redundancy but in return creates explicit IDs, Relationship semantics, Confidence Levels and source information.

Answer quality is also not the only aspect of utility when answering RQ2, because there is a substantial runtime difference between those representations. Of course, runtime depends on the used model, API and context size and should not be generalized into a general performance advantage. Still, in this prototype the structured representation uses the complete knowledge base much more efficiently in terms of runtime. Therefore, utility of a representation cannot be reduced only to the highest generated answer score

but also has to include how much context has to be processed to produce these answers.

These findings can also be compared to previous research concerning structured representations for LLM reasoning. Previous work reports that structured context and entity-relationship representations can improve reasoning performance and make complex Relationships more explicit [Zha+25; WT24]. Graph-oriented reasoning has similarly shown improvements for logical reasoning and multi-hop question answering [Han+25]. The findings of this thesis are generally consistent with these results, especially because the strongest advantage of explicit Relationships occurs for Direct and Indirect Relationship Reasoning. However, the PoC evaluation provides a less straightforward result because the unstructured representation still achieved the highest automated Overall Score. The contribution of structure should therefore not be understood as structure being universally better compared to natural language. Instead, the results support it especially for relational access, traceability and more efficient processing.

One important point here is also what exactly was evaluated as structured context. The knowledge base itself is stored in an OKF-based Markdown structure, but the Markdown files are not simply sent to the model. They are parsed into Python objects first and the actual LLM context is then constructed from those objects. Because of that, this thesis did not perform a direct comparison between Markdown, Python, JSON, YAML or any other possible serialization format. Previous research reported advantages for Python-based entity-relationship representations, but this experiment cannot confirm that Python itself is the reason for the observed results. The more relevant part for this thesis is the schema behind it, meaning the separation into Risks, Mitigations and Relationships and the additional contextual, evidence and provenance fields.

Another part that has to be considered is the amount of Relationships included in the knowledge base. The conservative dataset contains only the more strongly supported Relationships while the extended dataset additionally includes Medium-Confidence ones. Adding these Relationships makes the resulting Knowledge Graph substantially more connected and also shows more possible risk chains and Risk–Mitigation paths. At first this seems beneficial because more parts of the knowledge base can be connected to each other. On the other hand those added Relationships are not supported with the same confidence as the conservative core. Therefore, simply adding every possible Relationship would go against the idea of keeping evidence and Confidence Levels explicit.

More Relationships also directly increase the size of the context. This became a technical problem with the extended dataset because the combination of all Risks, Mitigations and extended Relationships approaches or exceeds the usable context size of the current implementation. For this reason, the extended Relationship dataset was analyzed as a graph but was not evaluated as another Full Structured Context configuration. The higher connectivity therefore only shows that more knowledge areas become connected. It does not show that an LLM would actually produce better answers with this larger dataset. This distinction is important because otherwise graph connectivity could easily be mistaken for LLM performance.

This context problem also points towards a possible later extension of the framework. Supplying the complete knowledge base works for the current proof of concept but will

become increasingly difficult when more Risks, Mitigations, Relationships or source information are added. A future version therefore does not necessarily need to decide between only structured knowledge or only complete raw text. The structured Risks, Mitigations and Relationships could be used to first select the relevant part of the knowledge base, while more detailed textual information is only supplied where it is needed for the actual question. A retrieval- or Graph-RAG-based implementation could use the existing Relationships for exactly this selection and would avoid passing every available entry to the LLM for every query.

Returning to RQ2, the findings support the three-object core summarized in Figure 6.1: separate Risk, Mitigation and Relationship objects with stable IDs and explicit Relationship directions. Contextual Constraints, Applicability Conditions, evidence, Confidence Levels and provenance should additionally remain part of the representation because not every finding is equally certain or applicable in every modernization. The evaluation particularly supports the usefulness of explicit Relationships for direct and indirect relational reasoning and they also make generated answers easier to trace back to the represented knowledge. At the same time, the strong automated result of Unstructured Full Context shows that textual detail still has value and should not simply be removed for the sake of creating more structure. Therefore, the answer to RQ2 is not to structure as much information as possible. The modernization knowledge should rather have an explicit and machine-readable core of Risks, Mitigations and Relationships while still retaining enough textual and contextual information to preserve what the original literature actually states.

9.3. Scientific Contribution

The thesis connects research on legacy system modernization risks with research on structured knowledge representation and LLM-based analysis. Existing modernization literature provides substantial knowledge about challenges, risks and mitigation strategies, while structured representation research investigates entities, relationships, Knowledge Graphs and LLM reasoning on the other hand. The thesis integrates these areas through an executable modernization risk knowledge base that is grounded on literature extraction.

Furthermore, the proposed framework introduces a relationship-oriented knowledge model. A central contribution is the independent modeling of them as first class knowledge objects. The model represents Risk–Mitigation effects, Risk–Risk cascades and Mitigation–Risk secondary effects. Furthermore, the relationship Evidence Type, Confidence Level, Applicability Conditions and Reasoning Notes provide information that cannot be adequately represented through simple cross-references between knowledge model entries. This makes uncertainty and interpretation within the relational knowledge explicit.

What this thesis also does is demonstrate how qualitatively extracted modernization knowledge can be transformed into a machine-readable and LLM representation. For further reasoning enhancement, Standardized Entries, Logic Patterns, IDs, contextual

fields and explicit Relationships are introduced specifically to support consistent machine processing. The resulting OKF-based Markdown representation remains human readable while allowing automated parsing into Python objects.

The conceptual knowledge model is implemented as a working research prototype rather than remaining solely a theoretical schema. This enables the proposed representation to be examined through actual LLM queries and graph analysis.

Additionally, the controlled proof-of-concept evaluation separates the effects of No Context, unstructured knowledge, structured entities without Relationships and the complete structured representation, therefore comparing different context conditions and their effect on the generated answer. This also makes it possible to investigate the specific contribution of explicit Relationship information rather than comparing only knowledge versus no knowledge. Furthermore, the completed expert evaluation complements the developer evaluation with an external practitioner assessment from six experts.

Answering what the scope of this contribution is, the thesis does not propose a universal causal theory of modernization risk and does not establish an operational risk-management standard. Moreover, its contribution is a structured synthesis and representation approach that makes heterogeneous modernization risk knowledge more explicit, traceable, relational, and accessible for LLM-based analysis.

9.4. Practical Implications

Out of these scientific contributions, there are several theoretical aspects that have practical implications, for example the support for early modernization analysis. The knowledeg base can support early stage exploration of potential modernization risks before detailed project risk assessment is performed. Practitioners can use the knowledge base to systematically identify relevant risk areas, possible mitigation measures, affected stakeholders or more like contextual constraints. This is consistent with the expert feedback, which identified possible use in architecture reviews.

Another use case is the slight support for traceable risk exploration, where stable Risk, Mitigation and Relationship IDs allow generated answers to directly reference knowledge entities. This can make LLM analysis more transparent as Paper IDs and retained source information additionally allow important findings to be traced back to the base literature level. This traceability and explicit Risk–Mitigation Relationships were also repeatedly identified as strengths by the experts.

Speaking of relationships, they lead to a general Value of relationship-oriented analysis. Explicit Relationships allow practitioners to examine not only individual risks but also potential risk chains and mitigation trade-offs. This can be useful when evaluating whether mitigating one risk could create another problem, leading to more possibilities in analysis. The graph representation can additionally provide an exploratory overview of connected modernization topics as well as possible chains.

However there remains a context dependence and human judgment need when working with the prototype as it is not a fully automated decision system. The decision if

a documented risk or mitigation applies to a specific organization requires project specific information and professional judgment by human experts. Confidence Levels and evidence information should therefore support interpretation, so the prototype is better understood as decision support and knowledge exploration than as an autonomous risk management mechanism. This boundary is also reflected in the expert feedback, which emphasizes continued technical review and governance controls are needed for practical use.

The implemented Web interface further demonstrates that the knowledge base can be accessed without manually navigating the underlying Markdown files and that the general expert ratings for Overall Quality, Usability, and Usefulness were consistently positive at approximately 82–84%. However, the current implementation remains a research prototype. Expert feedback identifies clearer source and evidence presentation, prioritization, responsible owners, next actions, business impact, more domain-specific examples as well as clearer applicability information as relevant improvements. Therefore, production use would additionally require even stronger semantic validation, access control, scalable retrieval for the bigger dataset, continuous knowledge maintenance and evaluation of use within real modernization projects.

9.5. Limitations of the Literature Base

Starting with limitations of the actual literature used in this thesis, it has to be mentioned that the literature search cannot guarantee that every existing modernization risk and mitigation strategy was found. Google Scholar was used as the main scientific search platform and the search itself was done iteratively, meaning that more specific searches were added whenever new risks, mitigations or terminology came up during extraction. Therefore, the resulting knowledge base should not be seen as an exhaustive representation of everything that exists about legacy modernization. It is rather a structured collection and consolidation of the literature that was included and analyzed in this thesis. The list of papers used for extraction can be found in the appendix.

Another problem when going through the literature was the terminology itself. Modernization papers do not always use Risk, Challenge, Problem, Barrier or similar terms in the same way. Sometimes two publications describe more or less the same problem with completely different names, while in other cases the same term is used for findings that are actually different. This is one reason why consolidation and standardized entries became necessary in the first place. Still, consolidating several findings into one Risk or Mitigation entry requires interpretation and cannot completely remove the ambiguity that already exists in the original literature.

Something that was also noticeable throughout the literature research is that not every modernization dimension receives the same amount of attention. Much of the existing work goes into technical modernization. Organizational, business and process aspects are also covered but not always with the same amount of detail. This was already visible in related work (Chapter 3), where especially technical modernization approaches make up a large part of existing research. Because the knowledge base is built from this

literature, it can naturally also contain more detailed information for some technical areas compared to less-researched organizational or process-related areas.

There is also a limitation in how strongly parts of the existing research field are empirically supported. The reviewed literature contains many proposed modernization approaches, while fewer of them are evaluated through actual modernization projects. For this thesis, this means that not every extracted Risk, Mitigation or Relationship can be based on strong empirical evidence. A finding coming from a proposed framework should therefore not automatically be interpreted in the same way as a finding that was observed repeatedly in real modernization projects.

The use of grey literature adds another limitation to this. It is especially useful for current modernization cases and practical problems, but the quality between those sources can differ considerably. Some sources have detailed evidence behind them while others are more guidance-oriented or come from companies that also have a commercial interest in a certain modernization approach. Authority, currency, neutrality and traceability were considered during source selection, but including grey literature still makes the overall literature base more heterogeneous.

Besides source quality, reporting itself can also create a limitation. Modernization literature can naturally focus more on successful approaches. Failed modernization attempts or mitigation strategies that simply did not work can be less likely to be documented in detail. Grey literature helps to find some of these practical failure cases but does not remove the problem completely. Therefore, if a specific Risk or failed Mitigation is not contained in the resulting knowledge base, this does not mean that it cannot occur in practice.

Lastly, the resulting literature base is also limited in time. Modernization changes quickly, especially when it comes to cloud platforms, architecture patterns, automation and now increasingly AI modernization tools. A technical Mitigation that is useful today could be less relevant within a few years, while more general problems such as missing knowledge, planning problems or organizational resistance may remain relevant for much longer. The knowledge base therefore represents the state of literature that was included during the extraction period of this thesis. Without further maintenance and additional extraction, it should not be interpreted as a permanently up-to-date modernization knowledge base.

9.6. Limitations of the Knowledge Extraction

After the literature research there is a parallel extraction coming with separate limitations, the biggest one being the manual extraction conducted by the developer. Literature coding, risk extraction, mitigation extraction, consolidation and Relationship modeling were performed manually, which is good for contextual interpretation but introduces researcher judgment into the resulting base. Extraction reliability can not be quantified through an inter-coder reliability measure because there was no second coder that could independently calculate it.

In this manual process, the task of distinguishing Challenges, root causes, trigger

events, Risks, etc. requires interpretation because publications do not consistently use these terms according to the definitions adopted in this thesis. Therefore, even individual passages can reasonably support more than one interpretation. The standardized extraction model does help to improve consistency but cannot eliminate conceptual judgment to the maximum.

As mentioned, semantically similar risks and mitigations were consolidated to reduce duplication – again manually. Consolidation may consequently remove source-specific details even when the overall meaning is retained, flattening the knowledge base to some extent. Hence, paper IDs and source quotations are important for preserving traceability to the original formulations.

One of the main limitations when looking into manual extraction is the relationship interpretation. Some Relationships are explicitly stated in individual sources, but most of them are derived by combining information from several passages or entries. Relationship modeling therefore contains a greater degree of interpretation than direct extraction of individual risks or mitigations, which is why Confidence Levels and Reasoning Notes were introduced to enhance transparent decisions. Nevertheless, qualitative Confidence Levels cannot remove the underlying uncertainty from these connections.

Furthermore, Standardized Entries and Logic Patterns are derived representations that summarize the other fields for a LLM rather than verbatim literature findings, again potentially compressing nuances of literature.

There is also a high degree of missing and unevenly documented information, for example not every source reports prerequisites, probability of success, effectiveness, effort, triggers, stakeholders or contextual constraints. Missing information was intentionally retained as unspecified instead of inferred, leading to gaps in the literature knowledge base and resulting entries consequently differing in completeness depending on the information available in the underlying sources.

9.7. Limitations of the Knowledge Model and Representation

Next up are the Limitations of the general knowledge model and corresponding template representation. The current knowledge representation does not constitute a formally defined ontology. Although entities, fields as well as Relationship Types are standardized, the model does not provide formal logical axioms or enforced semantic constraints comparable to a fully specified ontology.

The OKF format provides the syntactic structure used by the representation. The OKF Markdown serialization provides syntactic structure but does not itself enforce semantic consistency in fields. For example, the file format cannot independently verify whether a Relationship Type is compatible with its Source Type and Target Type. The consistency therefore depends on the correct extraction conventions, normalization, parsing and validation procedures.

Another limitation is that Evidence Type and Confidence Level preserve important uncertainty information but provide only a qualitative representation of uncertainty. A High-Confidence Relationship should not be interpreted as a quantified probability that

the relationship will occur in a concrete modernization project. The model therefore supports reasoning that considers evidence but not quantitative risk estimation in any way.

The schema that was used itself introduces the limitation of simplification, because representing the modernization knowledge through standardized fields necessarily simplifies some source-specific information. Complex Relationships may depend on several simultaneous organizational, technical and business conditions that cannot always be fully expressed through a single Relationship entry. The representation should therefore be interpreted as a structured abstraction of the literature and not a lossless semantic model.

9.8. Limitations of the Technical Implementation

The technical implementation working with the knowledge base brings more limitations with it, starting with the full context architectural limitation. The current prototype directly supplies the selected knowledge representation as LLM context and does not first filter the knowledge base with semantic retrieval or RAG. This approach is suitable for evaluating different complete-context representations but limits scalability as the knowledge base grows, directly leading to the next limitation: Context-window limitation.

Increasing the number of Relationships increases the size of the serialized context and therefore requires more space in the context window. Hence, the extended Relationship dataset approaches or exceeds the usable context window when combined with the complete Risk and Mitigation knowledge base, leading to an API context error. This just demonstrates that direct full context injection does not scale very well and is hard-capped at a given value. This is not optimal as every entry within the selected context configuration is supplied regardless of its relevance to an individual question. Retrieval-based or graph-based context selection could reduce context size and noise in future implementations and help with scaling issues and general speed.

Referring to the knowledge graph implementation, the Knowledge Graph is currently used primarily as a visualization and analysis representation. Graph traversal is not used directly to construct question specific LLM context or even graph reasoning as a secondary check option. However, Cluster metadata was derived from the Knowledge Graph and added to Risk and Mitigation entries as a later refinement. Nevertheless, because this extension was introduced after the recorded evaluation, its effect on LLM answer quality remains unknown.

One limitation that is present caused by the general construction of the knowledge base is the external API and model dependence. The prototype depends on externally provided LLMs and the corresponding API infrastructure, so model availability, rate limits, model updates, context-window sizes and provider changes are outside the control of the artifact. If models or equally other parameters change, reproducing identical outputs may therefore become even more difficult than in normal LLM blackboxes.

The Streamlit interface demonstrates the end-to-end feasibility of the approach but was simply developed as a research prototype rather than a production system, hence

production deployment would require additional work on it as the maturity and test levels are held basic.

9.8.1. Model-Related Limitations

Quickly covering more general model limitations, LLM-generated answers remain probabilistic and may vary between repeated executions even when the supplied context and prompt remain unchanged. This is a known "problem" in LLMs and nearly eliminates general reproducibility of answers regarding evaluation or validation. Grounding the model in an explicit knowledge base reduces unsupported generation but does not nearly guarantee deterministic or completely correct output. This however could be influenced by structured prompting techniques. Existing research additionally identifies limited control over what information LLMs learn during training and the possibility of undesirable outputs on different aspects [De 23]. The supplied knowledge base therefore also just constrains already learned models and inherits its limitations.

