AI-Generated User Stories: Are They Good Enough?

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

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

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

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

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

ABSTRACT

Large Language Models (LLMs), combined with advanced prompting techniques, have been used in requirements engineering, particularly in the automated generation of user stories. These stories are essential for agile software projects, but manual creation can be time-consuming and prone to inconsistencies, which has driven interest in automated approaches. Questions about the effectiveness and practical acceptance of these approaches, especially regarding quality and software professionals' perceptions, still remain. Furthermore, little is known about users' perspectives and the limitations of these new automation techniques. We conducted an empirical study with 24 participants who generated 457 user stories using the US-Prompt technique. We assessed the quality of the user stories using the QUS framework, and we analyzed the acceptance of the technique using the TAM3 model. Results showed that the US-Prompt method was effective, with 87.5% of the stories meeting more than 75% of the quality criteria. Participants found the technique easy to use and useful, although they identified limitations such as formatting inconsistencies and concerns about reliability for critical tasks. This study thus offers a provocative reflection on the use of LLMs and points to new directions for future research in this emerging area.

KEYWORDS

User Story, Large Language Models, Requirements Engineering.

1 Introduction

The increasing use of Large Language Models (LLMs) in software development tasks has significantly transformed the requirements engineering process. LLMs have become valuable tools for capturing and refining software requirements, streamlining and enriching the development process [1], including the creation of user stories.

A user story is a short and simple description of a software feature from an end user's perspective. They are extensively employed in agile software development projects to capture and convey end-user requirements effectively [7]. The quality of user stories is crucial, as it directly impacts system design and the final product [12]. However, the manual creation of user stories can be time-consuming, prone to inconsistencies, and often results in poorly written stories with inherent quality defects [6, 10]. To address the inefficiencies inherent in the manual creation, researchers conducted several studies on the automated generation of user stories using LLMs to streamline the process and improve quality [2, 9–11].

In this context, prompt engineering, with techniques such as Meta-Few-Shot Prompting, emerges as an advanced approach for automatically generating user stories [14], while maintaining quality standards based on the Quality User Story (QUS) framework defined by Lucassen et al. [6]. QUS consists of a set of 13 criteria that define the quality of user stories. However, beyond technical quality, it is essential to understand how future professionals in the field perceive this approach, as this perception may directly influence its adoption in real-world contexts.

This paper presents results from an empirical study with 24 participants, organized into six groups, in a Software Engineering course. The study examined the effectiveness and acceptance of a prompt for automated user story generation using ChatGPT, applied to different contexts through the Meta-Few-Shot Prompting technique. The prompt, referred to as the US-Prompt (User Story Prompt), includes: (1) Activity introduction, (2) Product vision, (3) Users and actions, (4) Template, (5) Quality criteria, (6) Few-shot examples, and (7) Generation request.

To assess the acceptance of the US-Prompt, we used the TAM3 model (Technology Acceptance Model 3) [16]. A total of 457 user stories were generated, 87.5% of which achieved success rates above 75% in adherence to the QUS quality criteria. The results indicate that, even in its standard version, the US-Prompt effectively generates consistent and high-quality user stories.

2 Background

Recent advancements in Large Language Models (LLMs), such as OpenAI's GPT [8], have opened new avenues in software engineering tasks, including automated requirements specification, where prompt engineering plays a crucial role in refining instructions to improve model performance. Prompt engineering is the process of crafting and adjusting input texts (prompts) used to guide the behavior of language models. Since the results are not always satisfactory on the first attempt, it is common to refine the prompt several times. An effective technique is the use of examples within the prompt itself, known as in-context learning, which helps the model better understand the task [15].

Well-crafted prompts steer the model toward the desired outcomes, improving the quality and relevance of the outputs. Techniques such as few-shot prompting and detailed instructions are particularly effective, as they help incorporate prior knowledge into the model, resulting in significant performance improvements and more accurate and coherent responses [17, 20].

Recent researches increasingly explore the integration of meta-learning and few-shot learning. Brown et al. [3] define meta-learning as a two-level learning framework (inner-loop/outer-loop), highlighting in-context learning during inference, in which models leverage skills acquired through unsupervised pre-training. This approach enables rapid adaptation to new tasks by identifying repeated patterns in sequences.

