CASE STUDY: PROTOTYPE OF QUALITY EVALUATION OF USER STORIES BASED ON REQUIREMENTS SMELLS DETECTED WITH THE AID OF GPT-3

Cláudio Ratke, Fabiano Oss, Giulia Larissa Conradi, Raquel Cristina Isensee da Silva, Robson Flávio Freitas and Kelvin Melim [1]

[1]Faculdade Senac Blumenau, Blumenau, Santa Catarina, Brasil
claudio.ratke@sc.senac.br

ABSTRACT

When developing software, it is essential to understand user needs and their interactions with the system. In this context, the user stories become tools used to transcribe functionalities and requirements of the software; in theory, this documentation should show the client's perspective in a clear and precise way. However, when written badly, they can cause misinterpretation and development issues that impact software quality, cause rework, and, in consequence, financial loss. Regarding this scenario, the proposal is to develop a system to analyse user stories using GPT-3 artificial intelligence, the same as that used by ChatGPT. The proposal presented in this project consists of the creation of a tool to diagnose and suggest corrections to grammatical, orthographic, and pattern errors in user stories written by the users in a web application that uses natural language in Portuguese and machine learning technology. This system is trained to recognize writing patterns and detect possible errors, applying the Requirements Smells concept and suggesting appropriate corrections that will improve the written quality and user experience when using the software. The collaboration with GPT-3 in this process is essential, given that it provides a set of advanced tools that guarantee its capability to work with a large variety of situations and writing contexts. In summary, the system of user story correction using GPT-3 artificial intelligence is an innovative solution capable of enhancing writing quality and improving the user experience.

KEYWORDS

User Stories, GPT-3, Agile Methodology.

1. INTRODUCTION

A user story is a technique for documenting software requirements in a natural language format. According to [1], user stories are an efficient and effective way to capture and document software requirements, providing a clearer understanding of the value that software is supposed to provide to the user.

However, poorly written user stories can lead to features that don't meet users' needs due to developers' misinterpretation of the requirements. As stated by [2], the success of a project depends on how you define the requirements. If you start with poorly defined requirements, the result will be poor.

For this reason, the concept of Requirements Smells [3] was created, with the aim of identifying common problems in requirements documentation that can negatively affect the development process.

Requirements Smells are indicators of quality problems in the elaboration of software requirements, as stated by [4], and their use can help in the quality assurance of tests and in good communication through the definition of writing rules for requirements. In other words, a requirements smell indicates a deficiency in the writing of the requirement, pointing out quality flaws, such as the presence of adjectives or adverbs that may cause ambiguity in understanding, as highlighted by [5]. A study conducted by researchers at Sannio University [6] evaluated the impact of requirements smells on software development by measuring time and quality through metrics such as defect rate and code complexity. The table below presents the results obtained after the end of the study, Table 1:

Table 1. Time and Quality of Software Based on Requirements Smells.

Time and Quality of Software Based on Requirements Smells		
Task Type a	**With Requirements Smells**	**Without Requirements Smells**
Average time taken (minutes) to complete the task	23,7 minutes	18,3 minutes
Average defect rate (quantity)	2,05	1,25
Average code complexity	4,21	3,63

2. REQUIREMENTS SMELLS

According to [7], user stories are one of the most used artifacts in requirements writing in agile scenarios to define requirements and specifications in a simple and clear way, from the perspective of the end user and using simple language. With one or more user stories for each item in the product backlog and a way to script the product deliverables, a project development journey is obtained.

To leverage the quality of user stories, forms of standardization and continuous improvement of the writing used in this tool are used. For example, the use of Smells Requirements as a qualifier for the writing of user history documents or the use of the ISO/IEC/IEEE 29148:2011 standard (Federal University of Mato Grosso do Sul. 2011). This standard aims to specify necessary processes that must be implemented in requirements engineering in software systems and products.

In addition, according to [8] the use of a natural language for specifications can result in requirement smells. This verification approach is based on a catalogue of poor-quality indicators, which can compromise the quality of requirements and trigger problems in other activities that depend on these specifications, such as implementation, testing, and planning of systems, as represented in Table 2 (Catalogue of Requirements Smells).

