Using LLMs for Requirements Specification in Data Science Projects: An Industrial Experience Report

Murilo Coelho
Atlântico Institute
State University of Ceará
Fortaleza, Ceará, Brazil
paulo_coelho@atlantico.com.br

Vitor Linhares
Atlântico Institute
State University of Ceará
Fortaleza, Ceará, Brazil
vitor_fontenele@atlantico.com.br

Dione Amancio
Atlântico Institute
State University of Ceará
Fortaleza, Ceará, Brazil
dione_amancio@atlantico.com.br

Matheus Paixao
State University of Ceará
Fortaleza, Ceará, Brazil
matheus.paixao@uece.br

Sávio Freire
Federal Institute of Ceará
Morada Nova, Ceará, Brazil
State University of Ceará
Fortaleza, Ceará, Brazil
savio.freire@ifce.edu.br

Paulo Maia
State University of Ceará
Atlântico Institute
Fortaleza, Ceará, Brazil
pauloh.maia@uece.br

Abstract

The integration of Data Science (DS) into complex domains, such as hydroelectric plants, presents unique challenges for Requirements Engineering, primarily due to a high degree of uncertainty and a strong emphasis on non-functional requirements. Consequently, the specification of requirements within these DS projects is often overlooked, heightening the risk of misalignment between technical execution and business value. This paper presents an industrial experience report on the adoption of Large Language Models (LLMs), specifically Gemini, to support requirements specification in a DS project within the power generation sector at the Atlântico Institute. We detail a structured, human-in-the-loop workflow that utilizes anonymized prompts to translate vague business goals into comprehensive technical specifications, including User Stories, Behavior-Driven Development scenarios, and acceptance criteria, while adhering to data governance policies. Our internal evaluation indicates that this LLM-assisted approach reduced the time required to write specifications and improves communication and enhances the quality of the requirements, ensuring that deliverables meet both rigorous technical standards and stakeholder needs.

Keywords

Large Language Models, Data Science, Requirements Specification, Industrial Settings

1 Introduction

The software industry has witnessed a significant surge in the integration of Data Science (DS) and Artificial Intelligence (AI) models into everyday business solutions [5, 11]. However, unlike traditional software engineering, where the scope, target users, and expected outcomes can often be defined with reasonable clarity upfront, DS projects are characterized by a high degree of uncertainty [6, 9]. The exploratory nature of data, combined with rapidly evolving business expectations, frequently results in vague initial scopes and unpredictable development lifecycles [12, 17]. This fundamental difference challenges standard software development practices, requiring teams to navigate an environment where solutions emerge iteratively as the understanding of the data deepens [10].

Within this uncertain landscape, Requirements Engineering (RE) emerges as a vital, yet often overlooked, discipline [18, 20]. While development teams frequently prioritize model performance and technical experimentation, the specification of requirements is often relegated to a secondary plane [20]. This lack of structure heightens the risk of misalignment between technical outputs and actual business value, resulting in solutions that fail to address the core needs of stakeholders [4]. Ultimately, it is the requirement itself that serves as the fundamental anchor; precisely defined requirements are the bridge that translates high-level business goals into actionable data tasks, providing the necessary guidance to navigate the technical complexities inherent in data modeling [3].

In recent years, the emergence of Large Language Models (LLMs) has introduced a transformative paradigm in Software Engineering, particularly within the scope of RE [1, 2, 15, 16]. These models, capable of processing and generating human-like text with high contextual awareness, offer significant potential to streamline the elicitation, refinement, and specification of requirements [15, 16]. In DS projects, where the translation of vague business goals into technical specifications is a major bottleneck, LLM-based tools act as a powerful catalyst [7, 8, 20]. By leveraging natural language processing to suggest, validate, and structure requirements, these tools help reduce the cognitive load on teams and minimize the semantic gaps between stakeholders and technical experts, fostering a more agile and assertive definition process [14, 19].

This paper presents an industrial experience report based on a DS project developed at the Atlântico Institute for a partner in the power generation sector. While the proposed LLM-assisted workflow is fundamentally a general-purpose RE technique applicable to broad software development scenarios, this report illustrates its practical effectiveness in a DS environment. Specifically, it demonstrates how these models can alleviate the burden of specifying complex, mostly non-functional requirements that demand advanced expertise in both Data Engineering and DS [13, 19]. The target audience for this report includes Product Owners, Requirements Engineers, and Data Science practitioners navigating the challenges of requirement specification in industrial settings. Furthermore, to ensure compliance and confidentiality, no sensitive or proprietary data from the partner was used in this report.

