Journal of Software Engineering Research and Development, 2019, 6:1, doi: 10.5753/jserd.2019.xxx

Generative Artificial Intelligence in Supporting Functional and Non-Functional Requirements: An Exploratory Study in Requirements Engineering

Josué Ferreira [**Universidade Federal do Pará** | ***josue.ferreira@icen.ufpa***]
Carlos Portela [**Universidade Federal do Pará** | ***csp@ufpa.br***]
Sandro Oliveira [**Universidade Federal do Pará** | ***srbo@ufpa.br***]

Abstract

Requirements Engineering (RE) is fundamental in software development, ensuring that systems meet the needs of stakeholders. This exploratory study analyzes the impact of Generative Artificial Intelligence (GAI) based on Large Language Models (LLMs) in the various stages of RE, with an emphasis on Functional Requirements (FRs) and Non-Functional Requirements (NFRs). Interviews with software engineers reveal that GAIs are more effective in eliciting and specifying requirements, increasing efficiency and reducing manual effort, especially for FRs. However, challenges such as dependence on refinement in prompts, generic responses and difficulties in dealing with complex NFRs still limit their use. The study points out opportunities for improvement, such as integration with RE artifacts and greater personalization of responses. It is concluded that, although promising, GAIs need advances to improve their support for RE and their adaptation to specific contexts.

Keywords: *Requirements Engineering, Large Language Models, Generative Artificial Intelligence, Functional Requirements, Non-Functional Requirements*

1 Introduction

Requirements Engineering (RE) plays a critical role in the development of high-quality systems. The efficient management of both Functional Requirements (FRs) and Non-Functional Requirements (NFRs) is essential to ensure that systems meet stakeholders' needs, providing robustness, scalability, and usability. As pointed out by Kamata and Tamai (2007), the quality of requirements directly impacts software success, influencing both its acceptance and alignment with organizational goals.

However, executing RE activities such as elicitation, analysis, specification, prioritization, validation, and management demands time, expertise, and intensive collaboration among stakeholders. To mitigate these challenges, a variety of techniques and tools have been developed over the years to optimize RE tasks and improve the resulting artifacts. Examples include tools for elicitation Ronanki et al. (2023); Ren et al. (2024), requirements analysis Mahbub et al. (2024), classification Kurtanović and Maalej (2017); Alhoshan et al. (2023), prioritization Felfernig et al. (2018); Vijayakumar and Nethravathi (2021), and requirements quality assurance Parra et al. (2015).

In this context, Artificial Intelligence (AI) technologies have recently seen exponential growth, particularly with the rise of Generative Artificial Intelligence (GAI) assistants. Large Language Models (LLMs), such as OpenAI's GPT-4, have opened new possibilities for RE activities Zhao et al. (2021); Ronanki et al. (2023). These tools, based on Natural Language Processing (NLP), offer innovative capabilities to generate, analyze, and refine software artifacts Yu et al. (2024). With advanced functionalities, LLMs have the potential to significantly reduce manual effort, promote efficiency, and minimize errors in complex tasks.

Despite these advances, a significant gap remains in understanding how LLMs can be optimally applied across the various phases of RE Zhao et al. (2021), especially in tasks related to FRs and NFRs. Several critical questions remain unanswered, such as: Which RE activities benefit most from the use of GAI? How do software engineers craft effective prompts for essential tasks, such as the initial generation of requirements and artifact traceability? Furthermore, how can these tools help ensure emergent properties, consistency, and completeness of documented artifacts, such as the Requirements Specification Document? These questions underscore the need for further investigations into the potential of LLMs and their implications in supporting RE activities, encompassing both FRs and NFRs throughout the process.

Given this scenario, this study explores the impact of generative AI on RE activities, with a focus on FRs and NFRs, from the perspective of software engineers. Through an exploratory approach, the objective is to examine how such assistants are being integrated into contemporary RE practices, including elicitation, analysis, specification, prioritization, validation, and requirements management. The aim is to contribute to the improvement of RE practices by promoting the efficient use of these technologies in the software development industry.

In addition to this introductory section, Section 2 presents the theoretical background on software requirements, requirements engineering, and an overview of generative AI. Section 3 discusses related work. Section 4 details the methodology adopted in this study. Section 5 highlights the results obtained. Section 6 provides a discussion aligned with the research questions. Section 7 presents the study's limitations. Finally, Section 8 concludes the paper.

