







# User Stories: Does ChatGPT Do It Better?

Reine Santos<sup>1</sup>, Gabriel Freitas<sup>1</sup>, Igor Steinmacher<sup>2</sup>, Tayana Conte<sup>1</sup>,  
Ana Carolina Oran<sup>1</sup> and Bruno Gadelha<sup>1</sup>

<sup>1</sup>*Instituto de Computação (ICOMP), Universidade Federal do Amazonas (UFAM), Av. Gal. Rodrigo Octávio Jordão Ramos, Manaus, Brazil*

<sup>2</sup>*Department of Computer Science, Northern Arizona University (NAU), 1900 S Knoles Dr, Flagstaff, AZ 86011, U.S.A. {rms, gabriel.freitas, tayana, ana.oran, bruno}@icomp.ufam.edu.br; igor.Steinmacher@nau.edu*

**Keywords:** User Story, Large Language Models, Requirements Engineering, Information System.

**Abstract:** In agile software development, user stories play a central role in defining system requirements, fostering communication, and guiding development efforts. Despite their importance, they are often poorly written, exhibiting quality defects that hinder project outcomes and reduce team efficiency. Manual methods for creating user stories are time-consuming and prone to errors and inconsistencies. Advancements in Large Language Models (LLMs), such as ChatGPT, present a promising avenue for automating and improving this process. This research explores whether user stories generated by ChatGPT, using prompting techniques, achieve higher quality than those created manually by humans. User stories were assessed using the Quality User Story (QUS) framework. We conducted two empirical studies to address this. The first study compared manually created user stories with those generated by ChatGPT through free-form prompt. This study involved 30 participants and found no statistically significant difference between the two methods. The second study compared free-form prompt with meta-few-shot prompt, demonstrating that the latter outperformed both, achieving higher consistency and semantic quality with an efficiency calculated based on the success rate of 88.57%. These findings highlight the potential of LLMs with prompting techniques to enhance user story generation, offering a reliable and effective alternative to traditional methods.

## 1 INTRODUCTION

In agile software development, requirements elicitation and specification are crucial for meeting end-user needs (Sommerville, 2011). However, the dynamic nature of agile environments, characterized by constantly evolving requirements and the need for rapid adaptation to customer needs, makes the creation of precise specifications challenging. This challenge is further intensified in interactive contexts, where development is both adaptive and iterative, requiring effective communication. The emphasis on quick deliveries can lead to misunderstandings and make it difficult to maintain a shared understanding among stakeholders (Rasheed, 2021).


Agile teams face challenges such as neglect-


ing quality requirements, excessive focus on minimal documentation, and poor prioritization, which may favor functional aspects. Frequent requirement changes cause instability, while inaccurate specifications and effort estimates complicate planning. Communication problems among stakeholders and inadequate architectural decisions also hinder progress (Hoy and Xu, 2023).


Requirements elicitation is often the most challenging process in software development, frequently cited as the main cause of project failure (Ferreira Martins et al., 2019). This process begins with interviews between analysts and stakeholders to capture domain requirements, but communication challenges can lead to misunderstandings and unclear requirements (Jaramillo, 2010).


Developing user stories is fundamental, especially in agile environments, where professionals use them flexibly to capture and express requirements. This practice facilitates communication between developers, clients, and other stakeholders, helping to prioritize and deliver features in development cycles and promoting continuous collaboration (Rahman et al.,


<sup>a</sup> <https://orcid.org/0009-0000-6060-228X>

<sup>b</sup> <https://orcid.org/0009-0008-0523-4531>

<sup>c</sup> <https://orcid.org/0000-0002-0612-5790>

<sup>d</sup> <https://orcid.org/0000-0001-6436-3773>

<sup>e</sup> <https://orcid.org/0000-0002-6446-7510>

<sup>f</sup> <https://orcid.org/0000-0001-7007-5209>

2024; Ronanki et al., 2024; Oswal et al., 2024).

Since these user stories are essential for guiding development and achieving the desired outcomes, it is crucial that they are well-crafted to avoid negative impacts on the client, project, and team (Oswal et al., 2024). However, user stories are often poorly written in practice and exhibit inherent quality defects (Lucassen et al., 2016b). The quality of user stories is crucial as it directly impacts system design and the final product. However, manually creating these stories can be time-consuming and prone to inconsistencies (Rahman et al., 2024; Ronanki et al., 2024).

With advancements in artificial intelligence (AI), particularly in Large Language Models (LLMs) like OpenAI's ChatGPT<sup>1</sup>, there has been significant potential to automate tasks that once demanded intensive human involvement (Belzner et al., 2023). By reducing human effort and the time spent on repetitive tasks, automation minimizes errors and increases efficiency, enabling developers to focus on higher-value activities (Yarlagadda, 2021). These models, skilled in interpreting prompts and generating text, enhance software engineering by providing expert knowledge and assisting developers throughout the software development lifecycle, from requirements engineering to system design (Belzner et al., 2023).

As a result, LLMs have become invaluable tools for capturing and refining software requirements, streamlining and enriching the development process (Belzner et al., 2023). Focusing on addressing the inefficiencies inherent in manual user story creation, various studies are being conducted on automated user story generation using GPT-based language models as a solution to streamline the process and enhance quality (Oswal et al., 2024; Rahman et al., 2024).

The central problem is to explore whether user stories generated by ChatGPT, utilizing prompting techniques, achieve higher quality than those created manually by humans. This comparison was carried out using the Quality User Story (QUS) framework (Lucassen et al., 2016a), which assesses key criteria such as atomicity, minimality, and soundness. By addressing this problem, the research aims to provide insights into the potential of AI to complement or surpass human efforts in generating user stories.

Thus, our research question is: **Do user stories generated by ChatGPT, using prompting techniques, exhibit superior quality compared to those manually written by humans?**

This research was divided into two studies: The first study compared manually created user stories with those generated by ChatGPT using free-form prompt. The second compared free-form prompt with

Meta-Few-Shot Prompt, a structured prompt composed of advanced prompting techniques designed to automate the generation of high-quality user stories.

