Exploring security requirements specification supported by LLM: Results of an exploratory study with entry-level software engineers

Marcelo M. Conti
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (ICMC-USP)
São Carlos, SP, Brazil
marcelo.mmartins@usp.br

Amália Melo
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (ICMC-USP)
São Carlos, SP, Brazil
amalia.melo@usp.br

Lina Garcés
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (ICMC-USP)
São Carlos, SP, Brazil
linagarces@usp.br

Abstract

Specifying security requirements is a difficult task for software engineers, involving the definition of mechanisms to guarantee properties such as confidentiality, integrity, and availability. Given the increasing adoption of Large Language Models (LLMs) to support various requirements engineering activities, we intend to investigate their feasibility for security requirements specification using quality attribute scenarios. In this exploratory study, we analyze entry-level software engineers' perceptions of using LLMs to generate quality scenarios for security requirements descriptions and the impact of this use on the activity. The exploratory study involved junior software engineers who first manually developed quality scenarios based on security requirements and then performed the same task with the aid of LLMs, using two distinct prompt engineering strategies. After each step, participants completed questionnaires containing quantitative and qualitative questions. The data were analyzed using descriptive statistics and open-ended response analysis to identify recurring advantages, limitations, and potential of LLMs as tools to support the specification of security requirements.

Keywords

security Requirements, Large Language Models, Software Engineering, Prompt Engineering, Quality attribute Scenarios

1 Introduction

Security is considered a quality attribute, a.k.a a kind of product non-functional requirement, of software systems [3] and is defined as the product's ability to protect information and data, ensuring that each stakeholder has the appropriate level of access and that the system can withstand attacks by malicious agents [7]. As a top-level quality attribute, security comprises eleven sub-characteristics, including: the CIA triad (Confidentiality, Integrity, and Availability), authenticity, accountability, hazard warning, secure integration, legitimacy, non-repudiation, operational restriction, resilience, fail-safety, and risk identification [7].

Security can be specified through quality attribute scenarios, i.e., a standardized, testable, and unambiguous way to describe a system's quality attributes [3]. A good scenario specifies the source of the stimulus (e.g., users input, external calls), the action that triggers the event (e.g., attack, request), the context in which it occurs (e.g., production, testing, staging, or development environment), the affected artifact (e.g., the whole system, a specific microservice), the expected system response (e.g., data processing as result of a request, failure prevention as result of an attack), and the criteria used to evaluate the effectiveness of that response (e.g., metrics as restoring data, proportion of failures prevented) [3].

In parallel, Large Language Models (LLMs) are among the main applications of machine learning (ML) within the connectionist paradigm, based on the Transformer deep neural network architecture [4]. These models are trained on large volumes of textual data and can understand and generate natural language by learning statistical patterns in this data [4]. In requirements engineering, LLMs techniques have been widely used to support activities such as requirements elicitation, specification, classification, and analysis [5]. With the widespread adoption of LLM-based tools, prompt engineering has emerged as a practice for guiding models in task execution, involving the design, refinement, and evaluation of instructions that generate LLM responses aligned with expected outcomes [9]. Consequently, the quality and structure of the prompt directly influence the quality, accuracy, and relevance of the model's responses [9].

In parallel, considering the increasing use of generative LLMs and prompt engineering techniques, it is necessary to establish standardized metrics and robust benchmarking frameworks to enable fair, unbiased comparisons across approaches. Highlighting this, the study by [10] investigated the use of prompt engineering techniques for generating user stories with the ChatGPT model, finding that the Meta-Few-Shot Prompt approach produced high-quality user stories, reinforcing the potential of prompt engineering to support requirements engineering activities.

In a previous study, [8] provided an overview of the state of the art in LLMs for security requirements engineering. Recent studies have investigated the use of LLMs to support different security-related requirements engineering activities. Hassine [6]investigated the use of LLMs to recover security-oriented traceability links between requirements and objective models. Other studies have employed LLMs in verifying compliance with security laws and standards. The work of Aberkane et al. [1] stands out, as he used ChatGPT to support the assessment of user story compliance with the General Data Protection Regulation (GDPR), employing prompt engineering techniques such as function prompts. The results of these studies demonstrated the potential of models as tools to support security requirements verification and validation activities [8].