Applying prompting techniques to the generation of software requirements highlights the need to align model outputs with established principles from Requirements Engineering. Just as prompt engineering uses structured inputs to improve model performance, Requirements Engineering relies on well-defined guidelines to ensure structure, presentation, and understanding in user stories.

General guidelines from Requirements Engineering, along with frameworks such as INVEST and the Quality User Story (QUS), provide structured criteria to guide the creation and evaluation of user stories [20]. Among these, the QUS framework proposed by Lucassen et al. [6] stands out for offering a comprehensive assessment of user stories through 13 quality criteria, covering aspects of structure and understanding. These criteria help ensure that development teams accurately interpret and implement requirements. In this study, the criteria from Ronanki et al. [12] were adopted: Well-Formed, Atomic, Minimal, Conceptually Sound, Unambiguous, Complete Sentence, and Estimable. Descriptions of the criteria are in the Supplementary Material[1].

2.1 Related Work

Regarding the use of LLMs for automated user story generation, recent research has explored frameworks to assess the quality of the outputs. Rahman et al. [10] introduced the tool "GeneUS," which uses GPT-4.0 and the "Chain-of-Thought Prompting" (CoT) technique to automate the generation of user stories. The "Refine and Thought" (RaT) strategy is employed to extract and refine requirements, and story quality is assessed using the RUST questionnaire (Readability, Understandability, Specificity, and Technical aspects).

Brockenbrough and Salinas [2] investigated the use of ChatGPT by Computer Science students for creating user stories, using the INVEST framework to assess the quality of the generated responses. The results indicated that using ChatGPT can improve understanding of requirements and increase efficiency in software development by producing more relevant and coherent stories.

In our previous study [14], we conducted two experiments assessing stories generated by ChatGPT: the first experiment found no significant difference between manually written stories and those generated using free-form prompts; the second, using prompt techniques, achieved 88.57% success, 22.72% higher than the free-form approach (65.85%), according to the QUS framework.

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

Compared to our previous study [14], which focused on a single scenario, this work differs in study design by comparing the performance of the US-Prompt across different types of software projects defined by the participants themselves. Furthermore, it incorporates an acceptance assessment based on the TAM3 model, an aspect absent from previous studies. Unlike Rahman et al. [10], who employed GPT-4.0 with the Chain-of-Thought approach and evaluated stories using the RUST instrument, we use the QUS framework and promotes active user participation in prompt construction. Lastly, while Brockenbrough and Salinas [2] explore the educational use of ChatGPT with the INVEST framework, their analysis does not address broader applicability or user acceptance of the prompt.

This study advances prior work by evaluating the quality and acceptance of automated user story generation with LLMs across different types of software projects, highlighting the technique's adaptability and the role of prompt customization in settings that resemble real-world scenarios.

3 Empirical Study

This study aims to answer the following research question: **"Is the adoption of the US-Prompt capable of generating high-quality user stories in different projects?"** To address this question, the technique was applied in an experimental setting, providing insights into its effectiveness and acceptance. The methodology follows the guidelines of Wohlin et al. [18] for conducting experimental studies, enabling a rigorous analysis of the results.

The study formulated the following hypotheses:

$\mathbf{H_{01}}$: There is no significant difference in the quality of user stories generated using the Standard US-Prompt compared to those generated using the Modified US-Prompt.

$\mathbf{H_{A1}}$: There is a significant difference in the quality of user stories generated using the Standard US-Prompt compared to those generated using the Modified US-Prompt.

The independent variable in the study is the method of automated user story generation using the US-Prompt, which includes two treatments: (1) the use of the Standard US-Prompt and (2) the use of the Modified US-Prompt. The dependent variable is the quality of the user stories, assessed using the QUS framework.