## 2 BACKGROUND

### 2.1 User Stories

The requirements of a system define what it should do, including the services it provides and operational constraints, reflecting customer needs for specific purposes like controlling a device or finding information. The process of identifying these requirements is called Requirements Engineering (RE) (Sommerville, 2011). In agile methods, RE is flexible and iterative, with requirements defined incrementally through epics and user stories (Alhazmi and Huang, 2020).

User stories, which are succinct descriptions of functionalities from the user's perspective, are a key element in this approach. They capture the essential aspects of a requirement, such as who the user is, what is expected from the system, and why it is important (Zhang et al., 2023).

This process refines high-level requirements into actionable tasks using a just-in-time model for development (Ferreira Martins et al., 2019), aligning the system's functionality with user needs in an iterative and flexible manner. The widely adopted standard format is (Lucassen et al., 2016a): "As a ⟨type of user⟩, I want ⟨goal⟩, so that ⟨some reason⟩."

Each user story should include acceptance criteria that define the conditions for the story to be considered acceptable, covering both functional and quality aspects (Zhang et al., 2023). Lucassen et al. (2016) show that applying templates and quality guidelines to user stories boosts productivity and enhances the quality of the final product.

Effective writing of user stories is essential, as they communicate user needs and guide the development team. General quality guidelines in requirements engineering, along with frameworks like INVEST and Quality User Story (QUS), provide criteria for evaluating story quality (Zhang et al., 2023).

INVEST (Independent – Negotiable – Valuable – Estimatable – Scalable – Testable) is an approach for creating effective stories in agile environments. By adopting these principles, it is possible to enhance the quality of stories and increase the efficiency of agile development, improving communication between the team and stakeholders (Buglione and Abran, 2013).

Lucassen et al. (2016) present the QUS framework, a collection of 13 criteria that focus on the intrinsic quality of the user story text and evaluate its

<sup>1</sup><https://chatgpt.com/>

quality across three main categories: syntactic, pragmatic, and semantic. The criteria are: Well-Formed, Atomic, Minimal, Conceptually Solid, Problem-Oriented, Unambiguous, Conflict-Free, Complete Sentence, Estimable, Unique, Uniform, Independent, and Complete (Lucassen et al., 2016a).

These approaches ensure that user stories are concise and clear, contributing to the success of software development projects and enhancing the user experience (Zhang et al., 2023).

## 2.2 LLMs and Prompt Engineering Techniques

With advancements in Large Language Models (LLMs) like OpenAI's GPT, new opportunities have emerged in software engineering. Prompt engineering plays a crucial role in refining instructions to improve model performance. Well-crafted prompts direct the model toward desired results, enhancing output quality and relevance. Techniques such as few-shot prompt and detailed instructions are particularly effective, as they help incorporate pre-existing knowledge into the model, leading to significant performance improvements and more accurate, coherent responses (Zhang et al., 2023; Vogelsang, 2024).

The combination of meta-learning and few-shot learning has been studied in recent research. Brown et al. (2020) define "meta-learning" as an inner-loop/outer-loop framework, highlighting "in-context learning" during inference, where models leverage skills acquired through unsupervised pre-training. This approach enables rapid adaptation to new tasks by identifying repeated patterns within sequences. The term "few-shot" refers to scenarios where only a few demonstrations are provided at inference, reducing the need for task-specific data and avoiding overly narrow fine-tuning distributions.

Hiraou (2024) applies the approach Few-Shot Meta-Prompting, iteratively improving prompt structures using few-shot examples while preserving linguistic styles and syntax. These advancements enhance models' ability to generalize, resulting in more accurate and versatile responses across contexts.

Free-form prompt, however, is an approach without specific instructions, allowing for more spontaneity. However, this can lead to varied responses, potentially impacting adherence to quality standards.

## 2.3 Related Work

Several studies investigate the use of LLMs in generating software requirements. Marques et al. (2024) highlights the benefits of ChatGPT in requirements

engineering, such as automating documentation, reducing errors, and increasing team efficiency. White et al. (2024) emphasizes the importance of well-structured prompts, presenting design techniques for prompts in software engineering.

Krishna et al. (2024) compares the automatic generation of Software Requirements Specifications (SRS) documents using language models, such as GPT-4 and CodeLlama, with the work of novice engineers. The results indicate that these models can produce comparable SRS drafts and identify issues.

Ronanki et al. (2024) explores the potential of ChatGPT in requirements elicitation, comparing the quality of the generated requirements with those formulated by experts. The requirements generated by ChatGPT were found to be abstract and understandable, but they exhibited limitations in terms of ambiguity and feasibility.

Rahman and Zhu (2024) introduced the tool "GeneUS," which utilizes 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 the quality of the stories is evaluated using the RUST (Readability, Understandability, Specificifiability, and Technical aspects) questionnaire.

Brockenbrough and Salinas (2024) investigated the use of ChatGPT by computer science students to create user stories, using the INVEST framework to assess the quality of the generated responses. The results indicated that using ChatGPT can improve the understanding of requirements and increase efficiency in software development, producing more relevant and coherent stories.

This research addresses gaps in generating high-quality user stories by comparing manual methods with ChatGPT to identify which yields better results.

## 3 METHOD

This research analyzes whether user stories generated using prompting techniques with ChatGPT achieve higher quality than those created manually by humans. To this end, the study evaluates the ability of each approach to produce user stories that best adhere to QUS quality criteria, i.e., identifying which technique demonstrates greater effectiveness in generating high-quality stories—whether they are manually created or assisted by ChatGPT.

To assess the effectiveness of the techniques, we followed the same evaluation process used in Ronanki's (2024) study, based on the Quality User Story (QUS) a holistic framework proposed by Lucassen et

al. (2016) (Lucassen et al., 2016b). These criteria offer a solid basis for comparing the methods, emphasizing the quality and compliance of the user stories with well-defined requirements. Table 1 presents the criteria, with detailed descriptions to ensure clarity and precision during evaluation.

