MIRA: Automated Generation of Measurable Non-Functional Requirements and Test Scenarios Using Multi-Agent Systems and RAG

Ana Klyssia Martins Vasconcelos[1], **Edson Rodrigo Pinheiro Moreira**[1], **Rubens Abraão da Silva Sousa**[1], **Gustavo Cesar Venancio Monteiro**[1] **Alan Portela Bandeira**[1], **Jerffeson Teixeira de Souza**[1] **Paulo Henrique Mendes Maia**[1], **Ismayle de Sousa Santos**[1]

[1]Programa de Pós-Graduação em Ciência da Computação
Universidade Estadual do Ceará (UECE) - Fortaleza, CE - Brazil

{ana.klyssia, edson.moreira, abraao.sousa, gustavo.cesar, alan.portela}
@aluno.uece.br

{ismayle.santos, jerffeson.souza, pauloh.maia}@uece.br

Abstract. *The specification of Non-Functional Requirements (NFRs) is critical to software quality, but suffers from human subjectivity and the hallucination tendency of generic LLMs. This work addresses the gap in the automated generation of accurate and testable NFRs from high-level user stories. We propose MIRA, a multi-agent architecture based on Retrieval-Augmented Generation (RAG) that orchestrates the contextualized production of requirements and test scenarios. From 11 representative user stories, MIRA automatically generated 25 NFRs and 49 test scenarios across 10 functional domains. Validation with 15 software professionals, totaling 165 evaluations per dimension, demonstrated high clarity ($\mu = 4.45$), measurability ($\mu = 4.39$) and test adherence ($\mu = 4.47$) on a 5-point Likert scale, with strong internal consistency (Cronbach's $\alpha > 0.84$). It is concluded that the approach mitigates generative inconsistencies, acting as an effective accelerator in the integration between requirements engineering and validation.*

1. Introduction

Non-Functional Requirements (NFRs) play a critical role in software quality, defining attributes such as security, usability, and performance. However, empirical studies show that NFRs are frequently specified late, described ambiguously, and lack measurable validation criteria, compromising traceability, testability, and software quality [Viviani et al. 2023, Kula et al. 2022].

Large Language Models (LLMs) have recently emerged as promising approaches to support software engineering activities, including requirements generation and test case creation [Khan et al. 2025]. Nevertheless, standalone LLMs remain prone to hallucinations and domain inconsistencies, often generating vague or non-verifiable NFRs [Gao et al. 2023]. This limitation is particularly critical in NFR specification, where precision and measurability are essential.

Recent studies suggest that combining Multi-Agent Systems (MAS) and Retrieval-Augmented Generation (RAG) can improve the robustness and contextualization of LLM-based systems. MAS enables task decomposition through specialized agents, while RAG provides access to external domain knowledge during generation [Yao et al. 2023, Gao et al. 2023]. However, the integration of these approaches for the automatic generation of measurable NFRs and corresponding test scenarios remains underexplored.

To address this gap, this work proposes MIRA (*Modelagem Inteligente de Requisitos e Avaliação*), a RAG-based multi-agent architecture designed to transform user stories into measurable Non-Functional Requirements and executable test scenarios. The following research question guides this study:

How can a multi-agent architecture supported by LLMs and RAG assist in the automated generation of measurable Non-Functional Requirements and their corresponding test scenarios?

To investigate this question, we propose MIRA (*Modelagem Inteligente de Requisitos e Avaliação*), a RAG-based multi-agent architecture designed to interpret user stories and simultaneously generate both the detailed NFR specification and the test scenarios required for its validation.

This paper is organized as follows: Section 2 presents the theoretical background and related work; Section 3 describes the MIRA architecture and the experimental protocol; Section 4 discusses the results and presents the concluding remarks and directions for future work.

2. Background and Related Work

The specification of measurable Non-Functional Requirements (NFRs) remains a persistent challenge in software engineering due to ambiguity, subjectivity, and the absence of objective validation criteria [Sommerville 2011, Viviani et al. 2023]. The ISO/IEC/IEEE 29148:2018 standard [ISO/IEC/IEEE 2018] establishes quality attributes such as clarity, completeness, and measurability, serving as the normative basis adopted in this work.