2 Background

2.1 Software Requirements

According to Sommerville (2018) and Pressman and Maxim (2021), system requirements can be grouped into three main categories: Functional Requirements, Non-Functional Requirements, and Domain Requirements. Functional Requirements (FRs) clearly and objectively specify the functionalities that the system must perform. They detail the operations the system must carry out in response to inputs provided by users or other systems. Non-Functional Requirements (NFRs), in turn, define constraints and conditions related to the system's operation, establishing standards for development, platform, response times, and access permissions. These requirements are associated with desired system qualities, such as performance, security, usability, and scalability. Finally, Domain Requirements address the specific characteristics of the domain in which the system will be applied, reflecting the particularities of its usage environment. They may impose constraints on functional requirements or provide specific calculations based on the domain context. These types of requirements play distinct yet interdependent roles in the development of robust and effective systems, ensuring alignment with the specific needs of the domain and stakeholders.

2.2 Requirements Engineering

Requirements Engineering (RE) plays a crucial role in the software development lifecycle, being fundamental for establishing high-quality requirements that ensure project success Lubos et al. (2024). According to Sommerville (2018), RE activities are responsible for detailing all the functionalities a system must provide, as well as specifying the services and constraints associated with its operation.

For Pressman and Maxim (2021), the goal of RE is to provide all project stakeholders with a shared understanding of the problem. To achieve this, various artifacts (work products) are used to ensure the quality and proper management of software requirements. Thus, RE activities can be organized into six phases: Elicitation, Analysis, Specification, Prioritization, Validation, and Management, as outlined below:

- **Elicitation:** Involves gathering and identifying stakeholders' needs;
- **Analysis**: Focuses on analyzing and refining the elicited requirements, resolving potential conflicts, inconsistencies, or ambiguities;
- **Specification:** Requirements are formalized and documented clearly and precisely, using structured text, diagrams, or formal models;
- **Prioritization:** Ranks requirements based on their value, impact, and urgency, to guide implementation order and allocate resources to the most critical functionalities;
- **Validation:** Ensures that requirements are complete, consistent, and aligned with stakeholders' expectations; and
- **Management:** Involves tracking and controlling requirements throughout the software lifecycle.

Given the complexity of the activities involved in RE and the significant effort required from stakeholders, technologies such as generative AI have emerged as promising tools to optimize these processes Ronanki et al. (2023).

2.3 Generative Artificial Intelligences

Large Language Models (LLMs) are advanced Natural Language Processing models capable of understanding and generating text in a manner similar to human language Min et al. (2023). These models enable the creation and refinement of various artifacts in software development, such as code in multiple programming languages and diagrams, based on the context provided by the user. To achieve this, the user interacts with the LLM through specific queries by embedding the question's context into prompts. The Generative AI then provides an output in natural language Reynolds and McDonell (2021), facilitating the analysis and interpretation of the results.

With recent technological advances, LLMs have emerged as support tools for addressing the challenges posed by Requirements Engineering, optimizing processes that were once highly manual and error-prone. Illustrating this potential, the study by Luitel et al. (2023) applied a model based on Bidirectional Encoder Representations from Transformers (BERT) to expand specific terms in a dataset. The results demonstrate that LLMs, such as BERT, can contribute to the semantic expansion of terms, reducing ambiguities and increasing the completeness and consistency of requirements specifications.

3 Related Works

Wei (2024) developed a customized LLM model to automate code generation from well-structured requirements, introducing the concept of Progressive Prompting. This approach enables stepwise collaboration, from requirements interpretation to test and code generation. Through a case study involving the development of a scheduling system, the model demonstrated the ability to understand complex requirements and produce robust solutions aligned with object-oriented design and test-driven development.

Zhang and Li (2024) investigated the application of LLMs in requirements classification, addressing both binary tasks (i.e., distinguishing between FRs and NFRs) and multi-class tasks (i.e., subcategories of NFRs, such as performance, security, and usability). Despite promising results, the authors identified challenges in more nuanced distinctions, such as specific subcategories. To improve accuracy, the study proposes enhancements in prompt engineering strategies, the use of domain-specific knowledge, and context-based learning techniques.

Hasso et al. (2024) presented Quest-RE, a methodology based on LLMs, such as ChatGPT-4, to enhance requirements elicitation and analysis through dynamic questions and structured strategies. This approach aims to identify gaps and dependencies, having been applied in real-world projects to improve requirements consistency. However, the authors emphasize the importance of human expertise when dealing with more complex contexts, where LLM limitations become evident.

While these studies explore specific applications of LLMs in individual phases of Requirements Engineering, this work distinguishes itself by adopting a more comprehensive approach. It examines the impact of different GAI assistants, such as ChatGPT, Gemini, and Perplexity, across multiple stages of the RE process (elicitation, analysis, specification, prioritization, validation, and management). Furthermore, it seeks to address gaps related to the use of such tools in both FRs and NFRs, analyzing how they can optimize RE practices in an integrated and effective manner.