2 Contextualization

This study was conducted at the Atlântico Institute, a Research and Technology Organization (RTO) located in Fortaleza. Atlântico serves as an Embrapii[1] unit for intelligent manufacturing, where Embrapii acts as a funding agency that fosters industrial innovation projects. The Institute possesses core expertise in DS and software development, leveraging the Scrum framework for agile project management. The industry partner operates in the power generation sector, specifically managing a hydroelectric plant. The primary objective of the project being developed at Atlântico is to harness data from multiple existing legacy and operational systems to drive strategic decision-making within the partner's operations.

The project proposes a continuous data ingestion workflow, centralizing information within a Data Lakehouse architecture. Following data processing and refinement, AI models are applied to provide insights that populate a dashboard tailored for the partner's stakeholders. Figure 1 illustrates the proposed reference architecture.

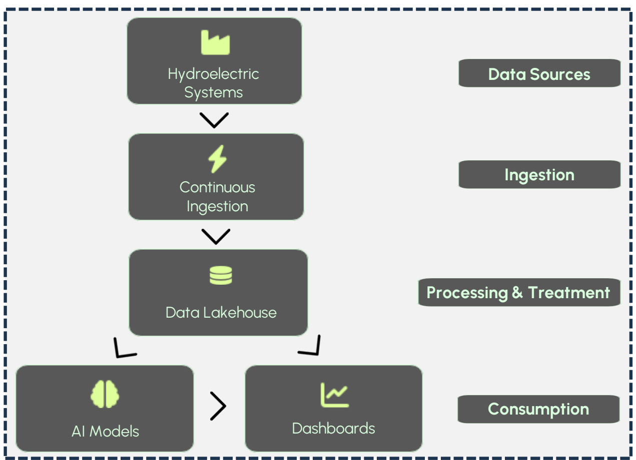

Figure 1: Base Project Architecture

As can be observed, a significant portion of the project's activities is intrinsically related to backend data processing, lacking direct interaction with a traditional end-user. This scenario poses a substantial challenge for requirements engineers with a limited background in DS. While continuous refinement meetings are essential to maintain business alignment, the textual specification of the requirements must be highly rigorous to prevent technical deviations during implementation. In this context, LLM-based tools emerge as potential allies, assisting teams in the accurate writing and technical specification of these requirements.

3 LLM-Assisted Requirements Specification Workflow

Regarding the RE process within the project, the workflow is structured into three main stages: (i) identification of the scope to be elicited based on the project roadmap and contract; (ii) external refinement with the partner to understand business needs; and (iii) internal technical refinement with the development team. Once these stages are completed, it is expected that all essential information for specifying the requirement has been successfully gathered.

Figure 2 illustrates the practical workflow we executed in this project for LLM-assisted requirements specification, which we structured into four sequential stages. During the **Preparation and Elicitation stage**, we actively gathered the necessary business and technical context by analyzing the project roadmap, reviewing the contractual scope, and holding refinement meetings with the partner and our internal team. This collaborative effort culminated in the design of an anonymized prompt. Next, in the **AI Generation stage**, we used these structured prompt segments to guide the LLM in producing an initial draft of the requirement artifacts. The core of our approach took place in the **Evaluation Loop stage**, where we prioritized human validation. We thoroughly reviewed the generated output to ensure contextual coherence, technical feasibility, and strict alignment with our prior discussions, iteratively refining the prompt whenever the results fell short of our expectations. Finally, during the **Waiting stage**, we restored the anonymized placeholders with the actual project data and converted the validated content into the official project artifacts ready for development.

During the execution of this workflow (between November and June 2026), we utilized corporate licenses of the Gemini model (version 3.1 Pro) provided by the Atlântico Institute. However, due to data privacy and governance policies, inserting sensitive or proprietary partner data into the tool is prohibited. To address this, the team established masking criteria: any information that identifies or characterizes the partner's infrastructure, specific systems, or encompasses proprietary logic that cannot be shared with third-party services must be anonymized. Therefore, the standard approach adopted by the team consists of crafting anonymized prompts focused exclusively on leveraging the LLM to provide the technical and textual structuring of the requirement. An example of this prompting approach is presented below:

> **Prompt:** I need to specify a requirement regarding the data ingestion from System [XXXXXX] into the controlled environment [XXXXXX], relating to alarms within the power plant.
>
> **Context:** Data must be ingested through a [XXXXXX] connection, arriving daily at 08:00 AM. Upon receiving this data, a pipeline must be created to feed the bronze, silver, and gold layers of our [XXXXXX] solution.
>
> **Format:** The requirement must be formatted as a User Story, following the "*As a... I want to... So that...*" structure. Additionally, it must include a BDD scenario in the "*Given... When... Then...*" format. Finally, based on the provided information, generate the acceptance criteria.
>
> **Note:** Maintain the [XXXXXX] placeholder pattern in the output so that sensitive names can be properly substituted later.

