Automation of Requirement Governance in Railway Communication Systems

AI-assisted Workflow for System Migration from GSM-R to FRMCS

Kristi Kapllani

Master thesis submitted to Delft University of Technology
in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE
Management of Technology

Faculty of Technology, Policy and Management
Delft University of Technology

Automation of Requirement Governance in Railway Communication Systems: AI-assisted Workflow for System Migration from GSM-R to FRMCS

Student name: Kristi Kapllani

Student number: 4539311

Graduation committee

Chair & First supervisor: Dr. Aaron Ding, Senior Associate Professor

Second supervisor: Dr. Igor Nikolic, Associate Professor

Acknowledgments

I would like to express my sincere gratitude to my first supervisor, Dr. Aaron Ding, for his guidance, patience, and continued support throughout the past seven months. Our weekly meetings played an important role in shaping this thesis. I especially appreciated his thoughtful feedback, not only on the larger research direction but also on the smaller details that often make the difference between good work and strong work. Whenever I felt lost, uncertain, or focused on the wrong things, he helped guide me back in the right direction with clarity and encouragement.

I would also like to thank Dr. Igor Nikolic for his direct and honest feedback during this process. His critical perspective challenged me to think more deeply, argue more carefully, and approach my research with greater academic rigour. I am grateful for the way he pushed me to strengthen both the quality of the thesis and the way I reflected on my work.

My sincere thanks also go to the Nokia employees, as well as those involved in arranging meetings and presentations throughout this process. Their time, openness, and practical insights gave me a much clearer understanding of how these issues are handled in industry. Their feedback helped me notice aspects of my research that I had initially overlooked and gave me the opportunity to validate my work against real professional experience.

I would also like to thank my employer for giving me the opportunity and flexibility to pursue this degree alongside my professional responsibilities. That support made it possible for me to continue my education while managing the demands of work.

Finally, I would like to thank my family for their constant love and support throughout this journey. They taught me the value of perseverance and reminded me not to give up, especially in the moments when things felt most difficult. In the times I felt weakest, they encouraged me to keep going and to push harder. Whether near or far, they stood by me through every stage of this process, and I am deeply grateful for their belief in me.

Statement on the Use of Generative AI

Generative artificial intelligence was used in two distinct ways during this thesis: as part of the research artefact itself and as a supporting tool during the preparation of the thesis.

First, generative AI formed an explicit component of the research artefact investigated in this study. GPT-4o was integrated into the three-agent n8n prototype to support requirement extraction and normalisation, comparison and clarification, and the generation of governance-oriented outputs. The design, configuration, use, and limitations of these AI-supported workflow stages are described in the methodology, artefact design, demonstration, evaluation, and appendices of this thesis. The outputs generated by the prototype were treated as research material to be evaluated rather than as inherently correct results, and human review remained part of the workflow design.

Second, ChatGPT and Claude (Anthropic) were used as supporting tools during the writing and development of the thesis. Their use included discussing and refining the structure of chapters, reviewing the logical consistency between research questions, design requirements, evaluation results, and conclusions, suggesting alternative wording for passages written by the author, assisting with language clarity and grammar, checking the internal consistency of the final manuscript, and troubleshooting LaTeX formatting.

Representative uses included requests to assess whether a chapter logically answered the relevant research question, to identify inconsistencies between prototype outputs and claims made elsewhere in the thesis, to perform final consistency and sanity checks on the complete manuscript, to improve the wording of author-written passages, and to help diagnose LaTeX formatting problems.

Generative AI was not used to generate interview data, fabricate empirical observations, or independently determine the findings of the study. Decisions regarding the research problem, methodology, artefact design, dataset selection, interpretation of practitioner input, evaluation of prototype behaviour, and final conclusions remained the responsibility of the author. Where AI-assisted discussion concerned academic literature or empirical evidence, the relevant claims and references were checked against the underlying sources and thesis material before inclusion.

The author takes full responsibility for the content, interpretation, analysis, and conclusions presented in this thesis.

Executive Summary

Modern railway systems depend on communication technologies that connect trains, infrastructure, operators, and control centres. When these systems are replaced or upgraded, organisations must interpret and coordinate large numbers of technical requirements originating from different documents and stakeholders. Much of this work still depends on engineers manually searching specifications, comparing information, resolving ambiguities, and determining responsibility for resulting decisions. This process is time-consuming and difficult to perform consistently, particularly when requirements change over time.

This thesis investigates how an AI-assisted workflow can support this requirements-governance process while keeping human experts responsible for consequential decisions. The study uses the European railway sector as its practical setting, specifically the transition from the existing GSM-R communication system to its next-generation successor, FRMCS. This transition illustrates a broader organisational challenge: information must be transformed from heterogeneous, document-based sources into structured requirements that can be reviewed, compared, assigned, and used in engineering and project decisions.

A three-stage workflow was designed to support this process. The first stage structures requirement information while preserving its original source. The second compares requirements and highlights areas requiring clarification or technical judgement. The third prepares information about responsibility, priority, and follow-up actions. Human review is placed between these stages so that AI-generated interpretations can be corrected, technically assessed, and authorised before influencing later decisions. The purpose is therefore not to automate requirements governance completely, but to reduce repetitive information-processing work and allow specialists to focus on tasks requiring expertise and organisational authority.

The prototype was demonstrated using official railway specification documents and evaluated against the design requirements developed in the study. The results show that the workflow can transform document-based requirement information into structured and traceable outputs while supporting comparison and review across multiple stages. The evaluation also exposed important limitations. In one case, the AI changed the meaning of a requirement during processing, subsequently producing an incorrect conflict indication. This illustrates why source information and human verification remain necessary even when much of the processing is automated. Industry consultation further indicated that practical adoption would depend on consistently high output quality, appropriate reference documents, and clearly assigned review responsibilities.

The main contribution of the thesis is a set of explicit requirements for designing an AI-assisted requirements-governance workflow, together with the rationale explaining why these requirements are needed and how they can be implemented. From a Management of Technology perspective, introducing AI into requirements processes is not only a technical implementation problem. Organisations must also determine which information is authoritative, which roles should review different outputs, and who retains responsibility for final decisions. The proposed workflow therefore combines automated information processing with organisational governance rather than attempting to replace it.

More broadly, the thesis shows how AI can act as a preparation and coordination layer between document-intensive engineering work and more structured requirements-management processes. Its value lies primarily in changing how work is distributed: machines can assist with repeated searching, structuring, and comparison, while human experts retain responsibility for interpretation, judgement, and authorisation.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
APA	American Psychological Association
CSV	Comma-Separated Values
DR	Design Requirement
DSR	Design Science Research
DSRM	Design Science Research Methodology
ERA	European Union Agency for Railways
ETSI	European Telecommunications Standards Institute
FRMCS	Future Railway Mobile Communication System
FRS	Functional Requirements Specification
GPT	Generative Pre-trained Transformer
GSM-R	Global System for Mobile Communications – Railway
LLM	Large Language Model
MBRE	Model-Based Requirements Engineering
MBSE	Model-Based Systems Engineering
MoT	Management of Technology
n8n	Node-based automation workflow tool
PDF	Portable Document Format
RE	Requirements Engineering
REC	Railway Emergency Communication
RQ	Research Question
SRS	System Requirements Specification
TPM	Faculty of Technology, Policy and Management
TU Delft	Delft University of Technology
UIC	International Union of Railways

1 Introduction

Software-intensive systems are increasingly embedded in complex organisational and operational environments. In such contexts, the success of a system depends not only on whether it is implemented correctly, but also on whether it addresses the actual needs, constraints, and goals of the stakeholders involved. For this reason, requirements engineering (RE) remains a foundational activity in systems and software development. As Nuseibeh and Easterbrook (2000) explain, RE is concerned with identifying stakeholders and their needs, documenting these needs in analysable forms, and supporting communication and implementation across the development lifecycle. RE is therefore not merely a documentation task, it is a problem-definition activity that connects real-world purposes to technical system behaviour.

At the same time, requirements engineering is inherently difficult. Requirements do not emerge from a stable and fully agreed problem space. Instead, analysts usually begin with incomplete, ambiguous, and sometimes conflicting views of what a system should do. Cheng and Atlee (2007) argue that RE differs from many other software engineering activities because it is rooted in the problem space rather than the solution space. Analysts must reason not only about software, but also about stakeholders, environments, assumptions, and changing operating conditions. Similarly, Zave (1997) characterises RE as a broad and interdisciplinary field that must handle communication barriers, incompleteness, inconsistency, and evolution. This difficulty is not accidental, it is a defining feature of the discipline.

A further challenge is that requirements rarely exist as one stable and uniform set. In complex engineering settings, they are distributed across levels of abstraction, from broad operational or organisational intent to more specific system-level and design-relevant statements. The relationship between these levels matters because requirements must often be interpreted, refined, compared, and acted upon across organisational boundaries rather than within one isolated technical task. In this sense, requirements engineering is not only about writing correct statements, but also about maintaining coherence between different forms of intent, representation, and responsibility (Dardenne et al., 1993; Letier & van Lamsweerde, 2004; van Lamsweerde, 2001).

In practice, this challenge is amplified by the fact that requirements work remains strongly document-heavy and coordination-intensive. Organisations still rely extensively on natural-language artefacts, stakeholder interactions, and partially structured records rather than on one fully integrated modelling environment. This makes requirements handling flexible, but it also makes clarification, comparison, and cross-document review labour-intensive, especially when requirements evolve across multiple actors and lifecycle stages (Franch et al., 2023; Heikkilä et al., 2017; Méndez Fernández & Wagner, 2015; Wagner et al., 2019).

The literature also shows that several well-known requirements engineering concerns converge here. Traceability remains difficult to maintain consistently across evolving artefacts and organisational settings. Inconsistencies and ambiguities are common rather than exceptional. Existing tools and model-based approaches offer important support, but they do so unevenly, and they often assume a degree of structure and process discipline that many real organisational environments do not fully maintain (Carrillo de Gea et al., 2012; Finkelstein et al., 1993; Gotel & Finkelstein, 1994; Hunter & Nuseibeh, 1998; Ozkaya et al., 2023; Ramesh & Jarke, 2001; Sadovykh et al., 2021). The practical challenge addressed in this thesis therefore lies partly in the transition between these two environments: converting heterogeneous, document-based requirement inputs into structured records that can feed more formal requirements-management processes without losing their source context or governance information.

Recent automation research, including work on large language models and agentic systems, suggests that parts of this broader problem can be supported computationally. However, most existing work still concentrates on isolated tasks such as extraction, validation, or consistency checking rather than on integrated workflow-level support that links intake, clarification, comparison, traceability, and review into one coherent governance process (Tang & Runkler, 2026; Umar & Lano, 2024; Zadenoori et al., 2025). This is the point at which the present thesis is positioned.

The central problem can therefore be stated directly: in complex multi-stakeholder migration settings such as the GSM-R to FRMCS transition, requirements governance across heterogeneous documents and abstraction levels remains fragmented, manual, and dependent on implicit knowledge that is rarely documented and frequently lost when requirements change hands. Current tools, methods, and automation support address parts of this problem in isolation but do not provide integrated workflow-level support for governing requirements across sources while preserving traceability, surfacing conflicts, and keeping humans in control of consequential decisions.

In this thesis, *requirements governance* refers to the organisational and procedural mechanisms through which requirement-related information is made traceable, compared across sources and abstraction levels, reviewed by appropriate roles, and translated into accountable decisions. It builds on conventional requirements engineering but places additional emphasis on how authority, responsibility, review, and decision rights are managed as requirements move across documents, stakeholders, and lifecycle stages. Requirements engineering therefore remains concerned with identifying, specifying, analysing, and evolving requirements, while requirements governance in this thesis focuses on how those requirements and the information surrounding them are coordinated and controlled across organisational boundaries.

The organisational problem addressed in this thesis is therefore not owned by a company as a single actor, but by the roles responsible for translating, coordinating, reviewing, and authorising requirements across technical and managerial boundaries. In the context studied here, these responsibilities are primarily represented by requirements engineers, system architects or technical managers, and project managers. These roles experience different parts of the same governance problem: requirements engineers must establish and verify the requirement record, technical decision-makers must interpret conflicts and dependencies, and project-level decision-makers must assign ownership, priority, and organisational commitment. They therefore constitute the primary organisational users and problem owners for the workflow investigated in this thesis.

This thesis is positioned in exactly that space. It takes as its case context the transition from GSM-R to FRMCS in railway communications. This transition provides a suitable setting because it involves multiple stakeholders, multiple document types, multiple abstraction levels, and a strong need for continuity between existing and future requirements. In such a setting, requirements do not only need to be elicited or documented; they need to be compared, refined, traced, checked, and prepared for decision-making across levels of detail. Existing literature provides substantial support for the individual concerns involved in this process, including goal refinement, traceability, inconsistency handling, model-based representation, and automation. However, the literature remains weaker on integrated, workflow-level support for handling these concerns together across heterogeneous requirements artefacts.

The gap addressed by this thesis is therefore not the absence of requirements engineering methods, tools, or automation as such. Rather, the gap lies in limited design knowledge about how to construct a coherent workflow that can systematically transform heterogeneous and partially specified requirement inputs into structured, comparable, and reviewable outputs while preserving traceability and keeping consequential governance decisions under human control. This distinction is methodologically important. The absence of a tool is a practical problem. The absence of design knowledge about how such a workflow should be structured, what information it must capture, how human review should be distributed across its stages, and under what organisational conditions it can function meaningfully constitutes a knowledge gap that the existing literature does not yet resolve (Gregor & Jones, 2007). This thesis addresses that design gap, using the GSM-R to FRMCS railway communication migration as the empirical context in which the artefact is designed and evaluated.

This formulation has one further implication that shapes the research design. Because the gap is a design knowledge gap rather than a tool availability gap, the contribution of the thesis is not a new software product. Rather, the core design contribution is a set of explicit requirements for a requirements-governance workflow, together with a documented design rationale. The requirements define what such a workflow must be capable of, while the rationale explains why each requirement is needed, how it responds to the identified governance challenges, and under what conditions the resulting artefact can be expected to perform usefully. This combination of explicit design requirements and their underlying rationale distinguishes the DSR contribution from a purely technical description of the implemented prototype (Gregor & Jones, 2007; Hevner et al., 2004).

To address this problem, the thesis adopts a Design Science Research perspective, in which an artefact is developed and evaluated in response to an identified organisational problem (Hevner et al., 2004; Peffers et al., 2007). In this study, the artefact is a document-grounded workflow for requirements governance across heterogeneous sources and levels of abstraction. The contribution is therefore both practical and methodological: the thesis proposes a workflow design, implements a bounded prototype, and evaluates the extent to which that prototype supports traceability, comparison, refinement, and review under human oversight.

In doing so, the thesis contributes to three related discussions. First, it contributes to the RE practice literature by focusing on a concrete coordination problem that persists despite extensive work on elicitation, modelling, and traceability. Second, it contributes to the tooling and automation literature by shifting attention from isolated task support toward workflow-level orchestration. Third, it contributes to the emerging discussion on AI in requirements engineering by positioning AI not as a replacement for requirements judgement, but as a supporting mechanism within a human-governed workflow artefact.

The value of the workflow investigated in this thesis is therefore not that it eliminates human involvement in requirements governance. Rather, it lies in the potential to change the *type* of human work required: from repetitive mechanical tasks such as locating source documents, establishing accountability, and formatting comparison outputs, toward targeted verification, conflict resolution, and decision-making that genuinely requires domain expertise and organisational authority (Shneiderman, 2020). In large multi-stakeholder projects, a disproportionate share of senior engineering time is currently consumed by preparation tasks that do not require expert judgment but must nonetheless be performed before judgment can be exercised. A structured, AI-supported workflow that absorbs this preparation layer, while keeping human reviewers in control of all consequential governance decisions, represents a form of process innovation with direct relevance to the management of technology in complex infrastructure migration programmes. From a Management of Technology perspective, the workflow can be understood as a mechanism for converting implicit organisational knowledge about requirements governance into an explicit, reusable, and auditable process artefact (Hevner et al., 2004).

Thesis Structure

The remainder of the thesis is organised according to the logic of the Design Science Research Methodology. It moves from problem definition and literature-based positioning, through artefact design and bounded prototype demonstration, to evaluation, discussion, and conclusion. Figure 1 summarises this flow.

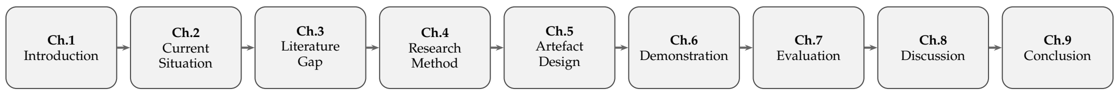

Figure 1: Thesis structure: nine chapters organised according to the DSRM process.

Chapter	Title	What it covers
Ch. 1	Introduction	Introduces the research context, explains why requirements governance in railway communication migration is difficult, and positions AI-assisted workflow support as a potential mechanism for addressing the identified governance problem.
Ch. 2	Current Situation	Reviews how requirements engineering is currently practised, where traceability and inconsistency remain persistent problems, and why existing tools provide uneven support.
Ch. 3	Literature Gap and RQs	Identifies the specific knowledge gap the thesis addresses and formulates four research questions that structure the investigation.
Ch. 4	Research Method	Explains the Design Science Research approach, defines the bounded case-based investigation, and describes the research procedure used to derive, develop, demonstrate, and evaluate the artefact.
Ch. 5	Artefact Design	Derives seven design requirements from the identified governance problems, explains the staged three-agent architecture and human checkpoint design, justifies the output structure, and identifies the organisational preconditions for workflow deployment.
Ch. 6	Demonstration	Shows the workflow running on two datasets: a synthetic stakeholder requirements list and a real section of the official FRMCS specification suite.
Ch. 7	Evaluation	Evaluates the prototype against the seven design requirements, compares its behaviour with a manual handling reference condition and a single-call AI reference condition, and reports the limitations of the bounded evaluation.
Ch. 8	Discussion	Interprets the role of AI and human checkpoints in requirements governance, examines organisational implications and prototype limitations and incorporates expert-review findings
Ch. 9	Conclusion	Answers the four research questions, states the core contribution as a set of explicit workflow design requirements together with their documented design rationale, presents practical recommendations for future application and identifies directions for future evaluation.

Table 1: Overview of thesis chapters and their content.

2 Current Situation in Requirements Engineering

2.1 Requirements engineering as a multi-level and socio-technical activity

Requirements engineering (RE) is widely recognised as the discipline concerned with identifying stakeholders' goals, functions, constraints, and needs, and with relating these to precise specifications and their evolution over time (Nuseibeh & Easterbrook, 2000; Zave, 1997). In principle, RE is not only about documenting what a system should do, but also about understanding why the system is needed, under what assumptions it will operate, and how software behaviour should relate to conditions in the surrounding environment (Nuseibeh & Easterbrook, 2000). This means that requirements engineering always involves both technical and organisational reasoning. It is not limited to software descriptions alone, but also includes domain knowledge, stakeholder intent, system boundaries, assumptions, and expected effects in practice.

A key feature of this situation is that requirements exist across multiple levels of abstraction. High-level requirements typically express strategic, operational, or organisational intent, while lower-level requirements move toward assignable, design-ready, or implementation-relevant forms. Goal-oriented requirements engineering has long shown that these levels should not be treated separately. van Lamsweerde (2001) argues that goals are central for eliciting, elaborating, structuring, analysing, negotiating, documenting, and modifying requirements. Similarly, Dardenne et al. (1993) show that requirements acquisition should start from goals, agents, and constraints rather than from isolated functions. This means that current RE practice, at least conceptually, is expected to move from higher-level intent toward lower-level requirements through structured refinement.

The literature also shows that this refinement process is not purely linear or mechanical. Letier and van Lamsweerde (2004) demonstrate that different refinements may satisfy higher-level goals to different degrees, especially when non-functional concerns and trade-offs are involved. Yu (1997) further distinguishes between early-phase RE, where stakeholder interests, rationales, and organisational dependencies are central, and later-phase RE, where the emphasis shifts toward precision, consistency, and formal verification. Gunter et al. (2000) add a further layer of clarity by separating domain knowledge, requirements, specifications, program behaviour, and the programming platform. Together, these studies show that the current understanding of RE is already strongly multi-level and socio-technical. Requirements engineering is therefore not simply about collecting statements, but about managing relationships between different forms of intent, knowledge, representation, and behaviour.

2.2 How requirements are commonly handled in practice

Although RE is conceptually rich, industrial practice remains far more document-heavy and context-dependent than many formal or model-based descriptions suggest. Survey-based and empirical studies consistently show that organisations still perform a large share of requirements work through interviews, meetings, stakeholder interactions, and textual artefacts rather than through one fully integrated modelling environment. Méndez Fernández and Wagner (2015) show that organisations continue to struggle with high-quality requirements engineering and report recurring practical problems such as communication flaws and moving targets. Wagner et al. (2019) reinforce this by showing that RE remains strongly shaped by local context, organisational arrangements, and human coordination, making it difficult to describe practice through one stable and universally applicable process model.

This practical reality becomes even clearer in the area of specification. Franch et al. (2023) show that, despite extensive academic work on notations, models, and tools, practitioners still rely predominantly on natural language and general-purpose supporting artefacts. Their interview-based results show that ambiguity, inconsistency, and incompleteness remain among the most common problems in day-to-day requirements specification work. In other words, the current situation is not one in which requirements are already born and managed inside a clean, formal, model-based ecosystem. Rather, they are usually created, revised, and interpreted across a mixed environment of documents, stakeholder communication, and partial structuring mechanisms.

Requirements quality research supports the same picture. Montgomery et al. (2022) show that empirical work on requirements quality concentrates strongly on ambiguity, completeness, consistency, and correctness. Importantly, their review also suggests that much empirical work still focuses on generic requirements rather than on the different artefact types that appear in real settings. This matters because current industrial RE rarely works with one single homogeneous requirement form. Instead, it often involves a range of requirement-like artefacts, such as goals, stakeholder requests, tables, narratives, change notes, technical constraints, or partially structured records.

Other empirical studies show that the challenge is not only the local quality of requirements, but the coordination of requirements across wider organisational processes. Heikkilä et al. (2017) show that, in large-scale development, requirements do not flow through one simple pipeline. Instead, they move through several linked processes that connect strategy, release planning, implementation, and feedback, each with its own coordination burden. Kirikova (2017) similarly argues that requirements engineering increasingly needs to be treated as continuous rather than front-loaded, especially in settings where business environments and development conditions change over time. This continuous character strengthens the case for workflow-level automation: when requirements are added, revised, or reinterpreted, activities such as source tracing, comparison, clarification, and review preparation must be repeated. A reusable workflow can reduce the burden of repeatedly performing these preparatory tasks while allowing human reviewers to concentrate on the new or changed decisions that require judgement. Sommerville and Ransom (2005) show that organisations attempt to improve RE process maturity, but that practical improvements remain difficult to relate cleanly to business outcomes. Unterkalmsteiner et al. (2015) add that coordination between requirements engineering and testing remains a substantial issue, indicating that requirements handling is deeply connected to downstream lifecycle activities rather than confined to one early development phase.

In complex railway communication programmes, requirements are not produced, interpreted, or acted upon by a single stakeholder group. They move between actors with different responsibilities, sources of authority, and decision rights. Table 2 summarises the main stakeholder groups relevant to the requirements-governance problem addressed in this thesis.