Recent studies have explored the use of Large Language Models (LLMs) for software engineering tasks, including requirements generation and test case creation [Khan et al. 2025]. Despite promising results, standalone LLMs remain susceptible to hallucinations, inconsistent outputs, and low coverage of edge cases, requiring mechanisms capable of improving contextual grounding and generation reliability [Wang et al. 2024, Li et al. 2025].

To address these limitations, recent approaches have combined Multi-Agent Systems (MAS) and Retrieval-Augmented Generation (RAG). MAS enables task decomposition through specialized agents, while RAG supports contextualized generation through external knowledge retrieval [Yao et al. 2023, Lewis et al. 2020]. However, the integration of these approaches for the automated generation of measurable NFRs and corresponding executable test scenarios remains underexplored in the literature.

This work addresses this gap through MIRA, a RAG-based multi-agent architecture designed to generate verifiable NFRs aligned with ISO/IEC/IEEE 29148 and their

corresponding test scenarios directly from user stories.

3. Method

This study followed the Design Science paradigm [Hevner et al. 2004] and was organized into two phases: (i) MIRA development and (ii) experimental evaluation.

3.1. MIRA System Architecture

MIRA comprises four components (Figure 1): the Integrator Agent, the RAG Generator Agent, the Data Layer, and the Client APP. User stories are retrieved from Jira and stored in both a relational database and a vector database to support semantic retrieval.

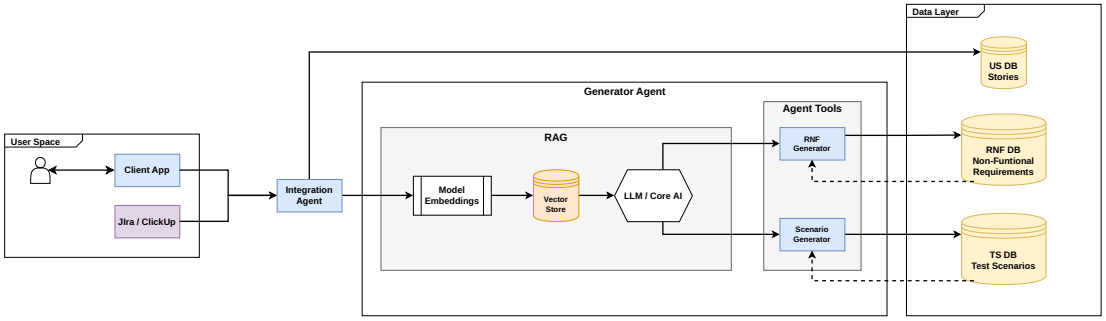

Figura 1. MIRA Architecture

The RAG Generator Agent retrieves contextual information and invokes Gemini 2.5 Flash through specialized prompts. Two specialized generators produce the artifacts: the NFR Generator and the TC Generator, responsible for generating non-functional requirements and corresponding test scenarios.

The Data Layer maintains traceability between generated artifacts and their originating user stories. Source code and artifacts are publicly available for reproducibility purposes [Vasconcelos et al. 2026].

3.2. Experimental Evaluation

The BADCamp dataset [Dalpiaz 2018] was adopted, comprising 69 user stories distributed across 10 functional domains. The dataset was selected due to the diversity of non-functional concerns it includes, such as security, access control, and availability.

Stratified sampling with analyst triangulation [Seaman 1999] was applied to select 11 representative user stories, covering all functional domains. Two researchers independently categorized the stories, with disagreements resolved by a third reviewer.

Fifteen software professionals participated in the evaluation, including QA Analysts, Requirements Analysts, and one Product Owner. Each participant evaluated the generated artifacts for all 11 user stories, totaling 165 observations per dimension.

NFR quality was assessed according to ISO/IEC/IEEE 29148:2018 [ISO/IEC/IEEE 2018] using five criteria: Clarity, Measurability, Relevance, Feasibility, and Completeness. Test scenarios were evaluated through Adherence, Clarity, Execution Viability, Coverage, and Condition Quality. All criteria were scored on a five-point Likert scale.

