AI-Driven Requirements Engineering: A Case Study on Automating Tasks and Improving Requirement Quality

Morten Westerheim
University of South-Eastern Norway
morten.westerheim@gmail.com

Gerrit Muller
University of South-Eastern Norway
Gerrit.Muller@usn.no

Kristin Falk
University of South-Eastern Norway
Kristin.Falk@usn.no

Abstract.

Practical Requirements Engineering is critical for ensuring traceability, standardization, and efficiency as the maritime industry shifts towards increasingly complex and digitalized systems. This study explores the potential of Artificial Intelligence to enhance Requirements Engineering practices in a Company that serves as a supplier to the maritime industry. Using AI to identify, analyze, and standardize requirements, this study examines how automation of early-phase tasks can address challenges such as fragmented processes, inconsistent tool usage, and low organizational maturity in Requirement Engineering. The research reveals the potential benefits and highlights the limitations and pitfalls of using AI tools in the Requirements engineering process.

Keywords.

Requirements Engineering, Requirements Management, Artificial Intelligence, EARS.

Introduction

The maritime industry is transforming rapidly, driven by increasing demands for complex, integrated, and digitalized vessel systems. As maritime markets evolve, so do stakeholder expectations for seamless functionality across navigation, propulsion, communication, and energy systems. This shift pressures internal product development processes, which must adapt to support more sophisticated and interconnected solutions. (Tijan, Jovi´c, Aksentijević, & Pucihar, 2021).

In such systems, the number and complexity of requirements rise considerably. Each subsystem must meet its own functional goals and align with the overall architecture of the system-of-system (SoS). This interdependency introduces a need for higher levels of coordination and traceability to manage potential overlaps, inconsistencies, and gaps in requirements. As a result, the ability of traditional document-centric practices regarding effective scalability is challenged.

Engineering organizations are exploring transitions toward more formalized and structured approaches to requirements engineering (RE) to meet these challenges. This includes adopting structured requirement templates, modeling techniques, and tools supporting traceability, clarity, and reusability across project lifecycles. In this context, integrating Artificial Intelligence (AI) into RE practices is beginning to emerge as an enabler for managing complexity early in the development process.

Company is a global provider of advanced technology solutions for maritime automation, navigation, energy, and propulsion systems. Over the years, Company has built its reputation on strong capabilities in delivery, customer service, and operational reliability. To maintain uptime and ensure compliance with industry standards, the company has traditionally prioritized incremental product improvements and efficient supply chain execution. While successful in many respects, this delivery-focused strategy has also contributed to accumulating technological debt (Li & Avgeriou, 2014). Particularly in areas related to legacy systems and early-phase development practices, it has created challenges in scaling innovation and responding to emerging market needs.

Company has taken essential steps to address these challenges by integrating Smart Agile frameworks, Lean principles, and traditional Systems Engineering approaches into its product development and management processes. These efforts have helped improve responsiveness and introduced new ways of working. Still, the transition has proven difficult due to the growth by acquisitions of other companies over the last decades, incorporating new organizations and cultures, and increasing product portfolios. The company still needs to make Improvements, especially when setting up standardized methods across teams and promoting a shared understanding of early-phase requirements.

Recognizing these limitations, Company's executive board has launched a broader digitalization strategy to harmonize internal work practices. This initiative seeks to enable more consistent collaboration across product lines and disciplines, break down silos, and establish a common foundation for tools and methods—especially in the early stages of system development, where alignment is often the weakest. By building a shared digital ecosystem, the company hopes to support long-term innovation while maintaining the reliability and quality on which its customers depend.

Figure 1 explains the different activities and workflows for early-phase Requirements engineering. This study's test case focused on the activities within the dashed lines. With some adjustments to fit the context of this study, the activities are based on the same principles as described by J.J. Norheim et al. (Norheim, Rebentisch, & Xiao, 2024). This study aims to determine how the company can apply AI to tasks associated with these activities. Specifically, extracting requirements from industry standard documents, analyzing content that indicates a requirement, and translating it into a standard syntax were the three tasks chosen for the test case described in this research.