The quality of user stories is a crucial element for effective communication between the development team and users, as pointed out by [9]. However, several requirements smell of requirements can compromise this quality, such as incompleteness, ambiguity, lack of clarity, redundancy, and contradiction, as highlighted by [10]. The presence of these issues can hinder developers' proper understanding of the requirements, resulting in software failures and delivery delays.

Table 2. Catalogue of Requirements Smells.

Anomaly	Definition
Adjectives and Adverbs Ambiguous	They are adjectives and adverbs that cause ambiguity in the understanding of the requirements. Example: If the quality (...) is too low, a fault should be written to the error memory.
Vague Pronouns	They are pronouns with unclear relationships. Example: The software must implement services for applications, which must communicate with controller applications deployed on other controllers.
Subjective Language	These are words whose semantics are not objective. Example: user-friendly, easy to use, economical.
Comparative	They are adverbs and adjectives, where the requirements express a relationship of the system with other specific systems. Example: better than, higher quality.
Superlatives	They are adverbs and adjectives, where the requirements express a relationship of the system with all other systems. Example: Better performance, shorter response time
Negative Affirmations	These are words used in functionalities that the system should not provide, as they can lead to a lack of explanation about the system's behavior in such cases. Example: The system should not accept visa credit cards.
Open or Unverifiable Terms	These are words that are difficult to verify because they offer several possibilities for running the system. Example: The system can only be activated if all the necessary sensors (...) work at sufficient measurement accuracy.
Gaps	These are words that make it possible for stakeholders to ignore the specifications. Examples: If possible, as appropriate, as applicable.
Incomplete References	These are references that readers cannot find.
Passive	Characterized by requirements where it is not clear which actor is performing a certain action in the system.
Duplicate Functionality	Characterized by repeating descriptions of the same interactions between systems and actors in various use case specifications.

Therefore, it becomes necessary to correct the Smells Requirements to ensure the quality of user stories and the delivery of value to the user. It is important that user stories are written concisely, using a natural writing pattern that follows the user's perspective and expectations. For this, collaborative work is needed between the development team and users, to ensure mutual understanding of the requirements [11].

Some warning signs of user stories are read by [12] as incompleteness or ambiguity. He also says that its simplicity can cause communication failures, such as misunderstandings. In addition, user stories are perceived as superficial approaches that disregard the complexity of the context and produce shallow documents for complex problems.

Despite the standards used in requirements writing and the best practices to develop them with quality, it is common for agile practitioners to have difficulty applying user stories as a requirement writing tool in agile projects. Below are some common factors that make it difficult to write and interpret user stories, all guided by the Requirements Smells approach presented in [12]:

- The breaches factor is characterized by scenarios that have one or more optional actions in the story.
- The incomplete references factor, when artifacts are cited as complementary content of the story, such as documentation, technical terms, or laws, but the references of these contents are not available or are incomplete.
- The problem-oriented factor refers to the scenario in which a story presents only the problem by specifying it without any solution.
- Comparative, superlative, and subjective language factors present the specifications in comparison or subjective interpretations of the user to their needs.
- Comparative, superlative, and subjective language factors present the specifications in comparison or subjective interpretations of the user to their needs.
- The ambiguity factor can arise in the lack of verification of words that are not specific in nature and tend to cause more than one interpretation and make parameterization difficult.
- The conflict factor occurs when two or more requirements cause inconsistencies with each other.
- The testable factor is detected in the absence or inaccuracy of the parameters specified in the histories that make it impossible to create metrics or parameters for testing.

3. RESEARCH

To compose the work, an interview was conducted with the public about technology and systems development in the region of Blumenau - Santa Catarina - Brazil. The answers were obtained through answers in a form available online. Seventy-one responses were collected.

Based on the interviews conducted with 71 individuals, it can be concluded that problems were identified in the writing of requirements in the form of user stories. In addition, there was an interest in improving the ability to write user stories more efficiently.

Based on a sample of 51 respondents, an analysis of requirements smells was carried out, which are problems or deficiencies found in the requirements of a system or software.

The frequency of each smell was categorized into five levels: "Very Frequent", "Frequent", "Occasionally", "Rarely" and "Never", as can be seen in Table 3.