4 Methodology

This study is characterized by a qualitative and exploratory approach, aiming to understand software engineers' perceptions regarding the impact of Generative Artificial Intelligence tools in supporting tasks related to Functional and Non-Functional Requirements in Requirements Engineering. The qualitative approach enables the investigation of subjective experiences and interpretations, allowing for an in-depth exploration of the nuances in participants' perceptions Denzin and Lincoln (2011). The exploratory nature of the study is justified by the emerging nature of the topic, making it possible to identify practices, challenges, and opportunities associated with the use of LLM-based tools. The following section presents the methodological flow of the study, composed of six stages, as shown in **Figure 1**.

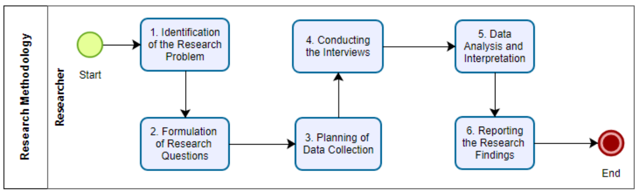

Figure 1. Research Methodological Process.

The study begins with problem definition (Stage 1), identifying gaps in the use of Generative Artificial Intelligence in Requirements Engineering, with a focus on analyzing these tools' support for Functional Requirements (FRs) and Non-Functional Requirements (NFRs). To guide the investigation, in Stage 2, three central research questions were formulated:

- **(RQ1)** How do Software Engineering professionals evaluate the impact of generative AI in supporting functional and non-functional requirements across the different phases of Requirements Engineering?
- **(RQ2)** Which types of functional and non-functional requirements are perceived by professionals to benefit the most from the use of AI assistants?
- **(RQ3)** What challenges and improvement opportunities are identified in the use of generative AI tools to support Requirements Engineering processes?

These questions guide the investigation into the application of generative AI in handling FRs and NFRs throughout the various RE phases (elicitation, analysis, specification, prioritization, validation, and management) where the tools are perceived as most useful (RQ1). Additionally, the research explores aspects related to user requirement generation, interfaces, and quality attributes such as security, performance, usability, and other essential properties for software success (RQ2) Malkawi (2013). Finally, it seeks to identify challenges and improvement opportunities in the use of such tools to maximize their contribution to essential RE processes (RQ3).

In Stage 3, related to data collection planning, semi-structured interviews were conducted with five professionals in the Software Engineering field who have experience both in applying RE processes and using generative AI tools for requirements-related activities. Participant selection was based on a minimum of one year of experience as a Software Engineer, and recruitment was carried out via LinkedIn and email invitations. The interviews addressed a variety of topics aligned with the study's scope, including: the most commonly used generative AI tools; perceived applicability in each RE phase; types of requirements most benefited; recommended prompts for improved accuracy; challenges encountered; and suggestions for improvement. The full interview questionnaire is available at `https://forms.gle/hBDKfAFuajvR9iVX6`.

The interviews were conducted following a structured protocol (Stage 4), with an estimated duration of 30 to 40 minutes. Sessions were held remotely using the Google Meet platform. With the participants' consent, interviews were recorded for documentation and later analysis, ensuring data anonymization and privacy protection.

Subsequently, in Stage 5, data analysis was performed through an interpretative approach, considering the research questions and the study's objectives. The process involved organizing responses according to the aspects mentioned by participants, describing their perceptions regarding the use of AI in Requirements Engineering, and correlating the findings with the theoretical framework adopted.

Finally, in Stage 6, the presentation of results synthesized the main findings concerning the relevance of generative AI tools in RE activities, highlighting the observed challenges, opportunities, and benefits of using these technologies.

5 Results

This section presents the results obtained through the execution of the study's methodology.

5.1 Participant Profiles

Table 1 presents the profiles of the five professionals interviewed. To ensure anonymity, the identifier "P" (for Professional) is used to reference each interviewee. Four participants work as Requirements Analysts, while one holds the position of Product Owner, this individual also has prior experience specifically as a Requirements Analyst.

Table 1. Interviewee Profile.

#	Gender	Age	Experience	AI Tool
P1	F	35	8 years	ChatGPT
P2	M	21	1 year	ChatGPT
P3	M	26	2 years	ChatGPT, Claude.ai
P4	F	33	8 years	ChatGPT/GPT4All
P5	M	29	8 years	ChatGPT

Although the sample size is small, the exploratory nature of the study aims to identify trends and generate initial insights, without requiring a large number of participants, as noted by Rego et al. (2018). It is also worth noting that the presence of senior professionals enriches the discussions, contributing years of experience in overcoming various challenges in Requirements Engineering.

