Adoption of LLMs in Requirements Engineering: What Practitioners Are Worried About?

Reine Santos
Institute of Computing
Federal University of Amazonas
Manaus, Amazonas, Brazil
rms@icomp.ufam.edu.br

Igor Steinmacher
School of Informatics, Computing,
and Cyber Systems
Northern Arizona University
Flagstaff, Arizona, USA
igor.Steinmacher@nau.edu

Tayana Conte
Institute of Computing
Federal University of Amazonas
Manaus, Amazonas, Brazil
tayana@icomp.ufam.edu.br

Ana Carolina Oran
Institute of Computing
Federal University of Amazonas
Manaus, Amazonas, Brazil
ana.oran@icomp.ufam.edu.br

Bruno Gadelha
Institute of Computing
Federal University of Amazonas
Manaus, Amazonas, Brazil
bruno@icomp.ufam.edu.br

ABSTRACT

Requirements engineers face persistent challenges related to stakeholder communication, information ambiguity, and the precise definition of system needs. In this context, user stories have gained traction as a structured, user-centered approach to representing requirements. At the same time, there is growing interest in using Large Language Models (LLMs) to automate such tasks. However, before proposing automated solutions, it is essential to understand how professionals currently elicit requirements, what techniques and tools they use, and how open they are to adopting LLMs for user story generation. To investigate this, an online survey was conducted with 21 software industry professionals experienced in requirements engineering. The questionnaire, divided into three sections, explored current practices in requirements elicitation and validation, as well as the use of LLMs in these activities. The results showed that 57.1% of participants currently use user stories in their work. Among them, 66.7% reported using LLMs, primarily ChatGPT, for tasks such as brainstorming and content review. Of these LLM users, 75% also apply them to user story generation, citing benefits such as increased speed, standardization, and improved story quality. However, limitations were also identified, including lack of context, potential over-reliance, and legal concerns. These findings indicate that there is growing interest in adopting LLMs in requirements engineering, including user story generation, as long as their use aligns with established practices and is supported by effective prompting, contextual input, and data privacy safeguards.

KEYWORDS

Requirements Engineering, User Stories, Large Language Models, Automated Requirements Generation, Survey.

1 Introduction

Requirements Engineering (RE) plays a crucial role in software development by ensuring that systems are aligned with stakeholders' expectations and needs [20]. Yet, several persistent challenges remain, particularly those involving communication with stakeholders, ambiguities in the conveyed information, and the accurate definition of system requirements [17]. Within agile methodologies, user stories have emerged as a widely used and adaptable approach for articulating requirements. They contribute not only to better communication among team and stakeholders but also support iterative development, prioritization, and ongoing collaboration [18]. Nevertheless, despite their popularity, teams often encounter difficulties in crafting user stories that are clear, cohesive, and consistent, issues that may compromise the quality of the resulting software product [10]. This situation highlights the importance of exploring how professionals perceive the integration of emerging technologies in the automation of user story generation.

Large Language Models (LLMs), in particular, have recently gained attention in the software engineering community for their capabilities in understanding, generating, and validating textual content [2]. In the context of RE, researchers have begun to investigate the applicability of LLMs to support the elicitation, representation, and maintenance of requirements in more effective ways [21]. Because they can produce clear and organized text, these models have the potential to help automate part of the user story creation process [11, 15, 18].

However, before advancing automated approaches, it is necessary to first examine how requirement elicitation is currently conducted in practice, what tools are already in use, and how professionals perceive and engage with LLM-based technologies in their workflows. Addressing this knowledge gap is the primary motivation of this study, which aims to offer a detailed understanding of the existing landscape, thereby paving the way for solutions that are more aligned with professional needs and expectations [24].

Given this context, it is necessary to investigate how industry professionals deal with requirements in their daily activities, for example, when eliciting and specifying requirements, and how they perceive the use of LLMs, including for the automated generation of user stories. This investigation aims to uncover the practices currently adopted, the main challenges faced, and the opportunities perceived by these professionals. To guide this exploration, the study is driven by the following research questions (RQs):

RQ1: How do industry professionals handle requirements in their daily activities?

RQ2: How do professionals use LLMs in the context of requirements engineering?

RQ3: What are professionals' perceptions of the automated generation of requirements, particularly user stories, using LLMs?

To explore these questions, we conducted an online survey with 21 software industry professionals with different years of experience in software development. The results show that professionals commonly use interviews, document analysis, prototyping, and stakeholder reviews for validation. While 57.1% use user stories, preferences for quality criteria differ: users favor practical structure (e.g., "Well-Formed," "Atomic"), while non-users value formal precision (e.g., "Unambiguous"). LLMs are already used by 66.7% of participants, mostly for creative support and technical tasks. Among those generating user stories with LLMs (75%), benefits include agility and standardization.

2 Background

Requirements Engineering (RE) process consists of identifying, modeling, communicating and documenting the needs of the system, as well as defining the usage context. RE plays a crucial role in traditional methodologies, in which requirements are defined and specified before the start of development. This leads to challenges such as communication gaps and excessive scope [1]. In traditional models like waterfall, RE follows predefined phases aimed at fully specifying the system before development, but often lacks adaptability. Agile methods, in contrast, emphasize collaboration and flexibility, with RE becoming iterative and evolving throughout the process. Requirements are frequently represented through user stories and refined just-in-time using techniques such as interviews, brainstorming, and case analysis [5].

Indeed, around 90% of agile practitioners use user stories to capture requirements [3]. User stories are concise descriptions of desired functionalities from the end-user's perspective. These narratives capture the essential elements of a requirement: the user role, the desired action, and the justification for the action [25]. The widely adopted format for these stories is: "As a⟨type of user⟩, I want⟨goal⟩so that⟨some reason⟩." Each story should be accompanied by acceptance criteria that define the conditions necessary for it to be considered satisfactory, covering both functional and quality aspects [25]. Research by Lucassen et al. (2016) shows that the use of templates and quality guidelines for user stories significantly contributes to increased productivity and improved product quality [10]. Effective writing of user stories is essential, as they guide the development team and clearly communicate user needs. In addition to general requirements engineering guidelines, frameworks such as INVEST [23] and the Quality User Story (QUS) [9] provide specific criteria for assessing and ensuring story quality.

The acronym INVEST (Independent, Negotiable, Valuable, Estimable, Small, and Testable) was proposed by Wake (2003) as a set of principles to support the creation of effective stories in agile settings. Applying these principles improves communication among team members and stakeholders and increases development efficiency. The QUS framework, proposed by Lucassen et al. (2016), includes 13 criteria for evaluating user story quality from syntactic, pragmatic, and semantic perspectives. These criteria include: Well-Formed, Atomic, Minimal, Conceptually Sound, Problem-Oriented, Unambiguous, Conflict-Free, Complete Sentence, Estimable, Unique, Uniform, Independent, and Complete [9]. These guidelines help ensure that stories are clear, concise, and suitable for development, contributing to project success [25].