These stakeholder groups do not operate independently. Requirements originating from operational, technical, or normative sources must be interpreted and refined across organisational boundaries before they can become design-ready commitments. This cross-stakeholder movement of requirement information is one reason why requirements governance extends beyond the specification task itself and includes questions of authority, ownership, review, and decision-making.

Taken together, these studies suggest that the current situation in RE is characterised by three persistent features. First, requirements work is still highly dependent on natural language and heterogeneous document-based artefacts. Second, requirements handling remains strongly de-

Table 2: Main stakeholder groups in railway communication requirements governance

Stakeholder group	Role in requirements governance	Typical interaction with requirements
Infrastructure managers	Define operational and infrastructure-related needs and ensure that communication systems support railway network operation.	Provide operational requirements, interpret infrastructure constraints, and review whether proposed solutions fit network-level needs.
Railway undertakings / operators	Represent operational users of the railway communication system, including drivers, controllers, and operational staff.	Provide user and operational requirements and assess whether system behaviour supports practical railway operation.
System suppliers and technology vendors	Translate higher-level operational and system requirements into implementable technical solutions.	Refine requirements, identify technical dependencies, resolve feasibility issues, and provide design-level evidence.
Standardisation and regulatory bodies	Define normative requirements, interoperability constraints, and sector-wide technical expectations.	Provide authoritative standards and specifications against which organisational and system requirements must be interpreted and compared.

pendent on coordination across actors, roles, and lifecycle stages. Third, manual effort remains central, especially when requirements must be interpreted, reconciled, refined, or checked across different abstraction levels.

2.3 Current handling of traceability, inconsistency, and quality concerns

A central part of current RE practice is the attempt to maintain traceability, consistency, and acceptable quality while requirements evolve. The literature makes clear that these concerns are not peripheral, but core elements of the way requirements are currently understood and managed.

Traceability is perhaps the clearest example. Gotel and Finkelstein (1994) define requirements traceability as the ability to describe and follow the life of a requirement in both forward and backward directions. Their distinction between pre-requirements-specification and post-requirements-specification traceability is especially relevant, because it shows that many traceability problems begin before requirements are stabilised in formal specifications. This is highly important for document-heavy environments, where many meaningful requirement transformations happen before a final structured artefact exists. Ramesh and Jarke (2001) extend this by showing that traceability is not practiced uniformly and that different organisational settings require structured models for managing trace links according to actual use.

Inconsistency handling is another persistent part of the current situation. Finkelstein et al. (1993) argue that large and complex systems are developed by multiple participants with different perspectives, responsibilities, and commitments, making inconsistency across viewpoints inevitable. They also argue that absolute consistency is not always desirable, because enforcing it too early may remove useful information or constrain development prematurely. Hunter and Nuseibeh (1998) build on this by suggesting that inconsistency should not only be detected, but also reasoned about and acted upon. This is important because it means that current RE practice cannot realistically assume clean inputs or total alignment. Instead, it must work with partial agreement, conflicting descriptions, and evolving interpretations.

Requirements quality introduces a third related concern. Montgomery et al. (2022) show that ambiguity, completeness, consistency, and correctness remain central quality dimensions in empirical RE research. In practice, this means that current requirements work is not only about collecting and recording information, but also about detecting weak, unclear, incomplete, or contradictory statements and deciding how they should be treated. Quality work therefore sits alongside traceability and inconsistency handling as a normal part of RE practice rather than as an optional extra.

The problem in the current situation is that these concerns are often addressed through separate lenses. Traceability is treated as one issue, inconsistency as another, and quality as another again. Yet in practical requirements work, they usually appear together. A weakly specified high-level requirement may lead to inconsistent refinements; missing traceability may make a conflict harder to diagnose; and low-quality source material may undermine both refinement and review. This suggests that although current RE practice recognises all three concerns, it still handles them in a way that is often fragmented rather than fully integrated.

2.4 Current tool support and model-based support

The current situation in RE is also shaped by a broad landscape of tools and methods designed to support requirements work. This is important because it means the thesis cannot claim that there are no existing solutions. On the contrary, the literature shows that a substantial body of support already exists.

Carrillo de Gea et al. (2012) show that many RE tools support parts of the RE process, but also identify continuing weaknesses in modelling, open data models, and data integration. Ozkaya et al. (2023) provide a more recent analysis of 56 RE tools and find that no single tool supports all important viewpoints. Their results show that project planning and execution are weakly supported in many tools, that natural language remains the dominant specification medium, that precise modelling is relatively rare, and that requirements analysis is not commonly supported in depth. They also note that requirements transformation mainly results in document production rather than richer downstream artefacts, and that extension of notations and domain-specific formalisation remain comparatively limited.

These findings matter because they show that current tool support is broad but uneven. Many tools are good at storing, organising, versioning, and documenting requirements. Some support collaboration, issue management, or trace links. However, fewer support cross-level reasoning, deeper requirements analysis, or coherent coordination across heterogeneous source artefacts. The literature therefore suggests that current tool support is substantial, but also fragmented.

Model-based requirements engineering (MBRE) attempts to address some of these weaknesses by introducing more structured representations and stronger support for traceability, automation, and analysis. Sadovykh et al. (2021) report that MBRE in large collaborative projects can improve scalability, heterogeneity handling, traceability, and automated documentation generation. Zhao and Zhou (2026) similarly show that MBRE has advanced significantly across modelling methods, RE activities, quality mechanisms, and application domains. In technically complex environments, model-based and MBSE-inspired approaches have also been argued to support consistency, lifecycle coherence, and change management more effectively than purely document-based practice (Windisch et al., 2022).

However, the current situation also shows a clear boundary here. MBRE and related approaches are often difficult to adopt broadly because they assume structured representations, modelling discipline, and a degree of process maturity that many real organisations do not consistently maintain. They are often strongest in environments that are already model-based, and weaker in environments where the input remains text-heavy, mixed-format, and only partially formalised. This creates a practical space between document-heavy industrial reality and fully structured model-based engineering. Existing tools and model-based approaches are therefore valuable, but they do not remove the need for support that can operate across heterogeneous artefacts without assuming that all requirements are already modelled in one coherent environment.

This thesis is positioned precisely at this boundary. Rather than replacing existing requirements-management or model-based approaches, the proposed workflow acts as an upstream preparation layer that transforms heterogeneous, document-based inputs into structured, source-grounded, and reviewable requirement records that can be more readily incorporated into established requirements-engineering processes.

2.5 Current automation support and the move toward AI-assisted RE

A more recent part of the current situation concerns automated support for requirements engineering. The literature shows that automation in RE is not hypothetical; it is already an active and growing area of research.

Umar and Lano (2024) review automated support for RE and show that a substantial body of work has emerged across phases and artefacts. Their review finds that automated tools most often support model generation, omission detection, consistency checking, and requirement validation, and that the analysis phase is the most widely automated. At the same time, the review also shows that most tools are semi-automated, that human validation remains necessary, and that future work should more explicitly address traceability between textual requirements and formalised models. This is especially relevant because it shows that automation has mainly been used to support individual tasks rather than fully integrated workflows.

The current literature on large language models (LLMs) in RE reinforces this pattern. Zadenoori et al. (2025) show that LLM-based applications in RE are growing rapidly, but remain concentrated mainly on elicitation and validation. They also observe that studies are usually evaluated in controlled settings, with limited industrial integration and limited support for broader workflow coordination. This suggests that current AI-assisted RE is promising, but still mainly oriented toward selected task support.

The agentic systems literature points to a further development in the current situation. Tang and Runkler (2026) describe LLM-based multi-agent systems as especially relevant for complex software engineering tasks that require role specialisation, collaboration, iterative reasoning, and tool use. They explicitly include requirements engineering within this landscape and note that agentic systems may support elicitation, analysis, model or specification generation, and validation. However, they also emphasise that challenges remain around human-agent coordination, evaluation, and integration across lifecycle stages (Tang & Runkler, 2026).

Taken together, the current situation in automation can be summarised as follows. There is already significant progress in task-specific automation, and there is a growing body of work on

LLMs and agentic approaches. However, support remains largely fragmented across functions, while more integrated workflow-level support remains comparatively underdeveloped.

Taken together, the current situation in requirements engineering is not one of theoretical absence, but of fragmented practical support. Requirements remain distributed across heterogeneous artefacts, coordination remains manual and role-dependent, and automation remains concentrated on isolated tasks rather than end-to-end governance. The unresolved problem is therefore not whether individual RE activities can be supported, but how they can be connected in a coherent workflow that preserves traceability, supports comparison and refinement across abstraction levels, and remains reviewable for human stakeholders. The next chapter formulates this more precisely as the literature gap addressed by the thesis. Table 3 provides a compact synthesis of the current situation in requirements engineering and the limitations that remain insufficiently addressed in the literature and in practice.

Table 3: Current situation in requirements engineering and its unresolved limitations.

Dimension	**What current practice/-support provides**	**What remains weak**	**Why it matters for this thesis**
Requirements handling in practice	Requirements are commonly elicited, discussed, and documented through natural-language artefacts, stakeholder interaction, and partially structured records.	Handling remains document-heavy, coordination-intensive, and dependent on manual clarification and comparison across roles and lifecycle stages.	The workflow must operate on heterogeneous, document-grounded inputs rather than assuming a fully model-based environment.
Traceability and inconsistency management	The literature recognises traceability and inconsistency handling as central concerns in RE.	These concerns are often treated separately, and in practice they remain difficult to maintain across evolving artefacts and organisational settings.	The workflow must preserve source grounding, surface mismatches, and make review-relevant governance information explicit.
Requirements quality management	Ambiguity, incompleteness, consistency, and correctness are well-established quality dimensions.	Quality problems remain persistent in practice, especially when requirements are informal, incomplete, or distributed across different artefact types.	The workflow must support clarification and structured review before weak requirement statements propagate downstream.
Tool and model-based support	A broad landscape of RE tools and model-based approaches already exists, with useful support for storage, versioning, modelling, and traceability.	Support remains uneven across the lifecycle and often assumes a degree of structure and modelling discipline that many real environments do not sustain.	The thesis cannot assume the absence of support, but instead addresses the gap between structured tools and messy document-based practice.

Continued on next page

Table 3 continued from previous page

Dimension	**What current practice provides**	**What remains weak**	**Why it matters for this thesis**
Automation and AI-assisted support	Automation and AI have shown promise for tasks such as extraction, validation, and consistency checking.	Most support remains task-specific and fragmented, with limited workflow-level integration across intake, comparison, refinement, and review.	This motivates the thesis focus on a document-grounded workflow artefact rather than on isolated automation of a single RE task.

3 Literature Gap and Research Questions

The literature reviewed in the previous chapter shows that requirements engineering is not weak in concepts, methods, or isolated forms of support. The field already offers substantial understanding of abstraction levels, traceability, inconsistency management, requirements quality, tooling, model-based approaches, and automation. What remains weaker is the integration of these concerns into one coherent workflow for governing requirements across heterogeneous artefacts while preserving traceability, surfacing inconsistencies and quality issues, and producing outputs that remain reviewable for human stakeholders.

This is the core literature gap addressed in the thesis. Existing work explains many relevant parts of the problem, but it does so mainly through separate streams. Refinement literature explains how higher-level goals may lead to lower-level requirements. Traceability literature explains how requirement relationships may be preserved. Inconsistency literature explains how conflicting perspectives may be tolerated and managed. Quality literature explains what weaknesses should be identified. Tool and MBRE literature explain what structured support is possible. Automation literature explains which tasks may be partly or fully assisted by machines. What is still insufficiently addressed is how these concerns can be handled together in one document-grounded workflow that begins from heterogeneous source artefacts and supports cross-level governance in a complex transition setting.

The gap becomes even more specific when seen from the perspective of current practice. In many engineering settings, requirements are not handled through one clean and unified environment. They are spread across legacy documents, transition studies, standards, technical reports, stakeholder notes, meeting outputs, and supporting tools. At the same time, governance tasks such as refinement, traceability preservation, duplicate/conflict detection, and review preparation still depend heavily on manual effort. This creates a situation in which the practical challenge is not only to represent requirements, but to orchestrate several linked tasks across heterogeneous artefacts and across abstraction levels. The literature reviewed above does not yet provide sufficiently strong, integrated support for this type of cross-level, document-grounded requirements governance.

The governance problem identified in the preceding sections can also be represented as an organisational information flow. In current practice, heterogeneous requirement inputs move between technical and managerial roles through a sequence of interpretation, comparison, clarification, and decision-making activities. Although these activities are connected, the information supporting them is often distributed across documents, tools, and informal communication. Figure 2 summarises this current-state process and highlights where governance information can become fragmented as requirements move between roles.

The figure is intended as a conceptual representation rather than a fixed organisational process model. Specific role titles and responsibilities may differ between organisations, but the underlying governance challenge remains: requirement information must move between actors with different technical knowledge and decision authority, while the supporting context is not consistently preserved in a structured form.

This gap directly motivates the main research question of the thesis. To align the wording with the literature, especially the emphasis on traceability and managed inconsistency rather than absolute elimination of conflict, the main research question can be formulated as follows:

> *How can a document-grounded workflow artefact support requirements governance across heterogeneous sources and levels of abstraction in the GSM-R to FRMCS transition, while preserving traceability, supporting comparison and refinement, and keeping humans in control?*

This formulation is deliberate. The thesis is not focused on achieving conflict-free requirements handling, which would be unrealistic in complex multi-actor settings. Instead, it focuses on improving requirements governance across heterogeneous sources and levels of abstraction by preserving traceability, supporting comparison and refinement, and keeping human review central to decision-making.

The first sub-question, RQ1, is exploratory: *How are requirements currently captured, clarified, compared, prioritized, and refined across documents, roles, and abstraction levels in the GSM-R to FRMCS transition, and where do the main manual bottlenecks remain?* This question addresses the need to understand how requirements governance currently happens in practice and where current approaches remain insufficient.

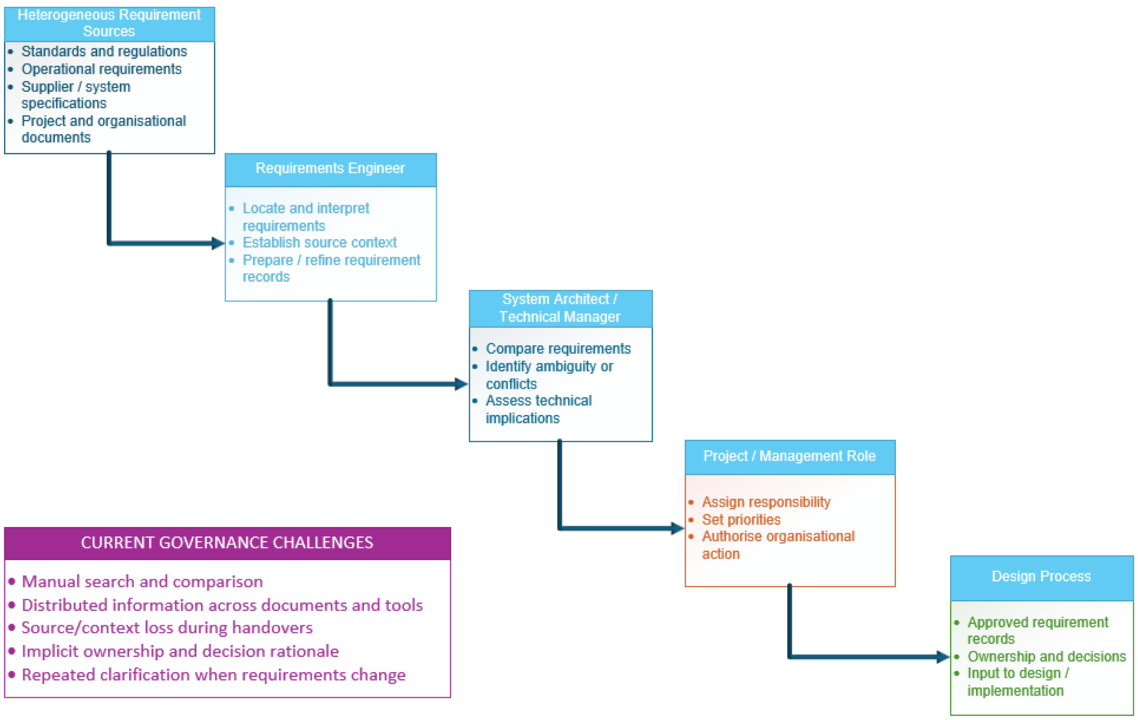

Figure 2: Conceptual current-state requirements-governance flow across heterogeneous sources and organisational roles.

The second sub-question, RQ2, is design-oriented: *What design requirements should a document-grounded workflow artefact satisfy to support traceable, reviewable, and human-controlled requirements governance across heterogeneous sources and abstraction levels?* This question translates the literature gap into concrete design requirements for the artefact.

The third sub-question, RQ3, concerns prototype capability: *To what extent can a prototype workflow transform heterogeneous and partially specified requirement inputs into structured, source-grounded, comparable, and actionable outputs that support clarification, refinement, and human review?* This question examines whether the proposed prototype can operationalise the workflow logic identified in the literature in a bounded and practically relevant way.

The fourth sub-question, RQ4, is evaluative: *How does the staged prototype compare with manual handling and a single-call AI reference condition within the bounded evaluation, particularly regarding traceability, reviewability, and observed processing limitations?* This question addresses whether the staged workflow provides added value over baseline approaches under realistic constraints.

Table 4 provides an overview of the research questions and shows how they structure the exploratory, design, demonstration, and evaluation components of the study.

RQ	**Type**	**Focus**	**Main role in thesis**
RQ1	Exploratory	Understand how requirements governance is currently handled across documents, roles, and abstraction levels, and identify the main manual bottlenecks.	Problem exploration and baseline diagnosis
RQ2	Design-oriented	Determine what design requirements a document-grounded workflow artefact must satisfy to support traceable, reviewable, and human-controlled requirements governance.	Derivation of design requirements
RQ3	Prototype-oriented	Examine whether a bounded workflow prototype can transform heterogeneous inputs into structured, source-grounded, comparable, and actionable outputs.	Demonstration of artefact capability
RQ4	Evaluative	Compare the staged prototype with manual handling and a single-call AI reference condition within the bounded evaluation, with particular attention to traceability, reviewability, and observed processing limitations.	Bounded comparison against reference conditions

Table 4: Research questions and their role in the thesis.

The same literature gap also explains why agentic AI is relevant to the thesis. The opening for agentic AI does not come from the claim that current RE has no tool support, nor from the assumption that multi-agent systems are inherently superior to traditional methods. Rather, it comes from the fact that the target problem is multi-step, decomposable, iterative, and dependent on intermediate artefacts and human review loops. Document-grounded requirements governance in this thesis includes intake of heterogeneous inputs, source grounding, clarification, comparison, refinement, and reviewer-oriented output generation. These are not independent activities; they must be coordinated. The recent literature on LLM-based and agentic systems suggests that such settings are especially suitable for agentic configurations, because different agents can be specialised for different roles while still operating within one broader orchestration framework (Tang & Runkler, 2026). At the same time, the literature also makes clear that human oversight remains necessary, which is why agentic AI is positioned in this thesis as a coordination mechanism under human supervision rather than as a replacement for requirements engineering judgement.

Taken together, the research questions therefore mirror the structure of the literature gap. RQ1 addresses the fragmented current situation. RQ2 derives the design requirements of a better workflow. RQ3 investigates whether such a workflow can transform incomplete and heterogeneous inputs into structured, comparable, and reviewable outputs under human control. The thesis contribution is thus not to add another isolated feature to the RE tool landscape, but to design and evaluate a document-grounded workflow artefact for requirements governance in a complex transition setting.

Accordingly, the final literature gap can be stated as follows: while the RE literature offers strong conceptual foundations, substantial empirical understanding, a broad tool landscape, and growing automation support, it still lacks sufficiently integrated workflow support for governing requirements across abstraction levels and heterogeneous artefacts while preserving traceability, managing inconsistency, surfacing quality issues, and producing outputs that remain reviewable for human stakeholders. This gap is especially significant in transition contexts such as railway communications migration, where legacy and future-oriented requirements artefacts, multiple stakeholder perspectives, and multiple abstraction levels must be connected coherently. The thesis responds to this gap by investigating how a document-grounded workflow artefact, supported by staged agentic AI under human oversight, can improve requirements governance under these conditions. The next chapter explains how this research problem is addressed methodologically through a Design Science Research approach, and how the artefact is developed, demonstrated, and evaluated within the bounded GSM-R to FRMCS case context.

4 Research Method

4.1 Research Design and Case-Based Approach

This study follows Design Science Research Methodology (DSRM), using the process proposed by Peffers et al. (2007) as the overall research logic. DSRM is appropriate because the objective of the study is not only to describe an existing requirements-governance problem, but to derive, develop, demonstrate, and evaluate an artefact intended to address that problem (Hevner et al., 2004; Peffers et al., 2007). The research therefore combines exploratory problem analysis with design-oriented artefact development and evaluation.

The research is conducted within a bounded, case-based investigation of the GSM-R to FRMCS transition in railway communications. The case provides a practical setting in which the characteristics central to the research problem are present, including heterogeneous requirements documents, multiple stakeholder roles, different levels of abstraction, and recurring requirements-governance activities. The case is used to develop and evaluate design knowledge under realistic conditions rather than to make statistical claims about railway projects in general.

The study distinguishes between the conceptual workflow artefact and its bounded prototype implementation. The conceptual artefact addresses the broader requirements-governance problem, whereas the prototype operationalises a selected subset of these functions within the scope and resources of the thesis. This distinction allows the proposed design to address the broader governance problem while keeping the empirical demonstration and evaluation sufficiently bounded.

The primary unit of analysis is the requirement-related statement contained in the selected source material. Relationships between requirement statements and their sources, including traceability, comparison, and refinement relationships, form a secondary unit of analysis. The study considers requirement information across high-, mid-, and low-level abstraction, while recognising that the implemented prototype demonstrates only selected transformations across these levels.

4.2 Research Procedure

The study followed five connected activities derived from the DSRM logic. First, the requirements-governance problem was explored through the literature and practitioner input in order to establish the practical and conceptual baseline. Second, the identified problems were translated into explicit design requirements for the workflow artefact. Third, these requirements guided the iterative development of a bounded prototype. Fourth, the prototype was demonstrated using both synthetic requirement material and selected real FRMCS specification documents. Finally, the artefact was evaluated against the derived design requirements and against manual and single-pass AI baseline approaches, complemented by expert review of selected prototype outputs.

The resulting design requirements and artefact architecture are presented in Chapter 5, the prototype execution is reported in Chapter 6, the evaluation results are presented in Chapter 7, and their broader implications are discussed in Chapter 8.

Problem exploration combined a literature-based analysis of current requirements-engineering practice with practitioner input and explicit case-boundary definition. The literature review was used to identify recurring governance difficulties relevant to heterogeneous, document-based requirements work. Preliminary semi-structured practitioner interviews were then used to assess whether these difficulties reflected the industrial setting and to refine the practical scope of the study. The interviews were used as contextual and design-oriented evidence rather than as a standalone qualitative dataset intended for systematic thematic analysis. An overview of the practitioner interviews is provided in Appendix G, and the corresponding interview guides are included in Appendix H.