Given this context, this work seeks to understand entry-level software engineers' perceptions of the impact of using LLMs to generate security scenario specifications. To this end, an exploratory study was conducted in which participants' perceptions and recurring themes in their reports were analyzed to understand how the use of LLMs can contribute to this activity and to identify opportunities for advancing research in this area.

Therefore, this article is organized as follows. Section 2 presents the methodology adopted for the exploratory study, describing the procedures, participants, materials, and how data were collected and analyzed. Section 3 presents and discusses the results obtained. Finally, Section 4 presents the work's conclusions, its limitations, and opportunities for future work.

2 Methodology

This study is an exploratory study with a predominantly qualitative approach, complemented by descriptive quantitative analyses. Figure 1 provides an overview of the method used.

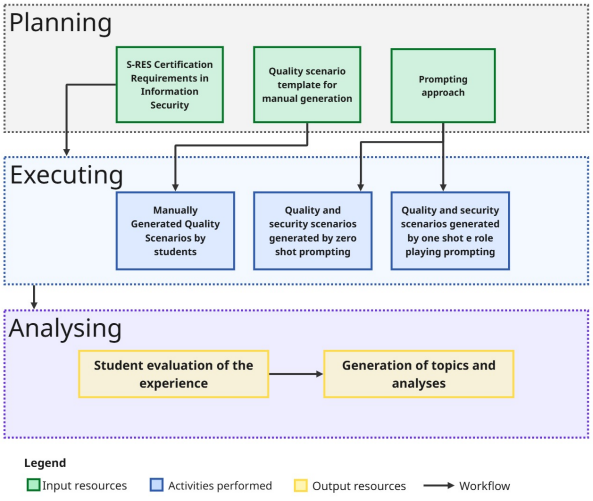

Figure 1: Overview of the study methodology

2.1 Planning

The exploratory study was planned to be conducted through practical activities focused on specifying security requirements using the quality-attribute scenarios template [3]. The participants, characterized as novice software engineers, already possessed prior knowledge of requirements engineering concepts such as: non-functional requirements, security as a quality attribute, and the specification of requirements through quality scenarios. Therefore, all participants had a common conceptual basis for carrying out the activities.

As a basis for all activities, manual specification, and specification with the aid of LLMs, participants selected a security requirement from the document [11]. This document was developed through a partnership between the Brazilian Society of Health Informatics (SBIS) and the Federal Council of Medicine (CFM), with the objective of defining criteria for evaluating and certifying security, privacy, and compliance aspects of Electronic Health Record Systems [12]. The document [11] presents each requirement with a coded identification, a title, and a description, including examples and explanatory notes when pertinent. The requirements are organized into security levels, each composed of thematic groups. These thematic groups include: software version control, user identification and authentication, authorization and access control, availability, communication between components, data security, auditing, documentation, time, integrity, and privacy.

Furthermore, this work explores two distinct prompt engineering strategies: (i) *One-shot with role-playing*, is a strategy in which the model receives a specific role ("security requirements analyst"), a task description, and an example of the expected response structure; and (ii) *Zero-shot* prompting, is a strategy in which the model receives only the instruction to develop a quality scenario based on the security requirement, without additional examples. We aimed to analyze participants' perception of how different forms of instruction influence LLMs' generation of quality scenarios for security requirements.

2.2 Executing

Forty-eight junior engineers participated in this study, which was comprised of two stages lasting two hours.

In the first stage, each participant manually created a security quality scenario using the template in Bass et al. [3] for the selected security requirement extracted from [11]. The scenario was required to encompass the six elements of a quality attribute scenario: source, stimulus, artifact, environment, response, and response measure.

As a second stage, the participants used the same security requirement to automatically generate security scenarios with the aid of an LLM. The prompts used in this stage are available at [2].