Table 1 also shows that the average age of the participants is 28.8 years, and the average professional experience is 5.4 years, suggesting a blend of perspectives from both junior and senior professionals. Interview durations ranged from 16 minutes and 46 seconds to 39 minutes and 2 seconds, with an average of 30 minutes and 39 seconds, falling within the expected timeframe as estimated in the Methodology section (see Section 3). Regarding the use of generative AIs, 100% of the interviewees confirmed their use in activities related to Requirements Engineering. Among the available options, ChatGPT is used by all participants. Additionally, one participant reported using Claude.ai, while no usage was reported for Gemini or Perplexity. In terms of frequency, 40% (2 professionals) use AI daily for RE-related tasks, while the remaining 60% (3 professionals) use it on a weekly, monthly, or occasional basis. These findings indicate a consistent adoption of such technologies within the software industry.

The following sections present the participants' responses to the study's research questions.

5.2 RQ1. How do Software Engineering professionals evaluate the impact of generative AI in supporting functional and non-functional requirements across the different phases of Requirements Engineering?

RQ1 aims to understand Software Engineering professionals' perceptions regarding the impact of generative AI on processes related to functional and non-functional requirements. This evaluation considers the various phases of Requirements Engineering, including elicitation, analysis, specification, prioritization, validation, and management.

Most interviewees (80%) considered generative AI most useful during the specification phase. Elicitation and validation were mentioned by 40% of the participants. Only one respondent (20%) cited the analysis phase, while prioritization and management were not mentioned by any of them. The predominance of specification as the main phase benefited by AI may be related to the fact that generative tools are effective at text formalization, requirements summarization, and documentation creation. In contrast, the limited mention of the analysis phase (20%) and the absence of prioritization and management suggest that these processes require more critical and contextual judgment, which may limit AI applicability.

The positive impact of generative AI on Requirements Engineering is evidenced by the fact that four interviewees rated its influence as '*Positive*' on a scale ranging from '*Very Negative*', '*Negative*', '*Neutral*', '*Positive*' and '*Very Positive*', while only one participant rated it as 'Neutral'. The use of AI was highlighted, for instance, for its ability to accelerate work during the requirements elicitation phase. However, professionals also emphasized that these tools still require improvements, as noted by P2: "*AI has helped me a lot in boosting productivity, but it needs improvements, such as greater accuracy in understanding the information*". P5 reinforces this view by stating: "*Generative AIs have undeniable potential, but there is still much to be done in terms of output quality and, above all, in terms of the sensitivity/dependence of these tools on the input prompts*".

Another point emphasized by professionals is the AI's ability to provide rapid responses, enabling requirements engineers to analyze a greater number of scenarios. P2 noted that their main activities with AI include elicitation, specification, and the creation of test scenarios, the latter being associated with the validation phase. P2 also reported a specific use case in which they used ChatGPT to support the elicitation phase:"*ChatGPT is a useful tool for the requirements elicitation phase, allowing documents to be uploaded and requirements to be extracted for the system efficiently, contributing significantly to this activity*". P4 used AI to optimize requirement specification and summarize documents during a project: "*During the planning phase of a project involving system development, I used ChatGPT to summarize the submitted documents and later to review the specified requirements*".

On the other hand, some limitations were noted. P5 pointed out that, in the elicitation/discovery phase for complex systems, AI may not fully grasp the project's needs, resulting in insufficient or biased suggestions: "*In complex systems, such as distributed systems, AI may not fully understand the needs of a large-scale project, even if the prompt includes detailed scenarios and business rules. Using AI may become unfeasible, as a major constraint is how much it can suggest in response to the demand, which may result in outputs that do not match the actual needs. This leads to the generation of insufficient or biased data*". P2 reinforced this concern by stating that tools should be integrated with RE artifacts to provide more complete and project-grounded responses: "*I believe that integration with artifacts collected during Requirements Engineering would give AI a better understanding, enabling it to deliver more complete and effective responses*".

Therefore, the overall evaluation by professionals is predominantly positive, though there is a clear need for improvements in response accuracy, adaptation to the specific context of each project, and support for the integrity of the data used. Additionally, the predominance of the specification phase as the one most benefited by AI reflects its effectiveness in documentation structuring, while its lower applicability to analysis and management may be associated with the need for deeper human judgment or intervention.