The design requirements were derived by translating the baseline pain points into explicit capabilities that the workflow artefact should provide. Each identified problem was linked to one or more design requirements and paired with a verification criterion so that the resulting requirements could later be evaluated rather than merely asserted. The resulting design requirements and their derivation from the identified governance problems are presented in Chapter 5.

Artefact development followed an iterative design-and-build process in which the derived design requirements were translated into a bounded workflow prototype implemented in n8n. The prototype architecture, agent responsibilities, output structure, and human-review logic were refined during development to operationalise the identified requirements within the practical scope of the thesis. These design decisions and their rationale are presented in Chapter 5.

The developed prototype was demonstrated on two bounded datasets: a synthetic stakeholder-style requirements dataset and a real-document dataset drawn from the FRMCS specification suite. The purpose of the demonstration was to establish whether the artefact could be applied coherently to representative heterogeneous requirement inputs before assessing its performance. Detailed dataset selection, workflow execution, and demonstration outputs are presented in Chapter 6. Demonstration was treated separately from evaluation, where the former established whether the artefact could be meaningfully applied to the bounded case, whereas the latter assessed how well it performed against explicit criteria and baseline approaches.

Evaluation combined three forms of evidence. First, the prototype was assessed against the design requirements derived earlier in the study. Second, its behaviour was compared with two baseline conditions: manual requirements handling and unstaged single-pass AI processing. Third, selected prototype outputs were reviewed by an industry FRMCS standardisation specialist to provide domain-informed assessment of comparison quality and identified discrepancies. The detailed evaluation criteria, bounded reference criteria, and assessment rubric are documented in Appendix F, while the evaluation results are presented in Chapter 7.

Evaluation evidence was analysed using a combination of structured indicators and qualitative interpretation, allowing directly observable properties of the workflow outputs to be assessed alongside reviewer interventions, identified errors, and practitioner feedback.

The practitioner interviews and expert-review interactions were conducted voluntarily and were treated as research material. Participants are referred to by professional role where appropriate, and raw interview transcripts are not reproduced in the main thesis. Publicly available specification documents were used for the prototype demonstration and evaluation, and any future use of confidential project material would require handling in accordance with applicable organisational and university data-management requirements.

5 Artefact Design

5.1 Deriving the Design Requirements

The design requirements for the workflow artefact are derived directly from the baseline limitations identified in Chapter 2 and the literature gap established in Chapter 3. Rather than selecting design features on the basis of technical convenience or implementation preference, this thesis follows a transparent derivation logic in which each requirement is traceable to a specific observed limitation in current practice and to a corresponding knowledge gap in the literature. This approach reflects the design science principle that artefact objectives must be explicitly grounded in the problem definition and paired with verifiable acceptance criteria (Hevner et al., 2004; Peffers et al., 2007). Gregor and Jones (2007) further argue that a design theory must articulate the principles of form and function that explain why an artefact takes the shape it does, not only what the artefact produces. Section 5.1 addresses that requirement by making the derivation chain explicit before any design or implementation decisions are described.

The baseline analysis in Chapter 2 identified three persistent characteristics of current requirements engineering practice that are particularly relevant to the governance problem addressed in this thesis. First, requirements work remains strongly dependent on heterogeneous, document-based artefacts expressed in different formats, at different levels of abstraction, and from sources carrying different degrees of formal authority (Franch et al., 2023; Méndez Fernández & Wagner, 2015; Wagner et al., 2019). Second, requirements handling involves substantial coordination effort across actors, roles, and lifecycle stages, with manual effort concentrated in tasks such as cross-document comparison, clarification of ambiguous inputs, and assignment of ownership (Heikkilä et al., 2017; Montgomery et al., 2022). Third, critical governance information, including the authority of a requirement's source, the identity of the requesting stakeholder, the rationale for a comparison decision, and the existence of unresolved ambiguity, is typically managed through informal communication, tacit knowledge, and undocumented assumption rather than through structured records (Gotel & Finkelstein, 1994; Ramesh & Jarke, 2001). When a requirement changes hands, this implicit information is frequently lost, making traceability, accountability, and cross-level comparison substantially more difficult than they would otherwise need to be (Ozkaya et al., 2023; Umar & Lano, 2024). These characteristics were corroborated by preliminary expert interviews with industry practitioners, whose accounts of current requirements governance practice confirmed that the identified pain points reflect real industrial experience (Kiss, 2026a; Lindner, 2026).

These three characteristics generate a specific set of governance challenges that current tool support and automation do not adequately address. As shown in Chapter 3, the literature gap is not the absence of requirements engineering methods or tools in general, but rather the limited availability of integrated workflow-level support for governing requirements across heterogeneous sources and abstraction levels while preserving traceability, surfacing inconsistencies, and producing outputs that remain reviewable for human stakeholders. From this gap, and from the three baseline characteristics described above, seven design requirements are derived for the workflow artefact. These requirements are expressed as functional capabilities that the workflow must be able to exercise. Each is paired with a verification criterion that is used in the evaluation phase in Chapter 7. The full derivation chain is presented in Table 5. Figure 3 summarises this derivation logic visually, showing how the identified practical problems motivate knowledge gaps, which in turn translate into the required capabilities of the workflow artefact.

Table 5: Design requirements derivation: from baseline pain points to required workflow capabilities

Baseline pain point	**Knowledge gap**	**Required capability**	**Verification criterion**
Heterogeneous inputs differ in source, format, abstraction level, and authority	Unclear what transformations and metadata fields make such inputs systematically comparable	**DR1 Source Grounding** Ingest and preserve heterogeneous inputs without losing source context	Source grounding correctness: does each output record correctly identify and preserve its origin?
Requirements arrive as partial statements without propagation information	Unclear what fields a record must contain to make downstream impact visible	**DR2 Governance Impact** Capture the governance context needed for downstream routing, interpretation, and review at the point of intake.	Are stakeholder type, abstraction level, and relevant intake metadata captured for each record?
Knowledge needed to review a requirement is distributed across a hierarchy with no single reviewer holding complete authority	Unclear how to design checkpoints that elicit specific domain knowledge without creating rubber-stamp approval	**DR3 Role-Appropriate Review** Structure human review checkpoints around role-appropriate judgment tasks	Checkpoint appropriateness: does each checkpoint elicit a judgment that a domain expert confirms is within their competence?
Critical governance information is left implicit and undocumented across organisational boundaries	Unclear which fields a structured record must explicitly capture to surface what current practice leaves implicit	**DR4 Governance Visibility** Surface implicit governance information explicitly in the requirement record	Field completeness: are source authority, stakeholder identity, comparison rationale, ambiguity, and ownership all captured?
All governance tasks are performed simultaneously in a single manual pass, making error propagation invisible	Unclear how to decompose governance tasks across staged processing in a way that prevents compounding errors	**DR5 Staged Processing** Decompose requirements governance tasks across discrete processing stages with explicit boundaries	Stage separation: does each agent operate only on the outputs of the preceding stage?
Human review is treated as a single undifferentiated approval step regardless of what type of judgment is needed	Unclear how to distinguish corrective, judgmental, and authorisation review within a staged workflow	**DR6 Review Differentiation** Make the type of human intervention required at each checkpoint explicit and role-specific	Checkpoint differentiation: can a reviewer identify what type of judgment is expected of them at each stage?
Outputs are structured for machine processing but not for human decision-making across reviewer profiles	Unclear what output format is needed to make a requirement record legible to all parties who must act on it	**DR7 Human-Readable Outputs** Produce outputs that are legible to human reviewers without requiring technical expertise to interpret	Output reviewability: can a domain expert (non-technical) act on the output without consulting the system?

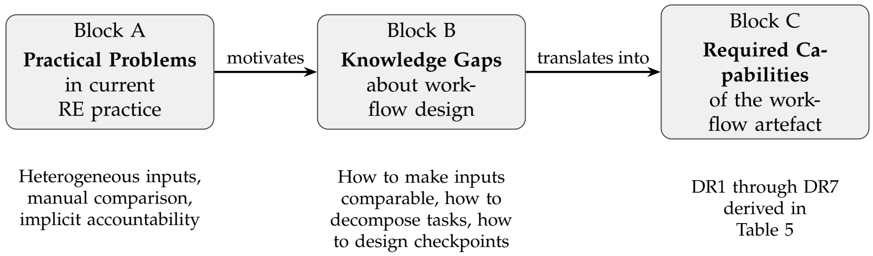

Figure 3: Derivation chain from practical problems to knowledge gaps to required workflow capabilities. Each block motivates the next, grounding the seven design requirements in observed baseline limitations.

The seven design requirements cover three distinct layers of the governance challenge. DR1 and DR2 address the input layer: the workflow must be able to process heterogeneous inputs and capture sufficient context at the point of intake for downstream governance functions to operate on a sound foundation. DR3 and DR6 address the human governance layer: the workflow must structure human review in a way that elicits specific, role-appropriate judgment rather than undifferentiated approval. DR4 and DR5 address the information architecture layer: the workflow must make explicit what current practice leaves implicit, and must do so through a staged architecture that prevents error propagation between governance functions. DR7 is a cross-cutting requirement that applies to all outputs: the workflow must produce artefacts that are actionable for human reviewers at each checkpoint, regardless of their technical familiarity with the underlying system.

It is important to note that these seven requirements do not describe the implementation of the prototype in isolation. They describe what any solution to the identified literature gap must be capable of doing. The prototype operationalises these requirements through a specific technical architecture, described in the sections that follow. The relationship between each design requirement and the specific prototype elements that address it is documented in the design-logic table presented in Section 5.5, and is used as the basis for the evaluation rubric in Chapter 7.

5.2 Architecture Rationale: The Three-Agent Staged Design

The workflow artefact described in this thesis is implemented as a staged pipeline of three specialised processing agents rather than as a single-pass or monolithic system. This architectural choice requires explicit justification because it is a design decision with significant consequences for both the quality of the outputs and the structure of human oversight.

5.2.1 Why a staged architecture rather than a single-pass approach

A single-agent approach, in which one processing call performs intake, source grounding, clarification, comparison, and output generation simultaneously, is technically feasible but unsuitable for the governance problem addressed in this thesis. The central difficulty is error propagation. When all governance tasks are performed in one undifferentiated pass, an error made during extraction becomes an invisible assumption in the comparison stage, which in turn produces a flawed ownership assignment. There is no natural boundary at which human judgment can intervene to correct the trajectory before the error compounds. This problem is not hypothetical. It corresponds directly to the baseline limitation identified in Chapter 2: that requirements governance currently fails not because individual tasks are poorly understood, but because the coordination across those tasks is fragmented and error-prone (Heikkilä et al., 2017; Wagner et al., 2019).

A staged architecture addresses this by creating explicit processing boundaries. Each boundary is a point at which the output of one stage is validated before it becomes the input of the next. This means that errors are caught early rather than compounding silently, and that human review is positioned at structurally meaningful points rather than only at the end of the process. The literature on agentic AI systems supports this rationale: Tang and Runkler (2026) argue that multi-agent configurations are especially suitable for complex tasks that are decomposable into sequential roles, because role specialisation improves reliability and supports intermediate verification. The requirements governance workflow addressed in this thesis exhibits precisely these characteristics.

5.2.2 Why three stages rather than two or five

The choice of three stages is not arbitrary. It reflects the minimum viable decomposition of requirements governance tasks into functionally distinct categories, each of which requires a different type of processing and a different type of human review.

The first stage addresses *intake and grounding*. Its function is to transform unstructured, heterogeneous requirement-related inputs into structured, source-grounded records. This stage must be completed and validated before any comparison or ownership work can begin, because comparison against a reference standard is only meaningful when the input has a stable, normalised representation and a verified source attribution. Combining intake with comparison in one stage would mean that a poorly grounded input could produce a misleading comparison note with no visible indication that the foundation was unreliable.

The second stage addresses *comparison and clarification*. Its function is to evaluate each grounded requirement against the reference specification, surface mismatches and open questions, and produce a clarified version of any requirement that was identified as ambiguous during intake. This stage depends on the outputs of the first stage and must be completed and validated before ownership and priority can be assigned, because assigning ownership to a requirement whose relationship with the governing standard is still contested is a governance failure rather than a governance decision.

The third stage addresses *actionable output generation*. Its function is to convert the validated, comparison-annotated requirement records into structured outputs that are legible and actionable for different reviewer profiles: a structured requirement register for engineering teams and a human-readable design brief for senior decision-makers. This stage depends on the outputs of the second stage and cannot be meaningfully executed until conflicts have been flagged and open questions surfaced.

Two stages would require either intake and comparison or comparison and output generation to be combined, reducing the separation between governance functions and the opportunity for intermediate review. Additional stages were not introduced because the derived design requirements did not identify further governance functions requiring a separate processing and review boundary. The three-stage decomposition was therefore selected as the most parsimonious architecture that separated source grounding, comparison and clarification, and actionable output generation while supporting DR4, DR5, and DR6.

5.3 Human Checkpoint Design Across Workflow Stages

Human review is incorporated into the workflow as an architectural component rather than as a single final approval step. The type of intervention required changes across the three processing stages because different errors and decisions require different forms of knowledge and organisational authority. This reflects the human-centred AI principle that oversight should be aligned with the nature of the decision being reviewed rather than treated as an undifferentiated safeguard (Shneiderman, 2020). The resulting checkpoint structure directly operationalises DR3 (Role-Appropriate Review) and DR6 (Review Differentiation).

The first checkpoint, positioned after Agent 1, serves a *corrective* function. Its purpose is to identify extraction, source-attribution, classification, or normalisation errors before they become inputs to the comparison stage. The reviewer therefore requires sufficient domain and system-boundary knowledge to verify whether the structured record faithfully represents the source material. In the GSM-R to FRMCS context, this role is represented by a Requirements Engineer with relevant railway-communication knowledge.

The second checkpoint, positioned after Agent 2, serves a *judgmental* function. At this stage, the workflow may surface conflicts, ambiguities, or differences between requirement sources that cannot be resolved from the documents alone. Resolving such cases may require knowledge of source precedence, agreed deviations, project constraints, or technical dependencies. The reviewer therefore requires broader technical visibility and decision authority, represented in the case context by a System Architect or senior technical lead.

The third checkpoint, positioned after Agent 3, serves an *authorisation* function. Ownership assignments, priorities, and proposed actions can influence organisational commitments and resource allocation, and should therefore not become binding solely because they were generated by the workflow. The reviewer at this stage requires sufficient organisational authority to confirm or modify these assignments. In the case context, this role is represented by a Project Manager or programme-level decision-maker.

The three checkpoints therefore differ not only in their position in the workflow but also in the type of judgement they require. Their purpose is to prevent the staged architecture from merely automating the transfer of uncertainty downstream: corrective review protects the quality of the input record, judgmental review addresses unresolved technical interpretation, and authorisation review retains human control over organisational commitments.

5.4 Output Field Design: The Requirement Record as a Communication Artefact

A central design decision in the workflow artefact concerns the structure of the requirement record produced across the three processing stages. The prototype produces a record containing twenty fields, distributed across the three agents: nine fields produced by Agent 1, six additional fields produced by Agent 2, and five additional fields produced by Agent 3. This progressive enrichment structure is itself a design choice that requires justification. The following subsections explain the logic behind the field selection, the rationale for distributing fields across stages rather than capturing all information at intake, and the governance purpose that each group of fields serves.

In addition to the progressively enriched requirement record, the final workflow stage produces a plain-language design brief derived from the same processed information. The structured requirement table and the design brief therefore represent two complementary outputs of the artefact: one intended for detailed engineering and governance work, and the other intended to communicate the main governance implications to a broader decision-making audience.

5.4.1 Fields as communication design rather than technical implementation

The twenty output fields of the prototype are not arbitrary technical details selected for convenience of implementation. They represent a deliberate communication design: a requirement record must carry sufficient context for every party who will act on it to understand its origin, its validity, its relationship to governing standards, and the action it requires. This framing follows directly from the baseline analysis in Chapter 2, which showed that requirements frequently stall or cause delays not because the technical work is unclear, but because the information needed to act on them is missing, scattered, or never reaches the right people (Heikkilä et al., 2017; Méndez Fernández & Wagner, 2015).

A minimal requirement record containing only an identifier, a textual statement, and an owner would capture the *what* but not the *why*, the *where*, or the *how*. Without source attribution, a reviewer cannot assess the authority of the requirement or trace it back to its origin if it is disputed (Gotel & Finkelstein, 1994). Without a comparison note, a reviewer cannot understand how the requirement relates to the governing standard without performing the comparison manually, inconsistently, and without leaving an auditable record (Ramesh & Jarke, 2001). Without an ownership confidence rating, a reviewer cannot identify which assignments require scrutiny and which can be approved efficiently. The field selection is therefore grounded in a specific analysis of what information is lost when requirements travel through current organisational practice, and what information must be explicitly captured to prevent that loss.

5.4.2 Agent 1 fields: establishing source and grounding

The nine fields produced by Agent 1 address two governance functions: normalisation of the requirement text into a consistent format suitable for downstream processing, and preservation of the source context that gives the requirement its authority and traceability.

The `id` field assigns a unique identifier to each requirement, enabling it to be tracked, referenced, and linked across all three processing stages and across the lifecycle of the project. The `text` field contains the normalised, rewritten version of the requirement in formal SHALL language. This normalisation is necessary because, as the baseline analysis showed, requirements in practice arrive in informal language, as partial statements, or as vague requests that cannot be compared or assigned without first being rendered in a consistent form (Franch et al., 2023; Montgomery et al., 2022).

The `sourcePassage` field preserves the exact original text from the source document before any normalisation. This field exists because normalisation risks subtly changing meaning, and the reviewer must be able to verify that the normalised text faithfully represents what the original document said. It is the audit trail that makes the normalisation step trustworthy rather than opaque (Gotel & Finkelstein, 1994).

The `sourceDoc`, `sourceSection`, and `sourceRef` fields together provide three levels of source attribution: the document, the section within the document, and the original reference code used in that document. These fields address a recurring failure in current practice: requirements arrive detached from their origin, making authority assessment and dispute resolution substantially more difficult (Ramesh & Jarke, 2001). In large projects, locating the source of a requirement without these fields can consume hours of manual searching through document management systems.

The `stakeholderType` field identifies the type of stakeholder who originated the requirement. In multi-stakeholder environments, the identity of the originating stakeholder determines how the requirement should be weighted in a conflict, who should be contacted when clarification is needed, and what kind of authority the requirement carries relative to other sources. Without this field, all requirements appear equally authoritative regardless of whether they originate from a regulatory body, a vendor specification, or an informal stakeholder request (Cheng & Atlee, 2007; Nuseibeh & Easterbrook, 2000).

The `abstractionLevel` field classifies each requirement as high, mid, or low level. This classification serves two governance functions. First, it enables like-for-like comparison: a high-level goal cannot be meaningfully compared directly against a low-level design constraint because they operate at different scopes (Letier & van Lamsweerde, 2004; van Lamsweerde, 2001). Second, it informs reviewer assignment at the first checkpoint: high-level requirements require strategic reviewers, while low-level requirements require technical specialists. Without this field, all requirements would be routed to the same reviewer regardless of their scope.

The `ambiguityFlag` field records whether Agent 1 identified the requirement as vague or unclear. This field is the primary input to Agent 2's clarification function. Without it, ambiguous requirements would be passed to the comparison stage unchanged, producing comparison results built on an unstable foundation. In current practice, ambiguous requirements are frequently resolved silently through assumption rather than through documented clarification (Franch et al., 2023). The ambiguity flag forces that assumption to become visible.

5.4.3 Agent 2 fields: comparison and validation status

The six fields added by Agent 2 address the comparison and clarification function of the workflow. They transform a source-grounded requirement record into a governance-annotated record that tells a reviewer not only what the requirement says, but whether it is clear, whether it aligns with the governing standard, and what action it requires.

The `clarifiedText` field contains the rewritten version of the requirement if it was flagged as ambiguous by Agent 1, or repeats the original text if the requirement was already sufficiently precise. This field creates a stable, unambiguous version of each requirement that Agent 3 can operate on. Without it, vague requirements would propagate through the workflow unchanged, producing ownership assignments and priority classifications based on an unclear statement.

The `comparisonNote` field contains a plain-language summary of how the requirement relates to the reference specification: whether it aligns, conflicts, or covers something the specification does not address. This field makes the comparison rationale explicit and auditable, replacing the informal, undocumented, and inconsistent manual comparison process that characterises current practice (Ozkaya et al., 2023).

The `mismatchFlag` field is a binary indicator of whether a conflict between the requirement and the reference specification was detected. Its function is triage: it allows a reviewer to filter immediately to the items requiring urgent attention without reading every comparison note in full. It also provides the primary metric for evaluating the accuracy of Agent 2's comparison function in the evaluation phase of the thesis.

The `stricterSource` field records which source is more demanding when a conflict is detected: the input requirement, the reference specification, or neither. This field exists because knowing that a conflict exists is insufficient for governance purposes. The reviewer needs to know which direction the resolution must go: whether to tighten the input requirement to meet the standard, or to flag the standard as inappropriately strict for this project context.

The `openQuestion` field captures any unresolved question that Agent 2 identified as requiring human judgment before the requirement can be finalised. This field operationalises a core governance principle: that AI processing should surface what it cannot resolve rather than silently making assumptions (Shneiderman, 2020). In current practice, ambiguities that resist automated resolution are frequently papered over and only surface as problems when they affect downstream implementation decisions.

The `reviewStatus` field provides the highest-level summary of each requirement's governance state after Agent 2 processing: *ok*, *needs review*, or *conflict*. This field is the primary triage tool for the System Architect reviewer at the second checkpoint. It reduces the cognitive load of review by allowing immediate sorting by urgency, and it provides a snapshot of the requirement's governance state at a specific point in time that supports traceability across the project lifecycle (Ramesh & Jarke, 2001).

5.4.4 Agent 3 fields: ownership, priority, and actionable output

The five fields added by Agent 3 transform the validated, comparison-annotated requirement record into an actionable governance document that specifies not only what the requirement says and whether it is valid, but who is responsible for it, how urgently it must be addressed, what evidence will be needed to verify it, and what the next concrete action is.

The `owner` field assigns each requirement to a responsible stakeholder: ERA, Infrastructure Manager, Train Operator, or Vendor. This field is the most fundamental governance output of the workflow. Without explicit ownership, accountability is diffuse and requirements fall through the gaps between organisational roles (Heikkilä et al., 2017). The field converts a requirement from a statement into an assignment.

The `ownerConfidence` field records Agent 3's confidence in the ownership assignment as high, medium, or low. Not all ownership assignments are equally straightforward. Some requirements map clearly to one stakeholder; others are genuinely ambiguous because they sit at the boundary between two organisational responsibilities. The confidence field signals to the Project Manager reviewer where human judgment is most urgently needed, rather than treating all assignments as equally reliable.

The `priority` field classifies each requirement as high, medium, or low priority based on its safety criticality and its review status from Agent 2. Requirements flagged as conflicts in Agent 2 are elevated in priority because unresolved specification conflicts carry governance risk. This field provides the basis for implementation sequencing and resource allocation decisions at the Project Manager checkpoint.