4. Results and Discussion

The 15 participants evaluated artifacts for all 11 user stories, totaling 165 observations per dimension. Full artifacts are available in the replication package [Vasconcelos et al. 2026].

4.1. Quantitative Evaluation

MIRA generated 25 NFRs and 49 test scenarios (avg. 2.27 NFRs/story; 1.96 scenarios/NFR), each NFR paired with at least one explicit validation scenario, addressing the recurring gap noted by [Viviani et al. 2023]. Strong semantic coherence was observed: financial stories yielded PCI DSS and AES-256 requirements (U.S.30), while navigation stories emphasized accessibility and performance, contrasting with the domain-context limitation of standalone LLMs [Khan et al. 2025]. Table 1 consolidates the results; all dimensions exceeded μ=4,0, with medians of 4 or 5.

Criterion	Mean	SD	Med.
NFRs ($\alpha = 0{,}846$)			
Clarity	4.45	0.75	5
Measurability	4.39	0.81	5
Relevance	4.44	0.75	5
Feasibility	4.25	0.85	4
Completeness	4.22	0.80	4
Test Scenarios ($\alpha = 0{,}853$)			
Adherence	4.47	0.71	5
Scenario Clarity	4.35	0.84	5
Execution Viability	4.41	0.82	5
Coverage	4.05	0.87	4
Condition Quality	4.05	0.95	4

Tabela 1. Evaluation results (Likert 1–5; $n = 165$/dimension).

Figure 2(a) shows similar profiles above 4.0 for both artifact types. The lowest scores, Completeness (μ=4,22) and Condition Quality (μ=4,05), indicate edge-case refinement as the primary improvement opportunity. Reliability was confirmed by Cronbach's Alpha (α>0,84) [Hair et al. 2019]; Figure 2(b) shows all Spearman correlations significant (p<0,001, ρ=0,45–0,75), confirming convergent validity. Performance was higher in well-characterized domains, Summit and Session Management (μ=4,55), Financial Processing (μ=4,43), and lower in generic ones such as Sponsorship Management (μ=3,99) [Wang et al. 2024], pointing to future work on enriching the vector database with domain-specific documentation. The Mann-Whitney U test (p<0,0001) confirmed that QA Analysts (μ=4,49) scored higher than Requirements Analysts (μ=3,97), reflecting their focus on executability, where MIRA performs best, versus stricter completeness and traceability criteria. Both groups position MIRA as an effective cognitive support tool, complementing rather than replacing human work.

4.2. Threats to Validity

Construct validity was addressed by grounding the instrument in ISO/IEC/IEEE 29148 [ISO/IEC/IEEE 2018] and SMART criteria [Doran 1981], confirmed by α>0,84 and Spearman correlations (ρ=0,45–0,75; p<0,001). Internal validity risks from sample size and the junior-heavy profile were mitigated by stratified

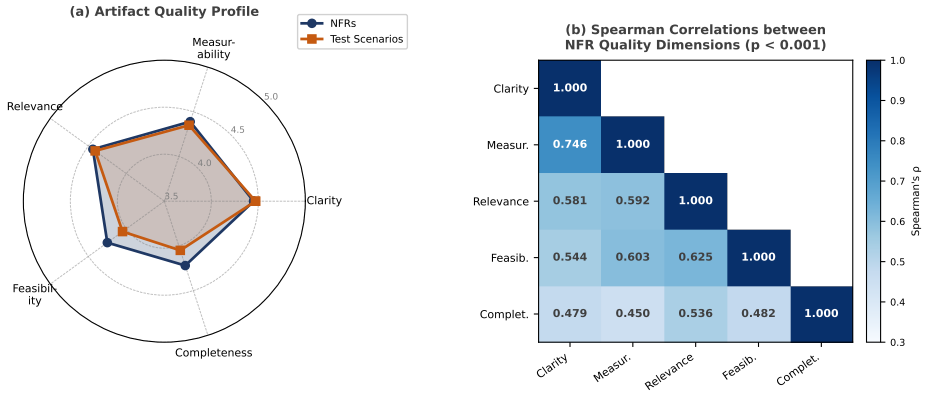

Figura 2. (a) Quality profile: NFRs vs. test scenarios. (b) Spearman correlation matrix (p<0,001, all pairs).

sampling and analyst triangulation [Seaman 1999], with the evaluator profile being representative of MIRA's intended audience. Results are limited to the BADCamp domain and should be generalized with caution. The absence of controlled baseline comparisons constitutes the main threat to conclusion validity, mitigated by the Design Science evaluation strategy and robust statistical evidence across all dimensions.