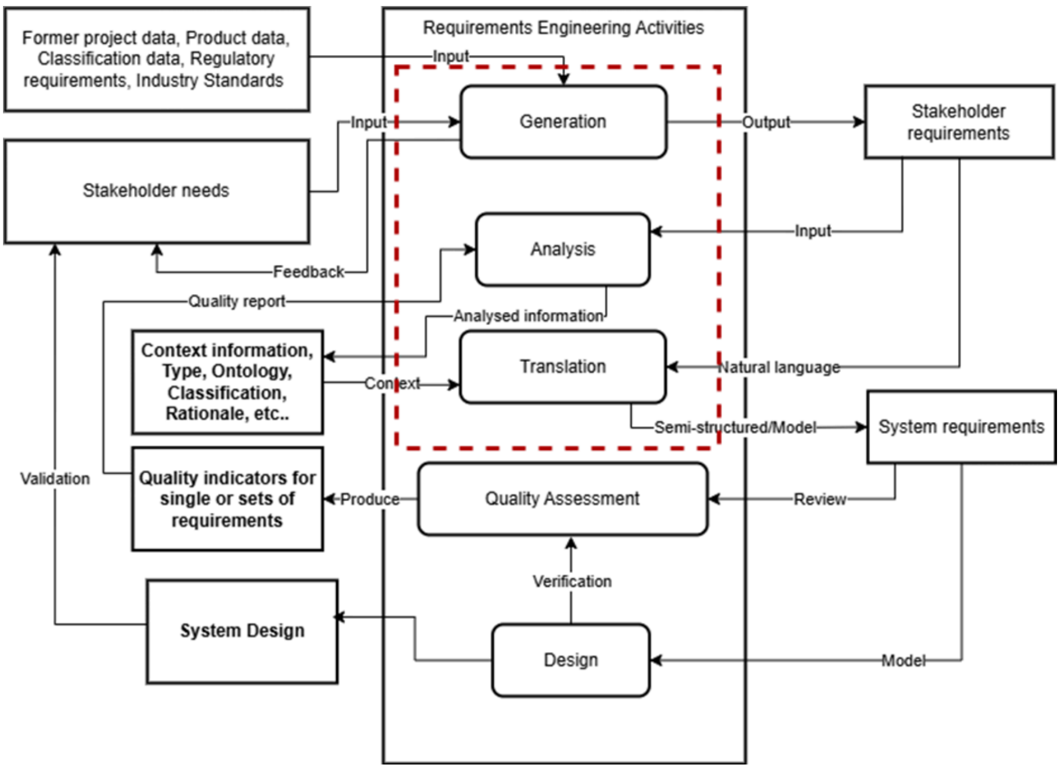

Figure 1: RE activities, inputs, and outputs (Westerheim, 2024)

The researchers chose the Easy Approach to Requirements Syntax (EARS) as the output format for the translation test. The EARS method aligns with the principles of reasonable requirements engineering (Mavin, 2009), emphasizing the need for clear, unambiguous, and verifiable requirements to reduce misunderstandings and rework. The structured language approach is designed to improve requirements' clarity, precision, and testability. It utilizes specific patterns and templates to standardize how requirements are written, making them less ambiguous and easier to understand. EARS promotes using modal verbs and defined sentence structures to ensure requirements are consistently phrased, which is particularly valuable in complex systems engineering projects. Specifically, EARS structures the requirements using the 'Ubiquitous' pattern, 'Event-Driven' pattern, 'State-Driven' pattern, and 'Optional Feature' pattern.

Problem statement. Company (Company) faces challenges in Requirements Engineering (RE), including a lack of standardized methods and fragmented processes due to document-centric approaches and inconsistent tool use. This results in low organizational maturity in RE practices and potential misunderstandings of requirements. This paper investigates how Artificial Intelligence (AI) tools can assist engineers by automating the extraction of requirements from standards documents and reformulating them into a standardized syntax (EARS). This research evaluates the effectiveness and limitations of different AI tool complexities (off-the-shelf vs. pre-trained) in addressing these challenges. It informs best practices for integrating such tools into Company's RE workflows.

To investigate these challenges and opportunities, this research focuses on three key research questions:

RQ1: Which activities within early-phase Requirements Engineering can AI tools be useful for Company?

RQ2: What are the observed differences in performance and reliability between off-the-shelf (Level 1) and pre-trained model (Level 2) AI tools for requirements engineering at Company?

RQ3: What technical and organizational enablers are required to adopt AI tools into Company's requirements engineering workflow?

Research Methods

The research combines a literature review, test case analysis, interviews, workshops, and workflow analysis. These methods integrate both qualitative and exploratory research approaches.

Interviews and workshops.

Talking to people in the organization has been an essential part of this study to establish a baseline and explore the context of early requirements development. Interviews can be completely free-form, where researchers are entirely perceptive, open to every piece of information. Alternatively, researchers can prepare themselves for the interview by writing down questions or keywords. More preparation tends to correspond with moving toward a standardized format on the spectrum axis. (Muller, 2013). The different methods of gathering information from people have been threefold in this study:

1. We interviewed sixteen stakeholders within the product development department in Company. (Johansson, 2024) and (Lunde, 2024) conducted and described the interviews in cooperation with the main author. The interviews aimed to understand the RE workflow and identify challenges and pain-points related to the requirements process.
2. We conducted Semi-structured interviews with stakeholders within a specific product development program to gain more practical and specific examples of struggles and opportunities to develop the test case. These roles included a Product line manager, a Project manager, a Release Manager, a Systems Architect, and two Domain experts. The goal for

these talks was to identify opportunities for AI tool integration.

3. To develop the test case, we conducted 10 iterative short workshops with a System Architect and Domain experts. The test case involved creating an AI tool and running tests, so a software development team joined these sessions. These workshops aimed to validate the possibility that we were addressing a real problem and that the tool was built to solve these problems.

Workflow analysis.

The researchers analyzed the Business Management System (BMS) to verify the findings in the pre-study interviews. They also examined the documented processes in the BMS to evaluate what the company had described as best practices.

Test case analysis.

To evaluate the performance of AI-powered tools and chatbots in automating tasks in the requirements engineering process. The researchers developed a test case using a collection of standard documents related to the design of Automatic Identification Systems (AIS) as a database for the test. The test's primary objective was to assess the system's ability to extract and reformulate the requirements according to the EARS model. It was also to learn about and measure the level of accuracy and at the same time, expose the limitations. The researchers also examined the level of fidelity during the test, and the prompt was optimized.

Use of AI tools.

In addition to traditional academic research methods, this study systematically incorporated AI tools to support various aspects of the research process. These tools were used as digital writing assistants to organize and structure chapter outlines, refine sentence formulations and technical phrasing (Grammarly, Chat GPT (OpenAI), Gemini Advanced (Google)), generate summaries of key concepts from reviewed papers (NotebookLM (Google)), recommendation of scientific papers to strengthen academic support (Keenious). The authors reviewed and revised all the AI-assisted content to keep it consistent with academic integrity requirements. Such tools align with emerging research practices and are transparently acknowledged in this study.

Method Limitations.

- The researchers evaluated outputs from the AI tools based on static test cases, including a specific scope of a larger system, rather than full-scale projects.
- The generalizability of findings is limited due to Company's specific domain and organizational context.
- The use of generative AI tools in academic writing is still experimental. However, including it as an evolving practice and interpreting it with proper transparency and critical thinking is essential. Nevertheless, integrating human-in-the-loop validation and reflective triangulation strengthens the study's methodological robustness.