The `evidenceType` field specifies what type of evidence will be needed to verify that the requirement has been met: a test report, a design document, a standard reference, a certification document, or another appropriate artefact. Specifying evidence type at the point of requirement capture, rather than at the point of verification, connects the requirement to its verification method from the beginning of the governance process and prevents the common failure mode in which verification planning begins too late (Unterkalmsteiner et al., 2015).

The `actionNote` field provides a single concrete sentence describing the next action required for this requirement. A requirement register that specifies what is needed without saying what to do next still leaves implementation teams without direction. The action note closes the gap between governance documentation and implementation planning, and makes the output of the workflow directly usable by engineering teams without requiring further interpretation.

5.4.5 Progressive enrichment as a design principle

The distribution of fields across three agents rather than their collection at a single intake stage reflects a deliberate design principle: progressive enrichment. Each stage adds only the governance information that can be reliably generated given the outputs of the preceding stage. Source attribution and normalisation must precede comparison, because comparison against a reference standard is only meaningful when the input has a stable and verified representation. Comparison must precede ownership assignment, because assigning ownership to a requirement whose conflict status is unknown risks committing resources to a governance decision that will need to be revisited. Ownership assignment must precede priority classification, because priority depends partly on the conflict status established by Agent 2.

This sequencing also determines the structure of the human checkpoints. Each checkpoint is positioned at the point where the information available to the workflow changes qualitatively: from source grounding to comparison annotation, and from comparison annotation to ownership and actionable output. The reviewer at each checkpoint can therefore make a genuinely informed judgment rather than being asked to validate outputs whose foundations they cannot yet assess. Table 6 summarises the complete field structure of the requirement record.

Table 6: Output fields by agent: governance purpose and design requirement addressed

Agent	Field	Governance purpose	DR addressed
Agent 1	`id`	Unique identifier enabling cross-stage tracking	DR4
	`text`	Normalised SHALL-format statement for consistent processing	DR1, DR4
	`sourcePassage`	Original text preserved as audit trail for normalisation	DR1, DR4
	`sourceDoc`	Document name enabling authority assessment and dispute resolution	DR1, DR4
	`sourceSection`	Section reference enabling precise location without manual search	DR1, DR4
	`sourceRef`	Original reference code preserving source document numbering	DR1, DR4
	`stakeholderType`	Stakeholder identity determining authority weighting and contact routing	DR2, DR4
	`abstractionLevel`	Scope classification enabling like-for-like comparison and reviewer routing	DR1, DR3
	`ambiguityFlag`	Ambiguity signal preventing vague requirements from entering comparison	DR4, DR5
Agent 2	`clarifiedText`	Stable unambiguous version enabling reliable comparison and assignment	DR4, DR5
	`comparisonNote`	Explicit comparison rationale replacing informal manual checking	DR4, DR7
	`mismatchFlag`	Binary conflict indicator enabling rapid triage and accuracy measurement	DR3, DR4
	`stricterSource`	Resolution direction indicator specifying which source demands more	DR4
	`openQuestion`	Unresolved ambiguity surfaced explicitly rather than assumed away	DR3, DR4
	`reviewStatus`	Highest-level governance state summary enabling checkpoint triage	DR3, DR6, DR7
Agent 3	`owner`	Explicit accountability assignment converting statement to governance commitment	DR4, DR6
	`ownerConfidence`	Confidence signal directing reviewer attention to uncertain assignments	DR3, DR6
	`priority`	Safety-informed sequencing basis for implementation planning	DR4, DR6
	`evidenceType`	Verification method specified at point of capture rather than retrospectively	DR4, DR7
	`actionNote`	Concrete next action closing gap between governance documentation and implementation	DR4, DR7

5.5 Mapping Prototype Elements to Design Requirements

The preceding sections established the design requirements for the workflow artefact and explained the rationale behind the staged architecture and the output field structure. This section makes the relationship between design requirements and prototype elements explicit through a design-logic table. The purpose of this table is to show that every element of the implemented prototype, covering every field, every output, every checkpoint, and the overall staged architecture, can be traced back to at least one of the seven design requirements derived in Section 5.1. This traceability is important to the design-science contribution because it distinguishes the artefact from a collection of implementation choices and makes the rationale behind its elements explicit. (Gregor & Jones, 2007; Hevner et al., 2004).

The design-logic table follows a structure proposed by Gregor and Jones (2007), who argue that a design theory must articulate not only what an artefact does, but why each design decision was made and what evidence would confirm that the decision was correct. Accordingly, each row in Table 7 contains four elements: the prototype element, the governance purpose it serves, the design requirement or requirements it addresses, and the evaluation criterion that will be used to assess whether it fulfils that purpose. The evaluation criteria in the final column correspond directly to the verification criteria established in Table 5 and are operationalised in the evaluation rubric presented in Chapter 7.

The rows of the design-logic table were selected according to one principle: completeness of accountability. Every deliberate design decision in the implemented prototype is included, and no decision is omitted on the grounds that it seemed self-evident. The table covers three categories of decision. The first category is output field design: all twenty fields produced across the three agents are included, because each field represents a choice about what governance information to capture and surface explicitly. The second category is human checkpoint design: all three checkpoints are included as design decisions in their own right, because their placement, function, and reviewer profile were derived from the design requirements rather than assumed as obvious. The third category is architectural design: the staged three-agent pipeline and the parallel output structure of Agent 3 are included because they are deliberate design choices with explicit governance consequences rather than incidental implementation details. Together, these three categories provide a complete account of the deliberate design decisions represented in the implemented prototype. No prototype element was excluded on the grounds that it seemed self-evident, and no element was included that does not correspond to a documented design decision traceable to the design requirements in Table 5.

Two observations about the table are worth making before it is presented. First, most prototype elements address more than one design requirement. This is expected: governance challenges are interconnected, and a field that surfaces implicit information (DR4) typically also supports human checkpoint design (DR3) by giving the reviewer the information needed to exercise the right type of judgment. Second, the table includes not only individual fields but also the staged architecture as a whole and the parallel output structure of Agent 3. These architectural decisions are design choices in their own right and must be accountable to derived requirements in the same way as individual fields.

Table 7: Design-logic table: prototype elements mapped to design requirements and evaluation criteria

Element	Governance purpose	DR	Evaluation criterion
Agent 1: Intake and Grounding			
`sourceDoc`	Requirements arrive detached from origin; authority cannot be assessed without knowing the source	DR1, DR4	Reviewer confirms document name matches actual source
`sourcePassage`	Normalisation risks subtly changing meaning; original text preserved as audit trail	DR1, DR4	Reviewer confirms normalised text faithfully represents original passage
`sourceSection` `sourceRef`	Manual location of source requirement in large documents consumes hours without precise reference	DR1, DR4	Reviewer can locate source requirement directly using section and reference fields
`stakeholderType`	Stakeholder identity determines authority weighting and contact routing; absent in current practice	DR2, DR4	Reviewer confirms stakeholder type matches document origin
`abstractionLevel`	Requirements at different abstraction levels cannot be directly compared; classification enables like-for-like assessment	DR1, DR3	Reviewer confirms classification matches their judgment of requirement scope
`ambiguityFlag`	Ambiguous requirements resolved silently through assumption in current practice; flag forces explicit surfacing	DR4, DR5	Reviewer confirms flagged requirements are genuinely ambiguous and non-flagged ones are sufficiently precise
Human Checkpoint 1 (Requirements Engineer)	Extraction and classification errors propagate invisibly if not intercepted before comparison stage	DR3, DR6	Assess whether the checkpoint enabled identification of extraction or classification errors before downstream processing
Agent 2: Comparison and Clarification			

Continued on next page

Table 7 continued

Element	Governance purpose	DR	Evaluation criterion
`clarifiedText`	Vague requirements produce unreliable comparison results; stable unambiguous version required before comparison	DR4, DR5	Reviewer confirms clarified text preserves original intent and is sufficiently precise for comparison
`comparisonNote`	Manual comparison is undocumented and inconsistent across reviewers; explicit rationale makes it auditable	DR4, DR7	Reviewer confirms note correctly characterises relationship between requirement and reference specification
`mismatchFlag`	Reviewers cannot read every comparison note; binary flag enables rapid triage of conflicts	DR3, DR4	Reviewer confirms flagged requirements are genuine conflicts; non-flagged ones are correctly assessed
`stricterSource`	Knowing a conflict exists is insufficient; resolution direction determines governance action	DR4	Reviewer confirms strictness assessment gives sufficient direction for conflict resolution
`openQuestion`	Ambiguities that resist automated resolution are papered over in current practice; field forces explicit surfacing	DR3, DR4	Reviewer confirms open questions are genuine and answering them would materially affect downstream processing
`reviewStatus`	Highest-level governance state summary enables efficient checkpoint triage	DR3, DR6, DR7	Reviewer confirms status correctly reflects combination of mismatch flag, open questions, and comparison note
Human Checkpoint 2 (System Architect)	Conflict resolution requires contractual and organisational knowledge that AI cannot access from documents	DR3, DR6	Assess whether the checkpoint enabled review of conflict cases and clarification issues before downstream ownership and priority decisions were made
Agent 3: Actionable Output			
`owner`	Without explicit ownership accountability is diffuse; field converts statement into governance commitment	DR4, DR6	Reviewer confirms assignment matches stakeholder authority and project structure

Continued on next page

Table 7 continued

Element	Governance purpose	DR	Evaluation criterion
`ownerConfidence`	Not all assignments equally certain; confidence signal directs reviewer attention where judgment is most needed	DR3, DR6	Assess whether low-confidence assignments are visibly identifiable as cases requiring closer reviewer attention
`priority`	Safety criticality and conflict status must inform implementation sequencing	DR4, DR6	Reviewer confirms priority reflects safety criticality and review status; adjusts where AI assessment is incorrect
`evidenceType`	Verification planning begins too late in current practice; specifying evidence type at capture connects requirement to its verification method	DR4, DR7	Reviewer confirms evidence type is appropriate and realistic for this requirement
`actionNote`	Requirement registers specify what is needed without saying what to do next; field closes gap between governance and implementation	DR4, DR7	Reviewer confirms action note is specific enough to act on and affected parties are correctly identified
Requirement table (CSV)	Machine-readable governance register enabling downstream analysis, audit, and reporting	DR4, DR5, DR7	Assess whether all fields are correctly populated and output is sufficient for downstream design and compliance activities
Design brief (plain text)	Senior decision-makers require a briefing format rather than a structured table; same data, different audience	DR4, DR7	Ask whether brief is understandable and sufficient for a senior stakeholder to make an informed governance decision
Human Checkpoint 3 (Project Manager)	Ownership assignments carry contractual and resource implications; authorisation requires organisational authority	DR3, DR6	Assess whether the checkpoint enabled review and correction of ownership assignments and priorities before outputs were treated as governance-ready
Overall architecture			

Continued on next page

Table 7 continued

Element	Governance purpose	DR	Evaluation criterion
Three-agent staged pipeline	Single-agent approach compounds errors without correction opportunity; staged design creates boundaries for human intervention	DR5	Assess whether errors caught at checkpoints would have propagated if checkpoint had not existed; whether removing any stage would make the next stage's output unreliable
Parallel output branch (Agent 3)	CSV and design brief serve different audiences from the same processed data; parallel generation avoids redundant processing	DR7	Assess whether the structured table and design brief are legible and useful for their intended review audiences under bounded conditions

The design-logic table makes the contribution of this thesis visible as a coherent argument rather than a collection of implementation choices. Each row shows that a specific governance problem, identified in the baseline analysis, motivated a specific design decision, which in turn can be evaluated against a specific criterion. The evaluation chapter uses these criteria as the basis for the assessment rubric applied to the prototype outputs.

5.6 Why Agentic AI Rather Than a Structured Manual Process

The design rationale presented in the preceding sections establishes what the workflow artefact must do and why each design decision was made. A further question requires explicit address: why is agentic AI the appropriate mechanism for implementing this workflow, rather than a well-structured manual process, a template-based tool, or an existing requirements management platform? This question is especially important in the context of this thesis because the workflow retains human reviewers at every checkpoint. If humans must verify all outputs, a sceptical examiner might reasonably ask what automation adds beyond additional complexity.

The answer has three distinct parts, each addressing a different dimension of the governance problem.

5.6.1 Natural language understanding at the intake stage

The first and most fundamental contribution of agentic AI in this workflow concerns the intake stage. A template-based approach requires inputs to already conform to a predefined structure: requirement statements must be entered into prescribed fields, source references must be manually populated, and abstraction levels must be assigned by the person entering the data. This assumption is rarely satisfied in practice. As the baseline analysis in Chapter 2 showed, requirements in multi-stakeholder projects arrive as plain text paragraphs, informal emails, workshop notes, and partially structured documents (Franch et al., 2023; Méndez Fernández & Wagner, 2015). A template cannot read a paragraph of free text and identify that the statement "the system should support operation in tunnels better than today" is a mid-level functional requirement from an infrastructure manager perspective that needs to be normalised into formal SHALL language before it can be governed. Only a system with natural language understanding capability can perform that inference reliably and at scale (Umar & Lano, 2024; Zadenoori et al., 2025).

This is not merely a matter of convenience. In large transition projects such as GSM-R to FRMCS migration, the volume of requirement-related inputs across heterogeneous document types makes manual structured entry economically impractical. The intake bottleneck, specifically converting informal inputs into governed records, is precisely the point at which current practice breaks down and manual effort becomes unsustainable (Heikkilä et al., 2017).

5.6.2 Consistency across large and heterogeneous input sets

The second contribution of agentic AI concerns consistency. A human reviewer applying manual judgment across a large set of requirements may face increasing consistency challenges when the same criteria must be applied repeatedly across many heterogeneous inputs (Montgomery et al., 2022). In a governance context where classification, comparison, and review decisions must remain stable across hundreds of requirements from dozens of documents, this variability can undermine the trustworthiness of the output. An AI system applies the same prompt and the same classification criteria to every requirement regardless of volume, producing an output whose consistency can then be checked and, where necessary, corrected at the checkpoint stage.

This consistency argument is particularly relevant for the comparison function of Agent 2. Manual cross-document comparison, which involves checking each requirement against a reference specification and documenting the relationship, is not only time-consuming but inherently variable: different engineers compare against different versions, apply different interpretive standards, and leave no auditable record of their reasoning (Ozkaya et al., 2023). The AI comparison function produces a documented, consistent comparison note for every requirement, creating an auditable record that manual practice does not.

5.6.3 Scalability across multiple teams and document sources

The third contribution concerns scalability. A shared template-based approach breaks down when many people with different roles, expertise levels, and document formats need to contribute simultaneously. In large migration projects involving regulatory bodies, vendors, and multiple infrastructure operators, forcing convergence on a single input format before processing can begin introduces coordination overhead that delays the governance process (Heikkilä et al., 2017; Wagner et al., 2019). The workflow designed in this thesis accepts heterogeneous inputs from multiple sources without requiring all contributors to conform to a shared format at the point of contribution. Normalisation and structuring are performed by the workflow itself, not imposed on the contributors.

Existing requirements management platforms such as DOORS, Polarion, and Jama Connect provide strong support for storing, versioning, and tracking requirements that have already been structured. However, they do not address the upstream problem of transforming heterogeneous, informal inputs into governed records (Carrillo de Gea et al., 2012; Ozkaya et al., 2023). The workflow described in this thesis is positioned precisely in that upstream space: it is not a replacement for requirements management tools but a preparation layer that makes their use more tractable by reducing the manual effort required to produce structured, source-grounded, comparable inputs.

5.6.4 The role of AI in changing the type of human work

Taken together, these three contributions point to a consistent principle: the value of agentic AI in this workflow is not that it eliminates human judgment from the requirements governance process. Rather, it is that it changes the *type* of human work required. Without AI support, senior engineers and architects spend a disproportionate share of their time on mechanical preparation tasks: locating source documents, converting informal statements into structured records, performing manual comparisons, and chasing accountability assignments across organisational boundaries. These tasks are necessary but they do not require the expertise that makes senior engineers valuable. With AI support, these preparation tasks are absorbed by the workflow, and human reviewers arrive at each checkpoint with a structured, pre-analysed input rather than a collection of raw documents. Their time is redirected toward the tasks that genuinely require human judgment: validating source interpretation, resolving specification conflicts, and authorising governance commitments (Shneiderman, 2020; Tang & Runkler, 2026).

From a Management of Technology perspective, this reallocation of cognitive effort represents a form of process innovation. The workflow does not automate requirements governance; it restructures the division of cognitive labour within it, enabling organisations to apply expert judgment more selectively and more effectively across a larger volume of requirement-related inputs than would be tractable through purely manual means (Hevner et al., 2004).

5.7 Organisational Preconditions for Workflow Deployment

The workflow artefact does not operate independently of the organisational environment in which it is deployed. Its ability to provide meaningful requirements-governance support depends on several conditions that cannot be enforced by the technical workflow itself. These conditions are therefore treated as design assumptions and boundary conditions of the artefact rather than as functions to be automated.

The first precondition concerns the authority and suitability of the reference specification used during comparison. Agent 2 assumes that an appropriate and organisationally accepted baseline exists against which incoming requirements can be assessed. If the selected reference is not relevant to the requirement domain, or if its authority has not been established, technically plausible comparison outputs may nevertheless be misleading. Selecting and approving the reference baseline therefore remains an organisational governance responsibility.

The second precondition concerns the authority of the input material. The workflow can technically process heterogeneous documents, but the governance weight of the resulting requirement record depends on the status of the source from which it was derived. Formal standards, contractual specifications, internal requirements, workshop notes, and informal communication do not carry equivalent authority. The workflow therefore assumes that source type and authority are established before its outputs are treated as governed requirement information.

The third precondition concerns the reviewer roles assigned to the human checkpoints. The three checkpoints require different combinations of domain knowledge, technical judgement, and organisational authority. Corrective review after Agent 1 requires sufficient knowledge to verify source interpretation; judgmental review after Agent 2 requires the ability to assess conflicts and dependencies; and authorisation after Agent 3 requires authority to confirm ownership, priority, and organisational action. The workflow can structure these review points but cannot guarantee that appropriately qualified reviewers are assigned to them.

Together, these preconditions define the organisational circumstances under which the artefact can reasonably be expected to support requirements governance. They also clarify an important boundary of the proposed design: the workflow can structure information and prepare decisions, but it cannot determine which sources are authoritative or replace the organisational authority required to validate and act on its outputs.

5.8 Artefact Availability

The implementation artefacts developed for this thesis, including the n8n workflow configuration and supporting files required to reproduce the prototype structure, are available in the accompanying repository at:

https://github.com/kristikapllani/frmcs-requirements-governance-workflow

The repository contains a README describing the included files and the steps required to inspect or reproduce the workflow. The thesis focuses on the design and evaluation of the artefact; detailed execution instructions are therefore provided in the repository rather than in the research methods chapter.

6 Prototype Demonstration

This chapter demonstrates the prototype workflow on two bounded datasets. The first is a synthetic stakeholder requirements dataset used to show end-to-end workflow execution under controlled conditions. The second uses publicly available material from the UIC FRMCS specification suite and provides the real-document evidence used in the evaluation presented in Chapter 7. The chapter reports the demonstration setup, workflow execution, and resulting outputs; their performance against the design requirements and baseline conditions is assessed separately in Chapter 7.

6.1 Case Datasets and Demonstration Setup

6.1.1 Synthetic dataset

The first dataset consists of a plain-text stakeholder requirements list containing twenty requirement-related statements compiled to reflect the heterogeneous character of requirements as they appear in real transition projects. The statements vary in abstraction level, wording style, and specificity, and several are deliberately formulated in informal or vague language to test the clarification function of Agent 2. The full input document is included in Appendix A. For the synthetic demonstration, Agent 2 compared the structured records against a researcher-constructed bounded reference specification created solely to exercise the comparison, clarification, and triage logic of the workflow under controlled conditions. The synthetic reference specification is reproduced in Appendix A.1.

6.1.2 Real document dataset

The second dataset uses two official, publicly available specification documents published by the International Union of Railways (UIC) as part of the FRMCS V2 specification suite. The input document is the FRMCS Functional Requirements Specification (FRS, v2.1.0, April 2025) (International Union of Railways [UIC], 2025a), from which Section 10.11, Railway Emergency Communication, was extracted as a bounded eight-page section. This section was selected because it is safety-critical, formally numbered, and relevant to the requirements-governance concerns identified during the preliminary practitioner consultation described in Chapter 4. The reference specification is the FRMCS System Requirements Specification (SRS, v2.1.0, April 2025) (UIC, 2025b), from which Section 7.1, On-Board FRMCS, was extracted as the initial comparison baseline for Agent 2.

The selection of requirements for the Agent 2 comparison stage followed an explicit criterion. Of the 85 records extracted by Agent 1 from Section 10.11, those carrying an explicit source reference in the format `10.11.x.x` were prioritised because they could be traced directly to numbered FRS clauses and therefore provided a clearer basis for bounded cross-document comparison. Records derived from explanatory or contextual passages were excluded from this comparison stage because they did not provide an equivalent clause-level reference. From the filtered set, the first ten requirements were selected for processing through Agent 2, reflecting the current scalability constraint of the prototype described in Section 7.3. The selected requirements span the selected REC behaviours of the REC-alert initiation and addressing process (REQ-011 through REQ-017) and the dynamic modification and monitoring requirements (REQ-033 through REQ-035), providing coverage of both the setup and the operational phases of Railway Emergency Communication.

Subsequent expert review identified SRS Chapter 7.1 as an inappropriate reference baseline for the REC-specific requirements extracted from FRS Section 10.11. Chapter 7.1 concerns On-Board FRMCS architecture and interfaces and does not provide a direct REC-specific counterpart to the selected FRS material. The comparison outputs reported in this thesis should therefore be interpreted within this limitation. Its effect on the evaluation is addressed in Chapter 7.3, and its broader implications are discussed in Chapter 8.

Using official specification material nevertheless provides a realistic document-based setting for examining source grounding, structured extraction, and the challenges involved in selecting and comparing authoritative requirement sources across abstraction levels. Table 8 summarises the two bounded datasets used in the demonstration chapter and clarifies the distinct role each one plays in the thesis.

Dataset	Purpose	Input type	Main output used	Role in thesis
Synthetic stakeholder dataset	Show coherent end-to-end workflow behaviour on a controlled bounded example	Informal stakeholder-style requirement list	Agent 1 records, Agent 2 comparison output, Agent 3 actionable output	Demonstration of the complete staged workflow in an easily interpretable setting
Real FRMCS document dataset	Show bounded application to official specification material in a realistic governance context	FRS/SRS specification sections from the FRMCS document suite	Source-grounded requirement records, comparison-annotated outputs, and downstream evaluation evidence	Bounded realism for the prototype and evidence base for Chapter 7

Table 8: Demonstration datasets and their role in the thesis.

6.1.3 Workflow execution

Both datasets were processed through the three-agent n8n workflow described in Chapter 5, using the GPT-4o language model for all agent processing steps. Human review checkpoints were applied between stages, with the researcher acting as reviewer at each checkpoint to validate prototype behaviour. The full workflow configuration is documented in Appendix B. Because the real-document run serves primarily as the bounded evidence base for the evaluation in Chapter 7, this chapter presents the synthetic dataset in fuller process detail. The real-document run is then summarised in Section 6.5.