Another known flaw that was reduced is hallucination, so the generation of nonfactual or unsupported statements [WT24]. The comparative evaluation shows that providing context reduces unsupported ID generation, but hallucination is not eliminated entirely. Additionally, the expert evaluation shows that Hallucination Resistance has to be interpreted separately from answer quality, as No Context received a comparatively high rating through safe refusal behavior while performing very poorly on Accuracy. The prototype therefore requires boundary instructions regarding knowledge and should not treat every generated statement as automatically supported.

There is also a difference between retrieval and reasoning. Correct retrieval of knowledge base entries does not automatically imply correct reasoning about their relationships. This gets worse when subsequent steps are relevant. For example, Multi-step risk chains are particularly sensitive to incorrect or inconsistent individual edges. Hence, distinguishing documented knowledge from LLM-generated synthesis remains necessary and the model has at this point of development no automatic checking module.

9.9. Threats to Validity

Starting with construct validity, this thesis uses concepts such as Risk, Mitigation, Relationship, Root Cause, Trigger, Impact, Evidence Type and Confidence Level to structure modernization knowledge. The literature however does not always separate those concepts in the same way. A challenge in one paper can for example be interpreted as a Root Cause or Risk in this thesis. Therefore, big parts of the final classification depend on the conceptual model used here. The same principle applies to the Overall Score in the PoC evaluation as it combines selected metrics and should therefore not be seen as a universal measurement for LLM answer quality. The expert criteria also measure different things, as for example a safe refusal can result in good Hallucination Resistance while still being a bad answer overall.

Another threat comes from the fact that the functional validation and comparative

evaluation were conducted by the developer of the artifact. This creates a possibility for assessment and confirmation bias, especially for answers where some qualitative judgment is required for the answer. Fixed questions, Gold Facts, predefined metrics and the same setup for every context configuration were used to reduce this problem. However, this cannot completely remove the influence of the developer. The expert evaluation with six external participants therefore adds another perspective that is independent from this part of the evaluation.

The knowledge base itself can also influence the observed results. During validation, individual Relationship entries with incorrect or unclear directions and types were found and corrected, however there can possibly be one or two that were missed due to the sheer size of the knowledge base. Especially for risk chains this is important because one incorrect Relationship can influence several follow-up steps. Therefore, a wrong answer does not always clearly show whether the problem comes from the LLM, the context representation or the underlying knowledge base itself. Those causes cannot always be separated completely.

Reliability is furthermore limited by the amount of LLM executions. LLM output is probabilistic and repeated execution of the same question can still result in slightly different answers. The conducted evaluation therefore only represents the recorded runs and not every possible answer of the configuration. More repeated executions per question would be needed to measure variance in a scientific way and check for upcoming differences.

A similar problem exists for the manual knowledge extraction. Extraction, consolidation and Relationship modeling were mainly conducted by one researcher and no second independent extraction was performed. Another researcher could therefore make different decisions when classifying findings, merging similar entries or assigning Confidence Levels to Relationships, generally every step that was done manually to gain the knowledge base. Paper IDs, source quotations, Citavi coding, standardized templates and Reasoning Notes all improve traceability, but they do not fully remove this researcher dependency.

Furthermore, external validity is limited by the used model and expert sample. The recorded comparative evaluation was performed using one selected LLM and model version, even if other models may handle structured and unstructured context differently, leading to a conclusion that the exact measured differences cannot simply be transferred to every LLM. The expert evaluation is similarly based on solely six experts. Even though their backgrounds cover different modernization areas, the result should still be understood as an additional practitioner perspective rather than a representative survey of modernization professionals or even full use of the prototype in practice, leading to the next limitation.

Most importantly, the prototype was not evaluated inside an actual running modernization project of a company. The experiments use predefined questions and therefore do not include all real project conditions but merely artificial scenarios. The results consequently show that the proposed framework can work on those theoretical scenes, but they do not prove that it already improves real modernization project outcomes.

Lastly, the Knowledge Graph comparison also has to be interpreted carefully. The extended graph becomes considerably more connected by including more Relationships in addition to the High-Confidence (conservative) ones. This shows the increasing structural effect of adding more Relationships, but higher connectivity does not automatically mean higher correctness of LLM answers. Especially because the extended dataset was not evaluated as another LLM context configuration, graph connectivity and LLM performance should be treated separately.

Overall, the use of source references, standardized templates, explicit Relationships, Confidence Levels, fixed evaluation questions and the external expert evaluation improve transparency of the thesis. However, the results should still be interpreted as evidence for the feasibility and usefulness of the developed approach and not as proof that the representation is universally better or already ready for operational modernization risk management.

10. Conclusion and Future Work

The purpose of this last chapter is to bring everything together in one final section, summarizing the overall research process and the main contributions of this thesis. It provides short and condensed answers to the research questions RQ1 and RQ2 that are based on the literature analysis, the developed knowledge model, the prototype implementation and the final evaluations. The chapter also presents the main conclusion regarding structured modernization risk knowledge and its derived use with LLMs. There is also a final section on possible extensions of the framework and a future outlook on the topic in general.

10.1. Summary

For its research approach, the thesis addressed this problem through a structured literature-based knowledge extraction process. For that purpose, Scientific and Grey Literature were analyzed in parallel to identify relevant Risks, Mitigation strategies and relationships between them. For structuring the extracted knowledge, three templates were developed corresponding to risks, mitigations and relationships. Those templates helped to distinguish between Risk, Root Cause, Trigger Event, Impact, Mitigation, contextual constraints, Evidence Type and other relevant attributes in the extraction. After finishing the extraction, similar findings from different sources were normalized to create consistent knowledge entries.

Having a conceptual knowledge base as a result that covers Technical, Organizational, Business and Process dimensions, this base additionally represents cross-dimensional dependencies instead of treating all risks as isolated entities. The resulting structured knowledge base contains 154 Risk entries and 166 Mitigation entries with a conservative Relationship dataset containing primarily High Confidence Relationships. An extended Relationship dataset was additionally created by incorporating Medium Confidence Relationships to investigate the effect of increased connectivity and the trade-off with precision. The Relationships in both datasets include Risk–Mitigation effects, Risk–Risk cascades, and Mitigation–Risk secondary effects.

Now that the knowledge base was present in a textual form, it needed to be restructured for enhanced reasoning purposes into an OKF-based Markdown representation. YAML metadata and Markdown sections now combine machine-readable structure with human-readable descriptions and Python parsers load this new representation into typed data objects used at runtime. The resulting objects can be combined dynamically into context representations for an LLM. Furthermore, a Streamlit Web interface was implemented to provide an accessible interaction layer where users can select different

available LLMs and different forms of context representation. For further visualization, a Knowledge Graph was additionally generated covering Risks, Mitigations and standard directed Relationships.

Once the working pipeline was established, it was first technically and functionally validated before conducting the comparative evaluation by the developer. The comparative evaluation was conducted, after small fixes were made, using the four context configurations where it was found that providing domain-specific modernization knowledge clearly improved performance compared with the No Context condition. More importantly, Explicit Relationships particularly improved the retrieval of connected knowledge and multi-hop reasoning information. The Unstructured Full Context achieved the highest average overall evaluation score in the current experiment but Full Structured Context showed a substantially lower runtime while additionally providing explicit IDs. The results therefore indicate that structured and unstructured knowledge representations provide different advantages rather than one representation being universally superior.

An expert evaluation was conducted as well, where the participating practitioners in the modernization industry found that the Full Structured Context performed best across nearly every examined metric. Especially for Direct and Indirect Relationship Reasoning, the explicit Relationships showed a clear advantage, while five of the six experts rated Full Structured Context higher than the structured variant without Relationships and one rated them equally.

Summarizing, this thesis developed an end-to-end process from unstructured literature knowledge identification to an executable LLM-supported modernization analysis prototype which can be used for risk analysis in modernization. Therefore, the main contribution is not only the collection of individual modernization Risks and Mitigations but their structured, traceable and relationship-oriented representation. The work demonstrates how heterogeneous modernization knowledge can be transformed into a reusable representation.

10.2. Answers to the Research Questions

This section provides the answers to the research questions presented in Chapter 1 based on the results of the thesis. RQ1 summarizes what was learned about modernization Risks and Mitigations in general on a theoretical level while RQ2 depicts what was learned about representing this knowledge for LLM use in a more practical manner.

10.2.1. Answer to RQ1

Repeating the research question as a reminder, RQ1 looked at the key risks and their mitigation strategies associated with holistic legacy system modernization across technical, organizational, business and process dimensions. The main findings in this thesis are that modernization involves interconnected Risks across Technical, Organizational, Business and Process dimensions, as the literature analysis showed that risk cannot be reduced to technical migration problems alone. Even though there is a dominance of

technical risks, they frequently interact with organizational knowledge, project processes, financial constraints and business continuity.

As mentioned above, a central result is that the four dimensions are analytically useful but strongly interconnected to the point of risks triggering and amplifying other risks. For example, loss of expert knowledge can increase misunderstanding of technical dependencies, which can lead to implementation errors, schedule delays and higher project costs. A holistic modernization risk analysis should therefore consider Risk Chains and dependencies instead of treating each Risk independently.

Mitigation strategies are then used to mitigate the probability of certain risks materializing. The literature identifies Preventive, Detective, Reactive and Recovery Mitigation strategies with common preventive approaches including extensive planning, architecture analysis, dependency analysis, documentation, Knowledge Recovery, stakeholder involvement, training, Proofs of Concept and incremental modernization. Detective measures include testing, monitoring, technical analysis and validation activities while Reactive and Recovery measures include Rollback mechanisms, fallback environments and contingency planning. Because the entities are interconnected in a many-to-many relation between risks and mitigations, many Mitigations address more than one Risk, while important Risks often require combinations of multiple mitigation measures.

Even though there are several combinations of these entities, no individual Mitigation strategy can be considered universally applicable. Its suitability depends on factors such as chosen modernization strategy, system architecture, project scope, available expertise, organizational readiness and technical constraints, meaning that each modernization context can require different mitigation strategies.

Furthermore, with the addition of mitigations, there are trade-offs and secondary risks because mitigations can reduce one Risk while creating additional effort, cost and/or complexity. Mitigation analysis therefore needs to consider both the intended Risk reduction and potential negative side effects.

As a condensed answer to RQ1, holistic legacy-system modernization is characterized by an interconnected network of Technical, Organizational, Business and Process risks where effective treatment of said dimensions requires combinations of context-dependent Preventive, Detective, Reactive and Recovery Mitigations. A useful modernization risk representation must therefore capture not only individual Risks and Mitigations but also Root Causes, Triggers, Impacts, contextual conditions, Risk Chains, Trade-offs and cross-dimensional Relationships. Finally, the identified knowledge should be interpreted as literature-grounded decision support knowledge rather than as universally valid causal rules or guaranteed project outcomes.

10.2.2. Answer to RQ2

After finishing the knowledge gathering part, RQ2 looks deeper into how to structure this knowledge by asking how modernization risk knowledge should be structured and formatted to maximize its utility for Large Language Models. The results support representing modernization knowledge through explicit and uniquely identifiable Risk, Mitigation and Relationship entities in a more structured form than basic text files. Risks and

Mitigations should not only contain short descriptions but also retain relevant context, evidence, provenance and applicability information that can be used in later analysis while stable IDs allow individual knowledge elements to be referenced directly in generated answers. This additionally improves traceability.

An important addition is an explicit relationship representation that overcomes implicit fields. These Relationships should be directed and use a consistent Source–Relationship–Target convention. Risk–Mitigation Relationships can represent prevention, reduction, detection or recovery effects that mitigations have on risks while Risk–Risk Relationships can represent cascading effects such as Triggers, Amplifies or Contributes To and Mitigation–Risk Relationships can represent newly introduced or amplified secondary Risks triggered by applying a mitigation strategy. The Relationship direction must be defined consistently to prevent semantic ambiguity.

Furthermore, for traceability, Evidence Type, Confidence Level, Applicability Conditions and Reasoning Notes are important components of the representation because they allow the system to distinguish directly documented Relationships from more interpretative ones. Additionally, information that is not available in the literature should remain explicitly unspecified instead of being completed through assumptions, supporting more transparent handling of uncertainty during LLM-based analysis. Paper IDs are used to trace back to the underlying literature if needed for back-checking the generated answers of an LLM.

All of this structure is implemented with the help of a new format: the open knowledge format (OKF). It demonstrates one possible representation combining machine readability and human readability as YAML metadata allows compact structured attributes to be parsed automatically, combined with Markdown sections which retain longer descriptions that are easier to inspect and modify by humans. Storing one knowledge object per file supports version control, manual maintenance and modular processing.

The evaluation demonstrates that providing domain-specific context clearly improves LLM performance compared with No Context. Because Explicit Relationships particularly improve the retrieval and coverage of relational knowledge the Full Structured Context showed substantially higher Recall than Structured Context without Relationships for tasks requiring explicit connections, demonstrating the importance of adding explicit relationships to the dataset. Nevertheless, the Proof-of-concept evaluation showed that detailed natural-language text descriptions retain information that can be valuable for LLM reasoning.

This indicates a trade-off between structured and unstructured representations as they provide different advantages. Structured knowledge improves semantic explicitness, relationship retrieval, traceability and efficiency immensely while on the other hand detailed natural-language context can preserve nuance and supporting information that may be lost through aggressive standardization. The most promising representation should therefore combine explicit structured knowledge objects with access to detailed contextual information if needed.

As a summary to this research question, Modernization risk knowledge should be represented using explicit Risk, Mitigation and Relationship entities, that should be typed

and directed with more detailed information with stable identifiers and a consistent schema. Standardized descriptions and Logic Patterns can support consistent LLM processing, while detailed source information should remain accessible to preserve context. Therefore, the results indicate that LLM utility is not maximized through structure alone but through a balance between explicit semantic structure, contextual richness, traceability, and efficient context selection.

10.3. Conclusion

The main conclusion of this thesis is that heterogeneous knowledge about legacy-system modernization Risks and Mitigations can be systematically transformed into a structured, traceable, and executable representation for LLM-supported analysis. The developed approach connects literature analysis, qualitative knowledge extraction, structured knowledge representation, technical implementation and an evaluation within one connected research process.

It stresses the importance of relationships in the dataset as separate entities, as collecting individual Risks and Mitigations alone is insufficient for holistic modernization analysis. Modernization projects are characterized by dependencies between technical, organizational, business and process factors. These dependencies are made accessible by providing explicit relationships thereby leading to an additional layer of knowledge rather than merely an organizational feature of a knowledge base.

The role and scope of the prototype is that of a research prototype and proof of concept that does not replace human expertise. It is not a complete operational risk-management platform and does not automatically predict the probability or financial impact of modernization Risks. Instead, it demonstrates how literature-derived modernization knowledge can be made systematically accessible for exploratory and analytical LLM-based interaction with possible practical uses, including early risk exploration or workshop preparation.

Concluding this thesis, structured external knowledge can improve the groundedness, traceability and relational capabilities of LLM-supported modernization analysis, in which explicit Relationships are particularly important for questions that require reasoning across multiple knowledge objects. However, structural formalization should preserve sufficient contextual detail rather than replacing it completely. The developed knowledge model therefore provides a foundation for more advanced retrieval-based, graph-based, and project-specific modernization decision-support systems.

10.4. Future Work

This final part of the thesis looks into what can be done in the future and further development of the prototype. First, the problem with the large dataset remains, calling for a replacement of complete context injection with selective retrieval, as the current approach becomes increasingly inefficient as the knowledge base and the number of Relationships grow. Retrieval could reduce context size, processing time, and the amount

of irrelevant information presented to the model and would additionally allow the knowledge base to scale beyond the size that can be provided within a single model context window.

Furthermore, knowledge retrieval could be implemented with semantic or graph-based counterparts. Embedding-based semantic search could be used to retrieve Risks and Mitigations that are conceptually related to a user query even when using different wordings. Metadata filters could additionally restrict retrieval according to Dimension, Modernization Strategy, Granularity Level, Evidence Type or Confidence Level. As the usable database contains a basic graph structure, graphs could be used to retrieve an initial relevant Risk and subsequently traverse connected Risks, Mitigations, secondary Risks, and supporting Relationships. Graph traversal could be particularly valuable for multi-hop questions and risk chain analysis. Both of these methods, however, need to be evaluated accordingly against the default full context scenario.

However, these approaches lead to using the extended relationship that is formulated but currently not in use due to context size restrictions. Retrieval architectures would allow the additional Medium Confidence Relationships to be used selectively, so future experiments should investigate whether the additional connectivity improves complex risk chain and scenario-based questions. It is important for accuracy reasons that High- and Medium-Confidence Relationships should remain distinguishable during retrieval and answer generation.