In recent years, the advancement of Large Language Models (LLMs) has introduced new opportunities for software engineering, including support for the generation and refinement of user stories [11]. These models stand out for their ability to learn advanced linguistic representations from large volumes of textual data. Models like BERT (Bidirectional Encoder Representations from Transformers) are trained to predict words in textual sequences, enabling them to capture complex patterns and nuances in natural language [4, 14]. With the emergence of models such as GPT, developed by OpenAI, new opportunities have arisen for applying LLMs to various stages of the software life cycle. In the context of requirements engineering, these tools have shown potential to support elicitation, generation, refinement, and even validation of requirements [21]. As such, LLMs emerge as promising allies in the pursuit of greater efficiency, clarity, and quality in requirements engineering, even supporting tasks such as user story generation.

3 Related Work

Prior surveys of RE practices indicate that user stories are primarily created for developers, rather than as a working representation for analyst–customer cooperation [6]. Wagner et al. [22] report that in agile settings, interviews, prototyping, and workshops are the most common techniques for eliciting requirements, while free-form textual models, structured lists, and use cases dominate documentation, with formal models rarely used. This suggests that although user stories are widely adopted, their utility in elicitation and validation remains limited. LLM-based tools have the potential to enhance the value of user stories by supporting analysis, refinement, and communication with stakeholders.

Similarly, Lucassen et al. [9] surveyed 182 professionals on user stories, noting that 94% use them with Scrum and prefer the Connextra template. User stories' simplicity fosters shared understanding and collaboration. Explicitly stating the "why" enhances quality, and following INVEST guidelines links to higher productivity and better deliverables, especially for less experienced teams. Non-technical stakeholders value these guidelines more, and experience with user stories correlates with positive perceptions of their effectiveness.

Given these limitations in elicitation and validation, recent research has explored how natural language technologies support software engineering activities, including requirements engineering. The study by Russo [19] is a survey investigating the adoption of Generative Artificial Intelligence tools in the context of software engineering. The author initially conducted a questionnaire with 100 software engineers, based on theoretical acceptance models. From the data analysis, the Human-AI Collaboration and Adaptation Framework was proposed, describing the main factors influencing the adoption of these tools. Contrary to traditional expectations, the results indicate that compatibility of AI tools with existing workflows is the primary driver of adoption, while perceived usefulness, social factors, and personal innovativeness have less impact than expected.

A systematic mapping study conducted by Zhao et al. [26] explores the field of Natural Language Processing for Requirements

Engineering (NLP4RE), identifying 404 studies published over the past 36 years. Most of these works focus on the requirements analysis phase, particularly on detecting quality defects in specification documents. While a majority of the proposals (67.08%) have been evaluated in laboratory settings, only 7% have undergone validation in industrial contexts. The study also identified 130 NLP4RE tools, but only 17 of them (13.08%) are publicly available, highlighting a gap between academic research and practical adoption.

Unlike previous studies, this work presents a survey conducted entirely with industry professionals, focusing on the intersection between requirements engineering, including user stories, and the use of LLMs. While Russo [19] explores AI adoption broadly in software engineering, and Zhao et al. [26] maps NLP techniques applied to requirements analysis, neither addresses the practical use of LLMs for requirements generation. Moreover, Lucassen et al. [9] investigates user story practices but does not consider automation. This study fills that gap by examining how professionals carry out requirements engineering activities and how they are already applying LLMs to user story generation.

4 Study Methodology

Surveys are among the most widely used research methods, especially in studies aiming to investigate opinions, attitudes, or behavior patterns within a group. More than just a questionnaire or checklist, a survey constitutes a comprehensive system for gathering information, with the goal of describing or explaining knowledge, perceptions and actions. This process involves a series of interdependent activities that structure a survey [7, 12].

In this study, an online survey was chosen due to its several advantages, as highlighted by Punter et al. [13] and Linåker et al. [8]. This survey format not only simplifies respondent access to the questionnaire but also reduces the effort required from participants to complete it. Moreover, features such as dynamic questionnaire navigation, adapting to previous answers, and the ease of analyzing electronically recorded data are additional strengths that make online surveys an efficient research tool. The following sections describe the survey design, execution, and data analysis.

4.1 Study Design

The design adopted in this study followed the guidelines for conducting research in the field of Software Engineering proposed by Kitchenham and Pfleeger [7], which initially recommend clearly defining the research objectives to ensure they are specific and measurable. Next, the research design is carried out, involving decisions about the methodological approach and the target population. With these defined, the research instrument, typically a questionnaire, is developed and must be carefully constructed. The following step consists of evaluating this instrument to ensure its clarity, validity, and reliability. After data collection, it is essential to guarantee that the obtained data are valid and consistent. Finally, data analysis is conducted, from which results are interpreted and used to answer the three research questions. Accordingly, the main issues related to the research design are the definition of the target population and characterization of respondents, as well as the questionnaire design, which will be explained below.

The first step of the technique involves making decisions about the methodological approach, such as defining the target population. The target population is the group or individuals to whom the research applies, selected to ensure that participants have sufficient experience to answer the proposed questions. Accordingly, the population consisted of industry professionals with prior experience in requirements elicitation activities. To reach this desired profile, subjects aligned with the study objectives were recruited. The study aimed to investigate how these professionals handle requirements in their daily work, particularly regarding elicitation and specification activities, and how they perceive the use of in these tasks.

Participants were selected based on eligibility criteria, such as professional experience in the software industry and prior involvement in requirements elicitation activities. Thus, a non-probabilistic convenience sampling was adopted, as characterized by [7], which is common in exploratory studies within Software Engineering when access to specialists with specific profiles is necessary. Although this type of sampling limits the generalizability of the results, it is suitable for obtaining qualified insights from experienced professionals regarding real-world requirements engineering practices.

With the research instrument for data collection, an online questionnaire was chosen, as it is an efficient tool to reach geographically distributed participants and offers greater convenience in responding. According to Linåker et al. [8], the use of self-administered online questionnaires preserves respondent confidentiality, prevents researcher influence over participants, and reduces operational costs. These characteristics were decisive in choosing this format, considering the target audience and the research objectives. Given these considerations and the nature of the study, the research was approved by the CEP of the Federal University of Amazonas (CAAE [829552724.0.0000.5020]).

To effectively collect the necessary information to answer the three research questions, a questionnaire structured into three sections was developed. Table 1 describes these sections in detail, while Figure 1 presents a flowchart representing the questionnaire's navigation logic. In this flowchart, diamonds indicate conditional questions that guide participants to different branches based on their responses (e.g., "Yes" or "No"). This scheme facilitates visualization of the progression and filters applied throughout the questionnaire completion process.

Below is a description of the three sections that make up the questionnaire, totaling 28 questions (Q1 to Q28). Analysis of the data collected through these questions provides important contextual support for analyzing responses related to RQ1, RQ2, and RQ3.

4.2 Study Execution

The online questionnaire was created using Google Forms, chosen for its ease of use, accessibility, and broad reach. After its design, a pilot study was conducted with two professionals experienced in requirements. This phase aimed to assess the clarity, relevance, and comprehensibility of the questions and identify ambiguities. Based on participants' feedback, questions were reordered for better flow, examples were added, and wording was refined to improve clarity. The pilot also confirmed that participants correctly understood

Table 1: Classification of the Questionnaire Questions by Section.