Different versions of the prompt were tested for two months before the final version was reached. Initially, providing instructions in a single, unstructured block resulted in generic outputs that frequently ignored the required format. In practice, we observed that strictly segmenting the prompt into 'Context', 'Format', and

[1] Brazilian Agency for Industrial Research and Innovation: https://embrapii.org.br/.

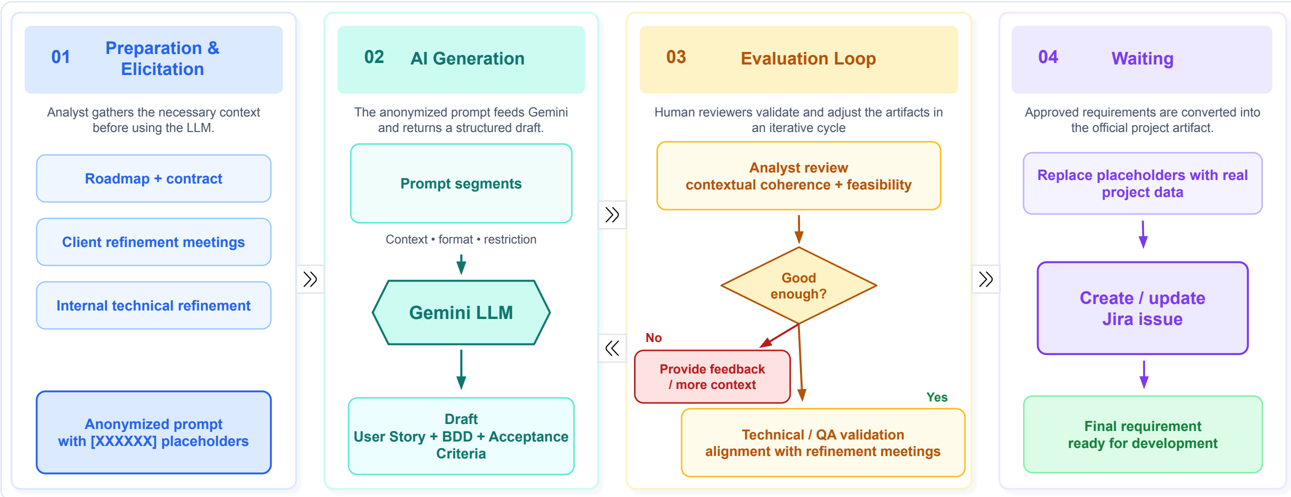

Figure 2: Overview of the LLM-Assisted Requirements Workflow Adopted in our Data Science Project

'Note' constraints yielded results much more aligned with our expectations. Furthermore, providing detailed context proved crucial to accurately feed the information into the generated User Stories, BDD scenarios, and acceptance criteria. Finally, the restrictive instruction acts as a safeguard, ensuring that the anonymized placeholders are strictly maintained in the output. Below is an example of the output generated by Gemini for the prompt above:

Title: Power Plant Alarms Ingestion via System [XXXXXX]

1. User Story
As a Data Engineer for the [XXXXXX] solution, **I want** to configure the automatic ingestion and layered processing (Bronze, Silver, and Gold) of alarm data from System [XXXXXX], **So that** the data is available and refined for power plant monitoring analysis.

2. BDD (Behavior-Driven Development)
Scenario: Successful daily alarm processing
Given that the [XXXXXX] connection to System [XXXXXX] is active;
And the alarm file was made available at 08:00 AM;
When the ingestion pipeline of the [XXXXXX] solution is triggered;
Then the raw data must be stored in the Bronze layer;
And processed into the Silver layer with appropriate cleansing and type casting;
And consolidated in the Gold layer for final business consumption.

3. Acceptance Criteria

- **Connectivity:** The pipeline must establish a connection to System [XXXXXX] via the [XXXXXX] method without authentication failures.
- **Scheduling:** The ingestion must be triggered automatically every day after 08:00 AM (expected data arrival time).
- **Traceability (Bronze):** Data must be persisted in the Bronze layer in its original format, maintaining load history.
- **Data Quality (Silver):** The pipeline must apply data cleansing rules in the Silver layer (e.g., deduplication, null field handling, and date standardization).
- **Business Rules (Gold):** The Gold layer must reflect the final view of the alarms, ready to feed dashboards or analytical models.
- **Monitoring:** There must be an execution log reporting the status (success/error) of each pipeline stage.
- **Error Handling:** If the data does not arrive at 08:00 AM, the system must generate a "data not found" alert for the operations team.