6.2 Agent 1: Intake and Grounding

Agent 1 was applied to the synthetic stakeholder requirements document. The agent processed the plain-text input and produced a structured CSV output containing all nine fields defined in the artefact design for each extracted requirement: a unique identifier, normalised SHALL-format text, the original source passage, source document name, source section, source reference, stakeholder type, abstraction level, and ambiguity flag.

An early version of the chunking logic in Agent 1 truncated the output after approximately twelve of twenty input statements, because a single GPT-4o call was asked to extract requirements from the full document at once. This was resolved by splitting the input into overlapping 3,000-character chunks and instructing the model to extract inclusively rather than restrictively. After this fix, nineteen of the twenty input statements were extracted. One statement concerning standards alignment rather than a directly expressed system behaviour was not extracted. Because the synthetic dataset was constructed to contain twenty requirement-related statements, this omission is reported as an extraction limitation rather than treated as a correct exclusion. This experience is itself informative for the thesis and is discussed further as a scalability limitation in Chapter 8.

Table 9 presents a representative sample of three records from the synthetic run. The human review checkpoint confirmed that the extracted records faithfully represented the source statements and that abstraction level classifications were appropriate; no corrections were required.

Table 9: Representative sample of Agent 1 output: synthetic dataset

ID	Normalised text	Abstraction	Stakeholder type	Ambiguity
REQ-001	The new onboard radio SHALL keep the active voice call running when the train moves between coverage areas, or SHALL recover quickly enough that the driver does not need to redial manually.	mid	Train Operator	no
REQ-004	The system SHALL support operation in tunnels and difficult coverage areas with fewer dropped sessions.	high	Infrastructure Manager	no
REQ-009	The system SHALL provide logs detailed enough for incident investigation, especially around failed registrations, interrupted calls, and service handovers.	low	Vendor	no

6.3 Agent 2: Comparison and Clarification

Agent 2 received the structured records from Agent 1 together with a reference specification and produced the six additional fields defined in the artefact design: clarified text, comparison note, mismatch flag, stricter source, open question, and review status. For the synthetic demonstration, Agent 2 was run on a bounded subset of twelve records in order to illustrate comparison, clarification, and triage behaviour without reproducing the full Agent-1 output in detail. A representative extract of Agent 2 output from the bounded prototype run is provided in Appendix E. In this section, however, the focus is on the synthetic demonstration run used to illustrate the comparison and clarification behaviour of the workflow.

On the synthetic dataset, Agent 2 was applied to a bounded subset of twelve structured records produced by Agent 1. Of these, six were assessed as *ok*, five were marked *needs review*, and one was identified as a genuine *conflict*. The conflict case concerned the requirement requesting logs detailed enough for incident investigation, while the reference specification required a minimum ninety-day retention period that the input did not specify. Agent 2 flagged this case as a conflict, assigned `stricterSource` as `spec`, and generated a comparison note describing the discrepancy. The remaining records illustrate that the contribution of Agent 2 in the synthetic demonstration was not limited to conflict detection, but also included clarification, triage, and explicit surfacing of cases requiring further human review. The human review checkpoint, applied by the researcher acting as System Architect, confirmed the conflict finding as genuine.

6.4 Agent 3: Actionable Output

Agent 3 received the annotated records from Agent 2 and produced two parallel outputs: a structured requirement table containing all twenty fields, and a plain-text design brief summarising the governance findings for a senior decision-maker audience.

On the synthetic dataset, ownership was distributed across four stakeholder categories, and priority classifications elevated the requirement flagged as a conflict by Agent 2. The design brief identified the logging conflict and recommended that the consortium resolve the retention discrepancy before design work on the logging subsystem begins. The human review checkpoint, applied by the researcher acting as

Project Manager, adjusted one ownership assignment on the grounds that identity handling across national boundaries falls within regulatory rather than vendor responsibility, illustrating the authorisation function of the third checkpoint.

6.5 Demonstration Summary

The demonstration on the synthetic dataset shows that the prototype workflow executes the intended governance functions coherently: heterogeneous plain-text inputs are transformed into structured, source-grounded records; comparison against a reference specification produces explicit alignment, review, and conflict classifications; and actionable outputs are produced with checkpoint oversight at each stage. Three limitations should be acknowledged: the dataset is small, the researcher acted as reviewer at all checkpoints, and an early truncation issue in Agent 1 indicates a current limitation in handling large single-pass extraction tasks.

Having confirmed that the workflow executes coherently on a synthetic dataset, the prototype was subsequently run on the real FRMCS document dataset described in Section 6.1. In that bounded run, Agent 1 extracted 85 source-grounded requirements from Section 10.11 of the FRS. From these, ten formally numbered requirements were selected for Agent 2 comparison under the current scalability constraint of the prototype. Of those ten, seven were assessed as *ok*, two were marked *needs review*, and one was flagged as *conflict*. These classifications are reported here as workflow outputs; their correctness and the subsequent expert interpretation of the conflict case are assessed in Chapter 7. The resulting comparison-annotated records and Agent 3 outputs formed the evidence base for the evaluation presented in Chapter 7 and are reproduced in representative form in Appendices C, D, and E.

7 Evaluation

This chapter evaluates the prototype workflow against the design requirements established in Chapter 5, using the real FRMCS document dataset described in Section 6.1. The evaluation follows the structure proposed by Hevner et al. (2004), who argue that design science artefacts can be validated through multiple methods, including analytical and descriptive evaluation against explicit design criteria, without requiring a full empirical field study at the prototype stage. Given the bounded scope and timeframe of this thesis, the evaluation presented here combines three complementary approaches: an artefact evaluation against the seven design requirements, a comparison against two baseline approaches, and an honest reflection on the limitations of the evaluation itself. The detailed assessment logic, including the bounded reference criteria and the rubric used for design-requirement judgement, is provided in Appendix F.

The evaluation evidence was analysed using both structured indicators and qualitative interpretation. Structured assessment was used for observable properties such as source grounding, field population, traceability, comparison outcomes, and the visibility of conflicts or open questions. Qualitative interpretation was used where assessment depended on contextual judgement, including reviewer interventions, normalisation errors, the practical usefulness of outputs, and findings from expert consultation. This combination reflects the fact that requirements-governance quality cannot be represented by a single quantitative measure: some properties can be verified directly, whereas others require reasoned assessment against the intended governance function of the artefact.

7.1 Artefact Evaluation Against Design Requirements

Table 10 evaluates the prototype against each of the seven design requirements derived in Table 5, using evidence from the real FRMCS run described in Section 6.1. For each design requirement, the table states whether it was met, partially met, or not met, and cites the specific evidence that supports this assessment. The comparison-annotated output generated during the bounded prototype run is reproduced in Appendix E. This appendix illustrates the structure and range of Agent 2 outputs on which the design-requirement assessment is based.

Table 10: Artefact evaluation against design requirements, using the FRMCS demonstration run as evidence

DR	Assessment	Evidence
DR1	Met. Heterogeneous input was ingested without losing source context.	Agent 1 produced 85 structured requirement-related records from FRS Section 10.11. Source-document and section metadata were retained for the extracted records, while clause-level source references were preserved where available. Representative records are reproduced in Appendix C.

Continued on next page

Table 10 continued

DR	Assessment	Evidence
DR2	Met. Downstream impact was made explicit at the point of capture.	Stakeholder type and abstraction level were assigned to every extracted requirement, enabling the first checkpoint to route requirements appropriately before comparison. Both fields are populated across all 85 extracted records, as confirmed by inspection of the Agent 1 CSV output in Appendix C.
DR3	Partially met. Checkpoints elicited role-appropriate judgment, but only one reviewer occupied all three roles.	The researcher, acting in turn as Requirements Engineer, System Architect, and Project Manager, engaged with three distinct review tasks across the workflow: intake verification after Agent 1, conflict interpretation after Agent 2, and ownership and priority authorisation after Agent 3. A concrete ownership correction at Checkpoint 3 was observed in the synthetic demonstration run in Chapter 6.4; in the bounded FRMCS run, the checkpoints were applied but no correction was recorded, and later expert consultation on REQ-035 showed that Checkpoint 1 would have required stronger domain knowledge to catch the normalisation issue during execution. Independent multi-reviewer validation was not possible within the scope of this thesis.

Continued on next page

Table 10 continued

DR	Assessment	Evidence
DR4	Partially met. Implicit governance information was surfaced, but a normalisation error in Agent 1 introduced a spurious conflict flag.	Agent 1 normalised FRS clause 10.11.2.10, which uses `should` and carries an O-Vx (optional) classification in the source document, into a formal `SHALL` statement. This normalisation caused Agent 2 to flag REQ-035 as a conflict between a mandatory input requirement and the reference specification. During expert consultation, an industrial FRMCS standardisation delegate confirmed that the source clause is optional in the FRS, showing that the `should`-to-`SHALL` normalisation altered its normative force (Kiss, 2026a). The expert review also identified the selected SRS Section 7.1 reference as unsuitable for direct REC-specific comparison. The resulting conflict classification therefore cannot be treated as evidence of a genuine specification discrepancy. Instead, the case demonstrates how incorrect source normalisation and reference selection can propagate into plausible-looking downstream outputs. The preserved `sourcePassage` and the first human checkpoint provide mechanisms for detecting such problems, although they were not sufficient to prevent them in this evaluation instance.
DR5	Met, with a scalability caveat. Tasks were decomposed across discrete stages with explicit boundaries.	Agent 1, Agent 2, and Agent 3 each operated only on the validated output of the preceding stage. However, Agent 2 could only process ten requirements per call against the full 85-item Agent 1 output without exceeding model output limits, indicating the current implementation does not yet scale the comparison stage to a full document section.

Continued on next page

Table 10 continued

DR	Assessment	Evidence
DR6	Met. The type of human intervention required at each checkpoint was distinguishable.	The three checkpoints were designed to support different intervention types: corrective review after Agent 1, judgmental review after Agent 2, and authorisation review after Agent 3. In the bounded run, this distinction was visible in the workflow structure and in the ownership correction made at Checkpoint 3. At the same time, the REQ-035 case showed that the corrective checkpoint was not exercised with sufficient domain knowledge during execution. This qualifies the strength of the evidence, but not the conceptual distinctness of the three checkpoint types.
DR7	Met. Outputs were legible to non-technical reviewers.	The design brief produced by Agent 3 summarised the flagged conflict and the associated ownership and priority assignments in plain language. The design brief generated during the bounded prototype run is reproduced in Appendix D. Its structure and language were assessed by the researcher acting in the Project Manager role as sufficient for a senior decision-maker to act on without requiring familiarity with the underlying twenty-field schema.

Five of the seven design requirements are assessed as fully met on the real FRMCS dataset. DR3 and DR4 are assessed as partially met. DR3 is partially met because a single researcher occupying all three reviewer roles introduces a risk of confirmation bias that an independent panel would not carry. DR4 is partially met because a normalisation error in Agent 1 introduced a spurious conflict flag that was only identified through external expert consultation rather than through the Checkpoint 1 corrective review process as designed. Both limitations are discussed further in Chapter 7.3.

7.2 Baseline Comparison

Following the evaluation procedure described in Section 4, the prototype workflow is compared against two baseline conditions: a manual baseline and a single-pass AI baseline.

7.2.1 Manual handling reference condition

The manual condition is used as a reference point rather than as a separately executed empirical benchmark. Under this condition, a Requirements Engineer would need to inspect the selected FRS material, transfer relevant requirements into a tracking structure, identify appropriate reference material, and document the relationship between the sources manually. The baseline analysis in Chapter 2 indicates that these activities are characterised by substantial search, coordination, and cross-document comparison effort (Montgomery et al., 2022; Ozkaya et al., 2023). No execution-time or accuracy measurement was collected for this manual condition, and it is therefore used only to contrast the type of work required rather than to claim a quantitative performance advantage for the prototype.

The REQ-035 case also illustrates a potential advantage of direct source inspection: a human reviewer reading clause 10.11.2.10 would encounter the original `should` wording and O-Vx classification directly. The workflow preserves this information through the `sourcePassage` field, but the evaluation showed that the corresponding corrective checkpoint must be exercised carefully if normalisation errors are to be prevented from propagating downstream.

7.2.2 Single-call AI reference condition

A second reference condition was observed during prototype development when Agent 1 attempted to process the synthetic twenty-item input through a single model call without chunking. Under this configuration, the output was truncated before all input statements had been processed. Introducing chunked extraction allowed nineteen of the twenty synthetic requirement-related statements to be extracted.

This comparison provides evidence for the need to decompose large extraction inputs within the implemented prototype. It should not, however, be interpreted as a controlled comparison between the complete three-agent architecture and a complete single-agent alternative, because only the extraction configuration was varied. The observation therefore supports the use of bounded processing steps but does not by itself demonstrate that a three-agent architecture is globally superior to every possible single-agent implementation.

7.2.3 Prototype workflow

The staged prototype successfully processed the bounded real-document case through the intended sequence of source grounding, comparison, and actionable output generation. Agent 1 extracted 85 requirement-related records from FRS Section 10.11. Under the current comparison-stage constraint, ten formally numbered requirements were subsequently processed by Agent 2 and Agent 3.

For these ten records, Agent 2 produced seven *ok*, two *needs review*, and one *conflict* classification. Subsequent expert consultation showed that the conflict classification associated with REQ-035 could not be treated as a reliable cross-document discrepancy. The source requirement had been normalised from `should` to `SHALL`, while the selected SRS reference section was also later identified as an unsuitable REC-specific comparison baseline. The case therefore demonstrates that plausible downstream outputs can still be produced when upstream normalisation or reference-selection decisions are incorrect.

Taken together, the reference conditions and prototype run provide evidence that staged processing can make intermediate transformations visible and reviewable. They do not establish a quantitative performance advantage over manual handling or over all possible single-agent implementations. The principal benefit observed in the bounded evaluation is instead the explicit separation of source grounding, comparison, and governance decisions, together with opportunities for human intervention between those stages.

7.3 Limitations of the Evaluation

Several validity considerations qualify the evaluation presented in this chapter. Construct validity was supported by deriving the evaluation criteria directly from the design requirements and the requirements-governance problems identified earlier in the study. Internal validity was strengthened by applying the bounded comparison conditions to the same task environment and by using an explicit assessment rubric. External validity remains limited by the use of a single case context and a bounded requirement sample, meaning that the findings support analytical rather than statistical generalisation. Reliability was supported by fixing the dataset boundaries, documenting the workflow configuration and outputs, and using explicit evaluation criteria that can be inspected and reproduced.

This evaluation should be read with four limitations in mind. First, the evaluation was conducted on a bounded section of one specification document (FRS Section 10.11), compared against a reference section (SRS Section 7.1) that was subsequently identified as not containing REC-specific system requirements (Kiss, 2026a). This means the comparison stage was not performed against the most appropriate reference material. The expert review confirmed that the SRS may not contain a direct system-level counterpart to FRS Section 10.11, suggesting that the relevant requirements may reside in a different document altogether (Kiss, 2026b).

Second, the researcher acted as the sole human reviewer at all three checkpoints. This was necessary given the timeframe of the thesis, but it means the evaluation cannot rule out the possibility that the researcher's familiarity with the prototype's intended behaviour influenced the checkpoint judgments. An independent review by a domain expert not involved in building the prototype would substantially strengthen this evaluation; such a review was sought from an industrial FRMCS standardisation delegate and is reported, where available, in Chapter 8.

Third, the comparison stage in the current prototype implementation is limited to ten requirements per call due to language model output constraints. The 85 requirements extracted by Agent 1 were not all carried through Agent 2 and Agent 3 in this evaluation run. This is reported transparently here rather than concealed, and is treated as a concrete scalability finding rather than a flaw to be hidden: it directly motivates the future work discussion in Chapter 8.

Fourth, the normalisation error on REQ-035 identified during expert consultation was not caught at the corrective Checkpoint 1 during the evaluation run, because the expert consultation occurred after the workflow had completed rather than at the checkpoint stage. This means the checkpoint functioned as designed in principle but was not exercised with sufficient domain knowledge in this evaluation instance. An evaluation involving qualified independent reviewers at each checkpoint would provide stronger evidence of the checkpoint structure's effectiveness (Kiss, 2026a).

The selection criterion applied to determine which ten requirements entered the comparison stage is documented in Section 6.1, ensuring that the bounded sample is not arbitrary but reflects a principled prioritisation of formally referenced requirements with direct clause-level traceability.

Taken together, the evaluation provides bounded evidence that the prototype can operationalise most of the derived workflow requirements, while also exposing important limitations in source normalisation, reference selection, reviewer independence, and comparison-stage scalability. The results therefore support the feasibility of the proposed workflow structure within the demonstrated case, but do not establish industrial-scale effectiveness or reliability. The next chapter interprets these findings more broadly and examines their implications for agentic AI, human oversight, and requirements governance.

8 Discussion

This chapter steps back from the artefact evaluation presented in Chapter 7 to interpret its broader implications. It considers the role of agentic AI when human review remains necessary, the function of differentiated human checkpoints, the organisational conditions required for workflow deployment, the limitations exposed by the prototype and its evaluation.

8.1 The Role of AI in Requirements Governance

The evaluation suggests that the central challenge addressed by the artefact is not primarily the automation of requirements engineering tasks, but the governance of requirement information as it moves across heterogeneous documents, levels of abstraction, and organisational roles. From this perspective, the value of the agentic workflow lies in its ability to structure and prepare information for governance activities that would otherwise require repeated manual search, comparison, clarification, and documentation. The role of AI is therefore best understood as supporting the preparation and coordination of requirement information rather than replacing the human judgement through which governance decisions are made.

A reasonable objection to the workflow proposed in this thesis is that, since a qualified human reviewer must validate the output at every checkpoint, the AI stages add complexity without removing the need for expert judgement. This objection misunderstands where the value of the workflow lies. The value is not that AI replaces the judgement exercised by a Requirements Engineer, a System Architect, or a Project Manager. The value is that it changes the *type* of work those reviewers spend their time on.

In current practice, a substantial share of the time spent on requirements governance is consumed not by judgement but by preparatory work: locating relevant passages in large specification sets, tracing requirement statements back to their sources, comparing information across documents, and determining which issues require attention from the responsible decision-maker. The organisational decision about which source or reference baseline is authoritative remains a human governance responsibility, as discussed in Chapter 5.7. None of these tasks require the expertise that makes a senior engineer valuable, yet they routinely consume a disproportionate share of that engineer's time. The evaluation presented in Chapter 7.2 illustrates this directly: locating the appropriate system-level reference for a given functional requirement within the broader FRMCS specification set is itself a non-trivial search task, before any comparative judgement can even begin.

By supporting much of this preparatory layer, the workflow allows the human reviewer to arrive at each checkpoint with a structured, pre-analysed input rather than two raw specification documents. The reviewer's attention is then directed toward the task that genuinely requires their expertise: confirming or correcting the classification at the first checkpoint, resolving the substantive question raised by a flagged conflict at the second, and authorising the resulting governance commitment at the third.

The scale at which this matters in practice was illustrated during a baseline interview with an industrial product manager responsible for GSM-R and FRMCS feature definition, who estimated that defining the requirements for a single new radio product involves on the order of four thousand detailed items to be considered against applicable railway and mission-critical specifications (Lindner, 2026). At this scale, manually tracking which standard applies to which requirement, and why, becomes a task whose difficulty is driven almost entirely by volume rather than by the difficulty of any single judgement. The same interviewee noted that a missed or poorly understood specification requirement can result in a product that fails certification by a Notified Body or fails to attract customer interest, directly consuming research and development resources without a corresponding return (Lindner, 2026). This grounds the governance problem addressed by this thesis not only as an engineering concern but as a Management of Technology concern: the cost of poor requirements governance is ultimately a cost borne by the organisation's resource allocation and product viability, not only by individual engineers' time. The value of the workflow, in short, is not that it eliminates human work. It is that it changes the type of human work required, from repetitive search and comparison at volume toward targeted verification and decision-making.

8.2 Why Human Checkpoints Remain Central

Section 5.3 established that the three checkpoints in the workflow serve qualitatively different functions: corrective, judgmental, and authorisation. This distinction is not only a design choice but a finding that the evaluation in Chapter 7 reinforces. Table 10 shows that DR3 was assessed as only partially met, precisely because a single researcher occupying all three reviewer roles cannot fully demonstrate that each checkpoint elicits the distinct type of judgement it is designed for. This is itself evidence for the argument being made here: if the three checkpoints were functionally interchangeable, this limitation would not matter. That it does matter is confirmation that the checkpoints are not interchangeable.

In practice, the third checkpoint, which serves the authorisation function, is the one most likely to be treated as a formality under time pressure, because ownership is often assumed to be informally understood within a project team. This assumption is itself a governance risk. Informal ownership is not the same as documented accountability: when a dispute arises later, an organisation cannot point to who authorised a given assignment if no such authorisation was ever recorded. The third checkpoint makes that authorisation step explicit within the workflow rather than leaving ownership entirely implicit. In the current prototype, however, the reviewer decision is not yet captured as a separate auditable event; this limitation is addressed in Section 8.4.1.

This view of human review as a structural necessity rather than a fallback is consistent with the position expressed by an industrial FRMCS standardisation delegate during the baseline interview phase of this study, who stated that human review is necessary at each step of a comparison process because even straightforward tasks such as transcription can introduce errors, and because definitional and contextual knowledge, the kind that determines whether a flagged discrepancy is a genuine conflict or an accepted exception, frequently cannot be resolved from the document text alone (Kiss, 2026a). The normalisation error identified on FRS clause 10.11.2.10 during the evaluation run illustrates this point directly: Agent 1 converted an optional `should` clause into a mandatory `SHALL` statement, which caused Agent 2 to flag a conflict that did not exist in the source document. A domain-knowledgeable reviewer at Checkpoint 1, comparing the normalised text against the `sourcePassage` field, would have been in a much stronger position to identify this error before it propagated.

Notably, this position was not shared in identical terms by every practitioner consulted during the baseline phase. An industrial product manager interviewed separately suggested that, while earlier involvement can improve precision, human review could in principle be concentrated at the end of the process, with the AI doing most of the work and the reviewer acting primarily as a final validator (Lindner, 2026). This divergence between two practitioners at the same company, occupying different roles in the requirements governance chain, is itself an informative finding rather than a weakness in the evidence base. It suggests that the appropriate placement and intensity of human review is not a settled question in practice and may vary with the reviewer's role, responsibilities, and priorities. In this case, the standardisation perspective placed greater emphasis on intermediate technical verification, whereas the product-management perspective allowed for greater concentration of review toward the final validation stage. The three-checkpoint design proposed in this thesis does not resolve this tension by assuming one view is correct. Instead, it attempts to balance the concerns represented by both: the corrective and judgmental checkpoints address the standardisation delegate's concern for per-item correctness, while the authorisation checkpoint addresses the product manager's concern for an efficient final point of commitment. The disagreement between these two practitioners therefore strengthens, rather than undermines, the case for a differentiated checkpoint structure over a single undifferentiated review step.

8.3 Organisational Implications for Requirements Governance

The findings of this thesis also have an organisational implication that extends beyond the technical design of the workflow. Introducing AI-assisted requirements processing does not by itself support stronger requirements governance. The workflow depends on organisational arrangements that determine which information is authoritative, who is responsible for validating it, and who has the authority to make decisions based on it. In this sense, the effectiveness of the artefact depends as much on governance structures around requirements data as on the performance of the AI components.