Literature Review

The application of AI tools in early-phase Requirements Engineering (RE) has gained significant traction in recent years, with Large Language Models (LLMs) such as ChatGPT appearing as assistants in RE tasks. (Marques, Silva, & Bernardino, 2024) Present a comprehensive review of ChatGPT's capabilities across various software RE activities, highlighting its potential for automating early-stage tasks. The study emphasizes that ChatGPT can improve efficiency in interpreting stakeholder needs, drafting initial requirement statements, and suggesting structured formats for requirements based on natural language inputs. This aligns with the need to maintain engineering rigor while leveraging the benefits of AI for increased productivity in early RE activities. (Dalpiaz & Niu, 2020) Highlight the historical development of AI methods in RE through the AIRE workshop series. Their findings point to specific RE tasks where the authors stress incorporating metadata (e.g., readability indices, traceability tags) into AI-driven RE pipelines to improve robustness and scalability in complex engineering environments.

In addition to AI-supported elicitation and classification, early-phase RE activities can also benefit from automated quality assurance mechanisms, particularly through the concept of "Requirements Smells." (Femmer, Fernández, Wagner, & Eder, 2017) Introduce this concept as a lightweight approach to detecting quality defects in natural language requirements, such as vagueness, loopholes, ambiguity, and non-verifiability, analogous to code smells in software development. The study proves how AI tools integrated into early RE phases support immediate feedback on requirement quality, thereby reducing costly rework in later stages of development. The prototype tool used in the study was based on NLP techniques to analyze requirement artifacts and highlight potential quality issues, which in the years that have followed have evolved into the more advanced LLMs, like ChatGPT, Gemini, and Co-Pilot, we know today.

Both studies emphasize that human oversight is still essential, particularly for validating the contextual relevance. Domain experts familiar with the specific system need to validate the AI-generated outputs. This ensures that text proposed by the AI tools is correct and contextually sound, especially in safety-sensitive maritime domains. It also communicates to employees that these tools are intended to support, rather than replace, engineering roles, thereby reducing perceived threats and fostering greater confidence, motivation, and willingness to engage with technology.

Integrating AI tools into existing requirements engineering workflows demands a structured approach to how requirements are specified, formalized, and managed throughout the development lifecycle. (Göhlich, Bender, Fay, & Gericke, 2021) This need is described by proposing a refined Product Requirements Specification (PRS) process, grounded in real-world observations from the automotive and rail industries. The study underscores the critical role of early-phase requirements activities such as task clarification, requirements refinement, and interface definition. While the study does not focus on AI, it identifies pain points—such as the lack of consistent structure, traceability, and reuse of requirements—that are well-suited for AI enhancement. Furthermore, the paper emphasizes the importance of aligning requirements processes with product and system architectures.

AI adoption is about automating isolated tasks and embedding intelligent assistance into a continuous, traceable, and context-aware requirements management workflow, particularly in domains where cross-disciplinary collaboration and compliance with standards are essential. Supporting this integration, (Vierlboeck, Nilchiani, & Blackburn, 2024) explore how Natural Language Processing (NLP) can be used to assess the structure and complexity of system requirements. Their case study demonstrates that NLP-enabled structural decomposition of requirements can enhance understanding of system interdependencies and complexity levels in early RE. Using metrics such as network density and spectral entropy, AI tools can help engineers identify structural bottlenecks and ambiguous linkages within requirements, supporting early design decisions. Similarly, (Johns, Carroll, Medina, Lewark, & Walliser, 2024) present an experimental AI-driven modeling assistant, AI-SME, integrated into a SysML-based MBSE6 environment. Their research shows that GPT-powered assistants can generate preliminary model art based on natural language inputs in fact, such as block definition and internal block inputs. (Siddeshwar, Alwidian, & Makrehchi, 2024) Presents a systematic review of AI-enabled frameworks in requirements elicitation, naming fifteen distinct elicitation tasks supported by AI tools and emphasizing the growing ecosystem of datasets and algorithmic techniques used in early RE automation. Their work provides a valuable benchmark for evaluating the practical implementation of AI solutions within an organization, where scalability, interpretability, and traceability remain foundational pillars of product development practices.

Test case

To determine the level of complexity of AI tools suited to assist in RE, the researchers designed a test case based on the stakeholders' needs in a specific product development project. Conducted two tests using tools within Levels 1 and 2, as described in Table 1.

Complexity	Description	Type of tools
1	Off-the-shelf tools	Chat GPT, Gemini Advanced
2	Pretrained model with several support tools, increasing data capacity for data handling and automation of tasks.	Snowflake Cortex, Phyton, Pandas, Beautifulsoup, PymuPDF, Numpy, Streamlit
3	Traditional machine learning built from scratch	Complex software development that takes significant resources and is therefore not explored in this paper

Table 1. Level of AI tool complexity.

The researchers trained the AI tools to work according to quality criteria derived from ISO/IEC/IEEE 29148 (ISO, 2018), emphasizing clarity, unambiguity, completeness, verifiability, traceability, and standard conformance (e.g., EARS templates). The researchers reviewed the AI-generated outputs to assess their utility in the engineering context.

Test case Level 1.

We define "Off-the-shelf solution" as web-based AI tools such as Chat GPT and Gemini Advanced. These tools were used to upload relevant documents. By prompting the tool to parse the documents into sizes according to the tool's token limits, the data was prepared for analysis by the AI tool. Each step, visualized in Figure 2, was conducted manually by data import/export and prompts. In this test, each clause that may contain several requirements was extracted for contextual and traceability purposes, one clause at a time. Whenever the researchers observed the level of fidelity deviating from the original, the tool was asked to run a diagnostic of its own process. The researchers instructed the tool to optimize the prompt based on the diagnostics data, and the clause was re-extracted. The translation into EARS format was tested using this tool; see further comments in the result section.