After completing each stage, participants answered a questionnaire comprising both quantitative and qualitative questions. The quantitative questions used a five-point Likert scale to assess: (i) the perceived difficulty of performing the manual task and (ii) the participants' level of confidence in using LLMs to support the creation of security attribute scenarios. The qualitative questions aimed to identify the tool and LLM model used, compare the quality of responses generated by different prompting strategies, compare the results obtained against the manual specification, and record the participants' perceptions. The forms are available at [2].

2.3 Analysing

The participants' open-ended responses were analyzed using an LLM (NotebookLM, Gemini 3.5) to support qualitative data analysis. To make the process more structured and reproducible, the analysis was broken down into four independent stages, each conducted by a specific prompt. This decomposition of the analysis reduced the complexity of the tasks assigned to the LLM, making each prompt responsible for a single analytical objective, thus promoting consistency in the results, as well as increasing the transparency and traceability of the process, since all interpretations were based on textual evidence extracted from the participants' comments.

In the first stage, each participant's preference regarding the investigated question was classified based on their comments. Next, an inductive thematic analysis was performed, in which emerging categories were identified from the responses. Subsequently, these categories were associated with preferences, thereby identifying the main criteria participants used to justify their choices. Finally, the results of the previous stages were consolidated, integrating the observed qualitative patterns and frequencies. The prompts and their responses are available in [2].

3 Results and Discussion

This section presents the results obtained from the analysis of the questionnaires answered by 48 participants after the completion of the proposed activities. The collected data were analyzed to identify patterns related to the perceived difficulty of the task, the LLM-based tools adopted, and the participants' confidence in using these technologies. In addition to presenting the results, possible factors that influenced the participants' perceptions are discussed, seeking to interpret the findings based on the literature and the context of the experiment. Tables 1 and 2 summarize the main results obtained during the data analysis.

Regarding the perceived difficulty in creating quality scenarios for security requirements, participants evaluated the activity using a five-point Likert scale, ranging from 1 (*very difficult*) to 5 (*very easy*). The responses indicate that the activity was perceived as having neutral difficulty, since approximately 68.75% (33/48) of participants rated it at level 3 on the Likert scale, corresponding to the perception that the task was neither very difficult nor very easy. Only for 12.5% (6/48) of participants was the manual specification of security requirements easy. This result can be explained by their prior knowledge of quality-attribute scenarios.

Regarding the **LLM tools** used during the experiment, participants were free to choose the LLM-based solution they preferred. As shown in Table 1, the Gemini was the most-used tool, followed by the free versions of ChatGPT and Claude, as well as other solutions. The greater adoption of Gemini maybe related to the availability of a basic plan accessible to all participants, which offered additional features compared to other free options.

The **participants' confidence in using LLMs** to support the specification of security scenarios was evaluated using a five-point Likert scale, ranging from 1 (*none confidence*) to 5 (*complete confidence*). As shown in Table 1, the results indicate a positive perception of these tools, with approximately 81.25% (39/48) of the participants reporting high confidence levels (ratings of 4 or 5) in using LLMs for this activity.

Table 2 presents the participants' preferences regarding the approach that produced the best quality scenarios for security requirements. The results demonstrated a preference for using LLMs over manual elaboration, with this option chosen by 83.3% (40/48) of participants, while only 8.3% (4/48) considered manual specification alone or the complementary (manual + LLM) option superior. Among the prompt engineering strategies, the *One-Shot approach with role-playing* was identified as the most effective by 91.66% (44/48) of participants, while only 4.16% preferred the Zero-Shot approach.

Table 1: Summary of the main results obtained from the questionnaire.