The organisational effect of the proposed workflow can be illustrated by comparing it with the current-state process introduced in Figure 2. Rather than replacing the organisational roles involved in requirements governance, the artefact inserts structured processing and review stages between heterogeneous source documents and downstream requirements-management activities. Figure 4 illustrates this proposed arrangement.

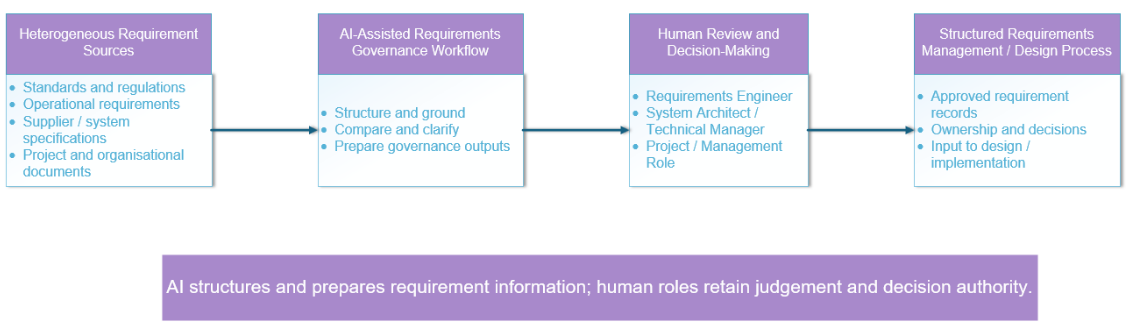

Figure 4: Proposed AI-assisted requirements-governance flow integrating automated processing with role-specific human checkpoints.

Compared with the current-state process, the proposed workflow does not remove the requirements engineer, technical decision-maker, or project-level authority. Instead, it redistributes preparatory work by using the AI stages to structure, compare, and organise requirement information before it reaches the corresponding human decision point. The organisational change is therefore not a transfer of governance authority to AI, but a change in how information is prepared and presented to the roles that already hold that authority.

Three organisational responsibilities are particularly important. First, organisations must establish ownership of the requirement sources and reference baselines used by the workflow. The evaluation showed that technically plausible comparison outputs can still be misleading when an inappropriate reference document is selected. Determining which document constitutes the authoritative baseline is therefore a governance decision rather than an AI task. Second, organisations must assign qualified reviewers with clearly defined decision rights to the different checkpoints. Corrective review, technical judgment, and final authorisation require different forms of expertise and organisational authority. Third, organisations should maintain an auditable record of human interventions, for example through the correction log proposed in Section 8.4.1, so that changes to AI-generated outputs become part of organisational learning rather than remaining isolated reviewer actions.

From a Management of Technology perspective, these findings suggest that the adoption of AI-assisted requirements governance should be treated as an organisational redesign problem rather than only as a technology implementation problem. The workflow changes how information, responsibility, and decision-making move between engineering and management roles. Its practical value therefore depends on aligning the technical workflow with data ownership, reviewer responsibilities, and organisational authority. The managerial recommendation is consequently not simply to introduce AI into existing requirements processes, but to establish the governance structure within which AI-supported outputs can be trusted, challenged, corrected, and authorised.

8.4 Limitations of the Prototype and Its Evaluation

Three limitations of the prototype, beyond those already noted in Chapter 7.3, deserve explicit discussion here.

8.4.1 The absence of a correction log

The current prototype does not record an audit trail of human reviewer decisions. When a reviewer at any checkpoint corrects, overrides, or confirms an AI-generated field, that action is applied to the output but not logged as a discrete event. The structured requirement table therefore does not distinguish between a field that was AI-produced and subsequently human-validated, and one that was simply passed through without correction.

This absence has three consequences. First, accountability is incomplete, since there is no record of which specific decisions were made by a human reviewer rather than generated automatically. Second, the accuracy of the AI stages cannot be measured systematically: the evaluation in Chapter 7.1 can report that one ownership assignment was corrected during the synthetic demonstration run, but it cannot report what proportion of AI-generated assignments require correction across a larger sample, because no mechanism currently captures that information. Third, the workflow cannot currently support iterative improvement, since there is no structured record of where the AI's outputs and the reviewer's judgement diverge.

The normalisation error on FRS clause 10.11.2.10, identified during expert consultation after the workflow had completed, illustrates the practical importance of recording reviewer interventions. A correction log would not itself have detected the incorrect `SHALL` normalisation; detection still depends on the reviewer comparing the generated `text` against the preserved `sourcePassage`. If the discrepancy had been identified at Checkpoint 1, however, a correction log would have captured the original AI-generated value, the reviewer correction, the responsible checkpoint, and the reason for the change. This would create an auditable record of the error and provide evidence for measuring recurring failure patterns across larger runs.

The importance of this gap is further reinforced by a trust concern raised independently during the baseline interview phase. An industry product manager described a negative experience with an internal, non-grounded AI assistant that produced unhelpful or unreliable answers, and argued that the trustworthiness of an AI tool depends heavily on whether its outputs can be checked against the sources and data it was grounded in (Lindner, 2026). The source-grounding fields produced by Agent 1 (`sourceDoc`, `sourceSection`, `sourceRef`, `sourcePassage`) already address this concern in part, by allowing a reviewer to verify any output against its origin. A correction log would extend this same logic of verifiability to the reviewer's own decisions, capturing, for each reviewed field, the requirement identifier, the field name, the AI-generated value, the reviewer-assigned value, the checkpoint stage, and the reviewer role. This is therefore identified as a priority extension for future versions of the prototype.

8.4.2 Risks in formal standardisation contexts

A further limitation concerns the deployment of AI-generated outputs within formal standardisation processes. The industry FRMCS standardisation delegate noted that contributions to bodies such as ETSI or 3GPP are subject to intellectual property provenance requirements that AI-generated text may not satisfy, since the origin and licensing status of the training data underlying a language model's output cannot currently be verified (Kiss, 2026a). This is an important boundary condition for the workflow proposed in this thesis: its governance value is clearest when used as an internal organisational tool for preparing and structuring requirements ahead of human decision-making, rather than as a means of generating content intended for direct submission to a formal standardisation body.

8.4.3 Independent expert validation

Chapter 7.3 noted that the evaluation presented in this thesis relies on the researcher as the sole reviewer at all three checkpoints, and that independent validation by a domain expert not involved in building the prototype would substantially strengthen confidence in the findings. A structured review session was arranged with the industry FRMCS standardisation delegate consulted during the baseline interview phase of this study, using the Agent 2 and Agent 3 output produced from the FRS and SRS dataset described in Section 6.1. Appendix G summarises the practitioner interviews drawn on in this discussion, including both the baseline interviews and the later expert review interaction.

Prior to the scheduled review call, the delegate examined the output materials and identified two specific issues (Kiss, 2026a). First, the delegate confirmed that FRS clause 10.11.2.10 uses `should` and carries an O-Vx classification in the source document, meaning the SHALL normalisation produced by Agent 1 was incorrect and the conflict flag on REQ-035 was a normalisation artefact rather than a genuine cross-document discrepancy. Second, the delegate confirmed that the SRS section used as the comparison baseline (Section 7.1, On-Board FRMCS) does not contain REC-specific system requirements, meaning Agent 2 was comparing FRS Section 10.11 requirements against a reference section that does not correspond to the same functional domain. Both findings are incorporated into the evaluation in Chapter 7 and into the limitations discussion in Chapter 7.3.

These findings qualify rather than invalidate the thesis contribution. They expose two failure modes that are directly relevant to requirements governance: incorrect normalisation can propagate into downstream comparison results, and an inappropriate reference baseline can make otherwise plausible comparison outputs misleading. The workflow contains mechanisms intended to mitigate these risks, but the evaluation also shows their limits. The corrective checkpoint and preserved `sourcePassage` provide an opportunity to identify normalisation errors, while the organisational preconditions in Chapter 5.7 place responsibility for reference-baseline selection outside the AI workflow itself. Neither mechanism prevented the problem in the bounded run. The expert consultation therefore provides independent scrutiny of selected prototype outputs and, importantly, helps establish the conditions under which those outputs should and should not be trusted.

During the scheduled review call, the delegate provided additional findings of direct relevance to the workflow design (Kiss, 2026b). On the question of output format, the delegate assessed the comparison table as partially useful in its current form, and confirmed that the `sourceDoc` and `sourceRef` fields, once identified, address the primary traceability concern by allowing a reviewer to locate the original clause directly. The delegate noted that they would additionally verify the `sourcePassage` field by copying it into the source document and searching for it, because AI tools have been observed to introduce errors even in simple text reproduction tasks. This conservative verification behaviour is consistent with the trust threshold described by this practitioner, who indicated that they would expect approximately 75 to 80 percent accuracy at minimum and ideally 95 to 99 percent over a sustained period before relying on such a tool in a real project context. The delegate observed that mandating the use of AI tools does not produce trust; consistent high-quality output over time does (Kiss, 2026b).

On the dual-output design, the delegate endorsed the parallel production of a structured CSV table and a plain-language design brief, applying the 80-20 Pareto rule: the targeting of different outputs to different audience groups is correct approximately 80 percent of the time, and providing both outputs allows the remaining cases to self-select the appropriate format.

On the question of where in the requirements lifecycle the tool would be most useful, the delegate indicated that both the initial capture phase and the conflict resolution phase during design are viable application points, and that the tool will likely be used wherever it can be applied, with a varying degree of success depending on the context. The delegate connected this to the broader industry context, noting that continuously changing requirements are a fundamental feature of modern product development rather than an edge case, and that any AI-assisted workflow must be capable of handling dynamic requirement environments rather than assuming stable inputs (Kiss, 2026b). The evidence that would be needed to validate the workflow beyond this bounded demonstration is taken up in Chapter 9.5.

9 Conclusion

This thesis set out to investigate how a document-grounded workflow artefact, supported by staged agentic AI under human oversight, can support requirements governance across heterogeneous sources and levels of abstraction in the GSM-R to FRMCS transition. This final chapter answers the four research sub-questions in turn, states the contribution of the thesis, and reflects on what has and has not been established.

9.1 Answering the Research Questions

RQ1 asked how requirements are currently captured, clarified, compared, prioritised, and refined across documents, roles, and abstraction levels, and where the main manual bottlenecks remain. The baseline analysis in Chapter 2, supported by interviews with two practitioners occupying different roles in the requirements governance chain (Kiss, 2026a; Lindner, 2026), showed that current practice remains document-heavy and dependent on manual search, comparison, and informal accountability. The most significant bottlenecks identified were the effort required to locate relevant material across large, multi-hundred-page specification documents, the difficulty of maintaining consistent comparison judgements across a high volume of requirements, and the absence of a structured record connecting a requirement to its source, its review status, and its owner.

RQ2 asked what design requirements a document-grounded workflow artefact should satisfy to support traceable, reviewable, and human-controlled requirements governance. Chapter 5 derived seven design requirements directly from the baseline pain points identified for RQ1, ranging from source-context preservation and explicit surfacing of implicit governance information, to role-differentiated human checkpoints and outputs legible to non-technical reviewers. These design requirements were not asserted but explicitly paired with verification criteria, allowing them to be evaluated rather than merely claimed.

RQ3 asked to what extent a prototype workflow can transform incomplete and heterogeneous requirement inputs into structured, source-grounded, comparable, and actionable outputs supporting clarification, refinement, and human review. The demonstration in Chapter 6 and the evaluation in Chapter 7 showed that the implemented three-agent prototype can transform bounded real-document input into structured, source-grounded, comparable, and reviewable outputs. In the real FRMCS run, Agent 1 extracted and structured 85 requirement-related records with source-grounding metadata from Section 10.11 of the Functional Requirements Specification, and ten formally referenced records were carried forward into the comparison stage. The evaluation also showed that the workflow can surface plausible conflict cases, while making clear that such outputs remain dependent on correct upstream normalisation and effective checkpoint review. Five of the seven design requirements were assessed as fully met on this real dataset, while two were assessed as partially met due to the normalisation issue on REQ-035 and the evaluation's reliance on a single reviewer occupying all three roles.

RQ4 asked how the staged prototype compares with manual handling and a single-call AI reference condition within the bounded evaluation. The comparison in Chapter 7.2 provided qualitative reference evidence rather than a controlled performance benchmark. The manual condition was not separately executed or timed, while the single-call AI condition concerned only the extraction stage on the synthetic input. Within these bounds, the staged prototype provided explicit intermediate outputs and review points, while the observed truncation supported the use of bounded processing steps for larger extraction inputs. The evaluation does not establish a quantitative advantage over manual handling or over all possible single-agent implementations.

9.2 Contribution of the Thesis

The core contribution of this thesis is a set of explicit design requirements for a requirements-governance workflow, together with a documented design rationale explaining why these requirements are needed, how they are operationalised in the artefact, and under what organisational conditions the workflow can provide governance value. (Gregor & Jones, 2007; Hevner et al., 2004). This contribution was demonstrated and evaluated, not merely proposed, using a bounded but real section of the official FRMCS specification suite, and was informed throughout by baseline interviews with industry practitioners directly involved in GSM-R to FRMCS requirements work. The findings further indicate that successful adoption requires organisational arrangements around data authority, reviewer responsibility, and decision rights; the technical workflow can structure these activities, but it cannot substitute for the governance structure that gives them legitimacy.

From a Management of Technology perspective, the thesis reframes requirements governance not only as an engineering activity but as an organisational resource allocation problem: the cost of unresolved or poorly tracked requirements conflicts is ultimately borne in development resources, certification risk, and product viability (Lindner, 2026). The workflow artefact proposed here is one way of converting implicit organisational knowledge about requirements governance into an explicit, reusable, and auditable process, without removing human judgement from the decisions that matter.

9.3 Target Audience and Practical Consequences

The findings of this thesis are relevant to three distinct audiences.

The first audience is requirements governance teams in organisations managing multi-stakeholder migration programmes such as GSM-R to FRMCS. For these teams, the workflow offers a structured approach to absorbing the preparation workload that currently consumes senior engineering time: document search, manual comparison, and informal accountability tracking. If adopted, the practical consequence is a shift in how governance effort is distributed. Engineers spend less time on mechanical retrieval and more time on the judgmental and authorisation decisions that require their domain expertise. If the findings are not acted upon, these teams will continue to rely on manual processes that scale poorly as the volume and complexity of specification documents grows.

The second audience is product managers and technology leaders responsible for resource allocation in infrastructure migration projects. For these stakeholders, the thesis grounds requirements governance as a Management of Technology concern: poor governance leads to missed specification requirements, failed certification by Notified Bodies, and wasted research and development resources (Lindner, 2026). The workflow proposed here offers a mechanism for reducing that risk by making conflicts and ownership gaps visible before design work begins rather than after rework has already started.

The third audience is the academic community working on AI-assisted requirements engineering and design science research. For this audience, the thesis contributes a documented design rationale showing how a multi-agent workflow can be derived from explicit design requirements, evaluated against real specification documents, and validated through expert consultation. The normalisation error identified during the evaluation and the trust threshold articulated by the domain expert (75 to 80 percent minimum accuracy, ideally 95 to 99 percent) provide concrete reference points for future work on AI-assisted governance tools (Kiss, 2026b).

9.4 Practical Recommendations

The findings of this thesis suggest several practical recommendations for organisations considering the adoption of AI-assisted requirements-governance workflows.

First, organisations should establish an authoritative document baseline before introducing automated comparison. The evaluation showed that technically plausible workflow outputs can still be misleading when the selected reference source is inappropriate. Responsibility for defining authoritative source documents should therefore remain an explicit organisational decision.

Second, human checkpoints should be assigned to roles with clearly differentiated responsibilities. Requirements engineers are best positioned to verify source grounding and extraction quality, technical decision-makers to assess inconsistencies and dependencies, and project-level roles to confirm ownership, priority, and organisational action. These checkpoints should not be treated as generic approvals, because the type of judgement required differs across workflow stages.

Third, AI-assisted processing should initially be introduced as a preparation and decision-support layer rather than as an autonomous requirements-management mechanism. The workflow is most useful where it reduces repetitive search, structuring, comparison, and documentation effort while retaining human authority over consequential decisions.

Fourth, organisations should maintain a correction and intervention log when deploying the workflow in practice. Recording when and why reviewers modify AI-generated outputs would provide evidence for accuracy, support organisational learning, and make it possible to identify recurring failure modes across successive workflow runs.

Finally, adoption should be evaluated iteratively using completed or well-understood projects as reference cases. Comparing workflow outputs against known outcomes would allow organisations to assess usefulness, accuracy, and effort reduction before relying on the workflow in active migration programmes.

9.5 Directions for Future Evaluation

The bounded evaluation presented in Chapter 7 provided evidence that five of the seven design requirements were fully met and two were partially met within the demonstrated real-document run. These results are subject to the limitations identified earlier, particularly the bounded sample, the unsuitable reference section used during comparison, and the reliance on the researcher as the checkpoint reviewer. A logical next step would therefore be to evaluate the workflow against three types of evidence at a larger scale.

The first is usefulness: whether Requirements Engineers, System Architects, and Project Managers would actually use the structured output and design brief in their daily work, or whether the tool would join the category of technically capable but practically avoided systems that already characterises much of the existing requirements management tool landscape (Carrillo de Gea et al., 2012; Ozkaya et al., 2023). Given the trust concern discussed in Section 8.4, establishing usefulness would require not only structured interviews and observed review behaviour, but also explicit attention to whether practitioners perceive the outputs as sufficiently source-checkable to be relied upon (Lindner, 2026). The delegate noted during the expert review call that trust in AI tools is built through consistent high-quality output over time rather than through mandated adoption, and identified a practical threshold of 75 to 80 percent minimum accuracy, ideally 95 to 99 percent, as the level at which practitioners would begin to rely on such a tool in a real project context (Kiss, 2026b).

The second is accuracy at scale: what proportion of AI-generated field values are corrected by reviewers across a much larger requirement set than the ten items processed by Agent 2 in this demonstration. This evidence type depends directly on the correction log mechanism discussed in Section 8.4, without which the rate and type of reviewer corrections cannot be measured systematically, as in the present evaluation.

The third is efficiency: whether the workflow measurably reduces the time between receiving raw, heterogeneous requirement inputs and producing a governance-ready output, relative to a directly observed manual benchmark. The manual condition used in the present evaluation was descriptive rather than a timed empirical benchmark, so a future study would need to measure both approaches on the same task and dataset.

The most concrete evaluation design proposed during the expert consultation involves using a completed product as a baseline (Kiss, 2026b). In this approach, the input requirements and process artefacts from a finished product are retained, and the entire requirements governance process is replicated using the AI-assisted workflow with minimal human intervention. The output is then compared against the known baseline on three dimensions: whether the same output can be reproduced, how long the process took, and how much human intervention was required at each stage. The delegate specifically recommended conducting this test on an existing completed product rather than a new one, because an existing product provides a known comparison baseline against which the AI output can be evaluated without ambiguity. This evaluation design directly addresses the three future evidence types identified in this section. Usefulness is assessed by whether practitioners judge the reproduced outputs and intermediate artefacts to be usable in relation to the known baseline and the original completed product. Accuracy is assessed by the proportion of AI outputs that required human correction, and efficiency is assessed by the time and intervention reduction compared to the original manual process (Kiss, 2026b).

Together, these three evidence types and the evaluation design proposed by the domain expert would provide a more complete basis for assessing whether the workflow's central claim holds at scale: that its value lies not in eliminating human work in requirements governance, but in changing its character, from repetitive search and comparison toward targeted verification, conflict resolution, and authorisation.

9.6 Closing Statement

The central claim of this thesis can be stated plainly. The value of the workflow investigated here is not that it eliminates human work in requirements governance. The value is that it changes the type of human work required: from repetitive reading, comparison, and formatting, toward targeted verification, conflict resolution, and decision-making. The evaluation presented in this thesis, bounded as it is, provides initial evidence that this shift is achievable in practice. Whether it holds at the scale of a full migration programme, on the order of thousands of requirements rather than tens, remains an open question for future work, guided by the correction log mechanism, multi-reviewer validation, and efficiency measurement directions identified in the preceding section.

References

Carrillo de Gea, J. M., Nicolás, J., Fernández Alemán, J. L., Toval, A., Ebert, C., & Vizcaíno, A. (2012). Requirements engineering tools: Capabilities, survey and assessment. *Information and Software Technology*, *54*(10), 1142–1157. https://doi.org/10.1016/j.infsof.2012.04.005

Cheng, B. H. C., & Atlee, J. M. (2007). Research directions in requirements engineering. *Future of Software Engineering (FOSE '07)*, 285–303. https://doi.org/10.1109/FOSE.2007.17

Dardenne, A., van Lamsweerde, A., & Fickas, S. (1993). Goal-directed requirements acquisition. *Science of Computer Programming*, *20*(1–2), 3–50. https://doi.org/10.1016/0167-6423(93)90021-G

Finkelstein, A., Gabbay, D., Hunter, A., Kramer, J., & Nuseibeh, B. (1993). Inconsistency handling in multi-perspective specifications. *Proceedings of the 4th European Software Engineering Conference (ESEC '93)*. https://www.researchgate.net/publication/2752629_Inconsistency_Handling_in_Multi-Perspective_Specifications

Franch, X., et al. (2023). The state-of-practice in requirements specification: An extended interview study at 12 companies. *Requirements Engineering*. https://link.springer.com/article/10.1007/s00766-023-00399-7

Gotel, O. C. Z., & Finkelstein, A. C. W. (1994). An analysis of the requirements traceability problem. *Proceedings of the First International Conference on Requirements Engineering*, 94–101. https://doi.org/10.1109/ICRE.1994.292398

Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association for Information Systems*, *8*(5), 312–335. https://www.researchgate.net/publication/201168908_The_Anatomy_of_a_Design_Theory

Gunter, C. A., Gunter, E. L., Jackson, M., & Zave, P. (2000). A reference model for requirements and specifications. *IEEE Software Requirements Engineering Conference*. https://ieeexplore.ieee.org/document/855609

Heikkilä, V. T., Paasivaara, M., Lassenius, C., Damian, D., & Engblom, C. (2017). Managing the requirements flow from strategy to release in large-scale agile development: A case study at ericsson. *Empirical Software Engineering*, *22*, 2892–2936. https://doi.org/10.1007/s10664-016-9491-z

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105. https://www.researchgate.net/publication/201168946_Design_Science_in_Information_Systems_Research

Hunter, A., & Nuseibeh, B. (1998). Managing inconsistent specifications: Reasoning, analysis, and action. *ACM Transactions on Software Engineering and Methodology*, *7*(4), 335–367. https://www.researchgate.net/publication/2635497_Managing_Inconsistent_Specifications_Reasoning_Analysis_and_Action

International Union of Railways. (2025a, April 29). *Future railway mobile communication system: Functional requirements specification* (UIC specification No. FU-7120).

International Union of Railways. (2025b, April 30). *Future railway mobile communication system: System requirements specification* (UIC specification No. AT-7800).

Kirikova, M. (2017). Continuous requirements engineering. *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech '17)*, 1–10. https://doi.org/10.1145/3134302.3134304

Kiss, Z. (2026a, May). Personal interview: Requirements governance in FRMCS standardisation [Nokia FRMCS Standardisation Delegate, ETSI TC RT. Interview conducted 12 May 2026. Transcript available with author.].