Furthermore, as the evaluation indicates that structured and unstructured representations have complementary strengths, a future architecture could combine concise structured Risk, Mitigation and Relationship objects with selected detailed literature passages or original extraction notes as an extension of the datasets that were developed in this thesis. Those detailed text passages could preserve contextual information and explanations that are difficult to capture completely in the standardized template fields. In general refinements of the template and the corresponding knowledge model are needed for basic improvement of answer quality.

Staying at the base level of the structured knowledge model, the current state is primarily based on manual literature screening, coding and structured extraction. This provides detailed control and traceability but requires considerable manual effort, so future work could investigate LLM-assisted extraction of candidates or, in general, (semi-)automatic extraction. Automatically generated candidates could subsequently be reviewed and approved by a human researcher or domain expert, so human involvement remains necessary. Modernization technologies, architectures, tools, and practitioner recommendations continue to evolve so a future version of the system could provide a workflow for continuously integrating newly published scientific and grey literature.

One central part in the prototype is the underlying system and combined user prompt that is provided to the LLM API. Therefore future work could systematically investigate alternative prompt structures because the current prompts provide explicit instructions regarding use of the knowledge base, ID referencing and unsupported information that is the same for every application scenario.

In terms of application, the most important point is testing of the proposed framework

in real-world legacy system modernizations. The system could be used during risk-identification workshops, architecture reviews, migration planning or strategy selection followed by an evaluation that could investigate whether relevant Risks are identified earlier or more systematically. Real-world evaluation would provide stronger evidence of practical usefulness than laboratory question answering alone. The expert evaluation could be expanded to a larger and more diverse participant group as well.

In this case, the prototype could be extended with dashboard functionality for practical modernization workshops. The interface could display mitigation coverage, unmitigated Risks, highly connected Risks and areas with weak evidence and would generally need more functionality to optimally aid a real modernization process. Future versions could for example also connect the framework to existing project-management, architecture-management, issue-tracking or risk-management platforms.

One other useful extension going hand in hand with real-world application is the Quantitative risk assessment. The current model intentionally focuses on qualitative knowledge and does not automatically calculate Risk probabilities or expected losses. Future research could investigate whether project-specific probability and impact information can be incorporated, however, this would require stronger empirical data than is currently available for many literature sources. Qualitative Confidence Levels should not simply be transformed into numerical probabilities.

Consequently, a long-term direction could be the development of a more comprehensive modernization advisory framework. All of the above could be combined in such a framework that can then speed up project planning and higher-level decisions. For that, the framework should be designed to adapt to organizations of different sizes, industries, technologies as well as modernization strategies. However, universal applicability across all companies should not be claimed without substantially broader empirical validation. The long-term goal should therefore be an adaptable decision-support framework rather than a universally valid autonomous modernization advisor.

Concluding this contribution, legacy system modernization will remain a complex and highly context-dependent problem where LLMs cannot remove the uncertainty associated with modernization decisions. However, structured, traceable and relationship-oriented knowledge representations can improve how existing modernization knowledge is accessed and explored. The developed prototype hence provides a foundation for more scalable Retrieval-Augmented, graph-based, and project-specific decision-support systems. Future work should therefore focus primarily on scalable retrieval, improved Relationship quality, hybrid context representations, broader evaluation and integration of organization-specific information as well as practical experience for the framework.

Declaration on the Use of AI Tools

AI tools, in particular ChatGPT by OpenAI, were used during the preparation of this thesis as supporting tools for language tasks. Their use was limited to improving wording, correcting grammar and spelling and refining formulations of text written by the author.

The identification and selection of literature, the extraction/interpretation of risks, mitigation strategies and relationships, the development of the knowledge model, the implementation of the prototype as well as the validation, evaluation and interpretation of the results were conducted by the author.

All AI assisted suggestions were critically reviewed, revised where necessary and verified by the author before being incorporated into the thesis. Responsibility for the content, correctness, argumentation and final wording of this thesis remains entirely with the author.

A. Literature Corpus and Search Documentation

A.1. Complete Extraction Corpus

The final extraction corpus contains 79 sources: 37 scientific-literature sources (P01–P37) and 42 grey-literature sources (G01–G42). The identifiers are the same identifiers used during extraction and analysis and therefore provide traceability from the reported findings back to the underlying source. Full bibliographic information is provided in the reference list; Tables A.1 and A.2 document the review-tracking information retained for each source.

The *Focus* column reproduces the thematic tags used during the review. The *Rel.* column reports the reviewer's screening-relevance score on a five-point scale (1 = low relevance to the thesis questions; 5 = very high relevance). This score was used as an internal prioritization aid and does not constitute a formal methodological quality appraisal of the source. The Notion tracking field *Status* is omitted because all 79 sources were marked as completed, and local file names are omitted because they are project-internal rather than bibliographic information.

Scientific Literature (P01–P37)

Table A.1.: Scientific-literature extraction corpus

ID	Title	Database / source	DOI / identifier / access	Focus	Rel.
P01	How Do Professionals Perceive Legacy Systems and Software Modernization?	ACM Digital Library	10.1145/2568225	Modernization	4/5
P02	Contemporary Software Modernization: Strategies, Driving Forces, and Research Opportunities	ACM Digital Library	10.1145/3708527	LLM; Modernization; Risk management	3/5
P03	Does Software Modernization Deliver What It Aimed for? A Post Modernization Analysis of Five Software Modernization Case Studies	IEEE Xplore	10.1109/ICSM.2015.7332499	Modernization; Risk management	4/5
P04	Investigating Systems Modernisation: Approaches, Challenges and Risks	Springer	Web / persistent link	Modernization; Risk management	5/5
P05	A Survey of Legacy System Modernization Approaches	DTIC	10.21236/ADA377453	Modernization	1/5
P06	Case Studies: Business and Technical Perspectives in Migration of Legacy Systems to Service Oriented Architecture	arXiv	arXiv:1412.7959	Modernization; Risk management	2/5
P07	Challenges in migrating legacy software systems to the cloud — an empirical study	ScienceDirect	10.1016/j.is.2017.03.008	Modernization; Risk management	4/5
P08	Modernizing Legacy Systems: A Scalable Approach to Next-Generation Data Architectures and Seamless Integration	Other	10.54660/.ijmrge.2023.4.1.901-909	LLM; Modernization; Risk management	5/5
P09	Navigating the Modernization of Legacy Applications and Data: Effective Strategies and Best Practices	Sage Journals	10.9734/ajrcos/2023/v16i4386	Modernization; Risk management	3/5
P10	Structuring the unstructured: an LLM-guided transition	Other	Web	LLM	1/5
P11	SR-LLM: Rethinking the Structured Representation in Large Language Model	ACM Digital Library	10.18653/v1/2025.acl-long.172	LLM	2/5
P12	Thinking with Knowledge Graphs: Enhancing LLM Reasoning Through Structured Data	arXiv	arXiv:2412.10654	LLM	5/5

Continued on next page

Table A.1 – continued from previous page

ID	Title	Database / source	DOI / identifier / access	Focus	Rel.
P13	A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT	arXiv	10.64346/plop2023p05	LLM	2/5
P14	Prompt Engineering for Large Language Models	Springer	10.2139/ssrn.4504303	LLM	2/5
P15	The Prompt Report: A Systematic Survey of Prompt Engineering Techniques	arXiv	arXiv:2406.06608	LLM; Risk management	2/5
P16	LLM-empowered knowledge graph construction: A survey	arXiv	arXiv:2510.20345	LLM; Risk management	1/5
P17	Reasoning with Graphs: Structuring Implicit Knowledge to Enhance LLMs Reasoning	ACM Digital Library	10.18653/v1/2025.findings-acl.1319	LLM	3/5
P18	Why Digital Transformation Fails: Human Resistance, Legacy Systems, and IT Modernization Challenges	ResearchGate	10.13140/RG.2.2.36366.11848	Modernization; Risk management	4/5
P19	Industrial perception of legacy software system and their modernization	Other	Web / persistent link	Modernization; Risk management	3/5
P20	The Roadmap to Mainframe Modernization: Bridging Legacy Systems with the Cloud	ResearchGate	10.32628/CSEIT25111214	Modernization; Risk management	5/5
P21	CMMI Compliant Modernization Framework to Transform Legacy Systems	Other	10.32604/iasc.2021.014280	Modernization; Risk management	3/5
P22	Software Maintenance and Evolution	ACM Digital Library	10.1145/336512.336534	Modernization; Risk management	2/5
P23	A Systematic Mapping Study on Legacy System Modernization	Emerald Insight	10.18293/seke2016-059	Modernization; Risk management	2/5
P24	Towards a Process for Migrating Legacy Systems into Microservices Architectural Style	Other	10.5753/eres.2020.13736	Modernization	1/5
P25	Migration of Existing or Legacy Software Systems into Web Service-based Architectures (Reengineering Process): A Systematic Literature Review	ResearchGate	10.5120/ijca2016907801	Modernization; Risk management	1/5
P26	Factors involved in Modernization Decision 2025	Sage Journals	Web / persistent link	Modernization; Risk management	3/5

Continued on next page

Table A.1 – continued from previous page

ID	Title	Database / source	DOI / identifier / access	Focus	Rel.
P27	AgentModernize: Preserving Business Logic in Legacy Modernization with Multi-Agent LLMs and Behavioral Specification Graphs	arXiv	arXiv:2605.17535	Modernization	2/5
P28	Migration Planning Framework for Legacy Systems' Cloud Migration	Other	Web	Modernization; Risk management	3/5
P29	SMART: Application of a Method for Migration of Legacy Systems to SOA Environments	Springer	10.1007/978-3-540-89652-4_60	Modernization; Risk management	3/5
P30	Modernizing Legacy Systems with Microservices: A Roadmap	ACM Digital Library	10.1145/3463274.3463334	Modernization	2/5
P31	A Systematic Mapping Study on the Modernization of Legacy Systems to Microservice Architecture	ResearchGate	10.3390/asi8040086	Modernization; Risk management	3/5
P32	A Survey on Knowledge Graphs: Representation, Acquisition, and Applications	IEEE Xplore	10.1109/tnnls.2021.3070843	Knowledge graphs; LLM	4/5
P33	Knowledge Graphs	ACM Digital Library	10.1145/3447772	Knowledge graphs; LLM	3/5
P34	Introduction: What Is a Knowledge Graph?	Springer	10.1007/978-3-030-37439-6_1	Knowledge graphs	2/5
P35	Migrating a Large Scale Legacy Application to SOA: Challenges and Lessons Learned	IEEE Xplore	10.1109/wcre.2013.6671318	Modernization; Risk management	4/5
P36	Contemporary Software Modernization: Perspectives and Challenges to Deal with Legacy Systems	arXiv	arXiv:2407.04017	Modernization; Risk management	5/5
P37	The risk management of everything	Emerald Insight	10.1108/eb023001	Risk management	5/5

Grey Literature (G01–G42)

Table A.2.: Grey-literature extraction corpus

ID	Title	Publisher / host	Access	Focus	Rel.
G01	Information Technology: Agencies Need to Develop and Implement Modernization Plans for Critical Legacy Systems	GAO / DTIC	Web	Modernization; Risk management	4/5
G02	Factors influencing the legacy system modernization strategy	JKU Linz repository	Web	Modernization; Risk management	4/5
G03	Legacy System Modernization Strategies	SEI / DTIC	Web	Modernization; Risk management	4/5
G04	Addressing Challenges on the Path to Success	ACT-IAC	Web	Modernization; Risk management	5/5
G05	Low-risk strategies for legacy system modernization in the public sector	OXD	Web	Modernization; Risk management	2/5
G06	EMRGF: A Practitioner Framework for Governance-Driven Enterprise Technology Modernization	arXiv	Web	Modernization	2/5
G07	The next step for your legacy workloads is AWS	AWS / IBM	Web	Modernization; Risk management	3/5
G08	Accelerating Legacy System Modernization in Government	ICIT	Web	Modernization	3/5
G09	The state of application modernization	Red Hat	Web	Modernization; Risk management	4/5
G10	Select your cloud migration strategies	Microsoft	Web	Modernization; Risk management	3/5
G11	The 6 Rs of application modernization	Microsoft	Web	Modernization	2/5
G12	Assess your workloads for cloud migration	Microsoft	Web	Modernization; Risk management	4/5
G13	Embracing the Strangler Fig pattern for legacy modernization Part one	Thoughtworks	Web	Modernization; Risk management	3/5
G14	Embracing the Strangler Fig pattern for legacy modernization Part two: Overcoming people and process challenges	Thoughtworks	Web	Modernization; Risk management	4/5

Continued on next page

Table A.2 – continued from previous page

ID	Title	Publisher / host	Access	Focus	Rel.
G15	Legacy Modernization Guide	Texas DIR	Web	Modernization	3/5
G16	Legacy Modernization: 6 architecture risks that could derail your program	Software Improvement Group	Web	Modernization; Risk management	4/5
G17	RECOMMENDATIONS TO MODERNIZE ARCHAIC AND INSECURE LEGACY SYSTEMS	MITRE	Web	Modernization; Risk management	4/5
G18	Searching for Systematic Reviews & Evidence Synthesis: Grey Literature	King's College London	Web	Grey Literature	5/5
G19	Grey literature: What it is & how to find it	Simon Fraser University	Web	Grey Literature	1/5
G20	Guide to Searching Grey Literature	University of Wolverhampton	Web	Grey Literature	4/5
G21	How to use grey literature	British Ecological Society	Web	Grey Literature	3/5
G22	Grey Literature: Finding grey literature	Monash University	Web	Grey Literature	2/5
G23	How Midsize Enterprise CIOs Fund Legacy Modernization	Gartner	Web	Modernization; Risk management	5/5
G24	Best Practices and Approaches to Application Modernization	SAP	Web	Modernization	3/5
G25	Why Delaying System Modernization is Your Biggest Business Risk in 2025?	SAP	Web	Modernization; Risk management	4/5
G26	Reimagine, Revamp, and Reinvent: A Dynamic Legacy Modernization Approach in 10 Steps	SAP	Web	Modernization	2/5
G27	Cloud Foundation	Oracle	Web	Modernization; Risk management	3/5
G28	What is COBOL modernization?	IBM	Web	Modernization	2/5
G29	What is mainframe modernization?	IBM	Web	Modernization	3/5
G30	From legacy systems to modern EDI solutions: A migration guide	IBM	Web	Modernization; Risk management	2/5

Continued on next page

Table A.2 – continued from previous page

ID	Title	Publisher / host	Access	Focus	Rel.
G31	Legacy application modernization: A comprehensive approach to modernize your business	IBM	Web	Modernization	4/5
G32	Strategic modernization with Oracle IAM	Oracle	Web	Modernization; Risk management	1/5
G33	Modernizing legacy systems	Fraunhofer IESE	Web	Modernization; Risk management	4/5
G34	Modernizing systems: Requirements analysis for legacy systems	Fraunhofer IESE	Web	Modernization; Risk management	2/5
G35	8 IT modernization traps CIOs must avoid	CIO	Web	Modernization; Risk management	4/5
G36	Overcoming the 6 barriers to IT modernization	CIO	Web	Modernization; Risk management	3/5
G37	How Much Risk Do Companies Incur by Avoiding IT Modernization?	LinkedIn	Web	Modernization; Risk management	4/5
G38	ISO 31000 - Risk management	ISO	Web	Risk management	3/5
G39	What is business risk management?	DNV	Web	Risk management	5/5
G40	Was versteht man unter Risikomanagement?	IBM	Web	Risk management	4/5
G41	Risikomanagement: Definition und leicht verständlicher Leitfaden zur Umsetzung	acquisa	Web	Risk management	3/5
G42	Technology Risk	KPMG	Web	Risk management	2/5

A.2. Search Strategy and Search Strings

The scientific literature search was conducted primarily through Google Scholar and was supplemented by direct access to publisher and repository platforms when relevant publications were identified. The documented search focus covered legacy systems, legacy-system modernization, modernization risks, modernization challenges, mitigation strategies, and, where required by the technical design of the thesis, knowledge graphs and large language models. The search was iterative: additional focused searches were performed during extraction when newly identified risks, mitigation strategies, terminology, or modernization concepts required further evidence or clarification.

Grey literature was searched separately through targeted web searches and practice-oriented sources. The final grey-literature corpus includes material from public-sector and standards organizations, research and advisory organizations, universities, and technology vendors. Examples include GAO, MITRE, ISO, Fraunhofer IESE, Gartner, Microsoft, IBM, AWS, Oracle, SAP, Red Hat, and Thoughtworks. Grey-literature sources were used where they contributed practice-oriented evidence, modernization guidance, implementation experience, risk descriptions, or concrete mitigation approaches that complemented the scientific literature.