5.3 RQ2. Which types of functional and non-functional requirements are perceived by professionals to benefit the most from the use of AI assistants?

RQ2 explores which types of functional and non-functional requirements stood out as being most positively impacted by the use of intelligent assistants. This analysis includes both quality attributes, such as security, completeness, and performance, and other perceived contributions, such as support in the initial generation of basic system functionalities.

The interviewees indicated that Functional Requirements (FRs) are the most benefited by the use of generative AI, particularly due to its strong application in elicitation, specification, and validation tasks. AI was used to map functional requirements, usability flows, and database structures by P5 in a system under development. They reported this practical case using Claude.ai: "*Using the Claude Sonnet tool, we performed a mapping of a general view of functional requirements for a system under development. In this scenario, we used the tool to define the interface scope, button actions, usability flow, and database structure*".

The predominance of FRs as the most benefited may be linked to the fact that engineers who already have a well-defined vision of the project tend to achieve better results when crafting more specific prompts. **Figure 2** illustrates which functional requirements are most supported by generative AIs.

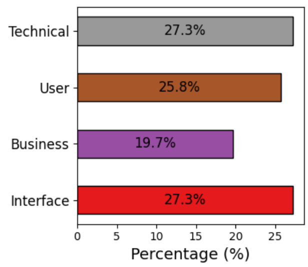

Figure 2. Perceived Support of Generative AI for FRs.

The data reveal that generative AIs assist in addressing all types of functional requirements, with particular emphasis on Technical, User, and Interface Requirements, which represent 27.3%, 25.8%, and 27.3%, respectively. The predominant recommendation highlights ChatGPT—used in 100% of the cases, as the preferred tool for supporting FRs, complemented by Claude.ai. The choice of ChatGPT reflects its ability to aid in defining technical requirements that meet client needs, as well as in synthesizing documents, reviewing specifications, and defining interface functionalities, aligning with the importance of technical, interface, and user requirements.

As for NFRs, AIs face more significant challenges, according to the participants' perceptions, as shown in **Figure 3**. The NFRs most positively impacted by generative AI were Performance and Usability (both with 16.1%). Security (12.5%) also stood out, reinforcing the concern for data protection. Meanwhile, Completeness, Portability, and Scalability (each with 8.9%) were cited as relevant for ensuring coverage, adaptability to different environments, and the ability to handle increasing demands. Maintainability, Operability, Compliance, and Availability requirements (7.1%) received less emphasis, despite being essential for the reliability and sustainability of the software.

As previously mentioned, limitations were emphasized regarding the ability of generative AIs to assist with more complex NFRs, such as security, scalability, and availability. P5 suggested that AI should include reflective questioning before providing answers in order to avoid overly generic responses: "*AI could implement questions, for example, if the user says: 'I want to create a monolith', the AI should ask in what context that statement applies, returning a question such as: 'What is the context for this monolith creation?'*".

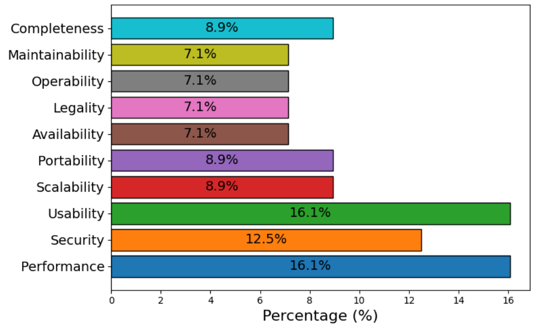

Figure 3. Perceived Support of Generative AI for NFRs.

A relevant point observed during the interviews is that professionals tend to assign less importance to non-functional requirements compared to functional ones. Reflecting this, P4 mentioned using generative AIs solely for performance and security requirements. This indication of a potential neglect of NFRs in software projects is a concern already noted in the specialized literature. Ramos et al. (2019) and Oliveira et al. (2024) emphasize that the lack of consideration for NFRs during the requirements analysis phase can lead to significant project failures.

Overall, the results indicate that while AIs are effective in generating well-structured functional requirements, their applicability to NFRs still requires refinement, particularly in understanding context and tailoring recommendations to the specific needs of a project. Additionally, the validation of NFRs was identified as a promising functionality, provided that AI tools are enhanced to better handle business rules and requirements related to compliance and security. Thus, functional requirements remain the most benefited by generative AI usage, while non-functional requirements could be better supported with improved AI capabilities for contextualizing product needs.

5.4 RQ3. What challenges and improvement opportunities are identified in the use of generative AI tools to support Requirements Engineering processes?