Evaluated aspect	Result	Count
Difficulty level	Level 1 (Very difficult)	2
	Level 2	7
	Level 3	33
	Level 4	6
	Level 5 (Very easy)	0
LLM used	Gemini	23
	ChatGPT	15
	Claude	7
	Other tools	3
Confidence in using LLMs	Level 1 (No confidence)	0
	Level 2	2
	Level 3	7
	Level 4	29
	Level 5 (Complete confidence)	10

The objective of this studyis understand participants' perceptions of the use of LLM-based tools in specifying quality scenarios for security requirements. However, the participants' preference for scenarios generated with the aid of LLMs suggests a perception of higher quality compared to manual specification. This result may be related to the complexity of the activity, which requires technical knowledge and experience in Requirements Engineering and software development to create adequate scenarios. Despite this potential, the results do not indicate that the use of LLMs should replace the role of the expert. Even with the confidence shown by the participants, the models can produce incorrect or inconsistent information.

Table 2: Distribution of participants' preferences.

Comparison	Alternative	Participants	Percentage
Manual vs. LLM	LLM	40	83.33%
	Manual	4	8.33%
	Complementary	4	8.33%
Zero-Shot vs. One-Shot	One-Shot	44	91.66%
	Zero-Shot	2	4.16%
	Equivalent	2	4.16%

Table 3 presents the categories identified during the qualitative analysis of the participants' responses, as well as their respective frequencies of occurrence. The results are organized according to comparisons between manual elaboration and the use of LLMs, and between the prompting approaches: Zero-Shot or One-Shot with role-playing.

The main aspects identified by the participants were generally predominantly associated with a preference for the One-Shot approach with role-playing. This preference is related to the perception that providing examples improves the quality of the generated specifications, functioning as a guiding mechanism for the model. According to the participants (2, 3, 9, 11, 12, 21, 23, 27, 29, 31, 36, 37, 40, 45), this approach produced more organized, consistent, and adherent responses to the expected format, in addition to better reflecting the objectives of the task. In contrast, the Zero-Shot approach was associated by the participants with the generation of more generic and less targeted responses.

Table 3: Categories identified through the thematic analysis of participants' responses and their associated preferences.

Study	Category	Freq.	Preference
Manual vs. LLM	Completeness and richness of detail	26	LLM
	Technical and structural quality	13	LLM
	Knowledge and access to information	9	LLM
	Time efficiency	5	LLM
	Human expertise and intentionality	9	Manual
	Human supervision and validation	6	Both
	Dependence on context and prompting technique	10	Both
Zero-Shot vs. One-Shot	Structural alignment and formatting	15	One-Shot
	Objectivity and conciseness	13	One-Shot
	Richness of detail and depth	12	One-Shot
	Model contextualization	10	One-Shot
	Technical quality and accuracy	7	One-Shot
	Adherence to the task objective	9	Both

The results indicate a positive perception among participants regarding the use of LLMs as tools to support the development of quality scenarios for security requirements. This perception was attributed to the models' ability to generate complete, detailed, and organized artifacts in the expected format. However, participants also highlighted the need for human review of the results. According to their reports (participants 5, 8, 17, 29, 31, and 42), the role of the expert remains essential to validate specifications, correct inconsistencies, and incorporate area-specific knowledge.Furthermore, the results show that the quality of the generated specifications depends on how the task is presented. In this context, prompt engineering techniques play a fundamental role in guiding the generation of more adequate, consistent, and reliable responses.

4 Conclusions and Future Works

This paper investigated the use of LLMs to support the specification of security requirements through quality attribute scenarios. The results indicate that LLMs can effectively assist requirements engineers in producing semi-formal security specifications, with participants generally perceiving the generated scenarios as useful and of good quality. Although participants expressed confidence when interacting with LLMs using prompts prepared by specialists, they reported difficulties in creating effective prompts on their own. This finding suggests that, beyond improving LLM capabilities, there is a need for better prompting guidance, reusable prompt templates, and training to enable practitioners to effectively incorporate LLMs into requirements engineering activities.

This study has several limitations, including the relatively small participant sample size, the controlled experimental setting, the use of expert-designed prompts, participants' freedom to choose different LLMs, and the evaluation of only one security requirement per participant. Future work will address these limitations by considering additional application domains, larger and more diverse participant groups, and real-world industrial settings. We also plan to investigate automated quality assessment methods for generated requirements, compare different prompting strategies and LLMs under controlled conditions, and evaluate LLM-based assistants integrated into software development workflows to better understand their reliability, usability, and practical value for security requirements engineering.