3.1 Empirical Study Design

We designed the study to simulate real-world scenarios typically encountered by novice software engineers, aiming to establish controlled and consistent data collection conditions across participant groups. Although the scenarios were created by students, each project was planned and implemented based on agile development practices, involving realistic problem definitions, requirements elicitation, and iterative implementation. The main goal was to evaluate the effectiveness of the US-Prompt in generating high-quality user stories, as well as to identify factors influencing its acceptance among future professionals in the field.

The study followed a three-phase structure: (1) evaluation of user stories generated using a predefined standard version of the US-Prompt; (2) evaluation of stories generated using a participant-modified version of the US-Prompt; and (3) assessment of US-Prompt acceptance using a TAM3-based questionnaire [16] using following constructs: Perceived Usefulness, Perceived Ease of Use,

and Behavioral Intention to Use, measured on a five-point Likert scale. Effectiveness was measured by the success rate, defined as the proportion of stories that met the QUS criteria.

We recruited participants from the Advanced Topics in Software Engineering, offered jointly to undergraduate and graduate students of the Software Engineering program at UFAM. In total, 29 participants agreed to take part in the study and signed an informed consent form, with the option to withdraw at any time, ensuring ethical compliance and voluntary participation. This research was approved by the CEP of the Federal University of Amazonas (CAAE [829552724.0.0000.5020]).

3.2 Execution of the Empirical Study

Initially, participants were self-organized into seven teams based on mutual affinities and availability. However, only six teams completed all the proposed phases of the study, totaling 24 participants. Of these, 6 are students who also work as software professionals.

We instructed the teams to define the context of a system they wished to develop. The project descriptions are available in the Supplementary Material (see footnote 1). Participants received training on specialized prompts. GPT-4o (Mini) was used with default settings (temperature 0.7, top-p 1.0, frequency and presence penalties 0) to generate user stories. The study comprises three phases:

Phase 1 – Generation with US-Prompt: Each participant individually generated a set of user stories using the standard US-Prompt provided by the study. Participants based their stories on the previously defined system context of each team. Since all 24 participants contributed a set of stories, this phase resulted in 24 sets of user stories, totaling **372 stories**, distributed among the teams as follows: Team A (59 stories), Team B (42 stories), Team C (69 stories), Team D (75 stories), Team E (51 stories), and Team F (76 stories).

Phase 2 – Generation with Modified US-Prompt: Participants worked as a group to collaboratively revise and enhance the standard US-Prompt. After modifying the prompt, each team generated a new set of user stories using their revised version. Five of the six teams produced 5 new sets of stories, resulting in 85 new stories. One team chose not to make any modifications to the prompt and retained the stories generated in the previous phase. All teams used the same system description for both phases.

Phase 3 – Acceptance Evaluation (TAM): After the story generation, the TAM3 model was applied to measure the acceptance of the US-Prompt, considering perceived usefulness, perceived ease of use, and intention to use it in the future.

3.3 Evaluation of the Generated User Stories

The user stories generated were evaluated through quantitative metrics, focusing on the effectiveness of the generation process.

We assessed each user story according to the QUS criteria using a binary scoring system: a score of "1" indicated compliance with the criterion, while "0" indicated non-compliance. This determined whether each story met the specific quality criterion. For example, the story from Group C, "As a cinephile, I want to filter the movie catalog by genre and director," should be split into two atomic stories: "As a cinephile, I want to filter by genre" and "As a cinephile, I want to filter by director." The overall effectiveness score for a set of user stories was calculated as a success rate, representing the percentage of criteria satisfied relative to the total evaluated. This rate served as the primary metric for comparing the performance of different prompts and analyzing the overall quality of the stories.

First, we summed the total number of satisfied criteria across all evaluated user stories. Then, we determined the maximum number of possible successes by multiplying the number of evaluated stories (N) by the number of quality criteria (C, which is 7). Finally, the number of criteria satisfied was divided by the total possible successes and multiplied by 100 to obtain the success rate. This rate reflects the overall adherence of the user stories to the QUS quality criteria and indicates the technique's effectiveness.

$$\text{Success Rate (\%)} = \frac{\sum_{i=1}^{N} \text{Criteria Met for Story}_i}{N \times C} \times 100$$

Further information and data can be found in the Supplementary Material (see footnote 1).