Each user story was evaluated against the QUS criteria using a binary scoring system, where a score of 1 indicated satisfaction and 0 indicated non-compliance, determining whether a user story meets or does not meet the assessed quality criterion, aligning with the study's purpose of measuring strict adherence to QUS standards. The total effectiveness score for a set of user stories was calculated as a success rate through three main steps. First, the total number of criteria met across all evaluated stories was summed. Then, the total possible successes were determined by multiplying the number of evaluated stories ( $N$ ) by the number of quality criteria ( $C$ , which is 7). Finally, the total criteria met was divided by the total possible successes and multiplied by 100 to obtain the success rate. This rate represents the overall adherence of the user stories to the QUS quality criteria, reflecting the effectiveness of the technique.

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

Similarly, success rates were calculated for stories generated using free-form prompt and Meta-Few-Shot Prompt, facilitating a direct comparison of the techniques. This approach enabled the representation of data as success percentages, providing a quantitative and objective basis for comparing the quality of user stories produced by the different techniques.

In this research, the focus was on evaluating the quality of the generated stories based on the QUS criteria, without the additional formalization of acceptance criteria or the definition of done, which allowed us to focus on analyzing the user stories in terms of their structural and qualitative characteristics.

The three methodological stages are outlined below, consisting of two comparative studies and the development of a new approach, as shown in Figure 1.

The research followed a systematic approach, organized in three main phases:

1. **Empirical Study 1 (Section 4):** This study focuses on comparing the quality of user stories generated manually versus those generated automatically using ChatGPT to determine which approach is more effectiveness. Two sets of user stories were created and analyzed, then used as benchmarks for further comparison.
2. **Meta-Few-Shot Prompt Design (Section 5):** This phase involved analyzing patterns and defi-

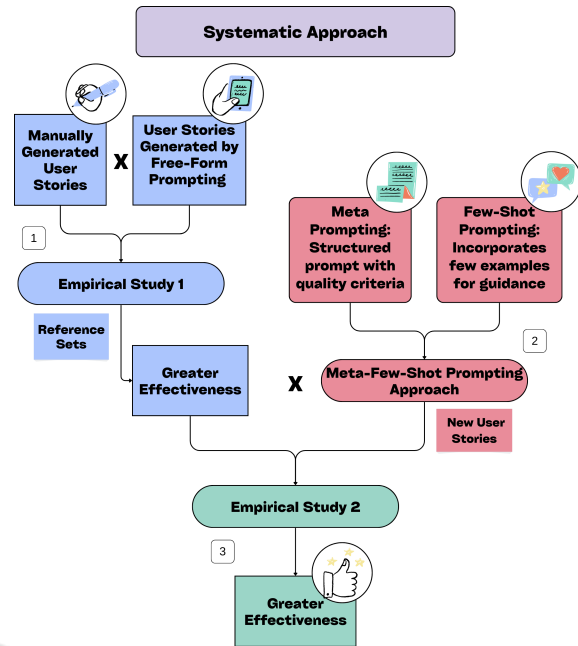


Figure 1: Diagram of the study's methodological steps.

ciencies in the user stories generated during Empirical Study 1, which led to the development of a structured prompt designed to guide ChatGPT in enhancing the precision and quality of user stories. The researcher applied the Meta-Few-Shot Prompt method to ChatGPT to generate a new set of user stories. This method combines:

- **Meta Prompting:** Utilizing structured prompts with specific quality criteria.
  - **Few-Shot Prompt:** Providing examples to guide the model's responses.
3. **Empirical Study 2 (Section 6):** In this study, a new set of user stories generated by ChatGPT using the newly developed Meta-Few-Shot Prompt method was evaluated and compared to the reference sets from Empirical Study 1. This comparison allowed for an assessment of the quality and effectiveness of the Meta-Few-Shot Prompt.

## 4 EMPIRICAL STUDY 1

This study was designed to simulate real-world scenarios faced by novice software engineers, ensuring consistent data collection conditions across all participant groups.

### 4.1 Study Design

The objective of this study is to compare the quality of user stories generated manually versus those gen-



Table 1: Description of the criteria derived from the Quality User Story (QUS) framework.

Quality	Criteria	Description
Syntactic	Well-formed	If the user story includes at least one role (persona), defining who is requesting the functionality, and one means (action), which is the specific activity or functionality that the system should allow.
	Atomicity	If the story addresses a single specific functionality.
	Minimality	If the story contains only one role, one means (e.g., “I want to open the map”), and one or more ends, which are the objectives the user aims to achieve using the functionality (e.g., “to easily locate tourist destinations”), without unnecessary information.
Semantic	Conceptually Sound	If the means express a functionality, and the ends explain the rationale behind that functionality.
	Unambiguous	If it avoids terms or abstractions that could lead to multiple interpretations.
Pragmatic	Full sentence	If the story is complete, well-formed, and provides sufficient context to be clearly understood.
	Estimability	If the story does not denote a coarse-grained requirement that is difficult to plan and prioritize.

erated using ChatGPT, focusing on the QUS framework. Specifically, we evaluate each method’s effectiveness, or success rate, in producing user stories that adhere to the established QUS quality standards.

This research seeks to determine if language model usage enhances user story generation within requirements engineering. The study addresses the following research question: What is the difference in quality between user stories generated manually and those created using free-form prompts in ChatGPT?

**Hypothesis:** The study was designed with the following null and alternative hypotheses.

- **Null Hypothesis (H01):** There is no significant difference in the quality of user stories generated manually compared to those produced using free-form prompt in ChatGPT.
- **Alternative Hypothesis (HA1):** There is a significant difference in the quality of user stories generated with free-form prompt in ChatGPT.

**Variables:** The independent variable is the method of user story generation, which includes two treatments: manual generation (treatment 1) and automated generation with ChatGPT (treatment 2). The dependent variable is the quality of the user stories, assessed based on adherence to the QUS framework.