Artifact Availability

Artifacts are available on Zenodo at [2].

Declaration on AI Use

AI tools were used for thematic analysis of participants' open-ended responses and for language refinement and grammar correction. They were not involved in the study design, data collection, or interpretation. All scientific content and conclusions are solely the responsibility of the authors.

Acknowledgements

The authors thank the PRPI-USP (Grant: 22.1.09345. 01.2) and Brazilian agencies for financial support: CNPq (Grant: 406941/2025-4, 420025/2023-5), and FAPESP (Grant: 2024/13482-8).

References

[1] Abdel-Jaouad Aberkane, Seppe vanden Broucke, Geert Poels, and Georgios Georgiadis. 2024. Leveraging ChatGPT for GDPR Compliance in Requirements Engineering: A Pilot Study. In *2024 IEEE International Conference on Security, Privacy, Anonymity in Computation and Communication and Storage (SpaCCS)*. 34–41. doi:10.1109/SpaCCS63173.2024.00012

[2] Anonymous. 2026. Exploring Security Requirements Specification Supported by LLM: Results of an Exploratory Study with Entry-Level Software Engineers. doi:10.5281/zenodo.21329935 Preprint. Acesso em: 12 jul. 2026.

[3] Len Bass, Paul C. Clements, and Rick Kazman. 2021. *Software Architecture in Practice* (4 ed.). Addison-Wesley Professional. 464 pages.

[4] Katti Faceli, Ana Carolina Lorena, João Gama, Tiago Agostinho de Almeida, and André Carlos Ponce de Leon Ferreira de Carvalho. 2025. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina* (3 ed.). LTC, Rio de Janeiro. https://app.minhabiblioteca.com.br/reader/books/9788521639213/ E-book. Acesso em: 2 jul. 2026.

[5] Alessio Ferrari and Paola Spoletini. 2025. Formal requirements engineering and large language models: A two-way roadmap. *Information and Software Technology* 181 (2025), 107697. doi:10.1016/j.infsof.2025.107697

[6] Jameleddine Hassine. 2024. An LLM-based Approach to Recover Traceability Links between Security Requirements and Goal Models. In *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering* (Salerno, Italy) *(EASE '24)*. Association for Computing Machinery, New York, NY, USA, 643–651. doi:10.1145/3661167.3661261

[7] International Organization for Standardization (ISO). 2023. ISO/IEC 42001:2023 – Information technology – Artificial intelligence – Management system. https://www.iso.org/standard/78175.html International Standard. Accessed: 2 Jul. 2026.

[8] Amália Melo, Lucas Almeida, and Lina Garcés. 2025. Investigating the use of Large Language Models in software security requirements: Results of a literature review. In *Anais do IV Workshop Brasileiro de Engenharia de Software Inteligente* (Recife/PE). SBC, Porto Alegre, RS, Brasil, 37–42. doi:10.5753/ise.2025.14892

[9] James Phoenix and Mike Taylor. 2024. *Prompt Engineering for Generative AI*. O'Reilly Media, Inc. 422 pages.

[10] Reine Santos, Gabriel Freitas, Igor Steinmacher, Tayana Conte, Ana Oran, and Bruno Gadelha. 2025. User Stories: Does ChatGPT Do It Better? (03 2025).

[11] Sociedade Brasileira de Informática em Saúde. 2021. Requisitos para Certificação de Sistemas de Registro Eletrônico em Saúde: Categoria Segurança da Informação. https://sbis.org.br/certificacao/v5.2/Requisitos_Certificacao_SBIS_Seguranca_V5.2.pdf Versão 5.2. Acesso em: 2 jul. 2026.

[12] Sociedade Brasileira de Informática em Saúde. 2024. Certificação de Software – Certificação de Sistemas de Registro Eletrônico em Saúde (S-RES). https://sbis.org.br/certificacoes/certificacao-software/ Disponível em: https://sbis.org.br/certificacoes/certificacao-software/. Acesso em: 12 jul. 2026.