4 Results and Discussion

The study analyzed both quantitative and qualitative data. Quantitative results were based on the success rates of user story generation. For qualitative data, open-ended responses from the TAM3-based questionnaire [16] were manually read and analyzed. Responses were grouped into emergent thematic categories such as clarity of instructions, perceived usefulness, and suggestions for improvement. Although no qualitative analysis software was used, the coding was performed manually to identify recurring patterns across participants' perceptions.

4.1 Quantitative and Qualitative Analysis

The analysis considered the quality evaluation of user stories generated with both the standard US-Prompt and the modified US-Prompt. The success rate for each quality criterion was calculated independently of the generation method.

The average number of user stories generated per participant was 14.88, with only small variations across groups. Although P6 produced only 9 stories, most participants, particularly those in groups A, D, and E, generated 14 or more, indicating consistent output using the US-Prompt. These results support the prompt version's capability to sustain a stable volume of user story generation.

Figure 1 illustrates the high effectiveness of the standard US-Prompt in generating user stories. Of the 24 analyzed sets, 21 (87.5%) reached or exceeded the minimum success rate of 75%. Only 3 sets (12.5%) fell below this threshold. The overall average success rate was 82.28%. These results reinforce the consistency and effectiveness of the approach implemented. The high success rate highlights the potential of the US-Prompt to generate user stories aligned with quality criteria, even before any adjustments or customization.

4.1.1 Analysis of Prompt Refinement and Performance Comparison. The groups started with the standard US-Prompt and applied distinct adaptations, which influenced the quality of the generated user stories as reflected in their QUS scores. The variations in refinement strategies reveal how different choices affected the prompt's effectiveness. Prompt modifications by each group are listed below:

Group A: Used an informal tone with emojis and a direct command ("CREATE") to clarify the task.

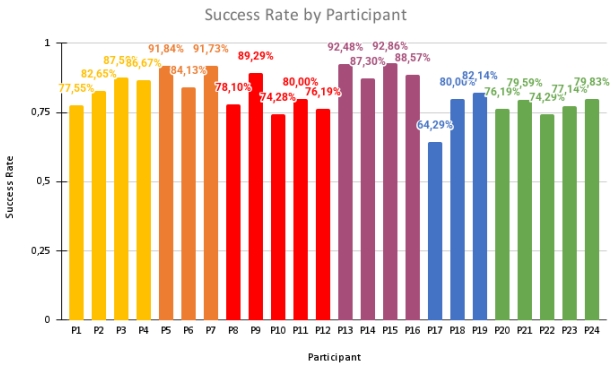

Figure 1: Participants' Individual Success Rates in User Story Generation with the US-Prompt Pattern

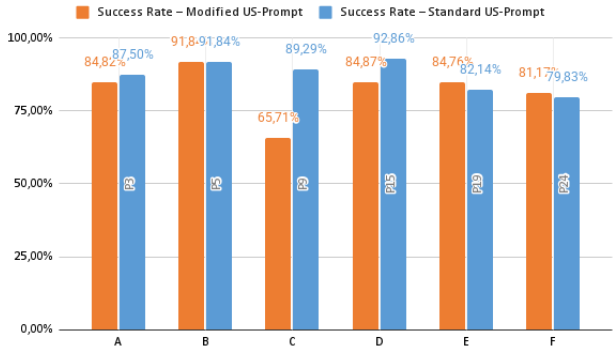

Figure 2: Comparison of Success Rates Between the Standard and Modified Versions of the US-Prompt.

Group B: Retained the original prompt unchanged.
Group C: Enriched the prompt with functionality examples (e.g., "registration form," "photo upload") to increase specificity.
Group D: Introduced proto-personas to contextualize story generation within realistic usage scenarios.
Group E: Expanded the prompt with a detailed product vision, defined roles, business rules, new fields, and contextual examples—focusing on functional clarity. The QUS criteria were removed to prioritize contextual richness, though prior exposure to the standard prompt may have helped maintain quality.
Group F: Segmented the generation into base and additional story versions to improve clarity and control. Added a second prompt to enhance atomicity by refining complex stories into simpler ones.