5. Conclusion

This work presented MIRA, a RAG-based multi-agent architecture for the automated generation of measurable NFRs and test scenarios from user stories. From 11 stories drawn from the BADCamp dataset, the agent generated 25 NFRs and 49 test scenarios, directly addressing the research question: LLMs structured within multi-agent architectures with RAG can automate the production of measurable and directly testable NFRs.

The evaluation conducted with 15 professionals (165 observations per dimension) yielded mean scores above 4.0 across all dimensions, with particular highlight for Adherence to NFR ($\mu = 4{,}47$) and Clarity ($\mu = 4{,}45$). Instrument reliability was confirmed by Cronbach's Alpha ($\alpha > 0{,}84$) and the convergent validity of the Spearman correlations ($\rho = 0{,}45$–$0{,}75$; $p < 0{,}001$). The dimensions of Completeness ($\mu = 4{,}22$) and Coverage ($\mu = 4{,}05$) indicate that the agent benefits from specialized review in contexts that demand greater exhaustiveness, positioning MIRA as a support tool for human work rather than a replacement for it.

As directions for future work, we intend to compare MIRA against controlled baselines, extend the evaluation to multiple domains, and investigate strategies for the automatic refinement of test scenarios, with a focus on exception case coverage.

Acknowledgments

This work was developed with support from the National Council for Scientific and Technological Development - CNPq (Process 312173/2025-3) and Brazilian Federal Agency for Support and Evaluation of Graduate Education - (CAPES)

Referências

Dalpiaz, F. (2018). Requirements data sets (user stories).

Doran, G. T. (1981). There's a s.m.a.r.t. way to write management's goals and objectives. *Management Review*, 70(11):35–36.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H., and Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Hair, J. F., Black, W. C., Babin, B. J., and Anderson, R. E. (2019). *Multivariate Data Analysis*. Cengage Learning, 8 edition.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

ISO/IEC/IEEE (2018). *29148:2018 – Systems and Software Engineering: Life Cycle Processes – Requirements Engineering*. International Organization for Standardization.

Khan, J. A., Qayyum, S., and Dar, H. S. (2025). Large language model for requirements engineering: A systematic literature review.

Kula, E., Greuter, E., van Deursen, A., and Gousios, G. (2022). Factors affecting on-time delivery in large-scale agile software development. *IEEE Transactions on Software Engineering*, 48:3573–3592.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, pages 9459–9474.

Li, Y., Liu, P., Wang, H., Chu, J., and Wong, W. E. (2025). Evaluating large language models for software testing. *Computer Standards & Interfaces*, 93:103942.

Seaman, C. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25:557–572.

Sommerville, I. (2011). *Engenharia de Software*. Pearson, São Paulo, Brasil, 9 edition.

Vasconcelos, A. K. M., Moreira, E. R. P., Sousa, R. A. S., Monteiro, G. C. V., Bandeira, A. P., de Souza, J. T., Maia, P. H. M., and Santos, I. S. (2026). Replication package for semish 2026. `https://anonymous.4open.science/r/replication_package_semish_2026-2FEC/README.md`. Public repository for replication package.

Viviani, L., Guerra, E., Melegati, J., and Wang, X. (2023). An empirical study about the instability and uncertainty of non-functional requirements. In *International Conference on Agile Software Development*, pages 77–93. Springer Nature Switzerland Cham.

Wang, J., Huang, Y., Chen, C., Liu, Z., Wang, S., and Wang, Q. (2024). Software testing with large language models: Survey, landscape, and vision. *IEEE Transactions on Software Engineering*, 50(4):911–936.

Yao, S. et al. (2023). React: Synergizing reasoning and acting in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.