Kiss, Z. (2026b, June). Expert review call: AI-assisted requirements governance workflow evaluation [Nokia FRMCS Standardisation Delegate, ETSI TC RT. Call conducted 24 June 2026. Recording and transcript available with author.].

Letier, E., & van Lamsweerde, A. (2004). Reasoning about partial goal satisfaction for requirements and design engineering. *Proceedings of the 12th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 53–62. https://www.researchgate.net/publication/221560194_Reasoning_about_partial_goal_satisfaction_for_requirements_and_design_engineering

Lindner, C. (2026, May). Personal interview: Requirements handling and ai-supported workflow needs at Nokia [Product Manager, Nokia railway communications group. Interview conducted 20 May 2026. Transcript available with author.].

Méndez Fernández, D., & Wagner, S. (2015). Naming the pain in requirements engineering: A design for a global family of surveys and first results from germany. *Information and Software Technology*, *57*, 616–643. https://doi.org/10.1016/j.infsof.2014.05.008

Montgomery, L., Fucci, D., Bouraffa, A., Scholz, L., & Maalej, W. (2022). Empirical research on requirements quality: A systematic mapping study. *Requirements Engineering*, *27*, 183–209. https://doi.org/10.1007/s00766-021-00367-z

Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: A roadmap. *Proceedings of the Conference on the Future of Software Engineering*, 35–46. https://doi.org/10.1145/336512.336523

Ozkaya, M., Kardas, G., & Kose, M. A. (2023). An analysis of the features of requirements engineering tools. *Systems*, *11*(12), 576. https://doi.org/10.3390/systems11120576

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

Ramesh, B., & Jarke, M. (2001). Toward reference models for requirements traceability. *IEEE Transactions on Software Engineering*, *27*(1), 58–93. https://www.researchgate.net/publication/3188133_Toward_Reference_Models_for_Requirements_Traceability

Sadovykh, A., Truscan, D., & Bruneliere, H. (2021). Applying model-based requirements engineering in three large european collaborative projects: An experience report. https://arxiv.org/abs/2105.03119

Shneiderman, B. (2020). Human-centered artificial intelligence: Reliable, safe and trustworthy. *International Journal of Human-Computer Interaction*, *36*(6), 495–504. https://arxiv.org/abs/2002.04087

Sommerville, I., & Ransom, J. (2005). An empirical study of industrial requirements engineering process assessment and improvement. *ACM Transactions on Software Engineering and Methodology*, *14*(1), 85–117. https://www.researchgate.net/publication/220403859_An_Empirical_Study_of_Industrial_Requirements_Engineering_Process_Assessment_and_Improvement

Tang, Y., & Runkler, T. (2026). *LLM-based agentic systems for software engineering: Challenges and opportunities* [Accepted at the GenSE 2026 workshop]. arXiv: 2601.09822. https://doi.org/10.48550/arXiv.2601.09822

Umar, M. A., & Lano, K. (2024). Advances in automated support for requirements engineering: A systematic literature review. *Requirements Engineering*, *29*, 177–207. https://doi.org/10.1007/s00766-023-00411-0

Unterkalmsteiner, M., Gorschek, T., Feldt, R., & Klotins, E. (2015). Assessing requirements engineering and software test alignment—five case studies. *The Journal of Systems and Software*, *109*, 62–77. https://doi.org/10.1016/j.jss.2015.07.018

van Lamsweerde, A. (2001). Goal-oriented requirements engineering: A guided tour. *Proceedings of the 5th IEEE International Symposium on Requirements Engineering*, 249–262. https://doi.org/10.1109/ISRE.2001.948567

Wagner, S., Méndez Fernández, D., Felderer, M., Vetrò, A., Kalinowski, M., Wieringa, R., Pfahl, D., Conte, T., Christiansson, M.-T., Greer, D., Lassenius, C., Männistö, T., Nayebi, M., Oivo, M., Penzenstadler, B., Prikladnicki, R., Ruhe, G., Schekelmann, A., Sen, S., . . . Winkler, D. (2019). Status quo in requirements engineering: A theory and a global family of surveys. *ACM Transactions on Software Engineering and Methodology*, *28*(2), 1–48. https://doi.org/10.1145/3306607

Windisch, R., et al. (2022). Model-based systems engineering in practice: Approach for model-based requirements engineering in agile mechatronic development. *Procedia CIRP*. https://www.sciencedirect.com/science/article/pii/S2212827122007429

Yu, E. S. K. (1997). Towards modelling and reasoning support for early-phase requirements engineering. *Proceedings of the Third IEEE International Symposium on Requirements Engineering*, 226–235. https://doi.org/10.1109/ISRE.1997.566873

Zadenoori, M. A., Dąbrowski, J., Alhoshan, W., Zhao, L., & Ferrari, A. (2025). *Large language models (LLMs) for requirements engineering (RE): A systematic literature review*. arXiv: 2509.11446. https://doi.org/10.48550/arXiv.2509.11446

Zave, P. (1997, February). *Classification of research efforts in requirements engineering* (tech. rep.). AT&T Laboratories—Research. https://www.researchgate.net/publication/220565934_Classification_of_Research_Efforts_in_Requirements_Engineering

Zhao, Z., & Zhou, Z. (2026). A systematic literature review on model-based requirements engineering. *Journal of Systems and Software*, *237*, 112836. https://doi.org/10.1016/j.jss.2026.112836

A Synthetic Stakeholder Input Document

This appendix reproduces the synthetic stakeholder input document used in the prototype demonstration described in Chapter 6. Unlike the real-document dataset used later in the thesis, this input was researcher-constructed for testing purposes. It contains twenty requirement-related statements written in mixed styles and at different levels of abstraction in order to test extraction, normalisation, and classification under imperfect input conditions.

The twenty statements below were intentionally written in varied styles and at different levels of abstraction. Some are relatively clear and specific, while others are incomplete, informal, or ambiguous. This reflects the practical reality that requirement-related inputs often arrive as plain text notes, stakeholder requests, or early requirement statements rather than as formally specified SHALL requirements.

1. The new on-board radio should keep the active voice call running when the train moves between coverage areas, or reconnect quickly enough that the driver does not have to redial manually.
2. We need a way for the driver to know immediately if the radio has dropped off the network, especially during busy operation.
3. The system should make it easier to hand over between national networks without service interruption.
4. The system should support operation in tunnels and difficult coverage areas with fewer dropped sessions.
5. Emergency communication must always take priority over normal operational calls.
6. It should be possible to see from the control side which train started the emergency communication and who is currently included.
7. The control centre wants emergency group calls to be set up fast enough that there is no confusion during an incident.
8. We probably need the addressing logic to work for trains that are crossing borders, but this still has to be checked with the national infrastructure managers.
9. The system should provide logs detailed enough for incident investigation, especially around failed registrations, interrupted calls, and service handovers.
10. If a train changes operational area, the radio should update the relevant communication profile without requiring manual intervention from the driver.
11. There should be one clear place to see which communication requirement came from which source document.
12. We need better ownership of communication-related requirements because right now it is often unclear whether something belongs to the supplier, operator, or infrastructure side.
13. If a requirement is stricter in a regulatory or standards document than in an internal design note, that difference should be made visible early.

14. The on-board unit should support railway emergency communication, including initiation, reception, and participation management.
15. We need a clearer way of marking which requirements are still open questions and which are already accepted.
16. The system should not depend on people remembering informal agreements from earlier meetings when assigning responsibility.
17. Voice quality should remain usable during handover and under degraded radio conditions, though the exact acceptable level still needs to be defined.
18. The product team wants a short management summary in addition to the detailed engineering register, because not everyone will read a full requirement table.
19. When requirements are rewritten into a more formal structure, the original wording should still be preserved somewhere so reviewers can verify that the meaning has not changed.
20. We should be able to compare an incoming requirement against the reference specification and quickly see whether it aligns, conflicts, or needs clarification.

A.1 Synthetic reference specification used for Agent 2 comparison

FRMCS Reference Specification — Migration Requirements

The FRMCS system shall provide continuous network coverage along all main railway lines, including tunnels and remote sections. The system shall support a minimum data throughput of 10 Mbps per train for critical applications. Emergency communication channels shall be established within two seconds of initiation under all network conditions.

The system shall authenticate all users and devices before granting access to communication services. All data transmitted over the network shall be encrypted using current industry standards. The system shall maintain a communication log for a minimum of ninety days, accessible only to authorised personnel.

The FRMCS infrastructure shall be designed to support a minimum of five simultaneous voice channels per train. The system shall provide guaranteed quality of service for safety-critical communications at all times. Network handover between base stations shall occur without any interruption to active communication sessions.

The system shall be capable of operating independently of public telecommunications networks in degraded mode. All onboard equipment shall comply with applicable European railway interoperability regulations. The system shall support remote configuration and firmware updates with zero service interruption. Infrastructure components shall have a minimum operational lifespan of fifteen years under normal railway operating conditions.

B Workflow Configuration and Execution Setup

This appendix describes the technical configuration of the bounded prototype workflow implemented for this thesis. Its purpose is not to document every low-level implementation detail of the n8n environment, but to provide a reproducible overview of how the three-agent workflow was operationalised, which files were used as inputs, how information moved between agents, and what outputs were generated at each stage.

The workflow was implemented in *n8n* as three linked but separately executable agent pipelines. Each agent was triggered manually during prototype execution and processed files stored in Google Drive. Across all three agents, the language model used for prompt-based processing was *GPT-4o*. The implementation was intentionally bounded in scope and designed to support the demonstration and evaluation activities described in Chapters 6 and 7, rather than to function as a production-ready system.

B.1 Overall workflow logic

The prototype follows the staged architecture described in Chapter 5. The workflow is divided into three agent stages:

1. **Agent 1: Intake and Grounding** Reads a bounded source PDF, extracts requirement-relevant statements, assigns initial metadata, and produces a structured CSV output.

2. **Agent 2: Comparison and Clarification** Reads the CSV produced by Agent 1 together with a selected reference specification PDF, compares the extracted requirements against the reference text, and produces a second CSV with comparison-related fields.

3. **Agent 3: Actionable Output** Reads the Agent 2 CSV output, assigns governance-related fields such as ownership and priority, and generates two parallel outputs: a structured requirement table and a plain-language design brief.

The implementation therefore reflects the progressive enrichment logic of the artefact. Each stage adds only the information that can be generated meaningfully from the validated output of the previous stage.

B.2 Input files used in the bounded prototype

Two official FRMCS specification excerpts were used as the main document inputs for the real-document demonstration and evaluation.

- **Source input for Agent 1:** a bounded PDF excerpt from the *FRMCS Functional Requirements Specification (FU-7120)* (UIC, 2025a), specifically Section 10.11 on Railway Emergency Communication. This excerpt covered pages 145 to 152 of the source document and served as the requirement intake document.

- **Reference input for Agent 2:** a bounded PDF excerpt from the *FRMCS System Requirements Specification (AT-7800)* (UIC, 2025b), specifically pages 45 to 70 containing Section 7.1 on On-Board FRMCS. This document served as the comparison baseline during the bounded prototype run.

In addition to these PDFs, Agent 2 used the CSV output generated by Agent 1, and Agent 3 used the CSV output generated by Agent 2.

It should be noted that, as discussed in Chapters 7 and 8, the selected SRS section later proved to be an imperfect reference baseline for Railway Emergency Communication. This is reported transparently in the thesis as a limitation of the bounded evaluation rather than hidden as an implementation detail.

B.3 Agent 1 configuration: intake and grounding

Agent 1 was implemented as a pipeline that begins with a manual trigger and then downloads the selected source PDF from Google Drive. The PDF text is extracted automatically and passed into a code node that performs character-based chunking before prompt-based extraction is executed.

The core implementation steps of Agent 1 are as follows:

1. manual workflow trigger,
2. document metadata assignment,
3. source PDF download from Google Drive,
4. PDF text extraction,
5. character-based chunking,
6. prompt-based extraction of requirement-relevant statements,
7. aggregation and parsing of extracted JSON arrays,
8. sequential renumbering of requirement identifiers,
9. SHALL-format normalisation,
10. CSV conversion and upload to Google Drive.

The chunking strategy used in Agent 1 was:

- **chunk size:** 3000 characters
- **overlap:** 200 characters

This strategy was introduced to avoid truncation and incomplete extraction when processing a longer document section in a single model call. Each extracted requirement record was required to contain nine fields:

- `id`
- `text`
- `sourcePassage`
- `sourceDoc`
- `sourceSection`

- `sourceRef`
- `stakeholderType`
- `abstractionLevel`
- `ambiguityFlag`

In the bounded real-document run, the metadata were fixed as follows:

- **Document name:** FRMCS Functional Requirements Specification (FU-7120)
- **Stakeholder type:** Standards body
- **Document section:** Section 10.11, Railway Emergency Communication

After extraction, the results from all chunks were merged, invalid or empty items were removed, requirement identifiers were renumbered sequentially, and a second model call rewrote the extracted requirement text into standard SHALL format while preserving the remaining fields unchanged.

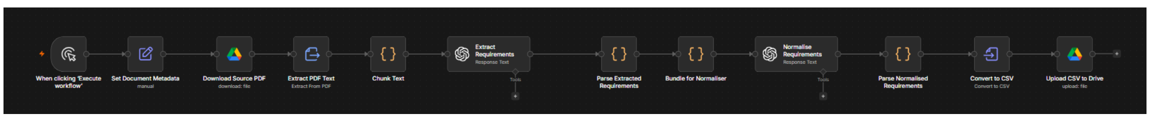

Figure 5: n8n implementation of Agent 1: intake and grounding.

B.4 Agent 2 configuration: comparison and clarification

Agent 2 was implemented as a second manually triggered workflow. It begins by downloading two inputs from Google Drive:

- the CSV output produced by Agent 1
- the bounded reference PDF excerpt from the SRS

Both files are parsed, merged, and prepared in a code node before being passed to GPT-4o for comparison.

The core implementation steps of Agent 2 are as follows:

1. manual workflow trigger,
2. download of Agent 1 CSV output,
3. download of SRS reference PDF,
4. file parsing and text extraction,
5. merge of requirement records and reference text,
6. filtering of requirements for comparison,

7. construction of comparison prompt,
8. GPT-4o comparison and clarification,
9. JSON parsing,
10. CSV conversion and upload to Google Drive.

A deliberate selection rule was applied before comparison. The workflow first filtered requirements to those with an explicit `sourceRef` containing `10.11`, reflecting formally numbered Railway Emergency Communication requirements in the FRS. From these, the workflow selected the first ten items for the comparison stage. This limit was imposed by current output-size constraints in the prototype and is discussed in the thesis as a scalability limitation rather than as a theoretical design limit.

The reference specification text was also trimmed before prompting. After extraction from the PDF, the relevant substantive content was located by searching for expected section markers and then truncated to a bounded text window for model input.

For each selected requirement, Agent 2 returned six additional fields:

- `clarifiedText`
- `comparisonNote`
- `mismatchFlag`
- `stricterSource`
- `openQuestion`
- `reviewStatus`

The resulting output was stored as a CSV file in Google Drive under the name `agent2_clarified`.

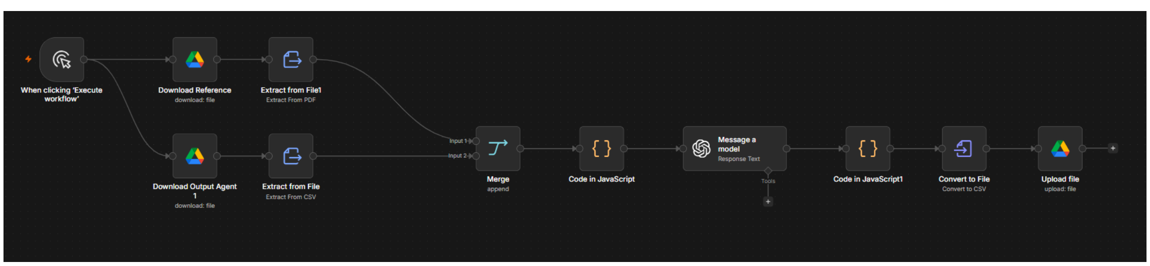

Figure 6: n8n implementation of Agent 2: comparison and clarification.

B.5 Agent 3 configuration: actionable output generation

Agent 3 was implemented as the final manually triggered workflow. It downloads the CSV output from Agent 2, parses the requirement records, and then executes two parallel branches:

1. one branch generates the final structured requirement table

2. the other branch generates the plain-language design brief

The core implementation steps of Agent 3 are as follows:

1. manual workflow trigger,
2. download of Agent 2 CSV output,
3. extraction and cleaning of structured requirement fields,
4. aggregation into one JSON array,
5. GPT-4o assignment of governance-related fields,
6. parsing of returned JSON,
7. CSV conversion and upload of the structured requirement table,
8. parallel GPT-4o generation of a design brief,
9. text parsing and upload of the design brief as a plain-text file.

The structured requirement branch preserved all previous fields and added five new fields:

- `owner`
- `evidenceType`
- `ownerConfidence`
- `priority`
- `actionNote`

The owner assignment prompt used a bounded set of stakeholder categories:

- ERA
- Infrastructure Manager
- Train Operator
- Vendor

The final structured CSV was uploaded under the name `agent3_requirement_table.csv`. In parallel, the design brief branch generated a shorter plain-language summary intended for management or review audiences, and uploaded it under the name `agent3_design_brief.txt`.

This parallel-output structure reflects the design principle that the same processed governance information may need to be presented differently depending on the intended audience.

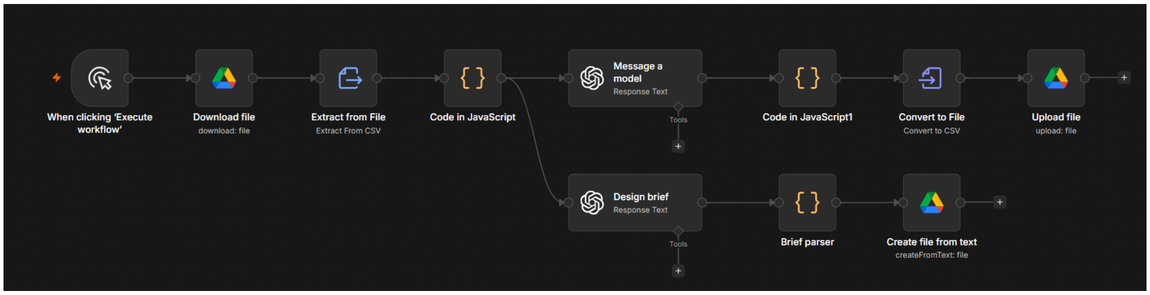

Figure 7: n8n implementation of Agent 3: actionable output generation.

B.6 Human review in the prototype execution

Although the implementation was technically automated between file-processing and model-calling steps, the prototype remained explicitly human-controlled. In the bounded thesis execution, the researcher manually reviewed outputs between stages and acted in the three checkpoint roles defined in the artefact design:

- Requirements Engineer after Agent 1
- System Architect after Agent 2
- Project Manager after Agent 3

This means the workflow as implemented was not a fully autonomous pipeline. It was a staged decision-support workflow in which automation prepared outputs for structured human review before the next stage proceeded.

B.7 Prototype outputs generated in the thesis

Across the bounded demonstration and evaluation runs, the workflow produced four main artefacts:

1. Agent 1 CSV output containing the initial structured requirement records
2. Agent 2 CSV output containing clarified and comparison-annotated requirement records
3. Agent 3 CSV output containing the full structured requirement table
4. Agent 3 plain-text design brief for review and decision support

These outputs correspond directly to the artefact logic described in Chapter 5 and the evaluation logic described in Chapter 7. The appendix does not reproduce the full contents of those outputs, since they are provided separately in the relevant appendices, but it documents how they were generated and how they relate to one another within the prototype workflow.

B.8 Implementation boundaries

This final section clarifies what the current workflow configuration does *not* yet provide.

First, the current prototype does not maintain a correction log that records which AI-generated field values were accepted, corrected, or overridden by human reviewers. Second, the current Agent 2 implementation is bounded to a subset of ten requirements per comparison run due to model output-size limitations. Third, the quality of comparison depends not only on the model output but also on the appropriateness of the selected reference specification. These boundaries are not hidden implementation defects, but part of the transparent bounded scope of the prototype and are discussed explicitly in Chapters 7 and 8.

Taken together, the configuration described in this appendix shows that the thesis prototype was implemented as a real, executable workflow rather than as a purely conceptual diagram. At the same time, it remains a bounded research prototype whose purpose is to demonstrate and evaluate workflow logic, not to claim production readiness.

C Representative Extract of Agent 1 Output

This appendix presents a representative extract of the structured output generated by Agent 1 in the bounded real-document prototype run described in Chapter 6. Agent 1 processed the selected excerpt of the *FRMCS Functional Requirements Specification (FU-7120)* (UIC, 2025a), specifically Section 10.11 on Railway Emergency Communication, and transformed requirement-related statements into a structured CSV format suitable for downstream comparison and review.

The full Agent 1 output for the bounded real-document run contained 85 extracted records. To keep the appendix readable, only a representative extract is reproduced here. The selected rows illustrate both kinds of records produced by Agent 1: requirement-related statements derived from explanatory passages in the section introduction, and formally numbered requirements extracted from clause 10.11.2.

C.1 Agent 1 field structure

Each Agent 1 record contained nine fields. Table 11 summarises the meaning of each field.

Table 11: Field structure of Agent 1 output

Field	Purpose
`id`	Sequential identifier assigned to each extracted record for downstream tracking across agents.
`text`	Normalised requirement statement in SHALL format.
`sourcePassage`	Original wording preserved from the source document for traceability and review.
`sourceDoc`	Name of the source document from which the record was extracted.
`sourceSection`	Section heading within the source document.
`sourceRef`	Original clause or requirement reference from the source document, where available.
`stakeholderType`	Type of stakeholder associated with the source document.
`abstractionLevel`	Classification of the requirement as high, mid, or low level.
`ambiguityFlag`	Indicator showing whether the extracted statement was judged to be vague or unclear.

C.2 Representative extract

Table 12 shows a representative extract of the Agent 1 CSV output. The table includes examples from the introductory explanatory material as well as formally numbered Railway Emergency Communication requirements. For readability, only selected fields are shown here.