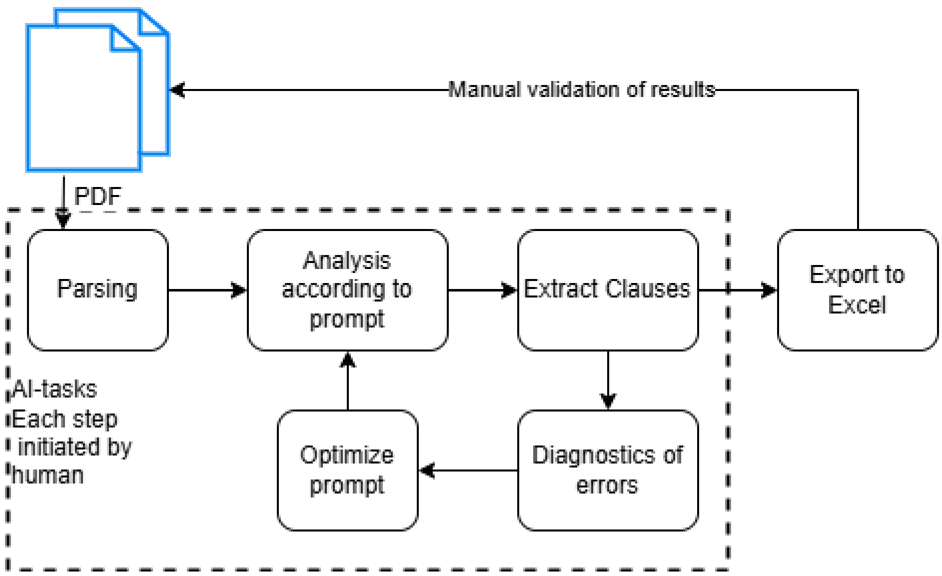

Figure 2: Test case level 1 (Author's illustration)

Test case Level 2.

This test setup integrated natural language processing (NLP), semantic embedding, and rule-based formatting. Various Python-supporting tools, including Pandas, Beautifulsoup, PymuPDF, NumPy, and Streamlit, were utilized with the cloud data platform Snowflake and Azure Blob Storage for cloud storage. The process commenced with the ingestion of PDF-format documents. Snowflake Cortex parsed these documents to extract textual content. Researchers performed functional testing to ensure the chatbot accurately and completely extracted requirement statements. Each requirement was furnished with a source clause, ensuring traceability and verification, and enabling tracking back to the source document for regulatory compliance.

The developers implemented a series of structured steps by integrating the tools mentioned in their research process. Figure 3 illustrates the Step-by-step workflow for extracting and processing system requirements from PDF documents. The process is divided into four stages: (1) identification and segmentation of requirement sections, (2) storage and linking in cloud environments, (3) conversion to text and preparation for analysis, and (4) AI-assisted analysis and translation into structured outputs using EARS elements. This diagram reflects the functional workflow used in the test case.

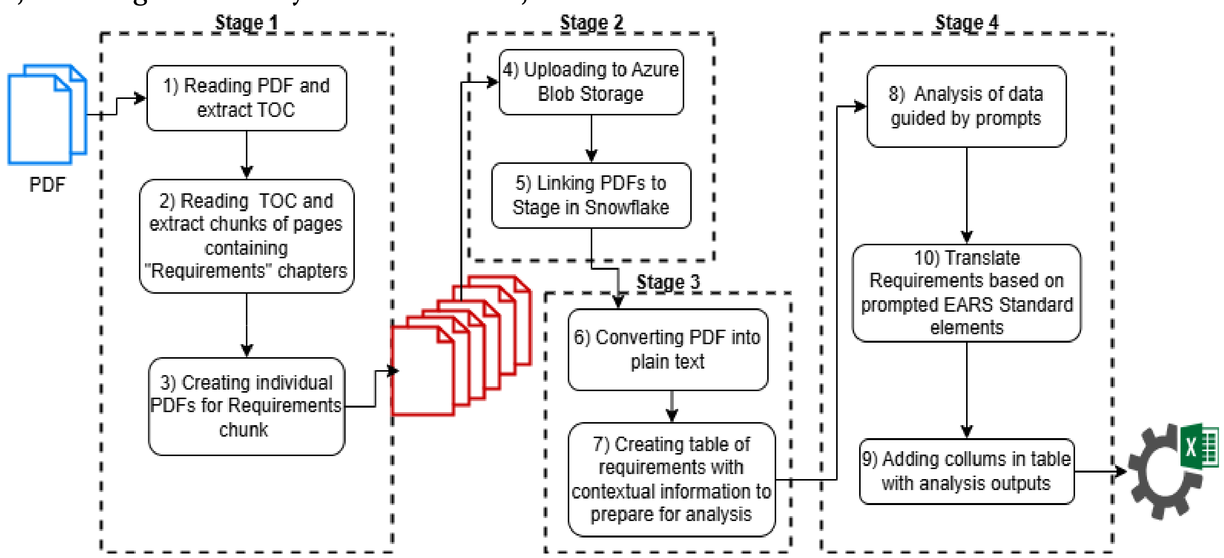

Figure 3: Test case level 2. (Author's illustration)

To manage the limitations of processing large documents directly with a language model (LLM), the workflow began by identifying relevant sections of each standard using the document's Table of Contents (TOC). This allowed for isolating only those chapters containing requirements, rather than loading the entire file. Each relevant section was extracted and saved as a separate PDF. These extracted PDF files were uploaded to Azure Blob Storage — a cloud-based service for storing large data volumes. To make the PDF files accessible for further processing, they were linked to Snowflake Stages, which function as secure gateways for data movement into the Snowflake environment.