**Selection of Subjects:** Participants were recruited through convenience sampling from the Requirements Engineering and Systems Analysis (ERAS) course within the Software Engineering program at the Universidade Federal do Amazonas (UFAM). Before participating, they received training on the QUS guidelines for creating effective user stories.

**Experimental Design:** All participants signed an informed consent form, which guaranteed their right to withdraw from the study at any time without penalty. Participants were divided into groups to develop user stories for a Travel Agency System using

two approaches: manual generation and automated generation with free-form prompts in ChatGPT. We anonymized the collected data to ensure its confidentiality. To ensure everyone had the same product vision, a scenario was created about a travel agency offering tickets, reservations, and tours, aiming to implement a management system to optimize operations and customer service. This scenario served as the foundation for the user stories and was essential in aligning all participants’ understanding of the system’s goals and functionalities.

## 4.2 Study Execution

The study execution was divided into two stages:

- **Stage 1 - Manual Generation:** The groups generated user stories manually from May 16 to May 21, applying the criteria learned during the classes. Each group submitted their stories through a provided digital platform;
- **Stage 2 - Automated Generation with Free-Form Prompt:** The groups were asked to use ChatGPT (version 3.5 or higher) to generate new user stories from May 27 to May 29. Participants provided basic and generic prompts to ChatGPT without examples or specific guidelines for a rigid and detailed structure, such as: “Generate a user story.” The intention was to simulate the freedom of a developer to create user stories autonomously.

Initially, twelve groups of participants were formed, but only seven, comprising 30 participants, completed both stages of the study.

The collected data included 14 sets of user stories from the seven groups. Each group provided one set of stories generated manually and another set generated with free-form prompt, totaling 126 stories (62 manual and 64 automated). Despite variations in story

length, all stories maintained similar levels of complexity, as all groups based their work on the same statement provided during the study.

### 4.3 Results and Discussion

The data were compiled from the evaluations of stories generated manually and automatically by participant groups. The analysis included calculating the success rate for each quality criterion met. The stories from each group, manually generated and created using free-form prompt in ChatGPT, were assessed based on the quality criteria.

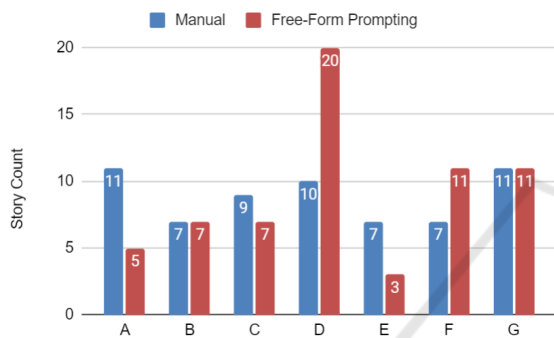


Figure 2: Comparison between the number of user stories generated by manual methods and Free-Form Prompt.

Figure 2 illustrates the number of user stories generated by the seven groups participating in the study, comparing the performance of the manual method with the use of ChatGPT utilizing free-form prompt. Among the analyzed groups, four (B, D, F, and G) generated a number of stories greater than or equal to those produced manually when employing the automated tool. These results suggest that ChatGPT increased the number of user stories generated. Although some groups faced challenges, the majority achieved a higher production with automation.

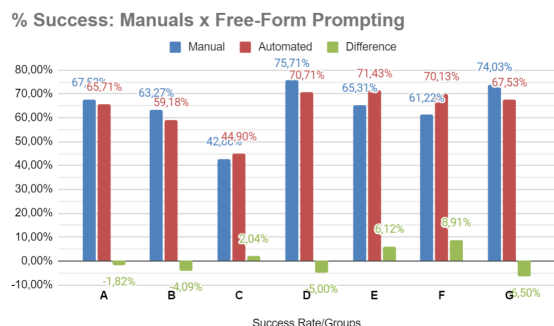


Figure 3: Comparison of success rates between Manual and Free-Form Prompt methods.

Subsequently, the success percentage was calcu-

lated for each group. Figure 3 presents the differences in the percentages of correct responses for each quality criterion of the generated stories, comparing the manual and automated (ChatGPT) methods using free-form prompt. The bars represent the percentage of correct responses for each group and method, with performance variations highlighted in green.

Overall, there is an observed improvement in performance in three out of the seven groups when using automated generation, with groups E and F achieving improvements of over 5%, with rates of 6.12% and 8.91%, respectively. Positive values indicate that ChatGPT outperformed the manual method in generating user stories, suggesting greater effectiveness in applying quality criteria such as soundness and atomicity. Conversely, negative values, such as those observed in group A, indicate that the manual method was more effective in certain cases, which may be related to the participants' experience or the quality of the prompts provided.

Negative variations in groups such as D and G, with values of 5.00% and 6.50%, respectively, may suggest that, although ChatGPT may have increased the number of user stories generated for some groups, the automated generation did not meet quality expectations. This variation could be attributed to specific factors influencing its performance, such as the complexity of the generated stories or differences in how groups interacted with the tool, ultimately affecting the final outcomes. In particular, the absence of clear and specific instructions in the prompts may have compromised the quality of the responses. Some prompts provided only the statement without additional guidelines, which may have led to a less effective generation of user stories.

Quantitative data was analyzed to assess the effectiveness of user story generation methods, using statistical analysis conducted in JASP software (version 0.19.1). The average effectiveness of manually generated user stories was calculated at 64.28% (rounded) and compared to the effectiveness of stories generated automatically using free-form prompt, which averaged 64.23% (rounded).

Although the mean of the manually generated stories is slightly higher, the statistical test did not find sufficient evidence to assert that the quality of stories generated by manual methods is significantly different from those generated automatically. Thus, it can be concluded that both methods produce comparable results in terms of user story quality.

The Shapiro-Wilk test was used to assess data normality, a prerequisite for applying parametric tests, such as the t-test, to ensure statistically valid results.

In this study, the p-value obtained for the Shapiro

test was ( $p = 0.463$ ). Since the p-value is greater than the established significance level of ( $\alpha = 0.05$ ), it can be concluded that the analyzed data exhibit a normal distribution. This justifies the use of Student's t-test.