4 Internal Evaluation

As previously discussed, the use of LLMs serves as a supportive tool for requirement specification and does not replace the critical analysis of the requirements engineer. The process still demands constant validation from both the engineer and the technical team. The tool's output is treated as a preliminary draft, a starting point to be further developed, improved, and polished. During internal evaluation, the engineer reviews the requirement for contextual coherence and the feasibility of the AI's suggestions. This process is iterative; if the output does not meet expectations, additional context is provided, and the prompt is re-executed. Once filtered and refined, the requirement is formalized in the project management tool (Atlassian Jira), and the previously anonymized data is replaced with the actual project information. The final stage involves technical validation with the team, where developers and Quality Assurance (QA) professionals ensure the specification fully aligns with the decisions made during previous alignment meetings.

This process has proven valuable for the requirements engineer, the technical team, and, consequently, the project. The positive impacts can be highlighted across three main fronts: Time, Communication, and Quality. Regarding Time, over a period of 14 sprints, all 142 project requirements were specified using this LLM-assisted workflow. Empirical observations by the team indicate a sharp reduction in writing effort: while drafting a complex data requirement previously took more than a full workday, tasks of similar complexity are now completed in just a few hours. In terms of Communication, the clear and structured writing of the requirements aligns with the technical team's practices, thus enhancing clarity and communication fluidity. Finally, there is a notable increase in the Quality of the requirements. The combination of the AI-generated draft with human validation has resulted in precise task descriptions and acceptance criteria. This has made the development cycle more accurate, ensuring that the final delivery meets both technical standards and the partner's actual business needs.

However, this LLM-assisted workflow also presented notable limitations and challenges. A frequent issue observed during the AI Generation stage was the artificial inflation of requirement complexity. In several instances, Gemini generated complex or entirely unnecessary acceptance criteria. Furthermore, the model occasionally exhibited "hallucinations" by being overly proactive, fabricating BDD scenarios and introducing domain specific information that was not provided in the original context. Consequently, these deviations required either manual pruning during the human validation stage or a complete re-execution of the prompt.

To provide broader context, the project maintained a 100% sprint acceptance rate over the 14 sprints, an average completion of 97% of planned story points, and a maximum partner recommendation score (10/10). While these overarching metrics cannot be attributed exclusively to the LLM, the consistent use of Gemini to specify all 142 requirements indicates that this structured approach effectively supports the project's execution and overall success.

5 Conclusion

This industrial experience report presents the adoption and practical impact of using LLMs, specifically Gemini, in the RE specification process for a project focused on DS at the Atlântico Institute. Faced with the challenge of specifying DS domains characterized predominantly by non-functional requirements, such as backend data processing in the power generation sector, the approach demonstrates how business needs can be transformed into technical specifications within this domain. The entire workflow built and established by our team, from the initial elicitation and anonymization, through AI-assisted generation, to the indispensable iterative cycle of human validation and subsequent formalization, is consolidated and synthesized in Figure 2. This structured process has proven to optimize communication, time, and the quality of deliverables, all while ensuring strict compliance with data governance policies.

As future work, we intend to conduct internal research across other projects within the energy sector, as well as in different domains, maintaining a focus on DS initiatives, to map how the requirement specification process is currently performed by those teams. Based on this mapping, the goal is to establish a cross-project standardization, creating comprehensive guidelines and a set of best practices for these specifications.

Artifact Availability

Due to confidentiality agreements and data privacy policies regarding the industrial partner's critical infrastructure, the datasets and full raw prompts used in this study are not publicly available.

Acknowledgements

This work was supported in part by the Atlântico Institute, the State University of Ceará (UECE), and the National Council for Scientific and Technological Development (CNPq - PDI 120070/2025-1). We used LLMs for language refinement, but all analysis and interpretation remain our sole responsibility.

References

[1] Camila Almeida, Isaque Copque, Alvaro Oliveira, Murilo Arouca, Adriano Barbosa, Sávio Freire, Manoel Mendonça, and Julio Cesar Leite. 2025. From Elicitation Interviews to Software Requirements: Evaluating LLM Performance in Requirement Generation. In *Workshop on Requirements Engineering (WER)*.

[2] Camila Almeida, Sávio Freire, Manoel Mendonça, and Julio Cesar Leite. 2026. Exploring Multiple Sources to Enhance LLM Effectiveness in Non-Functional Requirements Elicitation. In *2026 IEEE 34rd International Requirements Engineering Conference (RE)*. IEEE.