Figure 2 shows each group's highest QUS score from Phase 1, representing the best possible performance ("upper bound"), alongside the corresponding score in Phase 2. This integrated view highlights both the potential and variability of each prompting approach.

In most groups, the standard US-Prompt performed equally well or better than the modified version. The modified version only slightly outperformed the original in two cases (Groups E (P19) and F (P21)). In contrast, in three groups — A (P3), C (P9), and D (P15) — the Standard US-Prompt yielded significantly better results, with Group C exhibiting the widest difference (89.29% vs. 65.71%).

These findings suggest that while customization can lead to improvements, particularly when incorporating well-defined roles, business rules, and segmentation strategies, such gains are not guaranteed. For instance, Groups E and F benefited from enhancements such as increased contextual detail and technical structuring, including atomic segmentation and auxiliary prompts. Meanwhile, Group D also applied structural changes but did not achieve improved outcomes, reinforcing that greater complexity alone is insufficient.

Effective refinements require close alignment with the QUS framework's quality dimensions: syntactic clarity, semantic validity, and pragmatic relevance. The most successful strategies combined improved context, clear instructions, and quality-focused restructuring, supported by iterative validation and feedback.

To evaluate the effectiveness of user-story generation with the standard and modified US-Prompts, we conducted a statistical analysis using JASP (v 0.19.1). The standard US-Prompt achieved an average success rate of 87.24%, whereas the modified US-Prompt reached 82.20%, suggesting a slight advantage for the former.

To assess the normality of the data, we applied the Shapiro–Wilk Test, which confirmed that both datasets followed a normal distribution. Given this result, we proceeded with a paired t-test to compare the average success rates of the Standard and Modified US-Prompts. The test yielded a non-significant result ($p = 0.264$), which is greater than the conventional significance threshold ($\alpha = 0.05$), indicating that there is no statistically significant difference between the two prompt versions. Hence, the null hypothesis H_0, which states that there is no significant difference in the quality of user stories generated by the two prompts, was not rejected. This supports the conclusion that both prompts yield user stories of comparable quality. Additionally, the standard US-Prompt exhibited lower variability (SD = 5.252%) compared to the modified US-Prompt (SD = 8.792%), indicating more consistent results.

4.1.2 Assessment of Acceptance and Participant Perceptions: We used the follow-up questionnaire to gather data on the intention to use the proposed technique and identify improvement opportunities. The data was obtained through a TAM3-based questionnaire followed by an open-ended question: **"Do you have any comments or suggestions for improving the User Story Prompt?"**

The responses shown in the graph in Figure 3 indicate a positive trend regarding the construct of **Ease of Use** of the prompt. Most participants strongly agreed (72.4%) or agreed (20.7%) with the statement related to clarity of interaction (F1. "My interaction with the prompt for generating user stories was clear and understandable"). Regarding low cognitive effort (F2. "Interacting with the prompt to generate user stories did not require much mental effort"), 62.1% strongly agreed, and 20.7% agreed. Only 10.3% disagreed with the statement. For general ease of use (F3. "I find the prompt easy to use"), 79.3% strongly agreed. In terms of ease in generating user stories (F4. "I find it easy to use the prompt to generate user stories"), 62.1% strongly agreed, and 10.3% remained neutral, suggesting that some participants faced barriers when interacting with the prompt.

When asked about their experience with the prompt, participants shared comments at the end of the questionnaire. Participant P4 evaluated it directly and positively: *"... it was efficient and intuitive."* Participant P9 highlighted the ease of use: *"I found it simple and easy*

to use...". Participant P5 emphasized its usefulness: *"I found it easy and productive to use..."*. Participant P15 noted the supportive role of prompts: *"Using prompts facilitates the creation of user stories..."*. Participant P8 commented: *"... in this case I just copied and pasted, so I didn't make any mental effort like I usually would,"* acknowledging a reduced cognitive load—an indicator of positive usability.

However, participants P21 and P22 disagreed with the statement that interaction with the prompt required little mental effort, possibly due to a lack of familiarity with interacting with LLMs and using prompts. This may have led to additional cognitive effort or uncertainty regarding how to formulate good inputs.