The Student's t-test obtained p-value of ( $p = 0.983$ ) suggests no statistically significant difference between the two approaches. The null hypothesis ( $H_0$ ) was not rejected, indicating that both methods generate user stories of similar quality.

Additionally, the standard deviations were 10.9% for manual stories and 9.5% for automated stories. The lower standard deviation in the automated method suggests greater homogeneity in the responses generated with free-form prompt. This homogeneity means that, when using automated techniques, the stories tend to be more consistent in terms of quality and format, which can be advantageous in contexts where uniformity is desired.

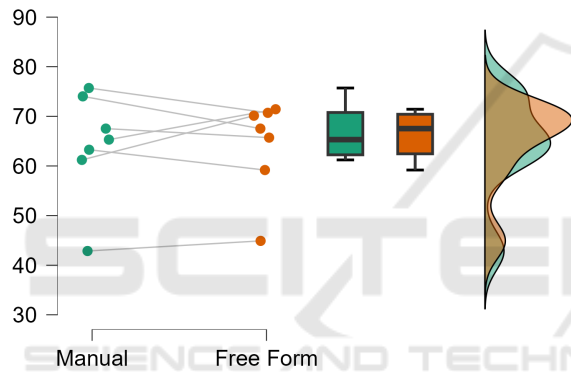


Figure 4: Graph evaluating indicators with visual representations: density plots, boxplots, and individual data points.

The graph in Figure 4 helps identify potential differences between the means and variability of the groups. The density curves illustrate the data distribution, highlighting a concentration of values near 70%, with overlapping distributions suggesting some similarity between the methods. However, differences in density peaks suggest variations in score distribution, supporting an approximate normal distribution.

The boxplots summarize the central distribution, showing similar medians and interquartile ranges for both methods. However, subtle differences are observed in the extreme scores and overall variation. According to the graph, free-form prompt has a median of 67.53%, compared to 65.31% for manual generation, indicating a slight effectiveness advantage for the free-form prompt method.

The individual data points, or 'jitter', show each data pair distinctly, providing a clear view of overall dispersion and facilitating the identification of descending patterns in the data.

In summary, while the overall effectiveness shows

no significant improvement over the manual method, the greater homogeneity of the automatically generated stories suggests that automated techniques may help increase the number of user stories generated and contribute to a more uniform and standardized generation. This result is consistent with findings from Krishna et al. (2024), which identified a marginal difference in quality between automatically and manually generated documents.

## 5 META-FEW-SHOT PROMPT APPROACH

The results from the empirical study in Section 4 showed that there was no statistically significant difference between user stories created manually and those generated automatically with free-form prompts by the participants. This highlighted the need to improve the approach for designing prompts to produce higher-quality user stories, leading to the development of the Meta-Few-Shot Prompt approach.

This approach combines Meta Prompting and Few-Shot Prompt to create a specialized prompt engineering technique aimed at enhancing the precision and quality of user stories generated by large language models. By leveraging Meta Prompting's structured guidance alongside the adaptive strengths of Few-Shot, Meta-Few-Shot Prompt enables the model to follow specific criteria through prompts that incorporate both explicit instructions and examples.

Using Meta Prompting, the model receives prompts containing quality criteria from the QUS framework by Lucassen et al. (2016), with instructions designed to meet standards. Simultaneously, Few-Shot Prompt provides a set of well-defined user story examples, both correct and incorrect, which enables the model to identify and replicate the desired patterns with greater consistency.

By integrating Meta-Few-Shot Prompt, the model effectively generalizes from a limited set of examples, adapting its output across diverse contexts while consistently producing user stories that meet established quality standards.

### 5.1 Approach Design

Based on the results of the empirical study in Section 4, the following steps were taken to design the Meta-Few-Shot Prompt approach for generating user stories. These steps aimed to create the prompt to improve the quality of the generated stories.

Prompt development followed these steps:

- **Analysis of Previous Stories:** The manually and automatically generated stories were evaluated according to the criteria of the QUS framework. The analysis revealed gaps, such as a lack of atomicity and minimality, indicating the need for adjustments in the automated generation process.
- **Creation of a Refined Prompt:** Based on the identified gaps, a detailed prompt was developed that provided ChatGPT with templates for the stories and specific guidance on the quality criteria. The Few-Shot approach included explicit examples of stories that met the quality criteria, while Meta Prompting offered a broader context regarding what was requested.
- **Testing and Refinements:** After creating the initial prompt, it was iteratively tested to generate new user stories. With each iteration, the prompt was adjusted to enhance the clarity of the instructions and improve the results. Additionally, both the evaluation criteria and the guidelines provided in the prompt were refined to ensure they were clear and precise, guiding the generation and review of these stories while ensuring they adhered to the established quality criteria. Examples of well-formed stories were included to facilitate understanding of what was expected.
- **Application of the Meta-Few-Shot Prompt Approach:** After iterations and refinements, the final prompt was executed to generate 15 new user stories with ChatGPT, which were then evaluated using the same criteria as the previous stages, ensuring consistency in quality assessment.

## 5.2 Approach Structure

The prompt was implemented in stages to enhance the creation of user stories. Summary of the approach:

1. **Orientation and Steps:** The document begins by instructing the reader to review all instructions.
2. **Activities:** The document breaks down the process into specific activities:
  - **Introduction of the Process:** The initial activity asks the user to communicate the software discovery context to ChatGPT, expecting a simple acknowledgment of “*Understood*”.
  - **Product Presentation:** This activity provides a detailed product vision for a software system, for example, a tourism agency’s management system, which ChatGPT must acknowledge.
  - **Requirements Presentation:** This activity presents the system requirements, including different user roles (as Travel Agents and

Clients), their actions, and functional requirements such as client registration, reservation management, and client-agent communication.

- **User Story Template:** This activity defines a template and specific criteria for user stories. Each story should include a persona, action, and goal and follow seven criteria: well-formed, atomic, minimal, conceptually solid, unambiguous, complete, and estimable.
- **User Story Generation:** It instructs ChatGPT to act as a Requirements Engineer, using provided information to generate user stories for the software based on the template and criteria, rather than responding with “*Understood*”.