Section	Type	ID	Full Question
Demographic Information	Open	Q1	Name:
	Open	Q2	Email:
	Closed	Q3	What is your current position?
	Closed	Q4	How long have you been working in the software development field?
	Open	Q5	What is the name of the company you work for?
	Open	Q6	Where is it located (city and state)?
Requirements Engineering Practices	Closed	Q7	How often do you participate in requirements elaboration in your projects?
	Open	Q8	What methods or techniques does your team use to elicit requirements?
	Open	Q9	What methods or techniques does your team use to validate the defined requirements?
	Closed	Q10	Does your team use user stories to structure and document the generated requirements?
	Closed	Q11	Besides user stories, what other forms or techniques does your team use to structure and document requirements?
	Open	Q12	What procedures or methods do you use to assess the quality of the created user stories?
	Open	Q13	Does your team use acceptance criteria for user stories? If so, what are the criteria and what information is included for a story to be considered complete?
	Closed	Q14	In your opinion, what is the biggest challenge when writing user stories in your team?
	Closed	Q15	Which of the following qualities do you consider important for a good user story?
	Closed	Q16	What forms or techniques does your team use to structure and document requirements?
Use of LLMs	Closed	Q17	Do you use Large Language Models (LLMs) in your professional activities?
	Open	Q18	Which LLM(s) do you use?
	Closed	Q19	Do you use LLMs for generating user stories?
	Closed	Q20	For which activities do you usually use LLMs?
	Open	Q21	How do you think the use of LLMs impacts your work? Explain the positive and/or negative impacts:
	Closed	Q22	Would you use LLMs to generate user stories?
	Open	Q23	What would be necessary for you to start using LLMs in generating user stories?
	Open	Q25	How do you think the use of LLMs impacts the generation of user stories? Explain the positive and/or negative impacts:
	Open	Q26	Why don't you use LLMs in your professional activities?
	Open	Q27	Do you use LLMs for any other types of activities?
Suggestions and Final Opinions	Open	Q24	Do you have any suggestions for improvements that would make the use of LLMs in user story generation more viable?
	Open	Q28	Do you have any suggestions or ideas to improve the use of LLMs in creating user stories?

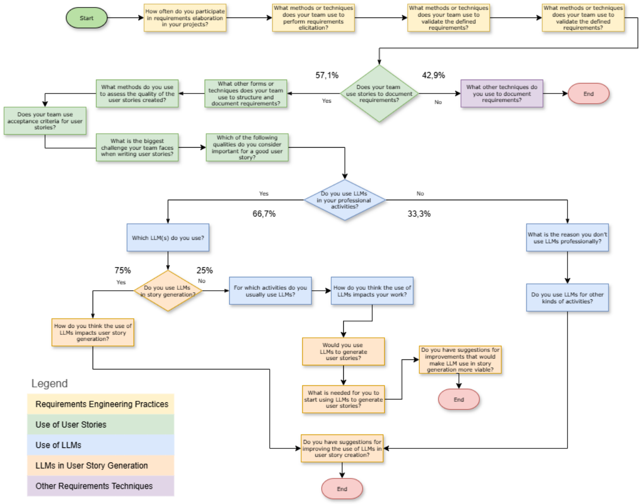
Start
How often do you participate in requirements elaboration in your projects?
What methods or techniques does your team use to perform requirements elicitation?
What methods or techniques does your team use to validate the defined requirements?
What methods or techniques does your team use to validate the defined requirements?
57,1%
42,9%
Does your team use stories to document requirements?
Yes
No
What other techniques do you use to document requirements?
End
What other forms or techniques does your team use to structure and document requirements?
What methods do you use to assess the quality of the user stories created?
Does your team use acceptance criteria for user stories?
What is the biggest challenge your team faces when writing user stories?
Which of the following qualities do you consider important for a good user story?
Yes
Do you use LLMs in your professional activities?
No
66,7%
33,3%
Which LLM(s) do you use?
What is the reason you don't use LLMs professionally?
75%
25%
Yes
No
Do you use LLMs in story generation?
For which activities do you usually use LLMs?
How do you think the use of LLMs impacts your work?
Do you use LLMs for other kinds of activities?
How do you think the use of LLMs impacts user story generation?
Would you use LLMs to generate user stories?
Do you have suggestions for improvements that would make LLM use in story generation more viable?
What is needed for you to start using LLMs to generate user stories?
End
Do you have suggestions for improving the use of LLMs in user story creation?
End
Legend
Requirements Engineering Practices
Use of User Stories
Use of LLMs
LLMs in User Story Generation
Other Requirements Techniques

Figure 1: Flowchart of the Survey Applied to Requirements Engineering Professionals.

each question, ensuring the instrument was well aligned with the research objectives.

The final version of the questionnaire was made available for a period of 14 days. The dissemination occurred through open posts on the LinkedIn platform and direct invitations via email and instant messaging applications, using convenience sampling. In all channels, it was requested that only professionals with experience in requirements elicitation participate in the survey, aiming to broaden the research reach and ensure the respondents' profiles aligned with the study's objectives. The complete questionnaire, including all applied questions and the conditional navigation logic, is provided in the Supplementary Material[1].

5 Study Results

Below are the results of the questionnaire, presented section by section. The survey had the participation of 21 respondents, and the data are detailed for each of the three areas.

5.1 Section 1 – Demographic Information

The results indicate that the participants predominantly hold the positions of Product Owner and Developer, each representing 28.6% of the sample. Additionally, 19% of respondents work as requirements analysts. These data demonstrate a predominance of professionals directly involved in the definition and implementation of requirements, which reinforces the sample's suitability for the research

[1] Supplementary Material: https://github.com/Reine66/SBQS25-Material

objective. Other participants hold roles such as Product Manager, tester, and technical specialist, each representing 4.8% of the sample. The presence of these professionals contributes to broadening the diversity of the sample by incorporating complementary perspectives on requirements engineering practices (Figure 2).

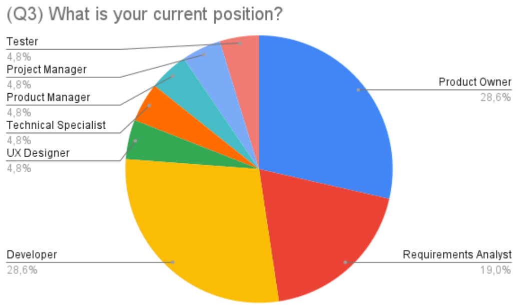

Figure 2: Distribution of study participants by role.

Regarding experience, the data reveal that 65% of participants have more than five years of experience in software development, with 23.8% having over 20 years, 23.8% over 10 years, and 14.3% over 5 years. The remaining respondents have between 1 and 5 years of experience, with 19% having 1 to 3 years and 14.3% having 3 to 5 years of experience (Figure 3). The vast majority of participants (85.7%) work for companies based in Manaus, Brazil. The others work remotely for companies in Florianópolis and Belo Horizonte.