Exact query strings were not consistently retained for every iteration of the search. Accordingly, this appendix reports the documented search themes, the complete final extraction corpus, and the source-level traceability information rather than reconstructing search strings retrospectively. This limitation should be considered when assessing the reproducibility of the search process.

A.3. Screening and Selection Information

Sources were included when they were directly related to legacy systems or legacy-system modernization, addressed risks, challenges, mitigation strategies, or supporting technical concepts relevant to the research questions, contained sufficient substantive information for extraction, and were traceable to an identifiable publication or web source. Sources were excluded when they were primarily promotional without substantive content, did not satisfy the defined language criteria, were insufficiently related to the research questions, or did not provide accessible full text or enough information for reliable extraction.

Grey-literature sources were additionally considered when they documented practitioner experience, modernization cases, recurring risk factors, implementation guidance, or concrete mitigation approaches. The relevance score shown in Tables A.1 and A.2 was used to prioritize sources during analysis; it was not used as a formal quality-assessment instrument or as an automatic exclusion threshold.

No complete quantitative record of all initially retrieved, deduplicated, screened, and excluded search results was retained. Consequently, exact screening-flow counts cannot be reported retrospectively. The final corpus of 79 included sources, the assigned source identifiers, and the traceability of extracted findings to these identifiers provide the available documentation of the selection and extraction process.

B. Extraction Templates

This appendix documents the line-oriented templates used during the manual extraction of modernization knowledge. The templates provide a stable structure for information that is reported with different terminology and levels of detail across the literature. Fields are only filled when they are supported by the underlying source or by an explicitly documented analytical classification. Missing information remains unspecified rather than being inferred. Standardized entries and logic patterns are normalized representations created for later LLM-oriented use, whereas quotations preserve the connection to the original source wording.

The templates shown below correspond to the raw text representation used as input to the conversion pipeline. Risk and Mitigation entries are extracted as independent knowledge objects. Relationships are represented separately so that the type, direction, evidence, applicability, and confidence of a connection are not reduced to a simple cross-reference. The cluster attribute is not part of the manual extraction template. It is added in the later OKF representation and therefore appears in the framework artifacts shown in Appendix C.

B.1. Risk Template

The Risk template captures an identified modernization risk together with its context, causal structure, consequences, evidence, and LLM-oriented normalization. Root cause, trigger, risk, and impact are treated as distinct concepts.

`Risk ID` Unique and stable identifier of the risk, using the prefix `R`.

`Paper ID` Identifier or identifiers of the scientific or grey-literature sources from which the entry was extracted.

`Risk Name` Short, standardized, descriptive name of the risk.

`Risk Description` Explanation of what the risk represents and why it is problematic during modernization.

`Modernization Strategy` Modernization approach or approaches in which the risk is relevant, for example migration, replacement, refactoring, rehosting, or another strategy used in the source material.

`Dimension` Primary modernization dimension: Technical, Organizational, Business, or Process.

`Granularity Level` Main level at which the risk occurs, for example Enterprise-wide, System-wide, Component-level, or Code-level.

`Root Cause / Challenge` Underlying condition, challenge, or problem that increases the possibility of the risk.

`Trigger Event` Action, event, or situation through which the risk can become active or materialize.

`Impact / Consequences` Reported or expected consequences if the risk materializes.

`Linked Risks` Lightweight references to related risks. Detailed causal semantics are represented through separate Relationship entries.

`Stakeholders` Actors affected by the risk or responsible for managing it.

`Contextual Constraints` Conditions under which the risk is particularly applicable or relevant.

`Evidence Type` Form of supporting evidence, for example Statistical, Case Study, Expert Opinion, or Anecdotal.

`Detection Source` Practical source from which the risk may be detected or assessed, for example Source Code, Project Plan, Architecture Diagram, Interview, or Operational Data.

`Key Terminology (Tags)` Important terms retained for filtering, retrieval, and later guidance.

`Standardized Risk Entry` One- or two-sentence normalized summary of the risk for consistent later processing.

`Key Quote for Prompting` Relevant quotation from the underlying literature that preserves traceability to the source wording.

`Logic Pattern` Conditional normalization in the form `IF [Condition] AND [Trigger] THEN [Risk]`.

B.2. Mitigation Template

The Mitigation template captures a modernization action or strategy independently from a specific risk. This is necessary because one mitigation may address multiple risks and may at the same time introduce additional trade-offs or secondary risks.

`Mitigation ID` Unique and stable identifier of the mitigation, using the prefix `M`.

`Paper ID` Identifier or identifiers of the scientific or grey-literature sources from which the entry was extracted.

`Mitigation Name` Short, standardized, descriptive name of the mitigation.

`Related Risks` Lightweight references to risks associated with the mitigation. The exact effect on a risk is represented through separate Relationship entries.

`Mitigation Description` Description of the action, strategy, or procedure used to address modernization risk.

`Strategy Type` Classification as Preventive, Detective, Reactive, or Recovery.

`Prerequisites` Conditions, knowledge, resources, or system characteristics that must exist before the mitigation can be applied.

`Tools / Technologies` Tools, frameworks, platforms, or technologies explicitly associated with the mitigation.

`Trade-offs / Side Effects` Costs, complexity, dependencies, disruption, or secondary risks that may be introduced by applying the mitigation.

`Effectiveness` Reported qualitative or quantitative information about the effect of the mitigation.

`Probability of Success` Qualitative success assessment: Low, Medium, High, or Not specified when supported by the source.

`Effort of Mitigation` Qualitative effort required to apply the mitigation: Low, Medium, or High when supported by the source.

`Stakeholders` Actors who implement, support, or benefit from the mitigation.

`Contextual Constraints` Conditions under which the mitigation is applicable.

`Evidence Type` Form of supporting evidence, for example Statistical, Case Study, Expert Opinion, or Anecdotal.

`Key Terminology (Tags)` Important terms retained for filtering, retrieval, and later guidance.

`Standardized Mitigation Entry` One- or two-sentence normalized summary of the mitigation for consistent later processing.

`Key Quote for Prompting` Relevant quotation from the underlying literature that preserves traceability to the source wording.

`Logic Pattern` Conditional normalization in the form `IF [Condition] THEN APPLY [Mitigation]`.

B.3. Relationship Template

Relationships are stored as separate knowledge objects because their attributes describe the connection between two entities rather than either endpoint in isolation. Three endpoint combinations are used in the raw extraction data. They share the same common relationship fields, while the identifier fields and allowed relationship types depend on the direction of the connection.

Risk–Mitigation Relationship

This variant links a risk to a mitigation that addresses it.

`Relationship ID` Unique and stable identifier of the relationship, using the prefix `Rel`.

`Risk ID` Referenced risk endpoint.

`Mitigation ID` Referenced mitigation endpoint.

`Relationship Type` Raw extraction relation, typically Prevents, Reduces, Detects, or Recovers. In the OKF representation, the relationship direction is encoded through explicit source and target identifier fields, while the stored relationship type itself is retained (for example, `Prevents`).

Risk–Risk Relationship

This variant represents a causal, amplifying, contributing, or dependency connection between two risks.

`Relationship ID` Unique and stable identifier of the relationship.

`Source Risk ID` Risk from which the directed relationship originates.

`Target Risk ID` Risk affected by the source risk.

`Relationship Type` Triggers, Amplifies, Contributes To, or Depends On.

Mitigation–Risk Relationship

This variant captures a secondary risk that is introduced or amplified by a mitigation.

`Relationship ID` Unique and stable identifier of the relationship.

`Source Mitigation ID` Mitigation from which the directed relationship originates.

`Target Risk ID` Risk introduced or affected by the mitigation.

`Relationship Type` Introduces or Amplifies.

Common Relationship Fields

The following fields are used across the relationship variants where applicable.

`Influence Strength` Qualitative assessment of the strength of the connection: Low, Medium, High, or Unknown.

`Secondary Risk Created` Additional risk or negative effect created by the relationship, where applicable.

`Applicability Conditions` Conditions under which the relationship is expected to hold.

`Observed Outcome` Effect reported in the literature or in a documented case when available.

`Evidence Type` Form of evidence supporting the connection, for example Statistical, Case Study, Expert Opinion, or Anecdotal.

`Confidence Level` Qualitative assessment of how strongly the available evidence supports the modeled relationship: Low, Medium, or High.

`Reasoning Notes` Explanation of the analytical rationale, especially when the relationship is inferred by combining findings rather than being stated directly in one source.

`Standardized Relationship Entry` One- or two-sentence normalized summary of the relationship for later LLM-oriented use.

B.4. Example of a Filled-Out Raw Extraction Entry

The following entry is reproduced in the same line-oriented form used by the raw text files. It uses Risk `R1` so that the subsequent OKF example can continue with the connected Mitigation `M1` and Relationship `Rel3`. The example therefore illustrates the transition from manual extraction to the structured representation without switching to an unrelated knowledge object.

```
Risk ID: R1
Paper ID: P01/P02
Risk Name: Incomplete or Erroneous Data Migration
Risk Description: Errors occurring during the transfer of data from old (often
non-relational) databases to modern systems, leading to data corruption or
system disturbance.
Modernization Strategy: Migration / Replacement
Dimension: Technical
Granularity Level: Component-level (Database)
Root Cause / Challenge: Use of obsolete database models that do not support
modern relational structures.
```

Trigger Event: Execution of the cut-over or data ETL (Extract, Transform, Load) phase.
Impact / Consequences: New system instability, loss of business-critical data, high regaining costs.
Linked Risks: System Downtime, Business Disruption.
Stakeholders: Database Administrator, Software Architect, Data Analyst
Contextual Constraints: Old non-supported databases, big data
Evidence Type: Case Study / Expert Opinion
Detection Source: Database Scheme Audit
Key Terminology (Tags): #LegacyModernization, #DataMigration
Standardized Risk Entry: Data migration is a critical risk factor where errors in transferring data from obsolete structures to modern environments can lead to system instability and business disruption.
Key Quote for Prompting: ``The main risk in modernization is that the data migration, which cannot be done perfectly. Errors are made and you have some risk that your new system is disturbed.''
Logic Pattern: IF [Non-Relational Database] AND [High Business Criticality] THEN [Incomplete or Erroneous Data Migration]

C. OKF-Based Representation Example

This appendix illustrates the OKF-based Markdown representation used by the implemented knowledge framework. In contrast to the raw extraction format documented in Appendix B, the OKF representation combines compact metadata in YAML frontmatter with longer descriptive content in named Markdown sections. The artifacts below reproduce the field names and values as they are stored in the framework.

The example uses Risk `R1`, Mitigation `M1`, and Relationship `Rel3`. The relationship explicitly connects `R1` to `M1` using `source_risk_id` and `target_mitigation_id`.

C.1. Risk Object: `R1.md`

```
---
id: R1
type: Risk
title: Incomplete or Erroneous Data Migration
cluster: C01
dimension: Technical
granularity: Component-level (Database)

papers:
  - P01
  - P02

strategy:
  - Migration
  - Replacement

evidence:
  - Case Study
  - Expert Opinion

tags:
  - LegacyModernization
  - DataMigration

---

# Description
Errors occurring during the transfer of data from old
(often non-relational) databases to modern systems, leading
to data corruption or system disturbance.

# Root Cause
```

```
Use of obsolete database models that do not support modern
relational structures.

# Trigger Event
Execution of the cut-over or data ETL (Extract, Transform,
Load) phase.

# Impact
New system instability, loss of business-critical data,
high regaining costs.

# Linked Risks
- System Downtime
- Business Disruption.

# Stakeholders

- Database Administrator
- Software Architect
- Data Analyst

# Contextual Constraints

- Old non-supported databases
- big data

# Detection Source
Database Scheme Audit

# Standardized Risk Entry
Data migration is a critical risk factor where errors in
transferring data from obsolete structures to modern
environments can lead to system instability and business
disruption.

# Key Quote for Prompting
> "The main risk in modernization is that the data migration,
which cannot be done perfectly. Errors are made and you have
some risk that your new system is disturbed."

# Logic Pattern
IF [Non-Relational Database] AND [High Business Criticality]
THEN [Incomplete or Erroneous Data Migration]
```

C.2. Mitigation Object: `M1.md`

```
---
id: M1
type: Mitigation
title: Data Migration Strategy
cluster: C01
```

```
papers:
  - P01

strategy_type: Proactive (Prevention)
probability_of_success:
effort: High

evidence:
  - Expert Opinion

tags:
  - MigrationStrategy
  - EarlyPlanning

related_risks:
  - R1
  - R4
  - R12
  - R15
  - R29

---

# Description
Develop a comprehensive strategy for data migration at the
very beginning of the project, including mapping and cleaning.

# Prerequisites
Deep understanding of the legacy data model

# Tools / Technologies


# Trade-offs / Side Effects
May increase initial project duration due to extensive early
planning requirements.

# Effectiveness
Medium

# Stakeholders
- Data Architect
- Project Manager

# Contextual Constraints

- Have to be done before migrating data

# Standardized Mitigation Entry
Developing a comprehensive data migration strategy early in
the modernization lifecycle is essential to mitigate the risk
of data loss and system failure.
```

```
# Key Quote for Prompting
> "If you are doing migration of your legacy, you have to
prepare a good data migration strategy"

# Logic Pattern
IF [Early Strategic Planning] THEN
[Reduced Risk of Migration Disturbance]
```

C.3. Risk–Mitigation Relationship: `Rel3.md`

The relationship is reproduced as represented in the framework: the risk is the source and the mitigation is the target. The implementation stores the relationship type as `Prevents`.

```
---
id: Rel3
type: Relationship

source_risk_id: R1 (Incomplete or Erroneous Data Migration)
target_mitigation_id: M1 (Data Migration Strategy)

relationship_type: Prevents
influence_strength: Medium
confidence_level: Medium

evidence_type: Expert Opinion

---

# Secondary Risk Created
May increase initial project duration.

# Applicability Conditions
Deep understanding of the legacy data model.

# Observed Outcome
Essential to mitigate the risk of data loss and system failure.

# Reasoning Notes
Proper data cleansing and schema mapping before migration
directly address the root causes of incomplete transfers.

# Standardized Relationship Entry
IF data is mapped and cleaned proactively THEN incomplete or
erroneous migration instances are heavily reduced.
```

C.4. Interpretation of the Example

The example demonstrates the separation of risks, mitigations, and explicit relationship objects in the implemented OKF representation. Risk `R1` and Mitigation `M1` are formally

connected through Relationship `Rel3`. The relationship object preserves the identifiers of the connected entities together with additional information such as relationship type, influence strength, confidence level, evidence type, applicability conditions, and reasoning notes.

D. Evaluation Materials and Detailed Results

This appendix contains the complete question and task material used for the functional validation, comparative proof-of-concept evaluation, and external expert evaluation. It additionally documents the detailed developer validation results, the question-level comparative evaluation results, the final expert questionnaire, and the individual anonymized expert ratings used for the aggregated results presented in Chapter 8.

D.1. Functional Validation Questions

The functional validation used 20 predefined questions. In line with the validation design described in Chapter 4, the questions cover basic retrieval, relationship retrieval, trade-offs and secondary risks, risk chains, simple consulting scenarios, and knowledge-boundary tests.

Basic retrieval.

1. What is risk R1, what causes it, and what consequences can it have?

2. What mitigation strategies are available for the scarcity of experts and documentation risk R2?

3. Describe mitigation M6, including its prerequisites, effort, effectiveness, stakeholders, and trade-offs.

Relationship retrieval.

4. Which mitigations are linked to risk R21, and how does each mitigation influence the risk?

5. Which risks are addressed by mitigation M1? For each risk, provide the relationship type, influence strength, and applicability conditions.

6. Explain the relationship between risk R1 and mitigation M1. Include the corresponding Relationship ID and supporting evidence.

Trade-offs and secondary risks.

7. Can mitigation M1 both reduce and introduce cost-related risks? Explain the different relationships using the relevant IDs.
8. Which risks can be reduced or prevented by component decoupling M14, and which new risks can it introduce?
9. What secondary risks or side effects may arise when applying a phased migration strategy?

Risk chains.

10. Which risks can be triggered by the scarcity of experts and documentation R2? Show the relevant risk-to-risk relationships.
11. Explain the risk chain from poor project planning R12 to cost overrun R4.
12. Construct a risk chain beginning with incomplete or erroneous data migration R1 and describe its possible downstream consequences.

Simple consulting scenarios.

13. A company has an undocumented legacy system, and its key developers will retire soon. What are the most relevant risks and which mitigations should be applied?
14. A company plans to migrate a mission-critical monolithic system to the cloud while minimizing downtime. Which risks and mitigation strategies are relevant?
15. A company wants to decompose a monolithic system into microservices. What risks should it consider, and what mitigations can reduce them?
16. A modernization project has poor documentation, inconsistent data, a limited budget, and a fixed deadline. Identify the three most relevant risks and recommend suitable mitigations.