Next, Snowflake Cortex's Parse Document functionality automatically converted each PDF into plain text. This was necessary because AI models operate on text, not PDFs. After conversion, the requirement sections were split into smaller, manageable text chunks. These text chunks were stored in a table structure for easy indexing and retrieval. Each of these text chunks was then processed using Snowflake Cortex Embed Text. This function creates a semantic embedding — a numerical representation of the meaning of the text. These embeddings are essential for understanding the words used and their context. This allows the system to compare meaning across documents, not just surface keywords.

To extract requirements from these sections, we used a carefully designed prompt. This prompt instructed the AI model to find only sentences explicitly labeled as "Requirement" and ignore unrelated content. If, for instance, tables were present in the Standard document, the prompt told the model to wrap them in <table> tags so that they could later be parsed using HTML tools like Beautifulsoup.

Finally, each requirement was reformulated using the EARS format to bring consistency and structure to the extracted content. Another structured prompt guided the model in translating the requirement into one of the standard EARS patterns, such as Event-driven, State-driven, or Unwanted behavior. Each reformulated requirement was mapped into a schema with elements like Subject, Verb, Object, Condition, and Constraint.

Results

In the pre-study, sixteen stakeholders with different perspectives on the requirements management process in product development were interviewed. Table 2. Describes their roles and involvement in the RE-process. The interviewees were asked the same set of questions about their views on the current situation, what they saw as challenging, and how they envision improvements to mend pain points. The researchers got a good picture of the activities and practices in requirements engineering and management. Analyzing the workflow described in the BMS, confirmed many statements. The most relevant challenge for this study was generally low maturity in working systematically with requirements management. Some pockets in the organizations had their own structures for organizing requirements. Still, these were spreadsheet-based primarily, making updating time-consuming with low traceability and no version control.

The BMS is relatively new, and its process descriptions are still a work in progress. The feedback from the stakeholders varied when it came to working in coherence with the processes. In general, there was no common practice across the organization. The confidence among the different disciplines in the correctness and completeness of the data received from others was low. The lack of proper guidelines and standard methods in RE led to many ways of writing and presenting requirements, which led to disputes and misunderstandings. Currently, no tools are commonly used to provide version control and traceability; many roles are left with time-consuming manual processes. In several cases, SA reported lacking contextual information, sometimes leading to unnecessary discussions and rework.

Nr./ role	Role	Involvement in RE-process
3	Lead System Architects (LSA)	Working with early phase requirements elicitation, verification, and validation at the Architecture level.
5	Systems Architects (SA)	Same as LSA, but also on the design level.
2	Test Managers (TM)	Working on testing implemented systems, but also involved in early validation of system requirements, focusing on testability.
1	Software Architect (SoA)	Primarily involved in establishing design requirements.
1	System design engineer (SDE)	Working on elicitation of system and subsystem requirements.
1	Cyber Security Architect (CSA)	Domain-specific to Cyber Security, working heavily with compliance towards industry standards.
1	Group Manager Hardware design GMH)	Involved with several parts of the development process, mainly overseeing the engineering team and working with industry-standard hardware requirements.
1	Product Development Manager (PDM)	Involved in the whole development process. Focused on the lifecycle of requirements.
1	Project Manager (PM)	Involved in the entire project phase, focused on progression in reviews and approval of requirements.

Table 2. Interviewees' roles and involvement

Based on these interviews, the researchers established hypotheses on how AI tools could mend some of the challenges mentioned. By having semi-structured interviews with a specific product development program, we established realistic objectives for the test case.

Analysis of Level 1 test case.

The structured evaluation of the off-the-shelf AI tool assisting in RE activities yielded several insights, revealing both capabilities and limitations of current mainstream AI tools when integrated into early-phase RE workflows. A basic preliminary test exploring the capabilities of document handling, Token sizes limitations and text interpretations were made for 3 candidates. Among Microsoft's Co-Pilot, Google's Gemini Advanced and OpenAI's Chat GPT, the latter was chosen for the test case based on the results.

A set of accuracy metrics based on the test results was established to evaluate the AI tool's performance by extracting requirements from structured documents. During the test phase, 70 clauses were assessed, representing a diverse sample of maritime standard content. Fifty clauses were extracted successfully, with no major omissions or structural errors observed. This resulted in an extraction accuracy of 70%. Twenty-one clauses were identified as having issues, leading to an overall error rate of 30%.

The extraction accuracy metric reflects the proportion of clauses in which the AI model correctly identified and presented the requirement statement and associated test criteria. It does not account for the degree of difficulty in each clause, nor does it differentiate between partial and complete errors. The error rate aggregates any forms of extraction failures, including incomplete text capture, low fidelity rate, formatting issues, missing lead-in sentences, clause segmentation problems, and limitations related to embedded tables or extended clause length.

While the AI tool could perform the majority of extractions without requiring intervention, no measurement of average extraction time or manual revision effort was included in the test scope. Therefore, metrics related to efficiency, throughput, or post-processing workload are not presently quantified, though they remain relevant for future evaluation. Similarly, the current analysis does not assess semantic comprehension or accurately assesses the presence, structure, and formatting of extracted content.

When applied to regulatory content, the metrics Table 3 provide a static, document-level perspective on the AI tool's extraction capability. While the findings are not meant to generalize beyond the tested dataset, they provide a clear baseline for assessing the system's operational accuracy under realistic engineering conditions.

Metric	Value	Description
Total Clauses Evaluated	70	Total number of requirement clauses included in the test set.
Successful Extractions	49	Clauses where the AI extracted complete and accurate requirement and test criteria content.
Failures/Issues Detected	21	Clauses where the extraction was incomplete, incorrect, or required prompt adjustment.
Extraction Accuracy	70%	Proportion of clauses with no issues (49/70).
Error Rate	30%	Proportion of clauses where the extraction had some defect or limitation (21/70).