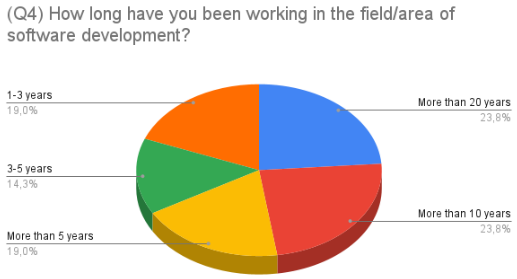

Figure 3: Experience level distribution of study participants.

5.2 Section 2 – Requirements Engineering Practices

Focusing on understanding how frequently participants engage in requirements engineering activities, it was observed that the vast majority of participants (95.2%) frequently participate in requirements elicitation in their projects. Only one participant (4.8%) reported occasional involvement. This result reinforces that the sample is composed mostly of professionals heavily involved in requirements engineering activities.

Through open and descriptive questions, it was possible to identify which methods or techniques their teams use for requirements elicitation and observe recurring patterns. The data reveal that interviews with stakeholders are the most commonly used elicitation technique, with 17 direct mentions, often described as an initial or continuous step, and frequently combined with other practices. Document analysis ranks second, with 12 mentions, employed to understand the context and complement the data collected. Other relevant techniques include brainstorming (7 mentions), common in workshops and ideation sessions, and Lean Inception (5 mentions), highlighted as a structured approach for strategic alignment. Direct observation (4 mentions) was also reported, mainly in scenarios requiring an understanding of the process in the field.

Regarding the requirements validation phase, the most common techniques are the use of prototypes, mockups, or wireframes, widely described (at least 12 responses), with emphasis on high-fidelity prototypes validated with users. Stakeholder reviews were also mentioned (about 10 responses), often associated with formal meetings or homologations. Other practices include document-based verifications (4 to 5 mentions), such as analysis of formalized requirements, comparison with regulations, and traceability. These findings indicate a strong adoption of iterative and collaborative approaches both in requirements elicitation and validation.

When asked about the use of user stories to structure and document the generated requirements, the data revealed a balanced scenario, 57.1% of participants affirming they use user stories, against 42.9% who reported not using them. Despite the narrow difference, the result demonstrates that user stories are more prevalent than previously thought, establishing themselves as an increasingly common practice in the structuring and documentation of requirements. The use of this approach indicates a trend toward maturation and broader adoption among development teams.

The data indicate that among the other forms or techniques teams use to structure and document requirements, diagrams are widely used, mentioned by 83.3% of participants, demonstrating a preference for graphical resources to describe functionalities, flows, and system architecture. Use cases and mockups/prototypes also stand out as recurring practices, both with 66.7% adoption, highlighting a mixed approach between structured textual descriptions and interface elements. These results show that, despite the growth of user stories, teams often adopt a complementary strategy, combining traditional techniques and visual resources to represent requirements in a clearer and more understandable way.

Among the 42.9% of participants who do not use user stories, more conventional methods were prevalent, since mockups and prototypes, mentioned by 77.8%, were the most commonly used technique, suggesting a strong preference for interface artifacts to explore and validate functionalities. Furthermore, diagrams and use cases were equally mentioned by 55.6% of respondents, reinforcing the role of these tools in communicating and structuring requirements, even outside the context of user stories.

Professionals who use user stories reported adopting various techniques to ensure that these stories are clear, feasible, and aligned with user and business needs. Among the most frequently mentioned techniques, peer reviews stood out, cited by nearly all respondents. This collaborative practice enables teams to identify ambiguities, inconsistencies, and opportunities for improvement before implementation. Another recurring technique is stakeholder validation, regarded as a central element in the processes adopted.

This validation occurs through meetings, prototype presentations, and direct interactions with end users, ensuring that user stories accurately reflect the actual project requirements.

Regarding the criteria prioritized during the evaluation of user stories, participants emphasized aspects such as clarity, alignment with user requirements, and technical feasibility. As illustrated by participant P10: "During the evaluation, we prioritize clarity, alignment with user requirements, and feasibility of implementation within agile development cycles." Participant P18 adds: "Peer reviews, stakeholder validation, acceptance criteria, prioritization, and continuous refinement."

In addition to these practices, complementary activities were observed throughout the development process, such as grooming and upstream meetings, used for technical detailing and defining acceptance criteria. Some teams also incorporate the refinement and discussion of user stories into daily stand-ups and retrospectives, supporting alignment with sprint planning and continuous stakeholder feedback. These findings highlight a consistent effort by teams to adopt sound requirement evaluation practices, combining collaborative approaches, user validation, and iterative cycles of continuous improvement.

Practices related to the acceptance of user stories were also identified. Professionals reported using a variety of criteria, and in many teams, this process was guided by checklists or a Definition of Done (DoD). Participant P1 stated that they consider as acceptance criteria all the elements that ensure the story is complete: "... everything I consider necessary for the user story to be regarded as complete. Business rules, behaviors, quality requirements."

According to P7, the criteria should be defined according to the project and must include as much relevant information as possible: "... the criteria depend heavily on the project, but we include as much measurable technical, business, and UX information as possible." P10 detailed two fundamental criteria for story validation: "Functionality Achieved – The story must correctly implement the described functionality, fulfilling the user's goal. Acceptance Tests – These must be defined and executable, ensuring that the functionality can be validated through automated or manual tests."

P13 emphasized that criteria are defined based on the specificity of each story: "Criteria are defined according to the specificity of each story." The involvement of the PO and the team was also highlighted, as explained by P15: "The criteria are defined by the PO. The team is consulted, and each member gives input on their story." Complementing this, P16 reported: "... We created a Definition of Done (DoD) section in the cards with the acceptance criteria."

Among the recurring elements in the criteria, P17 listed: "The user story must be described clearly and be understandable by all involved. Business rules and acceptance criteria must be defined, outlining the necessary conditions for the story to be considered complete. Any technical or functional dependencies must be documented to avoid surprises during implementation. The story must also be small enough to be completed in a single iteration (sprint) and deliver clear value to the user or business."

Planning and verifying the criteria are also part of the process. P19 explained: "The criteria are defined during each sprint Planning, where the team checks whether there is enough minimum information to start the task." P20 reinforced the idea of verification through a checklist: "... a story is accepted if it meets a checklist, which, for example, includes the acceptance criteria describing the conditions for considering the activity complete." Lastly, P21 pointed out that "acceptance criteria are applied through test scenarios that verify whether the result meets expectations."

The results indicate that the most frequently mentioned criteria for considering a story ready include clarity in its description, the definition of business rules, and well-defined acceptance tests. These elements ensure that the story is understood by everyone and can be properly validated. Other mentioned criteria include the story's suitability for sprint size, quality requirements (such as scalability and usability), and sound engineering practices, reinforcing teams' commitment to delivering well-defined stories aligned with both technical and business expectations.

In a multiple-choice question, participants were allowed to select more than one alternative regarding challenges faced when writing user stories within their teams. The main challenges identified were: lack of clear information about user needs (75%), communication issues between involved teams (58.3%), and lack of time to elaborate detailed stories (50%). These results suggest that main obstacles faced by teams are linked to limited understanding of user needs and organizational, process-related constraints. This highlights the importance of making the user story writing process clearer, more precise, and efficient, in order to support better alignment among stakeholders and enable more agile value delivery.