Knowledge-boundary test.

17. Is there a risk with the ID R999 in the knowledge base? Do not invent an answer.
18. Does the knowledge base provide an exact numerical probability that risk R2 will occur?
19. Recommend a mitigation for R1 using only explicitly documented relationships. Do not suggest unsupported mitigation strategies.
20. Which parts of your answer are directly stated in the knowledge base, and which parts are your own inference?

D.1.1. Detailed Functional Validation Results

The functional validation was evaluated directly against the modeled knowledge base and its Relationships. In addition to the question list above, Table D.1 documents the result for each individual validation question. The *Main result* column summarizes the generated answer, while the *Evaluation* column records the reason for the assigned score. The complete validation reached 94 out of 100 points, corresponding to an average score of 4.7 out of 5.

Table D.1.: Detailed results of the 20-question functional validation.

No.	Question	Main result	Score	Evaluation
1	What is risk R1, what causes it, and what consequences can it have?	The answer correctly identified R1 as incomplete or erroneous data migration and described its root cause, trigger, direct consequences, and possible downstream risks.	5/5	The answer was complete and well supported by the risk and relationship entries.
2	What mitigation strategies are available for risk R2?	M2, M4, M88, and M7 were identified as relevant mitigations for missing experts and documentation.	5/5	All named mitigations were connected to R2 through documented relationships.
3	Describe mitigation M6, including prerequisites, effort, effectiveness, stakeholders, and trade-offs.	The answer correctly described the Proof of Concept mitigation and included all requested fields.	5/5	The answer was complete and also included the documented initial cost and time trade-off.
4	Which mitigations are linked to risk R21, and how does each influence the risk?	M13, M17, and M19 were correctly identified as recovery, prevention, or prevention/recovery measures for downtime.	5/5	All explicit relationships to R21 were found and correctly described.
5	Which risks are addressed by mitigation M1?	The answer correctly listed R1, R4, R12, R15, and R29, including relationship type, strength, and applicability conditions.	5/5	The result matches the related risks and relationship entries of M1.
6	Explain the relationship between R1 and M1.	M1 prevents or reduces R1 through early data mapping, cleansing, and migration planning. Rel3 was correctly referenced.	5/5	The relationship type, strength, condition, and supporting evidence were accurately presented.
7	Can M1 both reduce and introduce cost-related risks?	The answer showed that M1 prevents R4 through Rel1 but may introduce initial cost pressure through Rel249.	5/5	The answer successfully demonstrated that a mitigation can have both positive and negative effects.
8	Which risks can be reduced by M14, and which risks can it introduce?	M14 prevents R24 and reduces R19, but it may introduce R18 and additional architectural complexity.	4/5	The main answer was correct. However, Rel250 contains an internal mismatch between its target risk and its written explanation, which the original answer did not mention.
9	What secondary risks or side effects may arise from a phased migration strategy?	The answer identified longer timelines, dual maintenance, staffing pressure, coexistence complexity, and synchronization risks.	4/5	The general conclusions were relevant, but some relationships applied specifically to the Strangler Pattern rather than directly to M17. These points should have been marked as inference.
10	Which risks can be triggered by R2?	R2 was correctly connected to R1 through Rel338 and R3 through Rel256.	5/5	Both explicit outgoing risk-to-risk relationships were correctly found and explained.
11	Explain the risk chain from R12 to R4.	Poor planning leads to unidentified complexity, unrealistic estimates, and finally cost overrun.	5/5	The causal chain was clearly explained and supported by Rel237 and Rel257.
12	Construct a risk chain beginning with R1.	The answer produced the chains R1 → R4 and R1 → R10 → R9 → R5 → R3 → R4, as well as an R1 → R29 branch.	3.5/5	Most links were correct. However, Rel255 contains contradictory source and target information, and the answer did not flag this inconsistency.

Continued on next page

Table D.1 continued from previous page

No.	Question	Main result	Score	Evaluation
13	A company has an undocumented legacy system and its key developers will retire soon. What risks and mitigations are relevant?	The answer identified R2, R94, R62, and R119 and recommended M88, M2, M7, M55, and M52.	4.5/5	The answer was practical and well grounded. M52 was relevant but had a more indirect relationship to the scenario.
14	A mission-critical monolith will be migrated to the cloud while minimizing downtime. Which risks and mitigations are relevant?	The answer covered downtime, fallback, resilience, compatibility, data, performance, security, compliance, and coexistence.	4.5/5	The answer showed strong advisory capability. A few recommendations were based on contextual synthesis rather than direct relationships.
15	A company wants to decompose a monolith into microservices. What risks and mitigations should it consider?	The answer covered feasibility, service boundaries, coupling, granularity, data, testing, API, team, and governance risks.	4/5	The result was comprehensive, although some mitigation mappings were inferred rather than explicitly documented.
16	A project has poor documentation, inconsistent data, a limited budget, and a fixed deadline. Identify three risks and mitigations.	R2, R48, and R100 were selected, together with M88, M2, M39, M1, M91, and M93.	4.5/5	The selected risks fit the scenario well. R3 and R4 would also have been reasonable alternatives.
17	Is there a risk with ID R999?	The model stated that the risk was not in the knowledge base.	5/5	The answer followed the fallback instruction and did not invent an entry.
18	Does the knowledge base provide an exact numerical probability for R2?	The model stated that this information was not in the knowledge base.	5/5	No unsupported numerical probability was generated.
19	Recommend a mitigation for R1 using only explicitly documented relationships.	M1 was recommended based on the explicit relationship Rel3.	5/5	The answer followed the restriction and did not suggest unsupported measures.
20	Which parts of your answer are directly stated in the knowledge base, and which parts are inference?	The answer distinguished stored fields and explicit relationships from synthesis, ranking, and contextual interpretation.	5/5	The response also correctly identified that the current API request did not contain the preceding answer.

D.2. Comparative Proof-of-Concept Evaluation Tasks

The comparative proof-of-concept evaluation used 35 predefined questions. The same questions were executed under all four context configurations while keeping the model, system prompt, and model parameters unchanged. The category assigned to each question corresponds to the evaluation dataset used for the question-level analysis.

Q01 *Direct Knowledge.* What are the root cause, trigger event, impact, and stakeholders of R1 – Incomplete or Erroneous Data Migration?

Q02 *Direct Knowledge.* What causes R2 – Scarcity of Experts and Documentation, and what consequences can it have for a modernization project?

Q03 *Mitigation Identification.* Which mitigations address R2 – Scarcity of Experts and Documentation, and how do they support knowledge recovery?

Q04 *Mitigation Identification.* Which risks are addressed by M6 – Proof of Concept? Include the Relationship types and applicability conditions.

Q05 *Comparison.* Compare R5 – Cultural and Staff Resistance with R9 – User Resistance.

Q06 *Direct Knowledge.* What are the main causes and consequences of R10 – Post-Modernization Performance Degradation?

Q07 *Mitigation Identification.* Which mitigation is documented for R18 – Orchestration and Complexity Explosion, and how does it control service complexity?

Q08 *Relationship.* Explain the Relationship between R1 – Incomplete or Erroneous Data Migration and M1 – Data Migration Strategy.

Q09 *Relationship.* How does M2 – Automated Logic Extraction address R2 – Scarcity of Experts and Documentation?

Q10 *Relationship.* How can M2 – Automated Logic Extraction help detect R8 – Transfer of Legacy Flaws?

Q11 *Relationship.* How does M5 – Digital Twin Simulation address R10 – Post-Modernization Performance Degradation?

Q12 *Relationship.* Explain how M6 – Proof of Concept detects R20 – API and Data Incompatibility.

Q13 *Relationship.* How does M7 – Staff Training and Developer Education reduce R5 and R9?

Q14 *Relationship.* Explain how M8 – Comprehensive Upfront Planning and Risk Analysis addresses R12 – Project Failure due to Poor Planning.

Q15 *Synthesis.* Which risks are addressed by M3 – Transition to Standard Software Products, and which secondary risks may this mitigation introduce?

Q16 *Risk Chain.* Create a risk chain from R2 – Scarcity of Experts and Documentation to R3 – Time Overrun.

Q17 *Risk Chain.* Explain how R12 – Project Failure due to Poor Planning can trigger R4 – Cost Overrun.

Q18 *Risk Chain.* Explain how R1 – Incomplete or Erroneous Data Migration can trigger R4 – Cost Overrun.

Q19 *Risk Chain.* How can R9 – User Resistance develop into R5 – Cultural and Staff Resistance?

Q20 *Secondary Risk.* Which risk may be introduced by M14 – Component Decoupling when microservices are created without strong governance?

Q21 *Secondary Risk.* Which risk may be introduced by M3 – Transition to Standard Software Products, and why?

Q22 *Risk Chain.* Explain how R16 – Lack of Legacy System Disposition Strategy can lead to R135 – Prolonged Dual-System Transition.

Q23 *Scenario.* A company migrates data from an obsolete non-relational database with incomplete documentation. Identify the relevant risks, mitigations, and Relationships.

Q24 *Scenario.* A monolithic legacy system is decomposed into many microservices without central governance or service-level agreements. Which risks may occur, and which mitigation should be applied?

Q25 *Scenario.* A mainframe application is rehosted to a distributed environment with additional firewalls and security layers. Which risk should be expected, and how can it be detected before full migration?

Q26 *Scenario.* Several legacy experts will retire before the modernization project is completed. The documentation is outdated. Which risks and mitigations are relevant?

Q27 *Scenario.* Employees resist a new graphical interface because they prefer the existing terminal-based workflow. Identify the risks, mitigation, stakeholders, and possible side effects.

Q28 *Scenario.* A company plans a large-scale modernization without a clear budget, roadmap, or legacy-system decommissioning plan. Identify the most relevant risks and mitigations.

Q29 *Scenario.* A Proof of Concept detects performance and API compatibility problems. Explain which documented risks were identified and how this information should influence the full modernization project.

Q30 *Scenario.* A company replaces custom legacy software with a standard commercial product. Identify both the risks that may be reduced and the new risks that may be introduced.

Q31 *Hallucination Test.* What is the exact percentage probability that M1 – Data Migration Strategy will prevent R1? State clearly when the information is not documented.

Q32 *Hallucination Test.* Does M6 – Proof of Concept guarantee that modernization project failure will be completely eliminated?

Q33 *Hallucination Test.* Does M14 – Component Decoupling prevent R18 – Orchestration and Complexity Explosion? Verify and correct the statement using the documented Relationships.

Q34 *Structured Output.* Return the risks, mitigations, Relationships, applicability conditions, secondary risks, evidence types, and confidence levels for a poorly documented data migration as valid JSON.

Q35 *Structured Output.* Create a machine-readable risk chain for R2 → R3 → R4, including all available Risk IDs, Mitigation IDs, Relationship IDs, Relationship types, and confidence levels.

D.2.1. Detailed Comparative Evaluation Results

The 35 comparative questions were evaluated with the same scoring procedure used in Chapter 8. The Overall Score combines identifier F1 (30%), Gold Fact Accuracy (25%), Gold Fact Coverage (20%), Groundedness (15%), and Format Quality (10%). Table D.2 first repeats the final configuration-level summary so that the detailed question-level values below can be interpreted directly. Table D.3 then reports the Overall Score for every individual question under all four context configurations. The more detailed Precision, Recall, F1, and ID Hallucination Rate values were used for the analysis in Chapter 8; the question-level Overall Score is shown here as the compact detailed result rather than reproducing twenty metric columns.

Table D.2.: Summary of the final comparative proof-of-concept evaluation.

Context configuration	Precision	Recall	F1	ID Hall.	Overall	Runtime
Full Structured Context	0.778	0.885	0.813	0.165	87.58	2min 45s
Structured Context without Relationships	0.801	0.614	0.667	0.170	75.12	2min 26s
No Context	0.608	0.444	0.497	0.249	46.87	1min 3s
Unstructured Full Context	0.877	0.952	0.893	0.123	92.72	23min 16s

Table D.3.: Question-level Overall Scores across the four context configurations.

ID	Category	Full Structured	No Relationships	No Context	Unstructured
Q01	Direct Knowledge	**100.00**	**100.00**	41.00	**100.00**
Q02	Direct Knowledge	**100.00**	88.50	78.00	97.50
Q03	Mitigation Identification	**92.05**	72.14	38.23	85.71
Q04	Mitigation Identification	**99.50**	72.69	23.87	93.50
Q05	Comparison	89.58	74.00	82.00	**97.00**
Q06	Direct Knowledge	**99.00**	97.00	54.50	**99.00**
Q07	Mitigation Identification	**100.00**	90.00	32.00	**100.00**
Q08	Relationship	88.00	81.50	65.50	**100.00**
Q09	Relationship	91.21	81.50	70.00	**100.00**
Q10	Relationship	**100.00**	77.50	70.00	**100.00**
Q11	Relationship	**93.21**	81.50	70.00	80.50
Q12	Relationship	**100.00**	85.00	70.00	**100.00**
Q13	Relationship	75.32	80.00	44.50	**97.00**
Q14	Relationship	89.50	83.50	72.00	**95.00**
Q15	Synthesis	86.40	70.95	27.84	**87.00**
Q16	Risk Chain	81.00	81.50	65.50	**100.00**
Q17	Risk Chain	**95.50**	81.50	70.00	80.50
Q18	Risk Chain	81.93	81.50	55.64	**100.00**
Q19	Risk Chain	**100.00**	76.45	68.00	91.50
Q20	Secondary Risk	98.50	32.00	30.50	**100.00**
Q21	Secondary Risk	**100.00**	91.00	41.00	93.00
Q22	Risk Chain	**100.00**	85.00	76.00	**100.00**
Q23	Scenario	**90.50**	63.07	49.32	83.32
Q24	Scenario	76.00	70.83	11.00	**84.64**
Q25	Scenario	**83.50**	62.50	13.50	**83.50**
Q26	Scenario	**100.00**	70.00	31.41	97.00
Q27	Scenario	85.50	**92.00**	11.00	85.50
Q28	Scenario	75.11	66.00	23.00	**81.40**
Q29	Scenario	**79.47**	73.35	29.00	79.00
Q30	Scenario	98.80	66.63	37.18	**100.00**
Q31	Hallucination Test	55.00	81.00	49.00	**91.50**
Q32	Hallucination Test	88.50	88.50	65.00	**100.00**
Q33	Hallucination Test	11.00	63.50	11.00	**86.00**
Q34	Structured Output	**88.95**	62.10	16.00	87.45
Q35	Structured Output	72.26	5.00	48.00	**88.86**
	Mean Overall Score	**87.58**	**75.12**	**46.87**	**92.72**
	Median Overall Score	90.50	80.00	48.00	95.00

Bold values indicate the highest Overall Score for the respective individual question. Ties are therefore shown with more than one bold value. These question-level differences are descriptive results of the controlled proof-of-concept run and should be interpreted together with the aggregate metrics and runtime comparison in Chapter 8.

D.3. External Expert Evaluation Tasks

The external expert evaluation reduced the interaction to six predefined task types so that the same main interaction patterns could be evaluated in a smaller setup. The following wording reproduces the task prompts provided to the experts.

Task 1 – Risk Information Retrieval. What are the root cause, trigger event, impact, and stakeholders of R1 – Incomplete or Erroneous Data Migration?

Task 2 – Direct Risk–Mitigation Relationships. Explain the documented relationships between:

- R1 – Incomplete or Erroneous Data Migration and M1 – Data Migration Strategy
- R2 – Scarcity of Experts and Documentation and M2 – Automated Logic Extraction

Task 3 – Indirect Risk Chain. Create and explain the risk chain:
R2 – Scarcity of Experts and Documentation → R3 – Time Overrun → R4 – Cost Overrun

Task 4 – Scenario-Based Analysis. A company migrates data from an obsolete non-relational database with incomplete documentation. Several legacy experts will also retire before the modernization project is completed.
Identify the relevant risks, mitigation strategies, and relationships.

Task 5 – Unsupported or Incorrect Information. Evaluate and correct the following statements. Clearly state when information is not documented:

(a) What is the exact percentage probability that M1 will prevent R1?

(b) M6 – Proof of Concept guarantees that project failure will be completely eliminated.

(c) M14 – Component Decoupling prevents R18 – Orchestration and Complexity Explosion.

Task 6 – Machine-Readable Output. Create the risk chain R2 → R3 → R4 as valid JSON.

Include all available:

- Risk IDs
- Mitigation IDs
- Relationship IDs
- Relationship types
- Confidence levels

Do not invent undocumented values.

D.4. Expert Questionnaire and Rating Criteria

The questionnaire first collected consent and a short professional background before the four context conditions were evaluated. Under *Consent to Participate*, the available responses were *No, I do not agree to participate.* and *Yes, I agree to participate.* The professional-background fields were *Professional role*, *Area of expertise*, and *Years of professional experience*.