RQ3 addresses the challenges and opportunities associated with the use of generative AI in supporting RE, highlighting technical, methodological, and usability barriers, as well as potential areas for enhancement.

The main challenges identified include:

- **Generic responses and the need for manual adjustments:** P1 noted that AI often generates shallow responses, requiring extensive rework: "*It provides generic information, which increases analysis time and, in some cases, demands complete revisions*". P2 reinforced that AI does not always interpret problems accurately, necessitating constant refinements: "*It's usually necessary to adjust the responses several times to get an acceptable result*".
- **Limitations in specialized domains:** P4, who works in the banking credit domain, pointed out the AI's difficulty in handling highly specialized knowledge: "*The responses are average because the system lacks deep knowledge on specific topics*".
- **Data security and integrity:** P3 raised concerns about privacy and legal compliance when using AI: "*Any AI tool must be approved by management before use, also considering information integrity*".

Despite these challenges, the professionals identified several opportunities for improvement:

- **Response precision and adequacy:** Improvements in technical writing, template generation, and support for requirements traceability (P1).
- **Integration with RE artifacts:** Linking to documents and requirements tools for greater contextual understanding (P2).
- **Information security:** Implementation of secure corporate sessions and greater control over data usage (P3).
- **User interaction optimization:** Providing examples of effective prompts (zero-shot, one-shot, and few-shot) to facilitate the generation of accurate responses (P4).
- **Adaptation to specific project contexts:** Integration with project management tools, wireframe design environments, and interactive support for validating non-functional requirements such as availability and scalability (P5).

P5 also suggested an enhancement to AI interaction by proposing that the system ask questions before delivering recommendations on NFRs: "*AI could include questions to better understand the context before responding, reducing the need for domain-specific expert systems for each area of software development*".

Therefore, while challenges related to accuracy, security, and contextualization persist, there is significant potential for improvement. Customizing AI tools, integrating them with existing RE environments, and enhancing their ability to interpret complex scenarios can transform their application in RE, making the process more efficient, secure, and aligned with project needs.

6 Discussion

The results show that 100% of the participants use generative AI tools in Requirements Engineering activities, with ChatGPT being the predominant tool (100%) and Claude.ai appearing as a complementary option (20%). In terms of frequency, 40% of professionals use AI daily, while 60% use it weekly, monthly, or occasionally, indicating a steady adoption in everyday project activities.

Data analysis reveals that among the various RE phases, specification (80%) and elicitation (40%) are the stages most effectively supported by AI. In contrast, phases such as prioritization and management were not mentioned by any participants, suggesting an uneven application of the technology throughout the process. This discrepancy highlights that, while the benefits are significant in document creation, template definition, and mapping of functional requirements, activities involving value-based decisions, requirements traceability, or change control still require additional capabilities to meet engineers' needs.

Interviewee testimonials illustrate practical uses of AI. For instance, one professional reported using ChatGPT for document summarization and requirement review during project planning, while another described using Claude.ai to define interface scopes, usability flows, and database structure. These use cases reinforce the notion that integrating AI with artifacts and support systems can enhance the effectiveness of RE processes, reducing the need for excessively detailed prompts.

Regarding the handling of FRs and NFRs, the data suggest that although AIs are effective in generating well-structured functional requirements, their application to NFRs still requires improvements to capture context and adapt recommendations to the specific characteristics of each project. Additionally, the validation of NFRs emerged as a promising area, provided that AI tools are improved to handle business rules and requirements related to compliance and security. In this regard, while functional requirements currently benefit most from generative AI, non-functional requirements may be better supported through enhanced contextualization capabilities.

Despite the benefits, participants reported significant challenges. The reliance on detailed prompts to obtain accurate responses often results in generic outputs, which can lead to rework. Moreover, the ability of AI to operate within highly specialized domains, especially those involving sensitive data or complex business rules, remains limited. To address these barriers, participants suggested several opportunities for improvement, such as providing prompt examples based on techniques like zero-shot, one-shot, and few-shot learning; and integrating AI with project management tools to enable a more contextualized approach.