Table 12: Representative extract of Agent 1 output

ID	Source ref	Normalised text	Level	Ambiguity
REQ-001	N/A	The REC FRMCS application and the common functions SHALL be configurable to fulfil local operational rules.	mid	no
REQ-003	N/A	The list of users addressed by the REC SHALL be determined and continuously adapted based on predefined addressing rules with a safety margin applied.	mid	no
REQ-005	N/A	The infrastructure manager SHALL define the size and shape of the initiating and addressed areas based on criteria such as line speed, track characteristics, user reaction time, and location information accuracy.	high	no
REQ-010	N/A	An external system SHALL be able to initiate a REC.	high	no
REQ-011	10.11.2.1	A controller SHALL be able to initiate a REC-alert.	low	no
REQ-012	10.11.2.2	A driver SHALL be able to initiate a REC-alert.	low	no
REQ-014	10.11.2.4a	A controller SHALL be able to initiate a REC-alert by selecting several addressed areas.	low	no
REQ-015	10.11.2.4i	When a controller initiates a REC-alert by selecting several addressed areas, the resulting addressed area SHALL be the union of the selected addressed areas.	low	no
REQ-033	10.11.2.9i	The system SHALL allow changing additional information during an ongoing REC-alert that is permanently presented to the recipients.	mid	no
REQ-034	10.11.2.9ii	The system SHALL allow adding additional information during an ongoing REC-alert if no additional information was inserted at setup.	mid	no
REQ-035	10.11.2.10	The system SHALL allow a controller to change the addressed area(s) during an ongoing REC-alert.	mid	no
REQ-036	10.11.2.11	The FRMCS system SHALL monitor user eligibility to participate in the REC-alert and adapt the recipients accordingly.	high	no

Notably, REQ-035 is included in this extract because it later became the key example of a normalisation error identified through expert consultation: the Agent 1 output rendered the source clause as a SHALL statement, while the original FRS clause used `should` and carried an optional classification. This makes the extract directly relevant to the evaluation discussion in Chapters 7 and 8.

This representative extract shows the twofold function of Agent 1 within the bounded prototype. First, it converts requirement-related material from the source document into a consistent SHALL-based representation. Second, it preserves the source grounding information needed for later comparison and review. In the full workflow, this structured output becomes the direct input to Agent 2, which adds clarification and comparison-related fields in the next stage of the workflow.

D Agent 3 Design Brief and Representative Output Extract

This appendix presents the management-facing output produced by Agent 3 in the bounded prototype run described in Chapter 6. Agent 3 received the comparison-annotated requirement records from Agent 2 and generated two parallel outputs from the same bounded requirement set:

1. a structured requirement table for downstream engineering and governance use
2. a plain-language design brief intended for management and review audiences

To keep the appendix readable, this appendix reproduces the full design brief and a short representative extract from the structured requirement table. Together, these two artefacts illustrate the dual-output logic of Agent 3: the same processed requirement set is presented in both a structured and a narrative format, depending on the needs of the intended audience. The outputs reproduced in this appendix are presented as they were generated during the bounded prototype run. They should therefore be read as *prototype outputs*, not as the final validated interpretation of the specification pair. As discussed in Chapters 7 and 8, subsequent expert consultation showed that the conflict associated with REQ-035 was caused by an upstream normalisation error in Agent 1 rather than by a genuine cross-document discrepancy, and also showed that the selected SRS reference section was not the most appropriate REC-specific comparison baseline.

D.1 Representative extract of Agent 3 requirement table

Table 13 presents a short extract from the Agent 3 requirement table. The selected rows show the range of governance states present in the bounded run, including requirements assessed as *ok*, *needs review*, and *conflict*.

Table 13: Representative extract of Agent 3 output table

ID	Owner	Owner confidence	Priority	Review status	Action note
REQ-011	Vendor	medium	medium	ok	Provide a certification document verifying REC-alert functionality for controllers.
REQ-012	Train Operator	high	high	ok	Conduct tests and produce a report verifying REC-alert initiation by drivers.
REQ-033	ERA	low	high	needs review	Provide standards references addressing interoperability for REC-alert updates.
REQ-034	ERA	low	medium	needs review	Investigate whether standards require a modular approach to REC-alert information updates.
REQ-035	ERA	medium	high	conflict	Review whether changing addressed areas during an ongoing alert should remain optional or become mandatory.

The extract shows how Agent 3 translated the comparison output into governance-oriented assignments at the time of execution. Some of these assignments, most notably the REQ-035 conflict case, were later reconsidered during expert validation. The appendix should therefore be read as evidence of workflow behaviour rather than as a final normative interpretation of the specification.

D.2 Full design brief produced by Agent 3

The text below reproduces the plain-language design brief generated by Agent 3 from the bounded requirement set. Minor formatting edits have been made for readability, but the content remains unchanged.

One presentational inconsistency in the brief should be noted. The stakeholder designations used in the design brief, such as Railway Operators, System Developers, and Safety and Compliance Teams, do not correspond one-to-one to the bounded owner categories applied in the structured requirement table (ERA, Infrastructure Manager, Train Operator, and Vendor). This occurred because the bounded owner list was enforced in the prompt of the table-generation branch of Agent 3 but not in the parallel brief-generation branch. The brief is nevertheless reproduced as generated, both to preserve the fidelity of the prototype output and because the discrepancy is itself informative: it shows that output constraints must be applied consistently across parallel generation branches if the two audience-specific outputs are to remain aligned. This observation is consistent with the broader finding, discussed in Chapters 7 and 8, that plausible outputs can diverge from intended constraints when those constraints are not enforced at every generation step.

Feature and Design Brief: GSM-R to FRMCS Migration

Key Functional Areas The requirements primarily focus on the initiation, management, and modification of REC-alerts within railway communication systems as part of migrating from GSM-R to FRMCS. Key functional areas include:

1. **Alert Initiation:** Controllers and drivers can initiate REC-alerts, with various options for specifying the alert scope, such as addressed or initiating areas and train functional identity.

2. **Alert Management:** Controllers can select multiple areas for alerts, creating unions of these areas, and are able to amend ongoing alerts by adding or changing information.

Critical Requirements

1. **Initiation Capabilities (High Priority):** Both controllers and drivers must be able to initiate REC-alerts effectively. Controllers have extended capabilities to specify multiple addressed areas.

2. **Information Modification (High Priority):** It is essential that systems allow the modification of ongoing REC-alerts, including changing areas and adding new information, catering to dynamic operational scenarios.

Conflicts and Mismatches

1. **Interoperability Concerns (REQ-033):** Clarification is needed on potential interoperability issues when modifying information during an alert.

2. **Modularity Requirement (REQ-034):** The requirement for adding information during alerts lacks clarity on whether a modular approach is necessary.

3. **Addressed Area Change (REQ-035 Conflict):** The requirement for changing addressed areas during active alerts is conflicting. The source treats this as optional, while the current requirement presents it as mandatory. Resolution is needed to align with operational priorities.

Stakeholder Owners and Responsibilities

1. **Railway Operators:** Ensure all staff are trained to initiate and manage REC-alerts effectively.
2. **System Developers:** Implement systems that meet the defined requirements, particularly focusing on dynamic alert management capabilities.
3. **Safety and Compliance Teams:** Review interoperability and modularity issues, ensuring that changes in alert information comply with safety regulations.
4. **Project Management:** Coordinate between stakeholders to resolve any conflicts and ensure timely and efficient handling of all requirements.

Recommended Next Steps To move forward effectively, resolve conflicts and open questions by conducting a stakeholder workshop. This will ensure alignment and clarity, especially regarding interoperability and modularity concerns. Following this, collaborate with development teams to prioritize high-criticality requirements, ensuring that the system design enhances alert management efficacy in the FRMCS framework.

D.3 Why both outputs matter

This appendix illustrates why Agent 3 was designed to produce both a structured table and a plain-language summary. The structured table supports detailed governance work, review, and downstream tracking, while the design brief supports communication with managers and decision-makers who need the key issues, responsibilities, and next actions in a shorter and more accessible form.

E Representative Extract of Agent 2 Output

This appendix presents a representative extract of the structured output generated by Agent 2 in the bounded prototype run described in Chapter 6. Agent 2 received the CSV produced by Agent 1 together with the selected reference specification excerpt and added comparison- and clarification-related fields to each structured requirement record.

Only a representative extract is reproduced here for readability. The appendix is intended to illustrate the structure and range of Agent 2 outputs, not to reproduce the full CSV generated during the bounded prototype run.

The outputs reproduced in this appendix are presented as they were generated during the bounded prototype run. They should therefore be read as *prototype outputs*, not as the final validated interpretation of the specification pair. As discussed in Chapters 7 and 8, later expert consultation showed that the REQ-035 conflict was caused by an upstream normalisation issue in Agent 1 rather than by a genuine cross-document discrepancy, and also showed that the selected SRS reference section was not the most appropriate REC-specific comparison baseline.

E.1 Agent 2 field structure

Each Agent 2 record added six fields to the structured output inherited from Agent 1. Table 14 summarises the meaning of these fields.

Table 14: Field structure of Agent 2 output

Field	**Purpose**
`clarifiedText`	Clarified version of the requirement text, or the original text if already sufficiently precise.
`comparisonNote`	Short explanation of how the requirement relates to the selected reference specification.
`mismatchFlag`	Indicates whether the workflow detected a conflict or inconsistency.
`stricterSource`	Indicates whether the input requirement, the reference specification, or neither was assessed as more demanding.
`openQuestion`	Records any issue that the workflow could not resolve automatically and that requires human judgement.
`reviewStatus`	High-level governance state after comparison, such as `ok`, `needs review`, or `conflict`.

E.2 Representative extract

Table 15 presents a representative extract of Agent 2 output. The selected rows show the range of states produced during the bounded run, including aligned cases, items requiring review, and the conflict case later revisited during expert validation.

Table 15: Representative extract of Agent 2 output

ID	Clarified text	Comparison note	Mismatch	Status	Open question
REQ-011	A controller SHALL be able to initiate a REC-alert.	Matches with the source passage in the spec.	no	ok	N/A
REQ-014	A controller SHALL be able to initiate a REC-alert by selecting several addressed areas.	Matches with the source passage in the spec.	no	ok	N/A
REQ-017	A controller SHALL be able to initiate a REC-alert by providing a specific train functional identity.	Matches with the source passage in the spec.	no	ok	N/A
REQ-033	The system SHALL allow changing additional information during an ongoing REC-alert that is permanently presented to the recipients.	Matches with source but lacks specific interoperability context.	no	needs review	Are there interoperability concerns for changing information during alerts?
REQ-034	The system SHALL allow adding additional information during an ongoing REC-alert if no additional information was inserted at setup.	Matches with source but lacks specific modularity context.	no	needs review	Is there a modularity requirement for adding information during alerts?
REQ-035	The system SHALL allow a controller to change the addressed area(s) during an ongoing REC-alert.	Contradicts spec as it is optional in source.	yes	conflict	Should changing addressed areas be mandatory?

REQ-035 is included in this extract because it later became the key example used in the evaluation and discussion chapters to show how an upstream normalisation error can propagate into a plausible but ultimately misleading conflict classification.

The extract shows how Agent 2 transformed the source-grounded output of Agent 1 into a comparison-annotated governance record. The extract also illustrates the three main states used in the workflow at this stage: aligned items, requirements needing human review, and flagged conflict cases. In the full workflow, this enriched output became the direct input to Agent 3, which added ownership, priority, evidence type, and action-oriented fields for governance use.

F Evaluation Rubric and Bounded Reference Criteria

This appendix explains how the bounded evaluation in Chapter 7 was structured. Its purpose is to make explicit what was being judged, how the bounded sample was defined, and how the prototype was assessed against the seven design requirements derived in Chapter 5.

The evaluation in this thesis was not designed as a statistical performance benchmark over a large industrial dataset. Rather, it was designed as a bounded design-science evaluation of whether the prototype workflow behaved in a way that was consistent with the artefact requirements established earlier in the thesis. For that reason, the gold-set logic and evaluation rubric were used primarily to support transparent, criterion-based judgement rather than large-scale numerical scoring.

F.1 Purpose of the bounded reference criteria

The gold-set in this thesis refers to the expected output structure and expected comparison behaviour for the bounded prototype run. It was used to determine whether the workflow produced outputs that were sufficiently correct, traceable, reviewable, and useful for governance purposes.

The gold-set did not take the form of a fully independent, pre-labelled benchmark dataset covering hundreds of requirements. Instead, it was defined through a bounded set of expectations tied to the selected prototype run:

- what fields should be present at each workflow stage
- what source grounding information should be preserved
- what kinds of comparison outcomes should be surfaced
- what kinds of outputs should be produced for review and action
- what kinds of evidence would count as meeting, partially meeting, or not meeting each design requirement

This bounded gold-set logic is consistent with the role of evaluation in design science research, where the key question is whether the artefact behaves in a way that satisfies its stated objectives under a realistic but limited use case.

F.2 Bounded evaluation sample

The bounded evaluation used the real-document prototype run described in Chapter 6. The sample was defined as follows:

- Agent 1 processed the selected FRS excerpt and produced 85 structured requirement records.
- Agent 2 did not process all 85 records in one comparison run. Instead, the workflow applied an explicit selection rule.
- Requirements with an explicit `sourceRef` containing `10.11` were prioritised.
- From this filtered set, the first ten requirements were selected for Agent 2 comparison.

This means that the evaluation of comparison, mismatch detection, and downstream output generation was performed on a bounded comparison subset rather than on the entire Agent 1 extraction output. This limitation is openly acknowledged in Chapter 7 and is part of the stated scope of the prototype rather than an unstated omission.

F.3 Evaluation evidence sources

The bounded evaluation drew on four sources of evidence:

1. **Workflow outputs** The CSV and text outputs produced by Agent 1, Agent 2, and Agent 3.
2. **Researcher checkpoint review** The researcher's own review of outputs at the three workflow checkpoints, acting in the bounded roles of Requirements Engineer, System Architect, and Project Manager.
3. **Cross-stage consistency checks** Inspection of whether later-stage outputs remained consistent with earlier-stage records and source-grounding fields.
4. **Expert consultation** Later validation input from an industrial FRMCS standardisation delegate, used to reassess the REQ-035 case and the appropriateness of the selected reference section.

Together, these sources were used to assess whether the workflow met its intended design requirements under the bounded case conditions.

F.4 Design-requirement assessment logic

The seven design requirements were not scored numerically. Instead, each one was assessed using a three-level judgement:

- **Met** The relevant workflow capability was clearly demonstrated in the bounded run, with no major contradiction between intended function and observed behaviour.
- **Partially met** The intended capability was visible in the prototype output, but the bounded run also revealed a limitation, inconsistency, or untested condition that prevented a full positive judgement.
- **Not met** The bounded run failed to demonstrate the intended capability, or the output clearly contradicted the design requirement.

Table 16 summarises this assessment logic.

Table 16: Assessment logic for design-requirement evaluation

Assessment	Interpretation
Met	The workflow demonstrated the intended capability clearly and produced output consistent with the relevant design requirement.
Partially met	The workflow demonstrated the intended capability in part, but a limitation, ambiguity, or evaluation constraint prevented a full positive judgement.
Not met	The workflow did not demonstrate the intended capability, or its output clearly failed to satisfy the requirement.

F.5 Rubric used for the seven design requirements

Table 17 shows the practical rubric used to judge the seven design requirements in the bounded evaluation.

Table 17: Practical rubric for evaluating the seven design requirements

DR	What was checked	Evidence of success
DR1	Whether heterogeneous input could be ingested without losing source context	Presence of correct source document, section, and reference information in Agent 1 output
DR2	Whether downstream governance impact was made explicit at the point of capture	Presence of stakeholder type, abstraction level, and related intake metadata in Agent 1 output
DR3	Whether checkpoints elicited role-appropriate judgement	Evidence that each checkpoint required a distinct type of human review rather than generic approval
DR4	Whether implicit governance information was surfaced explicitly	Presence of comparison rationale, ambiguity signals, ownership-related fields, and other governance-relevant metadata in later-stage outputs
DR5	Whether governance tasks were decomposed across explicit stages	Observable separation between intake, comparison, and actionable-output stages, with each stage operating on prior output rather than collapsing all tasks into one step
DR6	Whether the type of human intervention required at each checkpoint was clear	Ability to distinguish corrective, judgmental, and authorisation review functions across the three checkpoints

Continued on next page

Table 17 continued

DR	What was checked	Evidence of success
DR7	Whether outputs were legible to human reviewers without requiring technical system knowledge	Presence of a plain-language design brief and a structured output format suitable for review and action

F.6 Indicators used in the bounded evaluation

Technical correctness

This group concerned whether the workflow output was substantively correct enough to support governance:

- source-grounding correctness
- extraction and structuring accuracy
- comparison correctness
- mismatch and ambiguity detection
- traceability completeness

Human-review and practical usefulness

This group concerned whether the output would actually help a reviewer:

- output reviewability
- usefulness of the structured requirement table
- usefulness of the design brief
- presence of actionable next steps
- visibility of open questions and conflict cases

Effort

This group concerned whether the workflow changed the amount or type of manual work required:

- visible need for rework
- reviewer attention required per item in a bounded qualitative sense
- burden of manual search and comparison
- bounded feasibility of staged processing compared with a single-pass run

F.7 How the REQ-035 case was treated in the rubric

One specific point requires explicit clarification. During the bounded run, Agent 2 flagged REQ-035 as a conflict. At that stage of execution, this output was treated as a prototype conflict case and was reproduced accordingly in the later-stage outputs. However, later expert consultation showed that this case was caused by an upstream normalisation issue in Agent 1 and by the use of an imperfect comparison baseline. For that reason, the REQ-035 case was treated in the final evaluation not as proof of a genuine cross-document conflict, but as evidence of an important governance risk within the workflow.

This distinction matters for the rubric. The case did not lead to the conclusion that the workflow failed completely. Instead, it supported a *partial* rather than full positive assessment where the prototype behaviour was visible but later shown to require correction or stronger validation.

F.8 Role of expert consultation in the bounded evaluation

The expert consultation carried out after the prototype run did not replace the bounded evaluation rubric. Rather, it served as an external validation layer. Its main role was to test whether outputs that appeared plausible from the workflow perspective were also defensible from a domain-expert perspective.

This was especially important in two cases:

- reassessing the REQ-035 conflict interpretation
- reassessing the suitability of the selected SRS reference section

The expert input therefore strengthened the honesty and interpretive quality of the evaluation, even though it did not convert the bounded prototype study into a full industrial validation exercise.

F.9 Why this appendix matters

This appendix is intended to make the evaluation logic transparent. It shows that the assessments reported in Chapter 7 were not based on impression alone, but on explicit bounded criteria tied to the design requirements of the artefact. At the same time, it also makes clear that the evaluation remained a prototype-stage design-science assessment rather than a large-scale statistical benchmark.

G Interview Overview

This appendix provides an overview of the practitioner interviews used in this thesis. The interviews were conducted to support three purposes: problem exploration, workflow design, and bounded expert validation of prototype outputs. They were not treated as a standalone qualitative study, but as contextual and evaluative input within the broader design science research process.

All interviews were conducted through Microsoft Teams. To keep the focus on professional relevance rather than personal identity, the table below reports interviewees by professional role. All interviews were conducted with practitioners at a single multinational railway communications supplier, acknowledged earlier in this thesis; full interview attributions are provided in the reference list.

Table 18: Overview of practitioner interviews used in the thesis

Code	Role	Organisation	Date	Format	Main contribution
INT-01	FRMCS standardisation delegate, railway engineering role	Railway communications supplier	12 May 2026	MS Teams interview	Baseline problem exploration: standards, cross-specification inconsistencies, and reconciliation of requirements across UIC, ETSI, and 3GPP sources.
INT-02	Product manager, railway communications group	Railway communications supplier	20 May 2026	MS Teams interview	Current requirements handling practice: missing detail in incoming requirements, workflow usefulness, scale of requirement work, and management implications.
INT-03	FRMCS standardisation delegate, railway engineering role	Railway communications supplier	24 June 2026	MS Teams expert review call	Prototype validation: traceability expectations, trust thresholds for AI-supported output, REQ-035 interpretation, and suitability of the selected reference specification.

The first and third entries refer to two distinct interactions with the same practitioner and are treated separately because they served different functions in the thesis. The first interview contributed primarily to baseline exploration and problem framing, while the later interaction served as an expert review of generated workflow outputs.

The interview question guides used to support these conversations were semi-structured and adapted to the interviewee's role. Appendix H reproduces those guides. In the main thesis, interview findings are integrated where they informed the design requirements, evaluation interpretation, and discussion of organisational usefulness and limits.

H Interview Question Guides

This appendix reproduces the interview question guides used in the practitioner interviews conducted for this thesis. The interviews were semi-structured and the wording was adapted to the role of the interviewee. The first two guides were used for the baseline interviews, while the third guide was used in the later validation-oriented follow-up session.

H.1 Interview guide A: railway engineering role

1. Could you briefly describe your role and responsibilities as a railway engineer at the company?
2. How are you involved in requirements-related work in railway communication projects?
3. What kinds of requirement documents or inputs do you usually work with?
4. Do you mainly work with detailed technical requirements, or are you also involved in refining higher-level requirements into lower-level ones?
5. From an engineering perspective, what makes a requirement clear and actionable?
6. What kinds of ambiguities or missing details usually create difficulties during implementation or testing?
7. How are requirements currently reviewed, clarified, or handed over between roles or teams?
8. Is it difficult to trace a requirement back to its source or forward to implementation and evidence?
9. What kinds of duplicates, inconsistencies, or conflicts do you encounter in practice?
10. What type of output from a workflow tool would be most useful for you? For example, structured requirement sets, refined lower-level requirements, traceability links, or conflict flags.
11. At which points would human review definitely remain necessary, even if parts of the process were supported by AI?
12. What would make an AI-supported workflow helpful and trustworthy from your perspective?

H.2 Interview guide B: product manager role

1. Could you briefly describe your role and responsibilities as a product manager at the company?
2. How are you involved in requirements-related work in railway communication or product-related projects?
3. What kinds of inputs do you usually work with when requirements are being defined or discussed?
4. Do you mainly work with higher-level product needs and requirements, or are you also involved in their refinement into more detailed requirements?

5. How are product needs or stakeholder expectations currently translated into actionable requirements?
6. What are the main difficulties in this process? For example unclear wording, changing priorities, overlapping inputs, or missing details.
7. When requirements come from different stakeholders or documents, how are they currently compared, aligned, or prioritized?
8. What kind of workflow support would be most useful for you in this work?
9. What kind of output would help you most, for example structured requirement sets, grouped requirements, traceability tables, refinement suggestions, or gap/conflict lists?
10. At which points would human review definitely remain necessary, even if parts of the process were supported by AI?
11. What would make an AI-supported workflow useful and trustworthy from your perspective?
12. If the workflow produced incorrect or incomplete requirement outputs, what kinds of mistakes would be most problematic for your work?

H.3 Interview guide C: validation follow-up session

1. Looking at the comparison table I sent you, the one with fields like `comparisonNote`, `mismatchFlag`, and `reviewStatus`, is this the kind of structured output that would be useful to someone doing requirements governance work at Nokia? Or would a different format be more practical?
2. The workflow produces two parallel outputs: a structured CSV table for engineering teams and a plain-language design brief for senior decision-makers. Does that distinction make sense from your experience, or do the same people typically consume both?
3. What information would you expect to see in a comparison output that is currently missing from what I sent you?
4. You mentioned that traceability is paramount and everything AI produces needs manual verification. The workflow preserves the original source clause text in a field called `sourcePassage` alongside the normalised version. Does having that original text visible next to the AI interpretation address that concern, or would you need something more?
5. What would a practitioner at the company need to see in an AI-generated comparison output before they would trust it enough to act on it in a real project?
6. You described a brute-force cross-document comparison approach, comparing each chunk of one document against all chunks of the other. Is that how requirements comparison is currently done manually at the company, or is there a different process?
7. At larger scale, where in the process would a tool like this be most useful: at the beginning when requirements are first being captured, or later when conflicts need to be resolved?

8. If this workflow were to be tested properly in a real project context, what would you consider a meaningful test? What inputs, what outputs, and what criteria would tell you whether it actually added value?