The four context conditions were presented as follows:

- **Context 1 – Full Structured Context.** In this condition, the system receives the full structured knowledge context, including risks, mitigation strategies, and their documented relationships.
- **Context 2 – No Relationships Structured Context.** In this condition, the system receives structured information about risks and mitigation strategies, but no explicit relationships between them.
- **Context 3 – No Context.** In this condition, the system receives no additional domain-specific knowledge about legacy system modernization risks or mitigation strategies.
- **Context 4 – Unstructured Context.** In this condition, the system receives the modernization risk knowledge as an unstructured plain-text file rather than in a structured format.

For each context condition, the experts were asked the same three questionnaire items:

1. Which tasks did you complete under this context condition?
2. Briefly describe the tasks you completed. *Optional. A short description is sufficient.*

3. Please rate the generated answers from 0% to 100%. *Only rate the criteria corresponding to the tasks you completed. Leave all other fields empty.*

The six rating criteria supplied to the experts were:

- **Rating 1 – Accuracy.** Is the information correct, complete, properly assigned, and limited to documented information?
- **Rating 2 – Direct Relationship Reasoning.** Are both risk–mitigation relationships described completely and correctly without adding undocumented relationships?
- **Rating 3 – Indirect Relationship Reasoning.** Is the risk chain explained logically and correctly without unsupported intermediate steps?
- **Rating 4 – Scenario Mapping.** Are the relevant risks, mitigation strategies, and relationships identified and connected correctly?
- **Rating 5 – Hallucination Resistance.** Does the answer identify missing or false information and avoid inventing probabilities, guarantees, values, or relationships?
- **Rating 6 – JSON Validity.** Is the output valid, machine-readable JSON that correctly represents all available and undocumented information?

The rating scale used five reference points while allowing intermediate values:

- 0% – Completely insufficient: Incorrect, unusable, or largely invented.
- 25% – Mostly insufficient: Major errors, omissions, or unsupported claims.
- 50% – Partially sufficient: Partly correct but with noticeable weaknesses.
- 75% – Mostly sufficient: Largely correct with only minor weaknesses.
- 100% – Fully sufficient: Correct, complete, supported, and clearly reasoned.

After the four context conditions, the questionnaire stated: *Please provide a rating between 0% and 100%. These questions are optional.* The three general rating questions were:

- Overall quality of the program
- Usability and ease of interaction
- Usefulness for analysing risks in legacy system modernization

The optional free-text questions were:

- In which situations or professional activities could you imagine using the program?
- What were the main strengths of the program?
- What should be improved?
- Do you have any additional comments or recommendations?

D.5. Detailed Expert Ratings

The following tables contain the individual anonymized criterion ratings submitted by the six experts. In accordance with the analysis in Chapter 8, all non-empty ratings are retained and the task-selection checkboxes are not used as an additional filter. Empty survey fields are preserved as missing and shown as "–". This results in 107 submitted context ratings in total: 27 for Full Structured Context, 27 for Structured Context without Relationships, 27 for No Context, and 26 for Unstructured Context.

Table D.4.: Individual expert ratings for Full Structured Context.

Expert	Accuracy	Direct Rel.	Indirect Rel.	Scenario	Hallucination	JSON
E1	100%	100%	–	–	–	–
E2	75%	75%	100%	75%	100%	–
E3	90%	95%	90%	85%	100%	–
E4	85%	85%	90%	90%	95%	–
E5	95%	95%	85%	–	100%	100%
E6	90%	80%	95%	85%	100%	–

Table D.5.: Individual expert ratings for Structured Context without Relationships.

Expert	Accuracy	Direct Rel.	Indirect Rel.	Scenario	Hallucination	JSON
E1	100%	90%	–	–	–	–
E2	75%	75%	100%	75%	100%	–
E3	85%	75%	65%	80%	95%	–
E4	80%	65%	10%	75%	90%	–
E5	90%	75%	60%	–	80%	100%
E6	80%	60%	70%	75%	95%	–

Table D.6.: Individual expert ratings for No Context.

Expert	Accuracy	Direct Rel.	Indirect Rel.	Scenario	Hallucination	JSON
E1	0%	0%	–	–	–	–
E2	0%	0%	0%	0%	100%	–
E3	10%	0%	0%	15%	15%	–
E4	5%	0%	0%	20%	100%	–
E5	0%	0%	0%	–	100%	95%
E6	0%	0%	0%	10%	100%	–

For completeness, the optional general program ratings submitted by the same experts are shown in Table D.8. These ratings are independent of the context-specific task ratings above.

Table D.7.: Individual expert ratings for Unstructured Context.

Expert	Accuracy	Direct Rel.	Indirect Rel.	Scenario	Hallucination	JSON
E1	100%	90%	–	–	–	–
E2	50%	50%	75%	50%	50%	–
E3	80%	75%	70%	75%	85%	–
E4	70%	60%	–	65%	80%	–
E5	90%	85%	75%	–	90%	96%
E6	70%	55%	65%	60%	75%	–

Table D.8.: Individual optional general program ratings.

Expert	Overall Quality	Usability	Usefulness
E1	85%	90%	80%
E2	75%	75%	75%
E3	88%	82%	90%
E4	80%	86%	84%
E5	85%	90%	90%
E6	80%	76%	82%

E. Additional Knowledge Graph Results

This appendix provides full-scale visualizations of the two Relationship datasets discussed in Chapter 8 and summarized visually in Figure 8.1. For each dataset, an unclustered and a clustered graph representation is provided. The High-Confidence graph represents the conservative Relationship dataset, whereas the extended graph additionally includes the Medium-Confidence Relationships.

E.1. High-Confidence Relationship Graph

The following visualizations show the conservative Relationship dataset containing only High-Confidence Relationships. Both an unclustered representation and the corresponding clustered representation are provided to show the overall graph structure from two different perspectives.

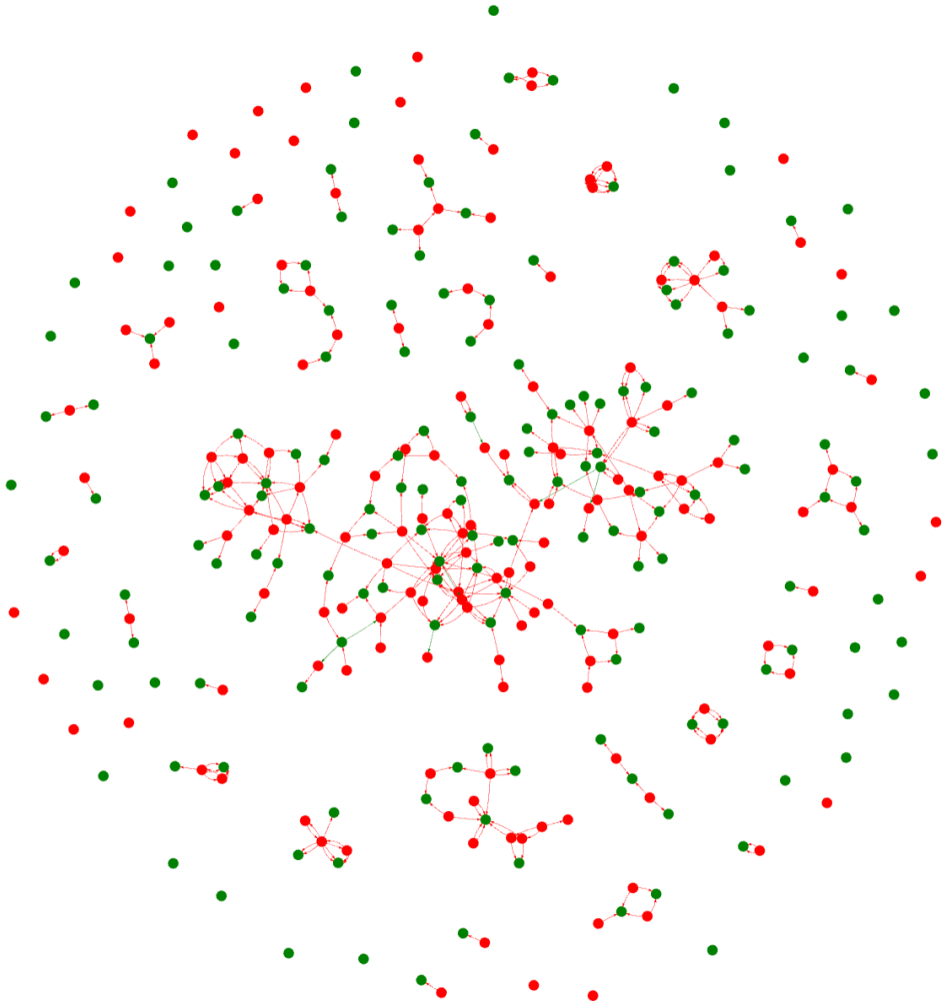


Figure E.1.: Knowledge graph of the conservative High-Confidence Relationship dataset without clustering.

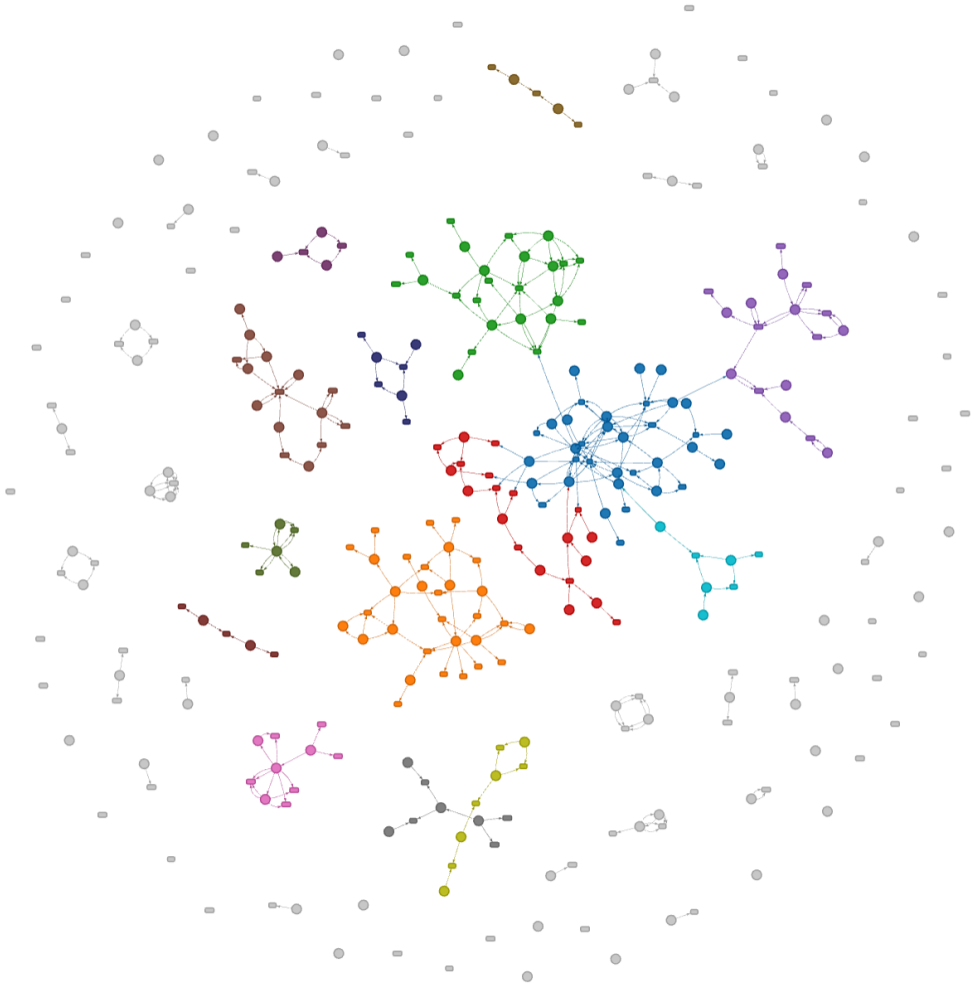


Figure E.2.: Clustered knowledge graph of the conservative High-Confidence Relationship dataset.

E.2. Extended High- and Medium-Confidence Relationship Graph

The extended Relationship dataset additionally contains Relationships classified with Medium Confidence. This increases the number of represented Relationships and consequently changes the connectivity and visual density of the resulting knowledge graph. Both the unclustered and clustered representations are shown below.

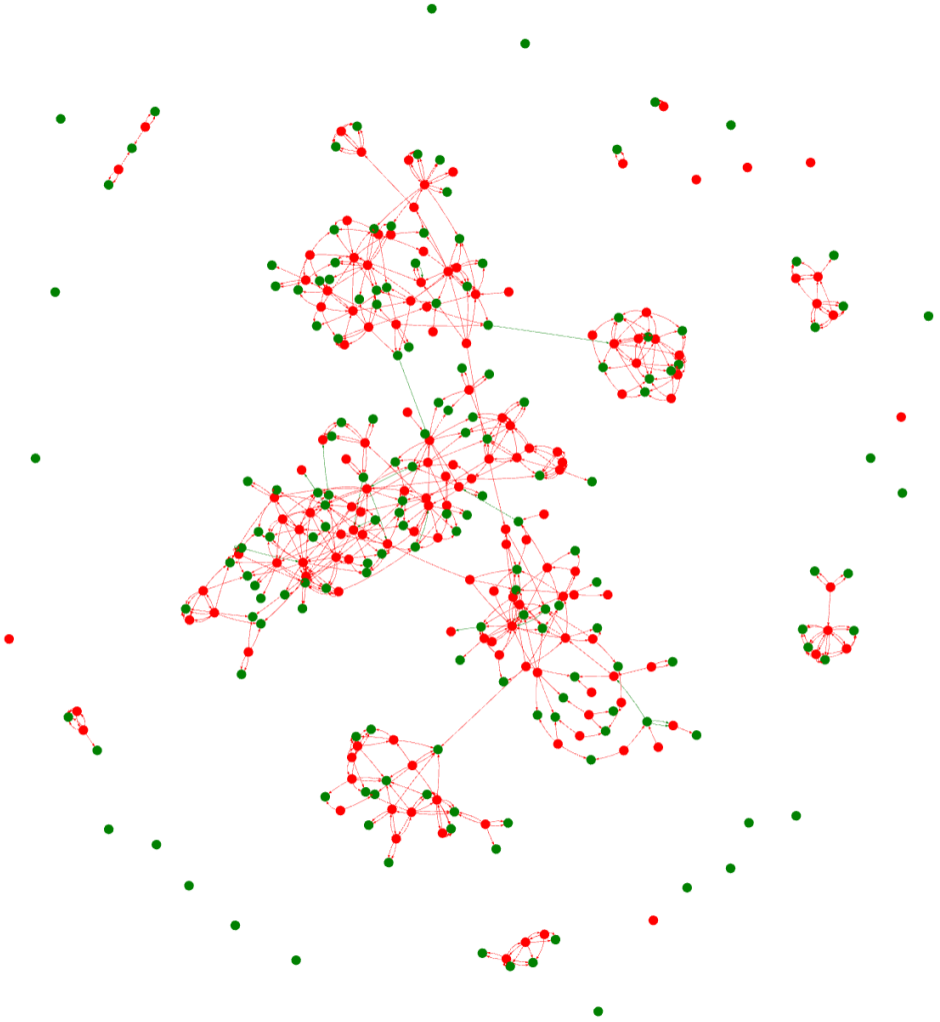


Figure E.3.: Knowledge graph of the extended High- and Medium-Confidence Relationship dataset without clustering.