Table 3. Metrics test case Level 1

Trials were made to convert the extracted clauses into separate requirements to the EARS standard requirements. This did not return any valid results as the AI tool had problems transitioning into the format while keeping the purpose of the requirement intact. On some occasions, it even changed the number of specific metrics required for standardizing the system.

Analysis of Level 2 test case.

The outputs from the process described in the Test case chapter included structured requirement statements traceable to their source sections. The EARS-formatted requirements were presented in an explanatory and structured way. For clarity, the researchers presented the results in 2 analysis levels: Clause and Requirement.

Metric	Value	Description
Total Clauses	107	The total number of requirement clauses is included in the original test document.
Complete Clauses	75	Fully extracted Clauses
Incomplete Clauses	28	Clauses where the extraction was incomplete
Missed Clauses	4	Clauses that were not detected
Extraction Accuracy	70%	Proportion of clauses with no issues (75/107).

Error Rate	30%	Proportion of clauses where the extraction had some defect or limitation (32/107).

Table 4. Clause-level analysis

Table 4 presents a manual validation was conducted to assess the completeness and reliability of the extracted requirements, where each clause was checked against the original document. Each of the 107 evaluated clauses was marked either "OK" (complete) or one of several "Lack" conditions representing different types of partial or missing content. The AI tool completed and correctly extracted 75 clauses (70%), while 28 (30%) exhibited partial extraction issues. These issues included missing first paragraphs, omitted content at the top of the following page, or skipped segments after enumerated or punctual lists. Four clauses (3.7%) were missing, meaning the AI tool extracted no relevant requirement content. The most common extraction issue was the omission of paragraph segments at the top of the next page, a problem observed in 19 entries. The errors connected to the introductory sentences and paragraphs, following punctuation, were less frequent but notable.

Metric	Value	Description
Total Requirement Lines Evaluated	400	Total number of entries processed as "requirements."
Valid Requirements (with modal verbs)	247	Lines with at least one normative modal verb.
Invalid Requirements (no modal verbs)	153	Non-directive lines lacking modal verbs; often descriptive or referential.
Estimated Extraction Issues (within valid)	21	Valid entries with formatting, scope, or semantic issues.
Successful Requirement Extractions	226	Valid requirements without extraction issues.
Requirement Extraction Accuracy	56.5%	Proportion of all extracted lines that are valid and issue-free (226/400).
Requirement-Level Error Rate	43.5%	Proportion of extracted lines that were invalid or flawed.

Table 5. Requirement-Level analysis.

As presented in Table 5, the AI tool divided the 103 extracted clauses into 400 text lines that the researchers evaluated for their suitability as formal requirements. Among these, 247 entries (61.8%) were identified as valid requirements by the AI based on the presence of normative modal verbs such as "shall," "must," "should," or "may." These modal verbs signal directive intent and are a standard feature of syntactically complete and testable requirements in technical standards.

For several reasons, the researchers evaluated 153 entries (38.2%) of the text lines as invalid. Either these lines lacked the necessary modal structure, were support text in the document (e.g., a footer or head note), or were parts of lists or tables that did not give meaning on their own, and therefore did not meet the criteria for enforceable requirements. Upon review, the researchers observed that many of these failed entries were extracted by the AI tool from sections of the standard document that are formally labeled "requirements", but the content itself was intended to provide clarification or supplementary information, not directive statements. Common patterns among invalid entries included descriptive commentary, contextual notes, and implementation guidance.

Additionally, many invalid entries consisted of reference citations or cross-references to other documents, clauses, or appendices. These did not contain independent obligations and thus failed to qualify as standalone requirements. The formatting of these entries also contributed to their misclassification. Structural artifacts introduced during automated parsing, such as bullet points, incomplete fragments, and inline citations, further obscured their intent.

Within the 247 valid requirements, a second-layer diagnostic identified 21 entries (8.5%) that exhibited structural or semantic issues despite containing modal verbs. These issues included ambiguous or overly dependent references to external clauses, inconsistent formatting, poorly defined subject-action relationships, or conflation of multiple directives into single, unstructured sentences. Such patterns risk accurate interpretation, testability, and automated downstream processing.

EARS Rephrasing

The test case also asked the AI tool to rephrase the language used in the standard document to formulate requirements into EARS format. Table 6 shows an example of this test's input, output, and breakdown.

<table>
<tr><td>Original extracted from IEC 62288</td><td>Rephrased to EARS</td></tr>
<tr><td>(MSC191/7.3.1.4) It shall be possible to add or remove information from the chart display by layer (IHO viewing group, see for example IHO S-52), but not as individual objects. It shall not be possible to remove information contained in the ECDIS Display Base (see IEC 61174) from the display. (See also 6.3.2.)</td><td>WHILE the chart display is active, the ECDIS system shall allow adding or removing information by layer (IHO viewing group), but not as individual objects, WHERE the information is not part of the ECDIS Display Base (IEC 61174).</td></tr>
<tr><td colspan="2">Breakdown according to EARS pattern:

• Condition: WHILE the chart display is active (state-driven pattern)
• Subject: the ECDIS system
• Priority: (not specified, assumed to be high)
• Verb: shall allow
• Object: adding or removing information
• Constraint: by layer (IHO viewing group), but not as individual objects
• Value: (not specified)
• * Optional element: WHERE the information is not part of the ECDIS Display Base (IEC 61174) (optional feature pattern)</td></tr>
</table>

Table 6. Example of rephrasing to EARS

The correctness and quality of these outputs need to be reviewed by a discipline expert for validation, which has not been done in this study. Results from EARS rephrasing should be viewed in this paper to demonstrate what is possible for further research and implementation.