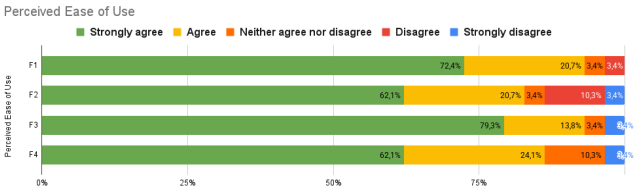

Figure 3: Distribution of responses regarding the Perceived Ease of Use construct.

The graph of Figure 4 also reveals a positive perception regarding the **Perceived Usefulness** construct of the prompt. A large portion of participants strongly agreed (64.3%) that using the prompt improves their performance (U1. Using the prompt improves my performance in creating user stories), 70.4% strongly agreed and 18.5% agreed that it improves productivity (U2. Using the prompt in my work could improve my productivity), and 60.7% strongly agreed that it improves effectiveness (U3. Using the prompt improves the effectiveness of generating user stories for the project).

However, compared to the Perceived Ease of Use construct, the proportion of neutral responses across all statements increased, especially in U3 (17.9%), indicating that although the prompt is perceived as easy to use, its usefulness was not clear to all participants.

Participant P9 highlighted that the prompt helped to speed up the story generation process: "*I believe it optimized time and the chat responses were satisfactory for our system...*", representing one of the main goals sought with the adopted approach, time saving.

P5 highlighted not only the ease of use but also the usefulness: "*I found it easy and productive to use... The Chat generated good user stories with it*," emphasizing both its usefulness and the quality of the generated stories. And P15 considers the approach as initial support: "*Using prompts facilitates the creation of user stories, serving as a starting point...*", highlighting its role in supporting story creation.

The reports indicate that, in the participants' perception, the prompt-based approach facilitates the creation process, reduces execution time, and increases productivity. Although most participants perceived the prompt as useful, some responses revealed divergent experiences, suggesting that usefulness may vary depending on context and user profile. As emphasized by participant P22: "I felt like I was already writing the user stories myself. Considering that, I think it's easier for me to just write them than to wait for it to do it, since everything is already broken down into actions." This perception of low usefulness, combined with frustration at not perceiving efficiency gains, contributes to a tendency toward non-adoption in the future.

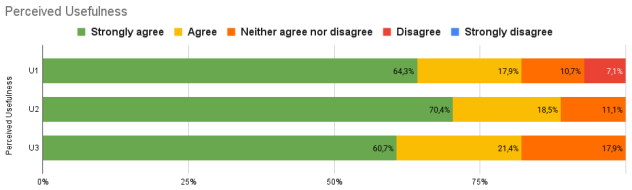

Figure 4: Distribution of responses regarding the Perceived Usefulness construct.

In the graph (Figure 5), referring to the construct **Behavioral Intention to Use**, the majority of participants (65.5%) strongly agreed with the possibility of using the prompt (I1. Assuming I have access to this prompt, I would use it to generate user stories), and 20.7% agreed with the statement, as well as for (I2. If I have access to this prompt, I intend to use it to generate user stories in future projects). However, it was observed that the proportion of neutral responses remained high (20.7%), and disagreements increased in (I3. I intend to use the prompt in the coming months to create user stories) to 13.8%, which may indicate a lower willingness to use the technique in immediate and real contexts.

Participants P17, P19, and P21 disagreed that they would use the prompt to generate user stories even if they had access to it. P17 and P22 declared that they do not intend to use the prompt in future projects, while P17, P19, and P22 stated that they have no intention of using it in the coming months. These data point to a lower intention for future usage among these participants despite their prior experience.

One possible explanation for this behavior is that, although the tool was considered useful and easy to use, there is uncertainty regarding its practical application in the short term. This may be related to the fact that many participants do not frequently have opportunities in their professional routine to create user stories. Thus, even though the experience with the prompt was positive, the future usage intention may have been impacted by the lack of immediate applicability of the technique, reinforcing the idea that the willingness to use is higher in hypothetical scenarios but decreases when faced with practical reality.