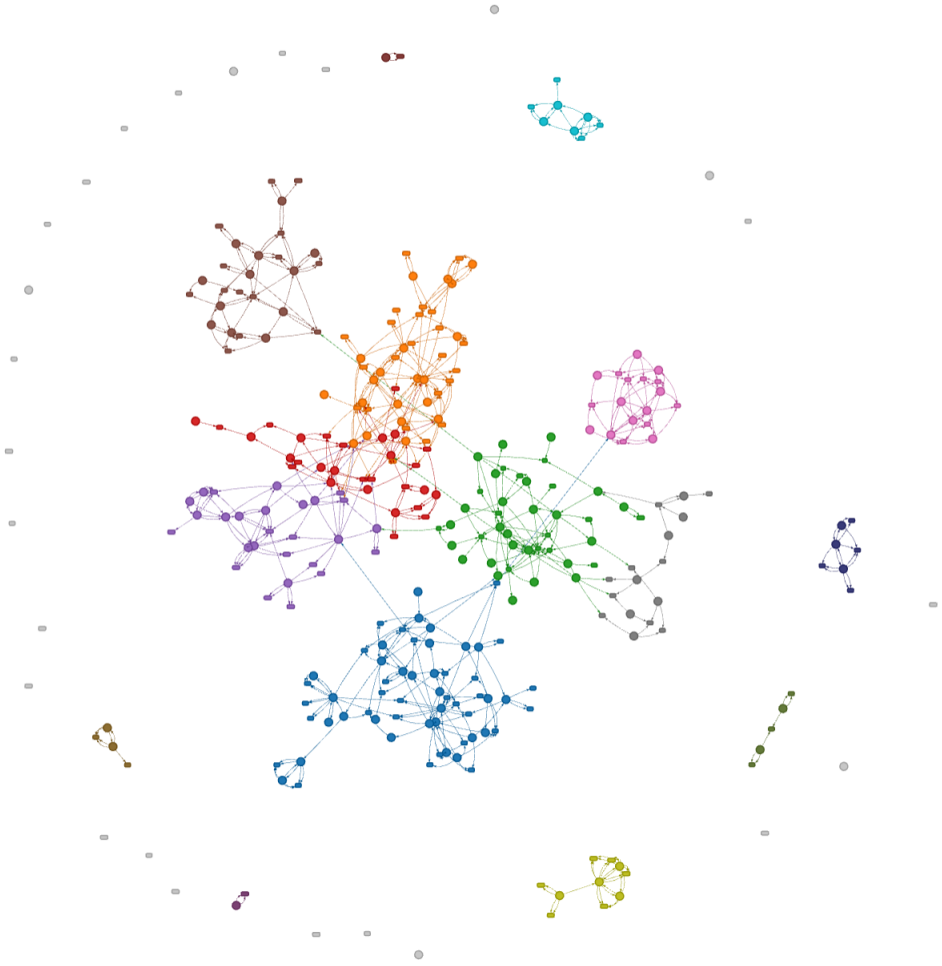


Figure E.4.: Clustered knowledge graph of the extended High- and Medium-Confidence Relationship dataset.

F. Repository and Prototype Usage

This appendix provides the practical information required to access and run the prototype described in Chapter 7. The complete implementation and the corresponding knowledge base are maintained in the following GitHub repository:

`https://github.com/Chrosmyc/Bachelor-thesis`

In addition to the source code, a deployed version of the Streamlit interface is available at:

`https://legacy-risk-evaluation.streamlit.app/`

The deployed evaluation version is protected by the password `LegacyEvaluation2026`.

F.1. Repository Structure

The repository contains the complete prototype together with the structured knowledge base. The main parts are separated according to their responsibilities. The `data` directory contains the Risk, Mitigation, and Relationship knowledge as well as the original raw text files. The `converter` directory contains the scripts used to convert the originally extracted data into the OKF-based Markdown representation. The `models` and `parser` directories contain the Python data classes and parsing logic used to load this representation. LLM communication is implemented in `llm`, while graph-related functionality is located in `graph`. The `evaluation` directory contains the evaluation questions, evaluation logic, ID extraction functionality, and stored evaluation outputs.

For normal local use, the central entry point of the complete framework is `main.py`. The separate modules do not have to be started individually. The main method combines context selection, knowledge loading, graph generation, model selection, and the interactive question-answering workflow. The repository additionally contains `streamlit_app.py`, which provides the web interface used during the expert evaluation.

F.2. Installation

A local installation requires Python and the dependencies listed in `requirements.txt`. After cloning the repository, the project directory can be opened with:

```
git clone https://github.com/Chrosmyc/Bachelor-thesis.git
cd Bachelor-thesis
```

The required packages are then installed once using pip:

```
pip install -r requirements.txt
```

The requirements file contains the dependencies used by the final prototype, including Streamlit, Requests, python-dotenv, PyYAML, PyVis, and NetworkX. A virtual environment can optionally be used, but is not required by the framework itself.

F.3. Configuration

For local execution, an API key for the KI:connect service is required. The implementation expects this key in an environment file located at:

```
llm/.env
```

This file has to contain the following variable:

```
KICONNECT_API_KEY=<your-api-key>
```

The actual API key is deliberately not part of the Git repository. Environment files are excluded through the repository's `.gitignore` configuration so that credentials are not committed together with the source code. The file therefore has to be created separately when the framework is executed on another machine.

The available language models are retrieved dynamically from the KI:connect API. This means that the model list is not hard-coded into the application and may change depending on the models currently provided by the external service.

For the deployed Streamlit version, no local environment file or user-provided API key is required. The deployed application uses the API credentials configured for the hosted instance through Streamlit secrets. Users of the deployed version therefore only need the application URL and evaluation password given above.

F.4. Running the Prototype

After the dependencies have been installed and the `llm/.env` file has been created, the complete local framework can be started directly from the repository root with:

```
python main.py
```

No separate parser, graph, or LLM script has to be started for the normal interactive workflow. The main method first asks the user to select one of the four context configurations. It then loads the corresponding knowledge representation and generates the Knowledge Graph for the selected structured context. Afterwards, the currently available models are requested from KI:connect and the user selects the model to use. The program then enters an interactive loop in which modernization questions can be entered until the program is closed with `exit`.

The four available context configurations are:

- Full Structured Context,
- Structured Context without Relationships,
- No Additional Context, and
- Unstructured Context.

These configurations correspond to the representations compared in the evaluation in Chapter 8. For the unstructured configuration, the raw Risk, Mitigation, and High-Confidence Relationship text files are loaded directly. For the structured configurations, the OKF-based Markdown files are parsed into the corresponding Python objects before prompt construction.

The Streamlit interface can also be started locally if the web-based representation is required:

```
streamlit run streamlit_app.py
```

However, this is not necessary for the basic local execution of the framework because the complete interactive workflow is already accessible through `main.py`. For normal use of the web interface, the deployed version linked at the beginning of this appendix can be used instead.

F.5. Deployed Prototype

The deployed prototype is available at:

`https://legacy-risk-evaluation.streamlit.app/`

with the evaluation password:

`LegacyEvaluation2026`

This version was used to provide the web-based interaction required during the external expert evaluation. It offers model and context selection, predefined evaluation tasks, free-form questions, and inspection of Risk, Mitigation, and Relationship IDs contained in the generated response. Since the API credentials are configured on the deployed instance, users do not need to provide their own KI:connect key when accessing this version.

F.6. Reproducing Graphs and Evaluation

The repository also contains the supporting functionality used for graph generation and the developer-conducted evaluation. Graph creation is implemented in the `graph` directory. The clustered representation reads the cluster information already stored in the Risk and Mitigation entries, while `update_clusters.py` can be used to recalculate the cluster assignments:

```
python update_clusters.py
```

The clustering procedure constructs an undirected NetworkX graph from the available entities and Relationships and applies Louvain community detection with a fixed random seed. The resulting cluster identifiers are written back into the corresponding Risk and Mitigation Markdown files. The graph visualization itself is generated with PyVis. Existing interactive HTML exports for the High-Confidence and extended graph variants are also contained in the `graph` directory.

The `evaluation` directory contains the predefined evaluation questions and the evaluation functionality used for the comparative proof of concept. The recorded outputs for the four context configurations are stored in `evaluation/output`. The default interactive execution through `main.py` does not automatically start the complete batch evaluation because the evaluation call is kept separate from the normal question-answer workflow.

When reproducing the reported results, the selected Relationship dataset must be treated as part of the experimental configuration. The comparative LLM evaluation described in Chapter 8 uses the conservative High-Confidence Relationship dataset. The extended graph variant additionally includes Medium-Confidence Relationships and was used for the separate graph comparison. Furthermore, because the LLMs are provided by the external KI:connect service, exact reproduction of generated natural-language responses cannot be guaranteed solely from the versioned source code.

Bibliography

[22] *Risk Management — Vocabulary*. Geneva, Switzerland: International Organization for Standardization, Feb. 2022. URL: https://www.iso.org/standard/79637.html (visited on 08/19/2026) (cit. on pp. 13, 14, 16).

[AAC16] E. Agilar, R. Almeida, and E. Canedo. "A Systematic Mapping Study on Legacy System Modernization." In: *Proceedings of the 28th International Conference on Software Engineering and Knowledge Engineering*. Knowledge Systems Institute, 2016, pp. 345–350. DOI: 10.18293/seke2016-059 (cit. on pp. 9, 28, 47).

[acq] acquisa. *Risikomanagement: Definition und leicht verständlicher Leitfaden zur Umsetzung*. acquisa. URL: https://www.acquisa.de/magazin/risikomanagement (visited on 07/21/2026) (cit. on pp. 15, 16).

[AG26] S. N. Ahmed and M. Galib. *AgentModernize: Preserving Business Logic in Legacy Modernization with Multi-Agent LLMs and Behavioral Specification Graphs*. May 17, 2026. arXiv: 2605.17535. URL: https://arxiv.org/abs/2605.17535 (visited on 07/21/2026) (cit. on pp. 3, 27, 28).

[AI23] Amazon Web Services and IBM. *The Next Step for Your Legacy Workloads Is AWS*. Industry eBook. Amazon Web Services and IBM, Aug. 2023. URL: https://pages.awscloud.com/rs/112-TZM-766/images/PTNR_IBM_Modernize_Workloads_eBook_Aug-23.pdf (visited on 07/21/2026) (cit. on p. 10).

[Ame17] American Council for Technology and Industry Advisory Council. *Legacy System Modernization: Addressing Challenges on the Path to Success*. Industry Report. Version 1.1. ACT-IAC, Jan. 12, 2017. URL: https://www.actiac.org/system/files/Legacy%20System%20Modernization%20Report%20v1.1%2001112017_0.pdf (visited on 07/21/2026) (cit. on pp. 13, 15, 22, 47, 50–52, 54, 57, 59).

[Ass+24] W. K. G. Assunção et al. *Contemporary Software Modernization: Perspectives and Challenges to Deal with Legacy Systems*. July 4, 2024. arXiv: 2407.04017. URL: https://arxiv.org/abs/2407.04017 (visited on 07/21/2026) (cit. on pp. 7, 14, 26, 55).

[Ass+25] W. K. G. Assunção et al. "Contemporary Software Modernization: Strategies, Driving Forces, and Research Opportunities." In: *ACM Transactions on Software Engineering and Methodology* 34.5 (2025), pp. 1–35. ISSN: 1049-331X. DOI: 10.1145/3708527 (cit. on pp. 2, 9–15, 22, 26, 50, 51, 53, 55).

[Bal+08] S. Balasubramaniam et al. "SMART: Application of a Method for Migration of Legacy Systems to SOA Environments." In: *Service-Oriented Computing—ICSOC 2008*. Vol. 5364. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 678–690. ISBN: 978-3-540-89652-4. DOI: 10.1007/978-3-540-89652-4_60 (cit. on pp. 23, 52).

[Bat+14] B. V. Batlajery et al. *Industrial Perception of Legacy Software Systems and Their Modernization*. Technical Report UU-CS-2014-004. Utrecht, The Netherlands: Utrecht University, Department of Information and Computing Sciences, 2014. URL: https://dspace.library.uu.nl/handle/1874/306443 (visited on 07/21/2026) (cit. on pp. 2, 7–9, 14, 26, 28, 47, 49, 59).

[Bia25] H. Bian. *LLM-Empowered Knowledge Graph Construction: A Survey*. Oct. 23, 2025. arXiv: 2510.20345. URL: https://arxiv.org/abs/2510.20345 (visited on 07/21/2026) (cit. on pp. 17, 26).

[BR00] K. H. Bennett and V. T. Rajlich. "Software Maintenance and Evolution: A Roadmap." In: *Proceedings of the Conference on the Future of Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2000, pp. 73–87. DOI: 10.1145/336512.336534 (cit. on pp. 3, 14, 15, 21, 48).

[Bri] British Ecological Society. *How to Use Grey Literature*. British Ecological Society. URL: https://www.britishecologicalsociety.org/content/how-to-use-grey-literature/ (visited on 07/21/2026) (cit. on pp. 34, 35).

[CIOa] CIO. *Eight IT Modernization Traps CIOs Must Avoid*. CIO. URL: https://www.cio.com/article/4176051/8-it-modernization-traps-cios-must-avoid.html (visited on 07/21/2026) (cit. on pp. 11, 52, 54).

[CIOb] CIO. *Overcoming the Six Barriers to IT Modernization*. CIO. URL: https://www.cio.com/article/3613163/overcoming-the-6-barriers-to-it-modernization.html (visited on 07/21/2026) (cit. on pp. 14, 15, 22, 51, 56).

[Com+00] S. Comella-Dorda et al. *A Survey of Legacy System Modernization Approaches*. Technical Note CMU/SEI-2000-TN-003. Pittsburgh, PA, USA: Software Engineering Institute, Carnegie Mellon University, Apr. 1, 2000. DOI: 10.21236/ADA377453. URL: https://www.sei.cmu.edu/library/a-survey-of-legacy-system-modernization-approaches/ (visited on 07/21/2026) (cit. on pp. 7, 8, 10, 12, 14, 48, 49, 54).

[De 23] A. De Bellis. "Structuring the Unstructured: An LLM-Guided Transition." In: *Proceedings of the Doctoral Consortium at ISWC 2023*. Vol. 3678. CEUR Workshop Proceedings. Athens, Greece: CEUR-WS.org, Nov. 7, 2023. URL: https://ceur-ws.org/Vol-3678/paper12.pdf (visited on 07/21/2026) (cit. on pp. 3, 17, 18, 115).

[DNV] DNV. *What Is Business Risk Management?* DNV. URL: https://www.dnv.com/assurance/articles/what-is-business-risk-management/ (visited on 07/21/2026) (cit. on p. 16).

[FAA25] L. F. Fávero, N. R. de Almeida, and F. J. Affonso. "A Systematic Mapping Study on the Modernization of Legacy Systems to Microservice Architecture." In: *Applied System Innovation* 8.4, 86 (June 20, 2025). ISSN: 2571-5577. DOI: 10.3390/asi8040086 (cit. on pp. 3, 9, 10, 26, 28, 47).

[Fen+20] D. Fensel et al. "Introduction: What Is a Knowledge Graph?" In: *Knowledge Graphs*. Cham: Springer International Publishing, 2020, pp. 1–10. ISBN: 978-3-030-37439-6. DOI: 10.1007/978-3-030-37439-6_1 (cit. on p. 27).

[Fraa] Fraunhofer IESE. *Modernizing Legacy Systems*. Fraunhofer Institute for Experimental Software Engineering IESE. URL: https://www.iese.fraunhofer.de/en/services/legacy-systems.html (visited on 07/21/2026) (cit. on p. 10).

[Frab] Fraunhofer IESE. *Modernizing Systems: Requirements Analysis for Legacy Systems*. Fraunhofer Institute for Experimental Software Engineering IESE. URL: https://www.iese.fraunhofer.de/en/solution/modernizing-systems.html (visited on 07/21/2026) (cit. on pp. 22, 23, 55).

[Gao23] A. Gao. *Prompt Engineering for Large Language Models*. Preprint. SSRN. July 8, 2023. DOI: 10.2139/ssrn.4504303. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4504303 (visited on 07/21/2026).

[Gar] Gartner. *How Midsize Enterprise CIOs Fund Legacy Modernization*. Gartner. URL: https://www.gartner.com/en/doc/753128-how-midsize-enterprise-cios-fund-legacy-modernization (visited on 07/21/2026) (cit. on p. 52).

[Gho+17] M. F. Gholami et al. "Challenges in Migrating Legacy Software Systems to the Cloud—An Empirical Study." In: *Information Systems* 67 (2017), pp. 100–113. ISSN: 0306-4379. DOI: 10.1016/j.is.2017.03.008 (cit. on pp. 9, 14, 22, 26, 47, 49, 56–59, 106).

[Goo26] Google Cloud Platform. *Open Knowledge Format (OKF) v0.2 Specification*. Google Cloud Platform. 2026. URL: https://github.com/GoogleCloudPlatform/open-knowledge-format/blob/main/SPEC.md (visited on 09/02/2026) (cit. on p. 17).

[GS14] M. Galinium and N. Shahbaz. *Case Studies: Business and Technical Perspectives in Migration of Legacy Systems to Service Oriented Architecture*. Dec. 26, 2014. arXiv: 1412.7959. URL: https://arxiv.org/abs/1412.7959 (visited on 07/21/2026) (cit. on pp. 8, 11, 21, 22, 49, 51–59).

[Han+25] H. Han et al. "Reasoning with Graphs: Structuring Implicit Knowledge to Enhance LLMs' Reasoning." In: *Findings of the Association for Computational Linguistics: ACL 2025*. Vienna, Austria: Association for Computational Linguistics, 2025, pp. 25698–25714. DOI: 10.18653/v1/2025.findings-acl.1319. URL: https://aclanthology.org/2025.findings-acl.1319/ (visited on 07/21/2026) (cit. on pp. 27, 28, 108).

[Hog+22] A. Hogan et al. "Knowledge Graphs." In: *ACM Computing Surveys* 54.4 (2022), pp. 1–37. ISSN: 0360-0300. DOI: 10.1145/3447772 (cit. on pp. 17, 18, 26, 27).