Subsequently, participants were asked to select all the criteria they considered relevant for a good user story. For this analysis, only the options selected by at least 50% of participants were considered. Figure 4 presents the most frequently mentioned criteria:

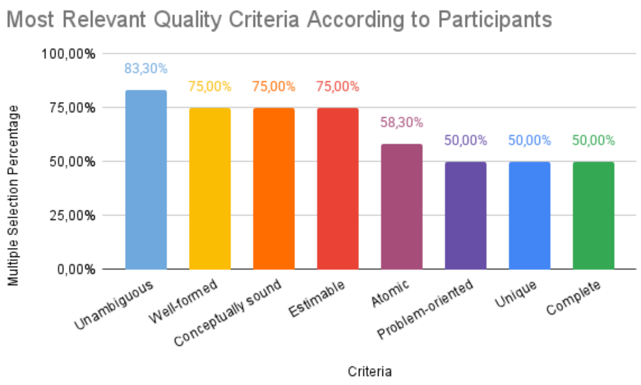

Figure 4: User story quality criteria selected by participants.

To aid selection, participants received a brief summary of each questionnaire criterion:

- **Unambiguous (83.3%):** avoids vague terms or abstractions that may lead to multiple interpretations.
- **Well-formed (75%):** explicitly includes a role (persona) and a specific action.
- **Conceptually sound (75%):** describes a functionality with a clear and well-explained purpose.
- **Estimable (75%):** represents a requirement that is not coarse-grained, making it feasible to plan and prioritize.

- **Atomic (58.3%):** addresses a single specific functionality, without mixing multiple goals.
- **Problem-oriented (50%):** specifies only the user's problem, without anticipating the solution.
- **Exclusive (50%):** ensures that each user story is unique, without duplications or redundancies.
- **Complete (50%):** contributes to the construction of a complete system, such that the implementation of the set of user stories results in a functional application with no missing steps.

5.3 Section 3 – Use of LLMs

Regarding the use of LLMs, the survey revealed that 66.7% of participants use Large Language Models (LLMs), such as ChatGPT, Gemini, LLama, and Copilot, in their professional activities, while 33.3% do not use these tools. ChatGPT was the most mentioned model, with 9 mentions, followed by Copilot, with 3 mentions. Other models mentioned include DeepSeek and Gamma.

Of the four participants who do not use LLMs in their professional activities, three justified their decision based on concerns related to security and privacy, particularly regarding the risk of violating the LGPD (General Data Protection Law). Participant P21 stated: "I don't know if I can use these tools without violating the LGPD." P15 pointed to organizational restrictions and the need to protect sensitive information: "There are company rules that limit the use of such tools." P13 was more direct, saying: "To avoid sharing confidential information." Overall, security and control over confidential data stood out as the main factors for not adopting LLMs. When asked about the use of LLMs in other activities, the same three participants gave different responses. Participant P15 stated that they do not use these tools for any other purpose. P21 reported limited use: "Only for personal things." Finally, P13 mentioned occasional use for language support: "Sometimes for text revision or other needs."

Among the participants who use LLMs in their professional activities, 75% reported employing them for generating user stories. The feedback indicates a broadly positive perception of the impacts these tools have on this process. Participants highlighted several positive impacts of using LLMs in user story generation. Agility and time-saving were among the main benefits mentioned. Participant P1 stated: "It reduces the work, I do it faster and with more details...". P10 emphasized: "Writing Agility – LLMs can quickly generate well-structured user stories, saving the team's time." Similarly, P18 mentioned: "Process acceleration...", while P19 pointed out: "Time gain is a very positive impact." Finally, P20 reinforced this view: "It speeds up and facilitates the work; currently, the use of LLMs is of great help given the increasingly tight deadlines."

Beyond agility, LLMs were also valued as support for creation and brainstorming. P18 reported: "I use it a lot as a support tool! Especially in user story generation, which speeds up the process a lot and helps inspire new ideas." P1 added: "Including ideas from the AI itself to complete, if I provide the proper context." According to P18 again: "Inspiration, brainstorming..."

Standardization was also mentioned as a relevant impact. P10 stated: "Consistency – They ensure a uniform standard, helping to maintain clarity and cohesion among stories." P18 reinforced: "Standardization," and P16 said: "It already sets up a standard model for stories," and "Helps with structuring." Support for story quality was another aspect highlighted. P10 mentioned: "Criteria Validation – They can automatically check if the stories follow best practices."

Among the negative impacts, the most recurrent limitation pointed out was the difficulty LLMs have in dealing with specific project context. P10 observed: "Lack of Specific Context – LLMs may not fully understand project nuances, generating generic or misaligned stories," and "Possible Inconsistencies – If prompts are not well crafted, the generated stories may not align with real product requirements." P18 also stressed: "But it is still necessary to refine to ensure they align with project context and business needs," adding: "Superficiality, lack of Alignment with the Business."

Some participants pointed out risks associated with excessive dependence. P16 warned: "It may make Product Owners and Requirements Analysts lazy or dependent on these tools, as with convenience they might not dedicate enough time to requirements analysis." P19 also highlighted: "Some professionals just copy ChatGPT suggestions without considering their context." Finally, P19 raised a concern related to organizational issues, noting: "In other contexts, it might be the license for ChatGPT use."

Overall, participants recognize the potential of LLMs to accelerate and support the writing process, especially in terms of agility, story structuring, and idea generation. However, the feedback also cautions the importance of critical review and alignment with project context to avoid overdependence or producing inappropriate and generic outputs.

Although two participants (P5 and P7) stated that they do not use LLMs for generating user stories, they reported employing them in other activities. Among these, the most common was generating initial ideas for brainstorming, mentioned by two respondents. Other activities cited included code generation, writing technical documentation, and requirements analysis or refinement, each mentioned by one participant. This suggests that LLMs are being used in a diverse manner throughout the development cycle.

Regarding positive impacts, these were mainly related to agility and time optimization. Participant P5 highlighted that "development becomes faster, optimizing delivery time," indicating productivity gains from using LLMs. Another positive point mentioned by P7 was the perception that results produced with AI assistance tend to be more complete.

However, some negative effects were also pointed out. P5 warned of the risk of "making the developer complacent, just pasting the generated code without really understanding the proposed solution." Another participant mentioned that although the results may be more complete, the process can end up taking more time, which contradicts the idea of efficiency in certain contexts. These observations emphasize the importance of critical and balanced use of LLMs in supporting software development.

Although both stated that they do not currently use LLMs to generate user stories, their responses indicate conditions that could motivate such adoption, revealing barriers related to limited time or structure, rather than rejection of the technology itself.

Participant P7 highlighted that it would be necessary to "have more time to dedicate to this and for the team to participate more in writing the story," suggesting that with a ready and validated technique, this barrier could be overcome. Participant P5 pointed out as the main need "to create the necessary prompt to extract a good user story," demonstrating a practical interest in using the tool, conditioned on the availability of an effective prompt model. These statements indicate that LLM adoption can be facilitated by ready and validated methodologies that reduce initial effort and foster confidence in the automated process.