In summary, this approach provides a procedural guide for systematically generating precise and structured user stories, adhering to predefined criteria, within the context of software product discovery.

The full approach are available in the Supplemental Material <sup>2</sup>.

## 6 EMPIRICAL STUDY 2

Building on the findings from the Empirical Study in Section 4, which revealed quality gaps in user stories generated through automated methods, this study introduces the Meta-Few-Shot Prompt approach. By combining the strengths of Meta Prompting and Few-Shot Prompt, as explored by Hiraou (2024), this technique directs ChatGPT to produce more accurate and higher-quality user stories.

### 6.1 Study Design

This study aimed to evaluate the effectiveness of the Meta-Few-Shot Prompt method in generating user stories, focusing on improvements in key aspects such as non-ambiguity, atomicity, and adherence to predefined quality standards. Unlike the previous study, which involved participant groups, this study was conducted solely by the researcher, who applied the Meta-Few-Shot Prompt method to generate new user stories and assess the results.

**Hypothesis:** To answer this question, the study was designed with the following hypotheses:

- **Null Hypothesis (H02):** There is no significant difference in the quality of user stories generated using free-form prompt compared to those produced using meta-few-shot prompt.

<sup>2</sup><https://figshare.com/s/eaf2e688f4e65afaf6a8>



- **Alternative Hypothesis (HA2):** There is a significant difference in the quality of user stories generated with meta-few-shot prompt.

**Variables:** The independent variable in this study is the method of user story generation, which includes two treatments: free-form prompt (treatment 1) and meta-few-shot prompt (treatment 2). The dependent variable is the quality of the user stories, assessed based on adherence to the QUS framework.

**Experimental Design:** Initially, two reference sets of user stories were established, consisting of stories generated manually and through free-form prompt (Empirical Study 1). These sets served as comparison points for evaluating the effectiveness of the Meta-Few-Shot Prompt method. The researcher then applied the Meta-Few-Shot Prompt method to ChatGPT to generate a new set of user stories. This method combined two main techniques: Meta Prompting (involving structured prompts with specific quality criteria) and Few-Shot Prompt (incorporating examples to guide the model's responses). The newly generated stories were then analyzed to evaluate their quality in comparison to the reference sets.

The quality of the generated stories was assessed using the same criteria established in the first study, ensuring a consistent evaluation for comparison.

A comparative analysis was conducted between the stories generated by the Meta-Few-Shot Prompt method and those from the manually and free-form prompt reference sets. A qualitative analysis examined improvements in story quality based on pre-defined criteria, and a statistical analysis assessed whether the methods produced significant differences in story quality.

## 6.2 Results and Discussion

The final prompt, utilizing the Meta-Few-Shot method, was applied to guide the model in generating user stories. This approach combines the structuring of Meta Prompting with the flexibility of Few-Shot Prompt, allowing the model to receive specific guidelines and examples of previous stories. The result was the generation of 15 stories, which were crafted to meet the established quality criteria.

The application of Meta-Few-Shot Prompt provided a rich and detailed context, helping the model capture important nuances of user requirements. Each generated story was designed to reflect the desired functionality in a simple, cohesive, and minimal manner. This technique aimed to enhance the relevance and quality of the stories, facilitating a more efficient and effective production process, as it not only ensures the creation of relevant and high-quality sto-

ries, but also facilitates the creation of these stories quickly, using only available resources, contributing to a more optimized and streamlined process in the agile development environment.

The resulting stories were then evaluated for compliance with the same quality criteria. The performance of these stories compared to those generated by free-form prompt is presented in Figure 5, highlighting the effectiveness of Meta-Few-Shot Prompt in generating high-quality stories.

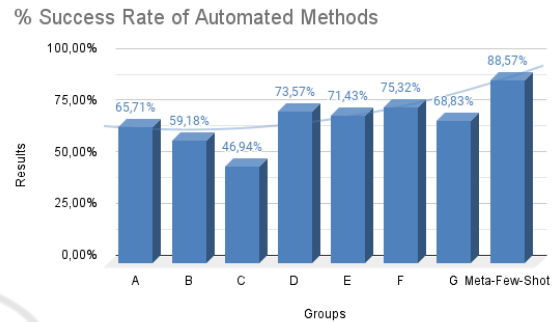


Figure 5: Success rate of Free-Form Prompt and Meta-Few-Shot Prompt methods.

The automated methods used in generating user stories showed efficiencies ranging from 46.94% to 75.32%. This indicates that, while these methods yielded varied results, they did not achieve consistently high efficiency.

In contrast, the Meta-Few-Shot Prompt method stood out by achieving an efficiency of 88.57%, calculated based on the number of correct responses relative to the total number of evaluated criteria. This metric provides a clear insight into the effectiveness in generating high-quality user stories. The significant difference in efficiency suggests that Meta-Few-Shot Prompt is a more effective approach for story generation compared to the simplified automated methods.

When comparing the average effectiveness of automated methods (65.85%) with Meta-Few-Shot Prompt, the latter was 22.72% more effective.

A statistical analysis was also conducted using the one-sample t-test. It was observed that when comparing manual generation with Meta-Few-Shot Prompt, the  $p$ -value was 0.001. Additionally, when comparing automated generation using the Free-Form Prompt method to Meta-Few-Shot Prompt, the  $p$ -value was  $< 0.001$ . These results provide sufficient statistical evidence to reject the null hypothesis (H02) and conclude that the success rate achieved by Meta-Few-Shot Prompt is significantly different from both the success rate of manual generation and that of automated generation using Free-Form Prompt. There-

fore, the alternative hypothesis (HA2) is valid.

This substantial difference highlights the potential of Meta-Few-Shot Prompt as a superior technique capable of generating user stories with higher quality and effectiveness.