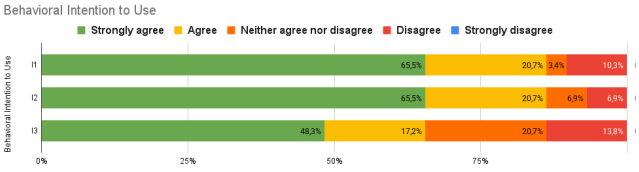

Figure 5: Distribution of responses regarding the construct Behavioral Intention to Use.

Although the prompt is considered useful and easy to use, its concrete applicability is not yet fully consolidated for all users, especially in terms of practical use in everyday life.

4.2 Improvement Recommendations

Several participants suggested improvements that could make the prompt usage even more effective. Participants focused their suggestions on three areas: exemplification of roles and contexts, input format, and story granularity. The recommendations are as follows:

Participant P2 recommends including clearer examples on how to properly specify user roles: *"...it could provide an example of how to list the users... so we can better understand how to put them."* P17 highlighted that the model went beyond the provided elements by including an unrequested actor. The participant further emphasizes this recommendation by suggesting the prior provision of a more detailed template on how to describe the system context: "A model of how the system context should be written to be inserted into the prompt could be provided beforehand."

These suggestions indicate that, for many participants, the effectiveness of the technique is directly related to the quality of prior instructions given. They reinforce the importance of offering clear and contextualized examples to reduce ambiguities and increase the model's adherence to the user's intentions. However, it is necessary to analyze the extent to which these examples may introduce bias.

P22 suggests changing the input format to avoid the feeling of "doing the work alone": "It would be better if activity 3 was filled with continuous text rather than a list of actions..." This implies that fragmented inputs reduce perceived value, as they make the model seem like it's just reorganizing user content. And P15 proposes a relevant improvement regarding the granularity of the results, noting that some generated stories could be more atomic. According to them, *"...I believe some could be broken down into more user stories..."*. These suggestions point to improvements in both the prompt's input and output, seeking better alignment with user expectations.

Ease of use alone does not guarantee prompt adoption. Lack of perceived benefits, quality issues, and unpredictability reduce trust. For repeated use, the prompt must deliver value, reliability, and meet user expectations.

5 Limitations

Although most of the sample consisted of Software Engineering students, prior research indicates they can adequately represent industry professionals [4, 13]. The Quality User Story (QUS) framework provided robust criteria for evaluating user stories but may overlook nuances in more complex contexts. The small sample size limits statistical power and may introduce biases, partially mitigated by review from two independent researchers. LLMs, such as ChatGPT, can produce incorrect information (hallucinations) [5]. Therefore, a requirements expert reviewed the stories during the QUS evaluation. The experiment did not fully control external variables, including participant experience, task environment, and prior tool familiarity, which may have influenced results. Future work should better control these factors to enhance validity.

6 Conclusion and Future Work

This study evaluated the effectiveness and acceptance of the US-Prompt technique for automated user story generation using LLMs, applying the method in different contexts defined by the participants. The results showed that the technique is capable of generating high-quality user stories, with 87.5% of the sets achieving over 75% adherence to the criteria of the QUS framework. Analysis of prompt modifications revealed that, although some adaptations improved generation, others compromised quality, highlighting the need for well-guided refinements based on clear guidelines.

These findings show the US-Prompt's strong potential as a robust, adaptable tool. Prior studies on LLM-generated user stories show good overall quality but highlight challenges in specifiability and technical details [10]. Brockenbrough and Salinas [2] report an 88% improvement in acceptance test criteria and story value. Yamani et al. [19] note human stories outperform LLMs in some aspects. Our QUS success rate reached 87.24% in top-performing groups, indicating strong and balanced quality.

The acceptance evaluation of the US-Prompt indicated that participants found the prompt easy to use and useful, mainly due to reduced cognitive effort and accelerated story creation. However, future usage intention varied, reflecting uncertainties regarding the technique's practical application in real-world environments.