5.4 Section 4 – Suggestions and Final Opinions

According to the participants, one of the most frequently mentioned ways to improve the use of LLMs in user story creation is related to enhancing the quality of the generated stories. Participant P1 believes that LLMs can be useful for reviewing and refining stories, suggesting that it is important to "use LLMs to review stories and suggest improvements based on acceptance criteria." He also highlights the potential to automate validations, recommending to "implement automatic checks to ensure that the stories are clear and testable." Participant P19 views LLMs as allies in optimizing time during this process: "using LLMs as an ally in creating user stories, so it is possible to save time."

One of the concerns raised by the participants relates to providing adequate context so that LLMs can generate user stories more aligned with the real needs of the project. Participant P1 emphasized that "it (the tool) needs to have context and possibly be linked with the story management tool," indicating the perception that integration with systems like Jira or Trello can facilitate the automatic provision of relevant information for story generation.

Participant P20 proposes a concrete approach in this regard: "provide, for training the LLMs, all possible artifacts for context." According to him, this practice has been actively adopted: "I have been using this strategy, not only to create the stories but also other artifacts that make up the project documentation." He further describes a more advanced application aimed at ongoing team support: "the creation of a prompt, trained with all project artifacts, so that developers can ask questions about requirements." This strategy suggests using LLMs as a contextualized point of consultation, contributing not only to story writing but also to the continuous understanding of requirements throughout development.

Participant P21 also reinforces the importance of context by suggesting the need to "train the model from the beginning of the project," highlighting that continuous information provision from early stages is essential for LLMs to adequately support task execution.

However, this ideal scenario faces significant limitations. As participant P18 points out, "AI receives less context" due to restrictions imposed by the LGPD (General Data Protection Law) on sharing sensitive and strategic data. This limitation "directly impacts the quality and alignment of the generated user stories" and, according to him, "can lead to generic results or stories misaligned with the real business needs." Furthermore, it requires additional effort to "anonymize and mask information without compromising the context" and reveals how critical this balance is, as "it is necessary to balance data protection with the clarity needed for AI to effectively contribute to accelerating the process."

Participant P13 also addresses this concern, noting that "using LLMs can be good, but at the same time it can expose technological advances or restricted information to the tool." He further stresses the importance of "understanding how to protect the type of information to bring more security to the user of a tool with this purpose, which helps adoption."

Thus, although provision of context is widely recognized as essential for LLM effectiveness, data protection requirements pose a significant challenge that must be addressed to ensure safe and responsible adoption of these tools in corporate environments.

6 Discussion of Study Results

This section presents the analysis and interpretation of the data collected in light of the research questions formulated for this study. The objective is to understand how industry professionals handle requirements engineering activities and how they perceive and use LLMs in their professional contexts. The discussion is organized based on the three guiding questions of the study, exploring current practices, the use of emerging technologies, and the perceived impacts of automation in the requirements specification process.

6.1 RQ1: How do industry professionals handle requirements in their daily activities?

The collected data reveal how industry professionals deal with requirements in their routines, highlighting practices, technical decisions, and adaptations according to the organizational context and the profile of those involved. Most participants in the survey have extensive experience in the sector: 65% have more than five years of experience in software development. This experience contributes to more mature practices and directly reflects how requirements are handled day-to-day.

Active involvement of professionals in the requirements engineering process is significant: all respondents reported frequent or occasional participation in requirements elicitation. Most reported using stakeholder interviews as the main elicitation technique, often conducted informally and continuously throughout development. Also, participants mentioned other approaches such as document analysis, brainstorming, observation, and Lean Inception.

Similarly, the open-ended responses regarding the methods and techniques used for requirements validation revealed that prototyping is the most recurrent and central practice in the validation process. More than 10 participants directly mentioned the use of prototypes, whether low or high fidelity, mockups, or even Excel representations, as a way to verify if the requirements meet the needs of the project and stakeholders.

Secondarily, reviews with stakeholders or users stand out as an essential step. These reviews take place both in formal meetings and informal interactions, where project participants evaluate and adjust the requirements based on their expectations and feedback.

Furthermore, the testing, acceptance, and validation phase in controlled environments was also mentioned, indicating that feedback obtained from usability tests, evaluations in builds or staging environments, as well as stakeholder acceptance processes, significantly contributes to verifying the quality and suitability of requirements before final implementation.

With respect to requirements structuring, user stories predominate, used by approximately 57.1% of teams, especially among professionals with more than five years of experience. Besides user stories, use cases, specification documents, and prototypes are also employed, indicating a combination of agile and traditional approaches depending on the team or project. More experienced professionals tend to adopt user stories more frequently, while less experienced ones still partially rely on traditional models.

Professionals who use user stories value criteria that combine good textual structure and practical applicability, such as "Well-Formed," "Atomic," "Estimable," and "Conceptually Sound." These criteria indicate that participants employing user stories focus on writing clear, well-defined, structured stories that fit agile planning and are feasible for teams to plan and execute.

Despite growing adoption of user stories, participants also report recurring challenges related to writing these stories, such as defining acceptance criteria and difficulty in assessing their quality. These challenges underscore that requirements engineering practice, even in agile environments, demands continuous attention to clarity, conceptual alignment, and planning capability.

In summary, industry professionals handle requirements by combining classical elicitation and validation techniques with agile documentation and management methods. Their practices are shaped by experience, organizational sector, and access to technology, resulting in a dynamic ecosystem where structured approaches coexist with experiments in agility and innovation.

6.2 RQ2: How do professionals use LLMs in the context of requirements engineering?

The data analysis reveals that approximately 66.7% of participants already use LLMs in their professional routines. The most mentioned tools include ChatGPT and GitHub Copilot, with occasional references to other models such as DeepSeek and Gamma. Still, one-third of respondents reported not using these tools, mainly due to concerns about privacy, information security, and compliance with the LGPD, factors especially relevant in organizational environments with stricter institutional policies.

Cross-referencing these data with professional experience shows that the use of LLMs is not limited to novice professionals. On the contrary, those with more than 10 years of experience make up a significant portion of users of these tools, evidencing openness to innovation even among professionals with established careers in the industry. On the other hand, professionals with less than 5 years of experience also appear among users, suggesting that exposure to LLMs can occur from early career stages, particularly in more innovative contexts.

The way professionals handle requirements also varies according to the sector in which they work. Most respondents (85,7%) work in private companies, environments that tend to be more open to agile approaches and the use of emerging technologies such as LLMs. A relevant example was the report from a participant in a public institution who described the combined use of user stories and LLMs for automated requirements generation, a practice not commonly expected, but one that points to a gradual adoption of intelligent solutions even in traditionally more conservative contexts.

Among professionals who use LLMs, reports indicate a variety of applications, such as support for technical writing, brainstorming ideas, code generation, and assistance in general development tasks. Although only one participant explicitly mentioned using LLMs in requirements documentation, this type of application still appears sporadically and unsystematically, suggesting that the direct use of LLMs in the domain of requirements engineering remains in its early stages. The most frequently cited use, brainstorming, reinforces the perception that professionals have primarily explored LLMs as tools for creative support and technical assistance, rather than as mechanisms aimed at formalizing requirements.

6.3 RQ3: What are professionals' perceptions of the automated generation of requirements, particularly user stories, using LLMs?