These results indicate that by applying the combination of Meta Prompting and Few-Shot Prompt, the model not only improves the quality of the generated stories but also demonstrates a significant increase in efficiency. These findings corroborate the results of Hiraou's (2024) study, which emphasizes how the application of advanced techniques such as Meta-Few-Shot Prompt can lead to significant improvements in the quality and effectiveness of responses generated by language models, highlighting its potential in practical contexts, such as requirements engineering and agile settings, where speed and quality are essential.

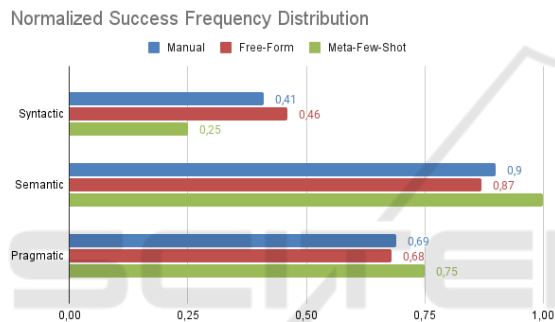


Figure 6: Methods comparison by quality categories.

In Figure 6, the normalized frequency of correct responses reveals the proportion of stories that met the quality criteria across three categories from Quality User Story (QUS) framework — syntactic, semantic, and pragmatic — for the manual, automated with Free-Form and Meta-Few-Shot Prompt methods.

Automatically generated stories showed slightly improved syntactic correctness, 0.46, compared to the manual stories, 0.41. However, small reductions were observed in the semantic 0.87 versus 0.90 and pragmatic 0.68 versus 0.69 categories, suggesting that semantic and pragmatic errors remain significant challenges despite an improvement in syntax.

Manual stories and generated with free-form prompt showed success rates of 0.41 and 0.46 for the syntactic category. While the Meta-Few-Shot method demonstrated a lower performance of 0.25 in this category, it compensated for this deficit with notable improvements in the semantic and pragmatic categories.

In the semantic category, there was a remarkable improvement, with the Meta-Few-Shot achieving a correctness rate of 1, in contrast to 0.90 and 0.87 for the manual and free-form methods, respectively.

In the pragmatic category, the Meta-Few-Shot also showed an increase, reaching a success rate of 0.75, compared to 0.69 and 0.68 for the manual and free-form prompt methods.

Thus, the Meta-Few-Shot Prompt method showed a substantial improvement in the success rate compared to the manual and Free-Form Prompt methods. Specifically, the Meta-Few-Shot achieved a correctness rate of 1 in the semantic category, surpassing the manual methods at 0.90 and Free-Form at 0.87. In the pragmatic category, the Meta-Few-Shot also showed an increase, with a success rate of 0.75, compared to 0.69 (manual) and 0.68 (free-form).

Sixty percent of the stories generated using Meta-Few-Shot Prompt fully met the established quality criteria, demonstrating improvements in semantic and pragmatic aspects. However, the most common syntactic error was the difficulty in ensuring atomicity, which is essential for clear and simple story formulation. Non-atomic stories compromised minimality, making them more complex and difficult to estimate and prioritize. This issue resulted in a 40% error rate across the generated stories. Half of these errors, specifically in three stories, were due to challenges in maintaining atomicity, directly impacting both syntactic clarity and the feasibility of implementing and prioritizing these stories within agile planning.

The results indicate that user stories generated by ChatGPT, particularly using the Meta-Few-Shot Prompt approach, are of higher quality than those written manually or generated through free-form prompt. This aligns with findings by Brockenbrough and Salinas (2024), who observed that user stories created with AI assistance surpassed those developed without such a tool, while extending their scope by incorporating structured prompts with clear examples.

Meta-Few-Shot Prompt effectively addresses aspects such as minimality, and semantic clarity, surpassing free-form prompt in both semantic and pragmatic quality. This underscores its ability to overcome the limitations of earlier methods and enhance the overall quality of generated user stories.

## 7 THREAT TO VALIDITY

All studies face threats that may compromise the validity of the results (Wohlin et al., 2012), which were categorized into internal, external, conclusion, and construct threats, with the corresponding mitigation strategies outlined below:

**Internal Validity:** The variability in participants' skills for manually and automatically generating user stories poses a threat. Although all participants re-

ceived prior training focused on creating user stories, the lack of a rigid prompt structure for some may have led to greater variability in the results, especially for those less familiar with generating stories automatically. One of the main benefits of using Meta-Few-Shot Prompt in the study is its ability to reduce this reliance on experience by providing a structured prompt that any participant can follow.

**External Validity:** The research sample was limited to students from the Software Engineering course, potentially restricting the generalization of competencies to roles beyond this academic setting. However, studies such as those by Höst et al. (2000) (Höst et al., 2000) and Salman et al. (2015) (Salman et al., 2015) have shown that students can adequately represent industry professionals, making this sample suitable for evaluating the framework.

**Construct Validity:** The metrics used to evaluate the quality of user stories, based on the QUS framework, provide a robust set of criteria to assess different dimensions of quality. While these metrics are reliable and ensure the overall quality of the generated stories, they may not capture all nuances in more complex contexts. However, the use of QUS significantly minimizes this risk.

**Conclusion Validity:** These include the small sample size, which limits generalizability, the variability in participants' skills, the risk of statistical errors due to the limited number of observations, and the low statistical power, as evidenced by the similarity in the results. Furthermore, bias in the analysis, due to the subjective evaluation of the researcher, can compromise the objectivity of the conclusions. To minimize the influence of the researcher, the author was responsible for identifying and analyzing all study results. Subsequently, all data and findings were reviewed and validated by two other researchers: the advisor, an expert in prompting techniques, and the co-advisor, an expert in RE, ensuring the validity and reliability of the conclusions.

**Threats from Hallucination:** A concern when using LLMs is hallucination, which refers to the generation of inaccurate or fabricated information that appears plausible but lacks a foundation. This occurs because models like ChatGPT are trained on vast amounts of online data, allowing them to extrapolate information, interpret ambiguous prompts, or modify data without solid grounding (Huang et al., 2024). As a result, the generated user stories may contain erroneous details that do not reflect reality. Therefore, it is crucial to include a final review stage by a requirements expert to verify whether the stories align with the project needs, ensuring their accuracy and applicability.