Furthermore, they proposed adapting AI tools to specific contexts, allowing them to ask reflective questions to better understand project needs. These strategies were implemented in the study by Hasso et al. (2024), which proposed customizing prompts to specific contexts, enabling the AI to formulate reflective questions and deepen its understanding of project needs. In their approach, the prompt was structured into two components: a fixed part covering general context and standard instructions, and a variable part that included project-specific information, such as requirements and operational scenarios. This prompt customization strategy contributed to deeper insights into project needs. Thus, the approach described in that study supports the findings of the present research, evidencing that prompt adaptation is a crucial element for enhancing tool effectiveness in specific contexts. Similarly, considering the professionals' observed neglect of NFRs, AIs could reinforce their importance in the project by offering guidance to prevent the formulation of vague or imprecise requirements. The Guide to the Software Engineering Body of Knowledge (SWEBOK) Bourque and Fairley (2014) emphasizes the need to avoid unverifiable requirements that rely on subjective judgment, such as "*the software must be reliable*" or "*the software must be user-friendly*". In this regard, AIs may provide mechanisms to minimize errors related to NFRs in projects.

In summary, to maximize the potential of generative AI tools in Requirements Engineering, it is essential to promote systematic integration with existing process artifacts. Such integration allows the AI to leverage previously collected information, such as documents, specifications, and models, to generate more accurate and contextual responses, reducing dependence on overly detailed prompts. As highlighted by studies in the field Ronanki et al. (2023) and corroborated by the professionals interviewed, incorporating artifacts into the analysis enhances the understanding of project needs, ensuring greater integrity and completeness in specifications. Moreover, integration with management tools fosters a more collaborative and contextualized approach, overcoming limitations inherent to isolated AI use. Therefore, the synergy between generative AIs and Requirements Engineering artifacts emerges as a promising strategy to improve the efficiency, accuracy, and robustness of RE processes, aligned with both FRs and NFRs.

7 Limitations

This study followed the guidelines proposed by Runeson and Host (2009) to discuss potential limitations in Software Engineering research within a qualitative context.

The main limitation of this study is the small number of participants, with only five interviewees, a consequence of the difficulty in recruiting Software/Requirements Engineers, despite widespread outreach. Although the exploratory nature of the research justifies a smaller sample size to identify trends and initial hypotheses, this limitation may reduce the diversity of perspectives and fail to capture all challenges and practices related to the application of generative AI. However, the presence of three senior professionals adds robustness to the findings, as they tend to identify more complex challenges, while junior professionals focus on immediate applicability.

The interviews did not explicitly distinguish among types of functional requirements (Interface, Business, User, Technical) or non-functional requirements (Scalability, Portability, Legal Compliance, Operability, Maintainability, among others). Following the strategy adopted in Ronanki et al. (2023), this approach was intentional, aiming to encourage responses based solely on participants' knowledge and experience.

This study also did not address Domain Requirements, although some limitations in specific domains, such as banking systems, were mentioned. This choice was made because domain requirements reflect the contextual particularities of each organization, which could limit the scope of participant responses and hinder the extraction of generalizable insights. In this sense, the study focused on functional and non-functional requirements, which are broader and more cross-cutting in nature, thus better aligning with the research objectives.

Furthermore, the analysis based on subjective testimonials may introduce individual biases related to each interviewee's experience and context, potentially impacting the validity of the insights. As such, the results should be interpreted as indicative and preliminary. Future studies with larger samples and mixed methodological approaches (both quantitative and qualitative) could validate and consolidate the identified trends, thereby reducing threats to validity and strengthening the robustness of the conclusions.

8 Conclusions and Future Work

The results indicate that generative AI tools, especially ChatGPT, have a positive impact on Requirements Engineering, with particular benefits for functional requirements, facilitating activities such as specification and elicitation. The high frequency of use and the practical examples reported demonstrate gains in efficiency and agility, even though stages such as prioritization and management are not directly supported.

The observed challenges, such as the dependence on detailed prompts and the tendency toward generic responses, highlight the need to improve response personalization and AI integration with support systems. To maximize benefits and mitigate these limitations, it is essential to develop more intuitive interfaces that guide users in formulating more precise instructions, as well as integrate AI with RE artifacts to deliver more contextualized outputs.

Despite the small sample size, the findings serve as a starting point for future investigations, which should expand the number of participants, integrate AI with RE artifacts, improve user interfaces, and provide guidelines for effective prompt formulation.

By enhancing response personalization and strengthening integration with support tools, generative AI can play a more strategic role in Requirements Engineering. This would enable not only the automation of repetitive tasks but also improvements in quality, ultimately contributing to the development of more efficient software aligned with stakeholder needs.

References

Alhoshan, W., Ferrari, A., and Zhao, L. (2023). Zero-shot learning for requirements classification: An exploratory study. *Information and Software Technology*, 159:107202.

Bourque, P. and Fairley, R. (2014). *SWEBOK: Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, Los Alamitos, CA, 3rd edition. Accessed: Mar. 01, 2025.

Denzin, N. K. and Lincoln, Y. S. (2011). *The Sage Handbook of Qualitative Research*. Sage.