The results indicate a broadly positive perception of using LLMs in the generation of user stories. It is observed that 75% of the participants who use LLMs in their professional routines apply these tools for this purpose, which demonstrates a significant adoption of this technology in the context of requirements engineering.

Professionals highlight clear benefits of using LLMs for this purpose, with emphasis on agility and time savings during story writing. Testimonials such as participant P1's statement, "Reduces the work, I do it faster and with more details," and P10's remark, "Writing Agility – LLMs can quickly generate well-structured user stories, saving the team's time," demonstrate this consensus. Additionally, LLMs are valued as tools that support creation and brainstorming, contributing to inspiration and the structuring of stories, as reported by participants P18 and P1.

Another relevant positive impact is the standardization provided by LLMs, which help ensure consistency and clarity in generated stories, as highlighted by participants P10, P16, and P18. Support for story quality through criteria validation is also an added benefit.

On the other hand, professionals recognize important limitations in the use of these technologies, especially the difficulty LLMs have in capturing the specific context of projects, which can lead to the generation of misaligned stories. This concern is evidenced in the reports of participants P10 and P18, who stress the need to refine and adjust stories to ensure alignment with the business. Furthermore, risks related to excessive dependence on the tools were pointed out, including the possibility of reducing analytical effort by professionals, as warned by participants P16 and P19.

In summary, the general perception is that LLMs offer significant potential to accelerate and support automated user story generation, but their use should include critical review and contextualization to ensure the quality and adequacy of generated requirements.

The perception of participants who do not yet use LLMs for this purpose does not express rejection of the technology but reveals practical barriers such as lack of time, absence of team participation in writing, and difficulty in crafting effective prompts. These reports suggest that adoption can be facilitated by ready and validated techniques, which reduce initial effort and increase professionals' confidence in the automated process.

Overall, the automated generation of user stories with LLMs is perceived as promising, especially when used as a support tool rather than a full replacement for human analysis. Critical and contextualized use is identified as an essential factor to ensure that benefits outweigh risks.

Moreover, analyzing respondents' suggestions and opinions in the final open question reveals a recurring emphasis on the importance of providing context as an essential factor for the good performance of LLMs in generating user stories. This provision can be automated, comprehensive, including all project artifacts, and initiated from the beginning of development. Such practice is seen as crucial to ensure that LLMs produce more relevant, clear, and aligned stories with the real needs of the project.

However, although the provision of context is widely recognized as essential for the effectiveness of these tools, data protection requirements, especially in private corporate environments, pose a significant challenge. Legal restrictions, such as those imposed by the LGPD, limit the sharing of sensitive information, requiring additional efforts to anonymize or mask data without compromising the necessary context.

These reports suggest that to enable safe and effective adoption of LLMs in corporate contexts, it is necessary to go beyond technical capacity of the tools, establishing clear strategies to guarantee privacy and information security. Concern about exposing sensitive data spans different experience levels and reinforces the importance of developing solutions that simplify this process, balancing protection and performance. This perception may reveal a barrier to adoption: distrust regarding data security remains an obstacle, even among those who recognize the potential of the technology, especially in corporate contexts dealing with confidential information or regulated by laws such as LGPD. To mitigate these risks, a viable solution is prompt engineering with minimal data, prioritizing essential context needed for the task. This includes only information such as user type, objectives, and general system scenario, avoiding names, identifiers, financial or personal data. Another complementary approach is use of local or private LLMs, which, when run in controlled environments (on-premises), ensure data is not transmitted to external services, increasing control and security over information used in automated requirements generation processes.

7 Threats to Validity

This study is subject to common threats to validity in survey research [16]. In terms of construct validity, terms such as "user story quality" may have been interpreted in different ways by respondents. To minimize ambiguity, a pilot study was conducted and the questionnaire was revised accordingly. Regarding internal validity, although the study is descriptive and does not seek to establish causal relationships, the inclusion of examples may have influenced responses, and open-ended answers remain subject to researcher interpretation bias. For external validity, the sample consisted of 21 participants recruited via convenience channels. Consequently, invitation counts and response rates could not be recorded, which limits the generalizability of the findings. Only professionals who self-declared experience in requirements engineering were included to ensure relevance. Most participants (85.7%) work for companies headquartered in Manaus, which may further limit the external validity of the results to other regions or organizational contexts. Regarding reliability, no formal statistical tests were conducted, but the questions were standardized and missing data were minimal, handled on a per-question basis. Finally, to ensure objectivity, all data collection and analysis procedures were documented and applied consistently to minimize potential researcher bias.

8 Conclusion

This study investigated how experienced requirements engineering professionals elicit and specify requirements, as well as their perceptions and use of LLMs, focusing on automated user story generation. Twenty-one professionals participated, most of whom were from Manaus-AM and had substantial field experience.

The most cited elicitation techniques were interviews, document analysis, brainstorming, and Lean Inception. For validation, participants emphasized prototyping, stakeholder reviews, and testing. A total of 57.1% of respondents reported using user stories, reflecting a trend toward consolidation, though with room for broader adoption. There is a preference for quality criteria that balance structure and utility (e.g., "Well-Formed," "Atomic," "Estimable").

Regarding LLM adoption, 66.7% of participants have integrated such tools into their routines, most commonly ChatGPT, for activities like brainstorming, technical assistance, and creative support.

The cross-analysis of the responses reveals significant differences in the adoption of user stories and LLMs according to professional experience. Early-career professionals (<5 years) show limited use of user stories and do not apply LLMs in story generation, highlighting the need for well-designed prompts to leverage the tool effectively. Among those with more than 5 years of experience, the adoption of user stories is universal, but only one uses LLMs, perceiving benefits such as increased speed, consistency, and criteria validation, although risks related to lack of context and inconsistencies are present. In the >10 years group, the use of user stories and LLMs is moderate, with LLMs applied occasionally in story generation, providing gains in inspiration, brainstorming, and standardization, but requiring refinement to avoid misalignment with business needs. Finally, professionals with more than 20 years of experience demonstrate a direct integration between agile practices and LLMs, applying them in story generation and perceiving clear positive impacts, such as reduced effort, faster writing, and creative support, while the associated risks involve potential overreliance.

Furthermore, participants note that LLM outputs often require manual refinement, limiting their reliability for direct adoption in critical RE tasks.

Suggestions from participants stress the importance of contextualizing LLMs with project artifacts to improve story relevance, and highlight data protection as a key challenge for corporate adoption. These insights reinforce the potential of LLMs to support user story generation, while also indicating the need for clear guidelines, secure practices, and further research into prompt strategies and enterprise-ready solutions. Future studies could expand the sample to include larger and more diverse participants across industries and regions, as well as complement perception-based findings with case studies or controlled experiments conducted in real companies to validate the results in practical settings.

ARTIFACT AVAILABILITY

Supplementary materials and research artifacts related to this study, including the survey instrument, survey flowchart and industry professionals' responses, are publicly available at the following link: https://github.com/Reine66/SBQS25-Material.