## 8 CONCLUSIONS

This research investigated whether user stories generated by ChatGPT using the Meta-Few-Shot Prompt exhibit superior quality compared to those written manually or generated through free-form prompt. Results showed that Meta-Few-Shot Prompt notably improved quality, particularly in semantic accuracy, eliminating errors in interpreting and translating requirements. 60% of the generated stories met all quality criteria, surpassing manual and free-form methods. However, challenges persisted in the syntactic category, mainly due to difficulties in ensuring atomicity, which impacted complexity and prioritization.

Meta-Few-Shot Prompt proved significantly more effective than previous methods, especially for semantic and pragmatic quality. This aligns with findings by Brockenbrough and Salinas (2024) that ChatGPT-driven user stories are of higher quality than those crafted without AI support. Future research should enhance syntactic quality and explore applications beyond requirements engineering, like automated test scenario generation, technical documentation, and refining non-functional requirements.

## ACKNOWLEDGEMENTS

We thank all participants in the empirical study and USES Research Group members for their support. This work results from the R&D project 001/2020, signed with the Federal University of Amazonas and FAEPI, Brazil, funded by Samsung and using resources from the Informatics Law for the Western Amazon (Federal Law n° 8.387/1991). Its disclosure complies with article 39 of Decree No. 10.521/2020. Supported by CAPES (Financing Code 001), CNPq (314797/2023-8, 443934/2023-1, 445029/2024-2), and Amazonas State Research Support Foundation (FAPEAM) through POSGRAD 24-25.

## REFERENCES

- Alhazmi, A. and Huang, S. (2020). Survey on differences of requirements engineering for traditional and agile development processes. In *2020 SoutheastCon*, pages 1–9. IEEE.
- Belzner, L., Gabor, T., and Wirsing, M. (2023). Large language model assisted software engineering: prospects, challenges, and a case study. In *International Conference on Bridging the Gap between AI and Reality*, pages 355–374, Cham. Springer Nature Switzerland.

- Brockenbrough, A. and Salinas, D. (2024). Using generative ai to create user stories in the software engineering classroom. In *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)*, pages 1–5.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Buglione, L. and Abran, A. (2013). Improving the user story agile technique using the invest criteria. In *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement*, pages 49–53.
- Ferreira Martins, H., Carvalho de Oliveira Junior, A., Canedo, E. D., Kosloski, R. A. D., Paldês, R. Á., and Oliveira, E. C. (2019). Design thinking: Desafios para elicitação de requisitos de software. *Informação*, 10(12):371.
- Hiraou, S. R. (2024). Optimising hard prompts with few-shot meta-prompting. arXiv:2407.18920. Retrieved from <https://arxiv.org/abs/2407.18920>.
- Höst, M., Regnell, B., and Wohlin, C. (2000). Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. *Empirical Software Engineering*, 5:201–214.
- Hoy, Z. and Xu, M. (2023). Agile software requirements engineering challenges-solutions—a conceptual framework from systematic literature review. *Information*, 14(6):322.
- Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., and Liu, T. (2024). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* Just Accepted.
- Jaramillo, C. M. Z. (2010). Computational linguistics for helping requirements elicitation: a dream about automated software development. In *Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas*, pages 117–124.
- Krishna, M., Gaur, B., Verma, A., and Jalote, P. (2024). Using LLMs in Software Requirements Specifications: An Empirical Evaluation. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 475–483, Los Alamitos, CA, USA. IEEE Computer Society.
- Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M., and Brinkkemper, S. (2016a). The use and effectiveness of user stories in practice. In Daneva, M. and Pastor, O., editors, *Requirements Engineering: Foundation for Software Quality*, volume 9619 of *Lecture Notes in Computer Science*, pages 187–202. Springer, Cham.
- Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M., and et al. (2016b). Improving agile requirements: The quality user story framework and tool. *Requirements Engineering*, 21(4):383–403.
- Marques, N., Silva, R. R., and Bernardino, J. (2024). Using chatgpt in software requirements engineering: A comprehensive review. *Future Internet*, 16(6):180.
- Oswal, J. U., Kanakia, H. T., and Suktel, D. (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)*, pages 913–920. IEEE.
- Rahman, T., Zhu, Y., Maha, L., Roy, C., Roy, B., and Schneider, K. (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)*, pages 791–801.
- Rasheed, I. (2021). Requirement engineering challenges in agile software development. *Mathematical Problems in Engineering*, 2021:1–18.
- Ronanki, K., Cabrero-Daniel, B., and Berger, C. (2024). Chatgpt as a tool for user story quality evaluation: Trustworthy out of the box? In Kruchten, P. and Gregory, P., editors, *Agile Processes in Software Engineering and Extreme Programming – Workshops*, volume 489 of *Lecture Notes in Business Information Processing*, Cham. Springer.
- Salman, I., Misirli, A. T., and Juristo, N. (2015). Are students representatives of professionals in software engineering experiments? In *2015 IEEE/ACM 37th IEEE international conference on software engineering*, volume 1, pages 666–676. IEEE.
- Sommerville, I. (2011). *Software Engineering*. Pearson, Boston, 9th edition.
- Vogelsang, A. (2024). From specifications to prompts: On the future of generative large language models in requirements engineering. *IEEE Software*, 41(5):9–13.
- White, J., Hays, S., Fu, Q., Spencer-Smith, J., and Schmidt, D. C. (2024). *ChatGPT prompt patterns for improving code quality, refactoring, requirements elicitation, and software design*, pages 71–108. Springer.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A., et al. (2012). *Experimentation in software engineering*, volume 236. Springer.
- Yarlagadda, R. T. (2021). Software engineering automation in it. *International Journal of Innovations in Engineering Research and Technology*. Retrieved from <https://ssrn.com/abstract=3797346>.
- Zhang, Z., Zhang, A., Li, M., and Smola, A. (2023). Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.