Discussion

The findings of this study reveal both opportunities and limitations in utilizing AI-assisted tools for early-phase requirements engineering (RE). From an organizational perspective, challenges arise from the variability in RE knowledge and practices among stakeholders in product development.

Consistency in the use of tools and methods is needed to support early-stage RE activities to reduce ambiguity, ensure semantic consistency, and enable traceability. An AI-based approach that begins with elementary tasks like extracting contextual data, identifying requirements, and rephrasing them into more formal formats has the potential to assist engineers in automating mundane, repetitive tasks. This aims to achieve the primary goal of better documentation of requirements and enhance their ability to manage the growing complexity of the systems they are building.

Most clause-level extraction challenges in the test case resulted from the AI tools' difficulties with pagination, multi-paragraph structures, and embedded lists. This highlights the limitations inherent in standard prompt engineering and layout-agnostic parsing methodologies. It calls for further research, testing, and experience-building, providing the organization with valuable insights. We are not just discussing knowledge of how to create and use support tools; the same functionality will also benefit the future development of products and customer support.

We must consider that the errors at the clause level occurred during the process of identifying and splitting into single requirements, which requires even more training and fine-tuning of the prompts. However, analyzing the structure and patterns of the errors shows that most of them can be resolved through better AI tool training, further development within prompt engineering, and adaptation to the rapid evolution of smarter LLMs.

The metrics show that the current level of the pretrained model yields only moderate accuracy. The requirement-level error rate of 43.5% reveals a critical gap in the AI tool's ability to distinguish between enforceable requirements and supporting or descriptive content. This accuracy rate may be insufficient for direct use in system-level specifications without further filtering in a regulated environment such as Company's product development. These results emphasize the need for a robust validation step following extraction, where non-requirement lines are flagged or filtered, ideally using a custom-trained classifier or rule-based post-processing. A semi-automated verification stage may also be needed to ensure readiness for downstream use.

The study's secondary objective was to test the integration of the EARS format for structural rephrasing. While the technical feasibility of EARS rephrasing was demonstrated in the study, the validity of the results heavily depends on the quality of the input requirements. Domain experts have yet to confirm whether the results hold a reasonable degree of value; however, the outcomes appear promising from a purely semantic perspective.

When analyzing the clarity of the patterned requirements, it was clear that the AI tool needs more training in utilizing the different patterns. The tool was fed a simple explanation of the EARS patterns and objectives. Providing the tool with a broader range of domain-specific data should be the next step to improving the output quality.

(Siddeshwar, Alwidian, & Makrehchi, 2024) Emphasize the broad applicability of AI methods in requirements elicitation but also point out that common limitations such as "low generalizability," "suboptimal results despite sufficient data," and "poor data quality" persist. In this study, the tools and prompts used in the test case may not work well in a different domain or with a different set of input documents. Even though we had clear clauses, optimized prompts, and structured inputs, the AI tool still made mistakes. Parts of the PDF inputs had formatting issues, long clauses, or embedded tables, which affected extraction quality. That reflects the exact type of input complexity Siddeshwar et al. are warning about.

From a practical standpoint, the study illustrates the potential for AI's role in accelerating RE processes for organizations like Company. However, to implement these tools in product development environments, one must strengthen the knowledge and practices for standardized pre-processing and general AI-based workflows that support human-in-the-loop validation.

While the test cases demonstrate that AI can automate early-phase requirements tasks with moderate success, the results also highlight the importance of human-in-the-loop validation to ensure contextual correctness and compliance. For Company to institutionalize this capability, the organization must assign responsibility for validating AI-generated outputs. This could involve creating cross-functional review teams composed of domain experts, systems engineers, and software architects. Guidelines would need to be developed to define acceptable levels of AI autonomy, revision protocols, and audit trails. Without this layer of organizational maturity in Requirements Engineering and the use and development of AI capabilities, the risk of over-relying on AI tools could undermine safety-critical engineering decisions.

To address the research questions, the research identified several activities to which AI tools could be applied.

RQ1. Within early-phase Requirements, these activities include extracting requirements from standards, analyzing content for requirement indicators, and translating into a structured syntax.

RQ2. The study highlighted performance differences between off-the-shelf AI tools and pre-trained models. Off-the-shelf tools offer quick initial exploration, while pre-trained tools demonstrate higher accuracy and automation potential. However, they require more setup and data preparation.

RQ3. To integrate AI effectively for Requirements Engineering, this research suggests 5 key points:

- A central place to store all requirements is essential for the AI tool to work on large quantities of data. In the test case, we only looked at a specific set of standard documents for a sub-system, which does not consider the broader context of the requirements in question.
- To train AI models using language specific to the organization and standard industry terminology is essential for the familiarity and adaptability of such a system within the organization.
- User guides for creating good prompts are needed to familiarize the employees with the potential and limitations of these tools for decision support, not for decision-making.
- Running small test projects like the one demonstrated in this study would be a good way forward. Developing similar test cases based on real challenges while adding the amount of data, new features, and more user-friendly interfaces, where outputs can be easily assessed.
- Having a team dedicated to overseeing the AI implementation, consisting of cross-functional disciplines, to ensure real usefulness for engineers and alignment with the organizational goals.

These steps are tangible options for AI integration into Company's current workflows for Requirements Engineering and other business areas.

Limitations. The empirical evaluations of AI performance were conducted using static test cases drawn from a specific segment of maritime standards, rather than encompassing the full scope of the operational project. It therefore does not capture the variability and complexity of requirements management across all of Company's diverse product lines.

The test did not measure time spent or cost of tokens to quantify the efficiency and effectiveness of using AI tools, thus limiting the insight into its practicality.