Felfernig, A., Stettinger, M., Atas, M., Samer, R., Nerlich, J., Scholz, S., and Raatikainen, M. (2018). Towards utility-based prioritization of requirements in open source environments. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 406–411. IEEE.

Hasso, H., Fischer-Starcke, B., and Geppert, H. (2024). Quest-re question generation and exploration strategy for requirements engineering. In *2024 IEEE 32nd International Requirements Engineering Conference Workshops (REW)*, pages 1–9. IEEE.

Kamata, M. I. and Tamai, T. (2007). How does requirements quality relate to project success or failure? In *IEEE International Requirements Engineering Conference (RE 2007)*, pages 69–78.

Kurtanović, Z. and Maalej, W. (2017). Automatically classifying functional and non-functional requirements using supervised machine learning. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 490–495. IEEE.

Lubos, S., Felfernig, A., Tran, T. N. T., Garber, D., Mansi, M. E., Erdeniz, S. P., and Le, V. M. (2024). Leveraging llms for the quality assurance of software requirements. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 389–397. IEEE.

Luitel, D., Hassani, S., and Sabetzadeh, M. (2023). Using language models for enhancing the completeness of natural-language requirements. In *International Workshop on Requirements Engineering Foundation for Software Quality (REFSQ)*, pages 87–104.

Mahbub, T., Dghaym, D., Shankarnarayanan, A., Syed, T., Shapsough, S., and Zualkernan, I. (2024). Can gpt-4 aid in detecting ambiguities, inconsistencies, and incompleteness in requirements analysis? a comprehensive case study. *IEEE Access*, 12:171972–171992.

Malkawi (2013). The art of software systems development: Reliability, availability, maintainability, performance (ramp). *Human-Centric Computing and Information Sciences*, 3:1–17.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., and Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys (CSUR)*, 56(2).

Oliveira, A., Correia, J., Assunção, W. K. G., Pereira, J. A., de Mello, R., Coutinho, D., Barbosa, C., Libório, P., and Garcia, A. (2024). Understanding developers' discussions and perceptions on non-functional requirements: The case of the spring ecosystem. *Proceedings of the ACM on Software Engineering*, 1(FSE):1–22.

Parra, E., Dimou, C., Llorens, J., Moreno, V., and Fraga, A. (2015). A methodology for the classification of quality of requirements using machine learning techniques. *Information and Software Technology*, 67:180–195.

Pressman, R. S. and Maxim, B. R. (2021). *Software Engineering: A Practitioner's Approach*. McGraw Hill Brasil, 9th edition.

Ramos, F. B. A., Pedro, A., Cesar, M., Costa, A. A. M., Perkusich, M. B., de Almeida, H. O., and Perkusich, A. (2019). Evaluating software developers' acceptance of a tool for supporting agile non-functional requirement elicitation. In *Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE)*, pages 26–42. KSI Research.

Rego, A., Pina, M., and Jr., V. M. (2018). Quantos participantes são necessários para um estudo qualitativo? linhas práticas de orientação. *Revista de Gestão dos Países de Língua Portuguesa*, 17(2):43–57.

Ren, S., Nakagawa, H., and Tsuchiya, T. (2024). Combining prompts with examples to enhance llm-based requirement elicitation. In *48th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1376–1381. IEEE.

Reynolds, L. and McDonell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems (CHI EA '21)*. Association for Computing Machinery.

Ronanki, K., Berger, C., and Horkoff, J. (2023). Investigating chatgpt's potential to assist in requirements elicitation processes. In *49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 354–361. IEEE.

Runeson, P. and Host, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14:131–164.

Sommerville, I. (2018). *Software Engineering*. Pearson, 10th edition.

Vijayakumar and Nethravathi, P. S. (2021). Use of natural language processing in software requirements prioritization – a systematic literature review. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 5(2):152–174.

Wei, B. (2024). Requirements are all you need: From requirements to code with llms. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 416–422. IEEE.

Yu, X., Liu, L., Hu, X., Keung, J., Xia, X., and Lo, D. (2024). Practitioners' expectations on automated test generation. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2024)*, pages 1618–1630.

Zhang, Y. and Li, Y. (2024). The classification of software requirements using large language models: Emerging results and future directions. In *2024 3rd International Conference on Cloud Computing, Big Data Application and Software Engineering (CBASE)*, pages 746–749. IEEE.

Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M. A., Chioasca, E.-V., and Batista-Navarro, R. T. (2021). Natural language processing for requirements engineering: A systematic mapping study. *ACM Computing Surveys (CSUR)*, 54(3):1–41.