ACKNOWLEDGMENTS

We thank the participants of the empirical study and the members of the USES Research Group for their valuable support. This work was financially supported by CNPq (grants 314797/2023-8, 443934/2023-1, and 445029/2024-2), CAPES (Funding Code 001), and FAPEAM through the POSGRAD 25-26 program. Funding was also provided by Project No. 017/2024 – DIVULGA CT&I/FAPEAM.

REFERENCES

[1] A. Alhazmi and S. Huang. 2020. Survey on differences of requirements engineering for traditional and agile development processes. In *2020 SoutheastCon*. IEEE, 1–9.

[2] L. Belzner, T. Gabor, and M. Wirsing. 2023. Large language model assisted software engineering: prospects, challenges, and a case study. In *International Conference on Bridging the Gap between AI and Reality*. Springer Nature Switzerland, Cham, 355–374. doi:doi/abs/10.1007/978-3-031-46002-9_23

[3] Fabiano Dalpiaz and Sjaak Brinkkemper. 2018. Agile Requirements Engineering with User Stories. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*. 506–507. doi:10.1109/RE.2018.00075

[4] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018). doi:10.48550/arXiv.1810.04805

[5] Hugo Ferreira Martins, Antônio Carvalho de Oliveira Junior, Edna Dias Canedo, Ricardo Ajax Dias Kosloski, Roberto Ávila Paldês, and Edgard Costa Oliveira. 2019. Design thinking: Desafios para elicitação de requisitos de software. *Informação* 10, 12 (2019), 371. doi:10.3390/info10120371

[6] Aleksander Jarzębowicz and Natalia Sitko. 2019. Communication and documentation practices in agile requirements engineering: a survey in polish software industry. In *EuroSymposium on Systems Analysis and Design*. Springer, 147–158.

[7] Barbara A. Kitchenham and Shari Lawrence Pfleeger. 2008. Pesquisas de Opinião Pessoal. In *Guide to Advanced Empirical Software Engineering*, Forrest Shull, Janice Singer, and Dag I. K. Sjøberg (Eds.). Springer, London. doi:10.1007/978-1-84800-044-5_3

[8] Johan Linåker, Sardar Muhammad Sulaman, Rafael Maiani de Mello, and Martin Höst. 2015. Guidelines for conducting surveys in software engineering. (2015).

[9] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper. 2016. The Use and Effectiveness of User Stories in Practice. In *Requirements Engineering: Foundation for Software Quality*, M. Daneva and O. Pastor (Eds.). Lecture Notes in Computer Science, Vol. 9619. Springer, Cham, 187–202. doi:10.1007/978-3-319-30282-9_14

[10] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and et al. 2016. Improving Agile Requirements: The Quality User Story Framework and Tool. *Requirements Engineering* 21, 4 (2016), 383–403. doi:10.1007/s00766-016-0250-x

[11] Jay U Oswal, Harshil T Kanakia, and Devvrat Suktel. 2024. Transforming Software Requirements into User Stories with GPT-3.5-: An AI-Powered Approach. In *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*. IEEE, 913–920.

[12] Shari Lawrence Pfleeger and Barbara A. Kitchenham. 2001. Principles of Survey Research: Part 1: Turning Lemons into Lemonade. *ACM SIGSOFT Software Engineering Notes* 26, 6 (2001), 16–18.

[13] Teade Punter, Marcus Ciolkowski, Bernd Freimut, and Isabel John. 2003. Conducting on-line surveys in software engineering. In *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings*. IEEE, 80–88.

[14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. 2019. *Language Models are Unsupervised Multitask Learners*. Technical Report. OpenAI. https://liuyehcf.github.io/resources/paper/Language-Models-are-Unsupervised-Multitask-Learners.pdf

[15] Tajmilur Rahman, Yuecai Zhu, Lamyea Maha, Chanchal Roy, Banani Roy, and Kevin Schneider. 2024. Take Loads Off Your Developers: Automated User Story Generation using Large Language Model. In *2024 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 791–801. doi:10.1109/ICSME58944.2024.00082

[16] Paul Ralph, Sebastian Baltes, Domenico Bianculli, Yvonne Dittrich, Michael Felderer, Robert Feldt, Antonio Filieri, Carlo Furia, Daniel Graziotin, Pinjia He, Rashina Hoda, Natalia Juristo, Barbara Kitchenham, Romain Robbes, Daniel Méndez Fernández, Jefferson Molleri, Diomidis Spinellis, Miroslaw Staron, Klaas-Jan Stol, and Sira Vegas. 2020. ACM SIGSOFT Empirical Standards. doi:10.48550/arXiv.2010.03525

[17] I. Rasheed. 2021. Requirement Engineering Challenges in Agile Software Development. *Mathematical Problems in Engineering* 2021 (2021), 1–18. doi:10.1155/2021/6696695

[18] K. Ronanki, B. Cabrero-Daniel, and C. Berger. 2024. ChatGPT as a Tool for User Story Quality Evaluation: Trustworthy Out of the Box?. In *Agile Processes in Software Engineering and Extreme Programming – Workshops (Lecture Notes in Business Information Processing, Vol. 489)*, P. Kruchten and P. Gregory (Eds.). Springer, Cham. doi:10.1007/978-3-031-48550-3_17

[19] Daniel Russo. 2024. Navigating the Complexity of Generative AI Adoption in Software Engineering. *ACM Trans. Softw. Eng. Methodol.* 33, 5, Article 135 (June 2024), 50 pages. doi:10.1145/3652154

[20] Ian Sommerville. 2011. *Software Engineering* (9th ed.). Pearson, Boston. https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf?utm_source=chatgpt.com

[21] Andreas Vogelsang and Jonas Fischbach. 2025. Usando Modelos de Linguagem de Grande Porte para Tarefas de Processamento de Linguagem Natural em Engenharia de Requisitos: Uma Diretriz Sistemática. In *Manual de Processamento de Linguagem Natural para Engenharia de Requisitos*, Alessia Ferrari and Gaurav Ginde (Eds.). Springer, Cham. doi:10.1007/978-3-031-73143-3_16

[22] Stefan Wagner, Daniel Méndez Fernández, Marcos Kalinowski, and Michael Felderer. 2018. Agile requirements engineering in practice: Status quo and critical problems. *CLEI Electronic Journal* 21, 1 (2018), 6–1.

[23] B. Wake and B. Wake. 2003. *Invest in good stories, and Smart Tasks*. https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/ Accessed: 28 April 2023.

[24] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, Anders Wesslén, et al. 2012. *Experimentation in software engineering*. Vol. 236. Springer.

[25] Z. Zhang, M. Rayhan, T. Herda, M. Goisauf, and P. Abrahamsson. 2024. LLM-based agents for automating the enhancement of user story quality: An early report. In *International Conference on Agile Software Development* (Cham). Springer Nature Switzerland, 117–126.

[26] Liping Zhao, Waad Alhoshan, Alessio Ferrari, Keletso J. Letsholo, Muideen A. Ajagbe, Erol-Valeriu Chioasca, and Riza T. Batista-Navarro. 2021. Natural Language Processing for Requirements Engineering: A Systematic Mapping Study. *ACM Comput. Surv.* 54, 3, Article 55 (April 2021), 41 pages. doi:10.1145/3444689