To fully validate the quality and correctness of the EARS translation, results need to be examined by a domain expert within the technological field in question, and this was not done in this study.

Other topics important to AI implementations that are not adequately addressed in this study are: Ethical considerations, Data bias, Transparency in AI-decision-making, Data security, cybersecurity, Training and user adaptation, Long-term maintenance, ROI, legal and regulatory compliance, and integration with existing tools.

Conclusion

This research has explored the potential of AI tools that can enhance early-phase RE processes at Company. We addressed specific challenges, including fragmented methodologies, inconsistent tool usage, and the need for improved organizational maturity regarding RE. We used a combination of interviews, workshops, and literature reviews as a foundation for a test case based on real-life challenges, where we evaluated off-the-shelf and pre-trained AI models. This study has provided insights into the practical application and limitations of AI-based extractions and management of requirements derived from industry standards. The research has revealed the potential for automating tasks related to requirement extraction and reformulation into standardized formats like EARS. Careful consideration must be given to these tools' accuracy, reliability, and integration.

The test cases demonstrated varying levels of success depending on the complexity of the AI tools utilized. Off-the-shelf solutions provided a quick and accessible method for initial explorations, but struggled with nuanced formatting, complex phrasing, and the conversion of requirements into specific formats like EARS. Conversely, leveraging more advanced processing and integration capabilities, the pre-trained model showed improved accuracy and automation potential. However, it also highlighted the importance of data preparation, prompt engineering, and iterative refinement to achieve reliable results. Both approaches emphasized the need for human oversight and domain expertise to validate the AI-generated outputs and ensure their relevance and accuracy for the specific operational context.

Further research. Based on the findings, limitations, and discussions in this study, there are, in particular, five areas to be explored in future studies:

- **Domain Expert Validation**. The AI-generated outputs, especially those reformulated into EARS syntax, should be evaluated by domain experts. Structured validation studies are needed to assess their interpretability, technical correctness, and usability in real engineering workflows.
- **Efficiency and Cost Assessment**. Measurements of time savings, token usage, and processing costs. This would enable a cost-benefit analysis of AI integration and support ROI-based decision-making.
- **Lifecycle Integration**. The current work focuses on early-phase requirements. Further research could track how AI-supported requirements evolve and trace across the full product lifecycle, including design, testing, verification and validation, quality assessment, and change management.
- **Cross-Domain and Cross-Standard Testing**. The test case focused on standards for one specific subsystem. Additional studies should evaluate the performance of AI tools across other standards and system domains.
- **Traceability Automation**. Further research should explore how automatically extracted requirements can be embedded with traceable metadata, enabling better linking between requirements, source documents, and verification activities.

References

Dalpiaz, F., & Niu, N. (2020). Requirements Engineering in the Day of Artificial Intelligences IEEE.Software 37, 4, pp7-10..

Femmer, H., Fernández, D. M., Wagner, S., & Eder, S. (2017). Rapid quality assurance with Requirements Smells. Journal of Systems and Software, 123 pp190-213

Göhlich, D., Bender, B., Fay, T.-A., & Gericke, K. (2021). PRODUCT REQUIREMENTS SPECIFICATION PROCESS IN PRODUCT DEVELOPMENT. IECD21, pp 2459-2470, Cambridge University Press

ISO. (2018). ISO/IEC/IEEE 29148:2018 Systems and software engineering — Life cycle processes — Requirements engineering. Retrieved on 26-10-2025, from https://www.iso.org/standard/72089.html

Johansson, S. (2024). Analyzing the Need for a Requirements Management System at a Maritime Technology Company. Master project paper at University of South-Eastern Norway

Johns, B., Carroll, K., Medina, C., Lewark, R., & Walliser, J. (2024, 7 2). AI Systems Modeling Enhancer (AI-SME): Initial Investigations into a ChatGPT-enabled MBSE. INCOSE 2024, Dublin, Ireland.

Li, Z., & Avgeriou, P. (2014, 12 27). A Systematic Mapping Study on Technical Debt and Its Management. Journal of Systems and Software, 101, pp 193-220

Lunde, N. W. (2024). Contribution of verification when establishing. Master project paper at University of South-Eastern Norway

Marques, N., Silva, R. R., & Bernardino, J. (2024, 5 21). Using ChatGPT in Software Requirements Engineering: A Comprehensive *Future Internet* **2024**, *16*(6), 180

Mavin, A. W., Wilkinsn, P., Harwoord, A., Noval, M., (2009). Easy Approach to Requirements Syntax (EARS). 17th IEEE International Requirements Engineering Conference. Derby: Atlanta, GA, USA, 2009, pp. 317-322

Muller, G. (2013, 3 19). Systems Engineering Research Methods. CSER 2013, Atlanta, USA

Norheim, J. J., Rebentisch, E., & Xiao, D. (2024). Challenges in applying large language models to requirements engineering tasks. Design Science 2024, Cambridge University Press

Siddeshwar, V., Alwidian, S., & Makrehchi, M. (2024). Systematic Review of AI-Enabled Frameworks in Requirements Elicitation. *IEEE Access*, vol. 12, pp. 154310-154336

Tijan, E., Jovi´c, M., Aksentijević, S., & Pucihar, A. (2021). Digital transformation in the maritime transport sector. Retrieved from https://www.sciencedirect.com/science/article/pii/S0040162521003115?via%3Dihub

Vierlboeck, M., Nilchiani, R., & Blackburn, M. (2024). Natural Language Processing to assess structure and complexity of system requirements INCOSE 2024, Dublin, Ireland

Westerheim, M. (2024, 11 29). Can AI assist in Requirements Engineering. Course paper at University of South-Eastern Norway

Biography

Morten Westerheim

Gerrit Muller

Kistin Falk