[Hog+24] G. Hogan et al. "Investigating Systems Modernisation: Approaches, Challenges and Risks." In: *Systems, Software and Services Process Improvement*. Vol. 2179. Communications in Computer and Information Science. Cham: Springer Nature Switzerland, 2024, pp. 147–162. ISBN: 978-3-031-71139-8. DOI: 10.1007/978-3-031-71139-8_10 (cit. on pp. 2, 3, 5, 7, 10, 11, 26, 49, 50).

[How] L. Howell. *Guide to Searching Grey Literature*. Study Guide. University of Wolverhampton. URL: https://www.wlv.ac.uk/lib/media/departments/lis/skills/study-guides/LS124-Searching-Grey-Literature.pdf (visited on 07/21/2026) (cit. on pp. 34, 35).

[IBMa] IBM. *From Legacy Systems to Modern EDI Solutions: A Migration Guide*. IBM. URL: https://www.ibm.com/think/insights/edi-solutions (visited on 07/21/2026) (cit. on pp. 10, 55).

[IBMb] IBM. *Legacy Application Modernization: A Comprehensive Approach to Modernize Your Business*. IBM. URL: https://www.ibm.com/think/topics/legacy-application-modernization (visited on 07/21/2026) (cit. on pp. 7, 8, 49, 52).

[IBMc] IBM. *Was versteht man unter Risikomanagement?* IBM. URL: https://www.ibm.com/de-de/think/topics/risk-management (visited on 07/21/2026) (cit. on pp. 15, 16).

[IBMd] IBM. *What Is COBOL Modernization?* IBM. URL: https://www.ibm.com/think/topics/cobol-modernization (visited on 07/21/2026) (cit. on p. 10).

[IBMe] IBM. *What Is Mainframe Modernization?* IBM. URL: https://www.ibm.com/think/topics/mainframe-modernization (visited on 07/21/2026) (cit. on p. 3).

[Ins] Institute for Critical Infrastructure Technology. *Accelerating Legacy System Modernization in Government*. Institute for Critical Infrastructure Technology. URL: https://www.icitech.org/post/accelerating-legacy-system-modernization-in-government (visited on 07/21/2026) (cit. on pp. 10, 23).

[Int] International Organization for Standardization. *ISO 31000: Risk Management*. International Organization for Standardization. URL: https://www.iso.org/publication/PUB100426.html (visited on 07/21/2026) (cit. on pp. 13, 16).

[Ji+22] S. Ji et al. "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications." In: *IEEE Transactions on Neural Networks and Learning Systems* 33.2 (2022), pp. 494–514. ISSN: 2162-237X. DOI: 10.1109/TNNLS.2021.3070843 (cit. on pp. 17, 18, 26).

[Kar+20] V. Karpukhin et al. "Dense Passage Retrieval for Open-Domain Question Answering." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. URL: https://aclanthology.org/2020.emnlp-main.550/.

[Kha+13] R. Khadka et al. "Migrating a Large-Scale Legacy Application to SOA: Challenges and Lessons Learned." In: *2013 20th Working Conference on Reverse Engineering (WCRE)*. IEEE, 2013, pp. 425–432. DOI: 10.1109/WCRE.2013.6671318 (cit. on pp. 2, 14, 28, 47, 52, 58).

[Kha+14] R. Khadka et al. "How Do Professionals Perceive Legacy Systems and Software Modernization?" In: *Proceedings of the 36th International Conference on Software Engineering* (Hyderabad, India). New York, NY, USA: Association for Computing Machinery, May 2014, pp. 36–47. ISBN: 978-1-4503-2756-5. DOI: 10.1145/2568225.2568318 (cit. on pp. 1, 2, 7, 8, 10, 11, 13, 21, 47–51, 54, 57, 59).

[Kha+15] R. Khadka et al. "Does Software Modernization Deliver What It Aimed for? A Post-Modernization Analysis of Five Software Modernization Case Studies." In: *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2015, pp. 477–486. DOI: 10.1109/ICSM.2015.7332499 (cit. on pp. 4, 7, 21, 47, 50, 52, 54, 58, 59, 106).

[Kha+21] M. Khan et al. "CMMI-Compliant Modernization Framework to Transform Legacy Systems." In: *Intelligent Automation & Soft Computing* 27.2 (2021), pp. 311–331. ISSN: 1079-8587. DOI: 10.32604/iasc.2021.014280 (cit. on pp. 7, 23, 25).

[Kin] King's College London Libraries and Collections. *Searching for Systematic Reviews and Evidence Synthesis: Grey Literature*. King's College London. URL: https://libguides.kcl.ac.uk/systematicreview/greylit (visited on 07/21/2026) (cit. on p. 34).

[KPM] KPMG. *Technology Risk*. KPMG. URL: https://kpmg.com/xx/en/what-we-do/services/advisory/risk-consulting/technology-risk.html (visited on 07/21/2026).

[Lan] E. Langbauer. "Factors Influencing the Legacy System Modernization Strategy." Johannes Kepler University Linz. URL: https://epub.jku.at/obvulihs/content/titleinfo/4587135 (visited on 07/21/2026) (cit. on pp. 4, 7–12, 15, 16, 22, 23, 51, 52, 55, 56).

[Lew+20] P. Lewis et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html (cit. on p. 19).

[Lin25] LinkedIn. *How Much Risk Do Companies Incur by Avoiding IT Modernization?* LinkedIn. June 26, 2025. URL: https://www.linkedin.com/pulse/how-much-risk-do-companies-incur-avoiding-modernization-bhkqe (visited on 07/21/2026) (cit. on pp. 3, 9, 10, 52).

[MH26] S. McVeety and A. Hormati. *Introducing the Open Knowledge Format*. Google Cloud. June 12, 2026. URL: https://cloud.google.com/blog/products/data-analytics/how-the-open-knowledge-format-can-improve-data-sharing (visited on 09/02/2026) (cit. on p. 17).

[Mica] Microsoft. *Assess Your Workloads for Cloud Migration*. Microsoft Learn. URL: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/plan/assess-workloads-for-cloud-migration (visited on 07/21/2026) (cit. on pp. 12, 23).

[Micb] Microsoft. *Select Your Cloud Migration Strategies*. Microsoft Learn. URL: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/plan/select-cloud-migration-strategy (visited on 07/21/2026) (cit. on p. 11).

[Micc] Microsoft. *The Six Rs of Application Modernization*. Microsoft Learn. URL: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/plan/select-cloud-migration-strategy (visited on 07/21/2026) (cit. on p. 11).

[MIT24] MITRE. *Recommendations to Modernize Archaic and Insecure Legacy Systems*. Technical Report PR-24-01820. MITRE, Sept. 2024. URL: https://www.mitre.org/sites/default/files/2024-09/PR-24-01820-recommendations-modernize-archaic-insecure-legacy-systems.pdf (visited on 07/21/2026) (cit. on pp. 4, 49).

[Mon] Monash University Library. *Grey Literature: Finding Grey Literature*. Monash University. URL: https://guides.lib.monash.edu/grey-literature/findinggreyliterature (visited on 07/21/2026) (cit. on p. 34).

[Ogu+23] O. Ogunwole et al. "Modernizing Legacy Systems: A Scalable Approach to Next-Generation Data Architectures and Seamless Integration." In: *International Journal of Multidisciplinary Research and Growth Evaluation* 4.1 (2023), pp. 901–909. ISSN: 2582-7138. DOI: 10.54660/.ijmrge.2023.4.1.901-909 (cit. on pp. 9, 10, 22, 55–57).

[Ora] Oracle. *Modernization Patterns*. Oracle Cloud Foundation. URL: https://docs.oracle.com/en/cloud/foundation/cloud_strategy/modernizationpatterns/index.html (visited on 07/21/2026).

[Ora24] Oracle. *Strategic Modernization with Oracle IAM*. Oracle. Oct. 8, 2024. URL: https://blogs.oracle.com/cloud-infrastructure/modernization-with-oracle-iam (visited on 07/21/2026).

[OXD] OXD. *Low-Risk Strategies for Legacy System Modernization in the Public Sector*. OXD. URL: https://oxd.com/insights/low-risk-strategies-for-legacy-system-modernization-in-the-public-sector/ (visited on 07/21/2026) (cit. on p. 28).

[Paz25] A. Pazova. "Factors Involved in Modernization Decision Making." In: *Softwaretechnik-Trends* 45.2 (2025), pp. 31–32. ISSN: 0720-8928. URL: https://gi.de/fileadmin/fg/sre/wsre2025/wsre2025_proceedings.pdf#page=16 (visited on 07/21/2026) (cit. on pp. 15, 21, 48, 52, 53, 57, 59, 105).

[PE23] S. Ponnusamy and D. Eswararaj. "Navigating the Modernization of Legacy Applications and Data: Effective Strategies and Best Practices." In: *Asian Journal of Research in Computer Science* 16.4 (2023), pp. 239–256. ISSN: 2581-8260. DOI: 10.9734/ajrcos/2023/v16i4386 (cit. on pp. 8–10, 14, 21, 22, 48, 49, 51, 53–57, 59, 106).

[Pow04] M. Power. "The Risk Management of Everything." In: *The Journal of Risk Finance* 5.3 (2004), pp. 58–65. ISSN: 1526-5943. DOI: 10.1108/eb023001 (cit. on p. 16).

[Pun26] H. Punihani. *EMRGF: A Practitioner Framework for Governance-Driven Enterprise Technology Modernization*. May 2026. arXiv: 2605.06703. URL: https://arxiv.org/abs/2605.06703 (visited on 07/21/2026) (cit. on pp. 47, 51, 53, 54).

[Rah26] A. Rah. *Why Digital Transformation Fails: Human Resistance, Legacy Systems, and IT Modernization Challenges*. Preprint. 2026. DOI: 10.13140/RG.2.2.36366.11848. URL: https://www.researchgate.net/publication/404348819_Why_Digital_Transformation_Fails_Human_Resistance_Legacy_Systems_and_IT_Modernization_Challenges (visited on 07/21/2026) (cit. on pp. 2, 50, 54).

[Red24] Red Hat. *The State of Application Modernization*. Industry Report. Red Hat, 2024. URL: https://www.redhat.com/en/resources/app-modernization-report (visited on 07/21/2026) (cit. on pp. 3, 13–15, 22, 28, 47, 48, 51–53, 57–59, 105).

[Rii23] S. Riihimäki. "Migration Planning Framework for Legacy Systems' Cloud Migration." Master's thesis. Turku, Finland: University of Turku, May 27, 2023. URL: https://urn.fi/URN:NBN:fi-fe2023060151601 (visited on 07/21/2026) (cit. on pp. 22–24, 26, 28, 52, 56).

[RS16] S. Rochimah and A. Sheku. "Migration of Existing or Legacy Software Systems into Web Service-Based Architectures (Reengineering Process): A Systematic Literature Review." In: *International Journal of Computer Applications* 133.3 (2016), pp. 43–54. ISSN: 0975-8887. DOI: 10.5120/ijca2016907801 (cit. on p. 11).

[SAPa] SAP Community. *Best Practices and Approaches to Application Modernization*. SAP. URL: https://community.sap.com/t5/technology-blog-posts-by-members/best-practices-and-approaches-to-application-modernization/ba-p/13320684 (visited on 07/21/2026).

[SAPb] SAP Community. *Reimagine, Revamp, and Reinvent: A Dynamic Legacy Modernization Approach in Ten Steps*. SAP. URL: https://community.sap.com/t5/enterprise-architecture-blog-posts/reimagine-revamp-and-reinvent-a-dynamic-legacy-modernization-approach-in-10/ba-p/13554847 (visited on 07/21/2026) (cit. on pp. 9, 10).

[SAP25] SAP Community. *Why Delaying System Modernization Is Your Biggest Business Risk in 2025*. SAP. 2025. URL: https://community.sap.com/t5/technology-blog-posts-by-members/why-delaying-system-modernization-is-your-biggest-business-risk-in-2025/ba-p/13966916 (visited on 07/21/2026) (cit. on pp. 2, 3, 9).

[Sch+24] S. Schulhoff et al. *The Prompt Report: A Systematic Survey of Prompting Techniques*. June 6, 2024. arXiv: 2406.06608. URL: https://arxiv.org/abs/2406.06608 (visited on 07/21/2026) (cit. on pp. 19, 27).

[Sea+01] R. C. Seacord et al. *Legacy System Modernization Strategies*. Technical Report CMU/SEI-2001-TR-025. Pittsburgh, PA, USA: Software Engineering Institute, Carnegie Mellon University, July 1, 2001. URL: https://www.sei.cmu.edu/library/legacy-system-modernization-strategies/ (visited on 07/21/2026) (cit. on pp. 12, 21–23, 50, 57, 58, 106).

[Sim] Simon Fraser University Library. *Grey Literature: What It Is and How to Find It*. Simon Fraser University. URL: https://www.lib.sfu.ca/help/research-assistance/format-type/grey-literature (visited on 07/21/2026) (cit. on pp. 34, 35, 49).

[Sof] Software Improvement Group. *Legacy Modernization: Six Architecture Risks That Could Derail Your Program*. Software Improvement Group. URL: https://www.softwareimprovementgroup.com/blog/legacy-modernization-6-architecture-risks-that-could-derail-your-program/ (visited on 07/21/2026) (cit. on pp. 5, 8, 14, 28, 47–49, 52, 59).

[Ste] D. Stevens. *Legacy Modernization Guide*. Guide. Texas Department of Information Resources. URL: https://dir.texas.gov/sites/default/files/Legacy%20Modernization%20Guide.pdf (visited on 07/21/2026) (cit. on pp. 15, 23, 26, 47, 51–53, 59).

[Thoa] Thoughtworks. *Embracing the Strangler Fig Pattern for Legacy Modernization: Part One*. Thoughtworks. URL: https://www.thoughtworks.com/en-de/insights/articles/embracing-strangler-fig-pattern-legacy-modernization-part-one (visited on 07/21/2026) (cit. on pp. 57, 58).

[Thob] Thoughtworks. *Embracing the Strangler Fig Pattern for Legacy Modernization: Part Two—Overcoming People and Process Challenges*. Thoughtworks. URL: https://www.thoughtworks.com/en-de/insights/articles/embracing-strangler-fig-pattern-legacy-modernization-part-two (visited on 07/21/2026).

[US 21] U.S. Government Accountability Office. *Information Technology: Agencies Need to Develop and Implement Modernization Plans for Critical Legacy Systems*. Testimony GAO-21-524T. U.S. Government Accountability Office, Apr. 27, 2021. URL: https://www.gao.gov/products/gao-21-524t (visited on 07/21/2026) (cit. on pp. 2, 3, 9, 11, 15, 23, 47, 51, 52, 59).

[Van25] N. R. Vanaparthi. "The Roadmap to Mainframe Modernization: Bridging Legacy Systems with the Cloud." In: *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 11.1 (2025), pp. 125–133. ISSN: 2456-3307. DOI: 10.32628/CSEIT25111214 (cit. on pp. 22, 53, 56, 57).

[Whi+23] J. White et al. "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT." In: *Proceedings of the 30th Conference on Pattern Languages of Programs*. Allerton Park, Monticello, IL, USA: The Hillside Group, 2023. DOI: 10.64346/PLoP2023p05 (cit. on pp. 17, 18).

[Wol+20] D. Wolfart et al. "Towards a Process for Migrating Legacy Systems into Microservice Architectural Style." In: *Anais da IV Escola Regional de Engenharia de Software (ERES 2020)*. Sociedade Brasileira de Computação, 2020, pp. 255–264. DOI: 10.5753/eres.2020.13736. URL: https://sol.sbc.org.br/index.php/eres/article/view/13736 (visited on 07/21/2026).

[Wol+21] D. Wolfart et al. "Modernizing Legacy Systems with Microservices: A Roadmap." In: *Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineering*. New York, NY, USA:

Association for Computing Machinery, 2021, pp. 149–159. DOI: 10.1145/3463274.3463334 (cit. on p. 26).

[WT24] X. Wu and K. Tsioutsiouliklis. *Thinking with Knowledge Graphs: Enhancing LLM Reasoning Through Structured Data*. Dec. 14, 2024. arXiv: 2412.10654. URL: https://arxiv.org/abs/2412.10654 (visited on 07/21/2026) (cit. on pp. 4, 17–19, 27, 28, 108, 115).

[Zha+25] J. Zhang et al. "SR-LLM: Rethinking the Structured Representation in Large Language Model." In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vienna, Austria: Association for Computational Linguistics, July 2025, pp. 3443–3462. DOI: 10.18653/v1/2025.acl-long.172. URL: https://aclanthology.org/2025.acl-long.172/ (visited on 07/21/2026) (cit. on pp. 19, 27, 28, 108).