This study contributes by demonstrating the potential of the US-Prompt as an effective starting point for automated user story generation, integrating objective quality assessment and user perception. As future work, we plan to redesign US-Prompt by integrating participant feedback and utilizing other techniques, alongside the assessment, to develop a more powerful and intuitive tool. This improved prompt will be tested in real-world scenarios with professional teams, with the goal of expanding the boundaries of automated user story generation.

ARTIFACT AVAILABILITY

Artifacts used and produced during the experiment are available at: https://github.com/Reine66/Supplementary-Material. The repository includes the prompts, generated user stories, quality evaluations (QUS), analysis of modifications, the TAM questionnaire, and the adopted framework.

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] 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

[2] Allan Brockenbrough and Dominic Salinas. 2024. Using Generative AI to Create User Stories in the Software Engineering Classroom. In *2024 36th International Conference on Software Engineering Education and Training (CSEET)*. 1–5. doi:10.1109/CSEET62301.2024.10662994

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates,

Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[4] Martin Höst, Björn Regnell, and Claes Wohlin. 2000. Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. *Empirical Software Engineering* 5 (2000), 201–214.

[5] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.* (Nov. 2024). doi:10.1145/3703155 Just Accepted.

[6] 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

[7] Garm Lucassen, Fabiano Dalpiaz, Jan Martijn E. M. van der Werf, and Sjaak Brinkkemper. 2016. The Use and Effectiveness of User Stories in Practice. In *Requirements Engineering: Foundation for Software Quality*, Maya Daneva and Oscar Pastor (Eds.). Springer International Publishing, Cham, 205–222.

[8] OpenAI. 2024. ChatGPT (versão GPT-4). https://chat.openai.com/. Acesso em: 19 maio 2025.

[9] 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.

[10] 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

[11] Vijayalakshmi Ramasamy, Suganya Ramamoorthy, Gursimran Singh Walia, Eli Kulpinski, and Aaron Antreassian. 2024. Enhancing User Story Generation in Agile Software Development Through Open AI and Prompt Engineering. In *2024 IEEE Frontiers in Education Conference (FIE)*. 1–8. doi:10.1109/FIE61694.2024.10893343

[12] 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

[13] Iflaah Salman, Ayse Tosun Misirli, and Natalia Juristo. 2015. Are students representatives of professionals in software engineering experiments?. In *2015 IEEE/ACM 37th IEEE international conference on software engineering*, Vol. 1. IEEE, 666–676.

[14] Reine Santos, Gabriel Freitas, Igor Steinmacher, Tayana Conte, Ana Oran, and Bruno Gadelha. 2025. User Stories: Does ChatGPT Do It Better?. In *Proceedings of the 27th International Conference on Enterprise Information Systems - Volume 2: ICEIS*. INSTICC, SciTePress, 47–58. doi:10.5220/0013365500003929

[15] Jose Sousa, Cristian Souza, Raiza Hanada, Diogo Nascimento, and Eliane Collins. 2024. Generation of test datasets using LLM - Quality Assurance Perspective. In *Anais do XXXVIII Simpósio Brasileiro de Engenharia de Software* (Curitiba/PR). SBC, Porto Alegre, RS, Brasil, 644–650. doi:10.5753/sbes.2024.3587

[16] Viswanath Venkatesh and Hillol Bala. 2008. Technology acceptance model 3 and a research agenda on interventions. *Decision sciences* 39, 2 (2008), 273–315.

[17] Andreas Vogelsang. 2024. From Specifications to Prompts: On the Future of Generative Large Language Models in Requirements Engineering. *IEEE Software* 41, 5 (2024), 9–13. doi:10.1109/MS.2024.3410712

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

[19] Asma Yamani, Malak Baslyman, and Moataz Ahmed. 2025. Leveraging LLMs for User Stories in AI Systems: UStAI Dataset. In *Proceedings of the 21st International Conference on Predictive Models and Data Analytics in Software Engineering* (Trondheim, Norway) *(PROMISE '25)*. Association for Computing Machinery, New York, NY, USA, 21–30. doi:10.1145/3727582.3728689

[20] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic Chain of Thought Prompting in Large Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=5NTt8GFjUHkr