[3] Hamza Hussein Altarturi, Keng-Yap Ng, Mohd Izuan Hafez Ninggal, Azree Shahrel Ahmad Nazri, and Abdul Azim Abd Ghani. 2017. A requirement engineering model for big data software. In *2017 IEEE conference on big data and analytics (ICBDA)*. IEEE, 111–117.

[4] Darlan Arruda. 2018. Requirements engineering in the context of big data applications. *ACM SIGSOFT Software Engineering Notes* 43, 1 (2018), 1–6.

[5] Abdullah M Baabdullah. 2024. The precursors of AI adoption in business: Towards an efficient decision-making and functional performance. *International Journal of Information Management* 75 (2024), 102745.

[6] Hoa Khanh Dam. 2019. Artificial intelligence for software engineering. *XRDS: Crossroads, The ACM Magazine for Students* 25, 3 (2019), 34–37.

[7] Luigi Di Caro, Amon Rapp, Vijayan Sugumaran, and Farid Meziane. 2026. From Natural Language to Knowledge: The Role of LLMs in Conceptual and Data Modeling.

[8] Xavier Franch, Andreas Jedlitschka, and Silverio Martínez-Fernández. 2023. A requirements engineering perspective to AI-based systems development: A vision paper. In *international working conference on requirements engineering: foundation for software quality*. Springer, 223–232.

[9] Marcos Kalinowski, Tatiana Escovedo, Hugo Villamizar, and Hélio Lopes. 2023. *Engenharia de Software para Ciência de Dados: Um guia de boas práticas com ênfase na construção de sistemas de Machine Learning em Python*. Casa do Código.

[10] Mary Beth Kery. 2018. Towards scaffolding complex exploratory data science programming practices. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 273–274.

[11] Archana Mantri and Rahul Mishra. 2023. Empowering small businesses with the force of big data analytics and AI: A technological integration for enhanced business management. *The Journal of High Technology Management Research* (2023).

[12] Iñigo Martinez, Elisabeth Viles, and Igor G Olaizola. 2021. Data science methodologies: Current challenges and future approaches. *Big Data Research* 24 (2021).

[13] Patrícia Mello, Pablo Ximenes, Rousiany Lemos, Ana Luíza Bessa, Mariela I Cortés, and Cleilton Lima Rocha. 2018. On the applicability of BDD in a Business Intelligence project: experience report. In *Proceedings of the XVII Brazilian Symposium on Software Quality*. 296–304.

[14] Devon Myers, Rami Mohawesh, Venkata Ishwarya Chellaboina, Anantha Lakshmi Sathvik, Praveen Venkatesh, Yi-Hui Ho, Hanna Henshaw, Muna Alhawawreh, David Berdik, and Yaser Jararweh. 2024. Foundation and large language models: fundamentals, challenges, opportunities, and social impacts. *Cluster Computing* 27, 1 (2024), 1–26.

[15] Cristian Rotar and Qingyu Zhang. 2025. A design science research approach to Large Language Model-Based Agents for Requirements Specification (LLMBA4RS) in low-code applications. *Requirements Engineering* 30, 4 (2025), 399–422.

[16] Malik Abdul Sami, Zheying Zhang, Muhammad Waseem, Kai-Kristian Kemell, Zeeshan Rasheed, Tomas Herda, Md Toufique Hasan, Jussi Rasku, and Pekka Abrahamsson. 2025. A multi-agent llm system for automated requirements analysis: a study on user story generation and prioritization. In *Euromicro Conference on Software Engineering and Advanced Applications*. Springer, 178–187.

[17] Christoph Schröer, Felix Kruse, and Jorge Marx Gómez. 2021. A systematic literature review on applying CRISP-DM process model. *Procedia computer science* 181 (2021), 526–534.

[18] Eurizene Sousa, Murilo Coelho, Roberta Braga, Matheus Paixao, Rodrigo Spínola, Manoel Mendonça, Julio Cesar Leite, and Sávio Freire. 2024. What do linkedin brazilian ads ask of requirements engineers?. In *Proceedings of the XXIII Brazilian Symposium on Software Quality*. 331–340.

[19] Lorijn Van Rooijen, Frederik Simon Bäumer, Marie Christin Platenius, Michaela Geierhos, Heiko Hamann, and Gregor Engels. 2017. From user demand to software service: using machine learning to automate the requirements specification process. In *2017 IEEE 25th international requirements engineering conference workshops (REW)*. IEEE, 379–385.

[20] Andreas Vogelsang and Markus Borg. 2019. Requirements engineering for machine learning: Perspectives from data scientists. In *2019 IEEE 27th international requirements engineering conference workshops (REW)*. IEEE, 245–251.