Requirements Smells	**Frequency**				
	Very frequent	**Frequent**	**Occasionally**	**Rarely**	**Never**
Ambiguity	2%	15.7 %	51%	27.5%	3.9%
Vague pronoun	0%	37.3 %	39.2%	21.6%	2%
Subjective language	21.6%	35.3 %	25.5%	17.6%	0%

Comparative	7.8%	21.6 %	29.4%	29.4%	11.8%
Superlatives	13.7%	29.4 %	17.6%	25.5%	13.7%
Negative Affirmations	17.6%	37.3 %	27.5%	11.8%	5.9%
Open terms	9.8%	25.5 %	31.4%	25.5%	7.8%
Gaps	7.8%	21.6 %	23.5%	29.4%	17.6%
Incomplete references	2%	17.6 %	31.4%	35.3%	13.7%
Passive	3.9%	25.5 %	41.2%	21.6%	7.8%
Duplicate functionality	7.8%	27.5 %	45.1%	13.7%	5.9%

4. PROTOTYPE PROPOSITION

Based on the results obtained through the research carried out and the analysis of the information collected on the use of user stories, in which the participants voluntarily answered the questionnaire, there is evidence of a significant possibility in the adoption of user story analyzer software, based on the concepts of requirements smells.

Flow of the project's system architecture: it is possible to visualize the operational flow of the prototype, which promotes the interconnection between the users and the internet with the application of artificial intelligence, as can be seen in Figure 01.

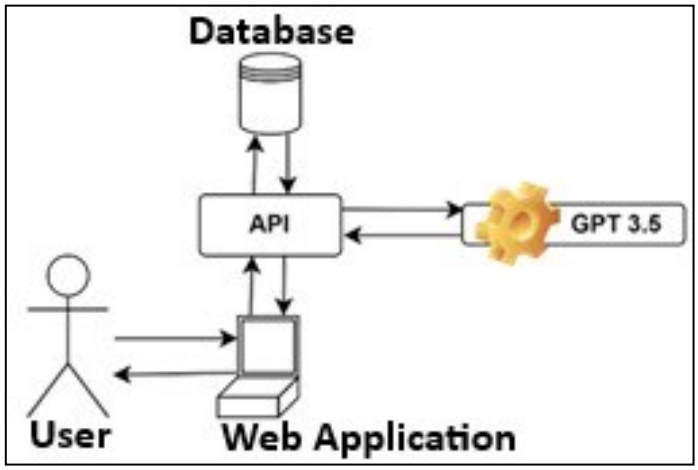

Figure 1 – Project System Architecture

For this flow to be possible, it was necessary to create tables in the MySQL database, following the MER (Entity Relationship Model) represented in Figure 2.

In addition to the database, it was necessary to create the page available on the Web as shown in Figure 3. On this page, the user will include the user stories, and in the fields on the side, the results will be presented, namely: standardization and requirements smell results.

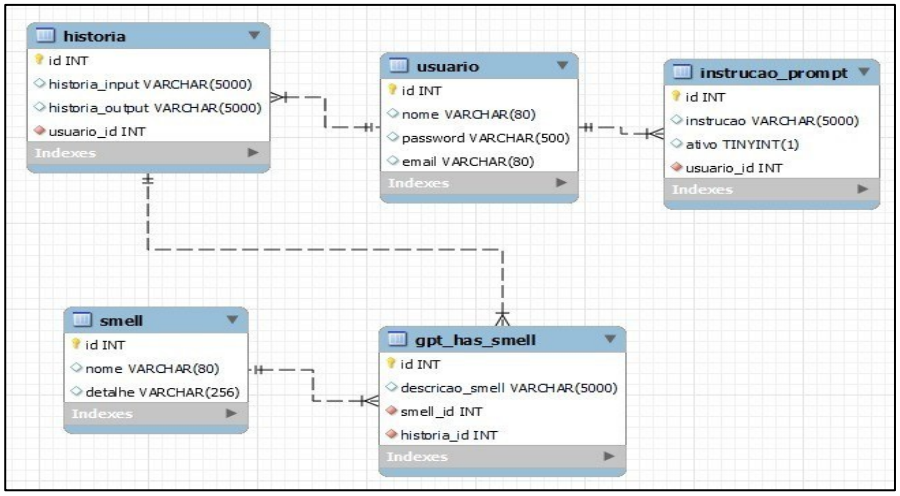

Figure 2 – Entity relationship model

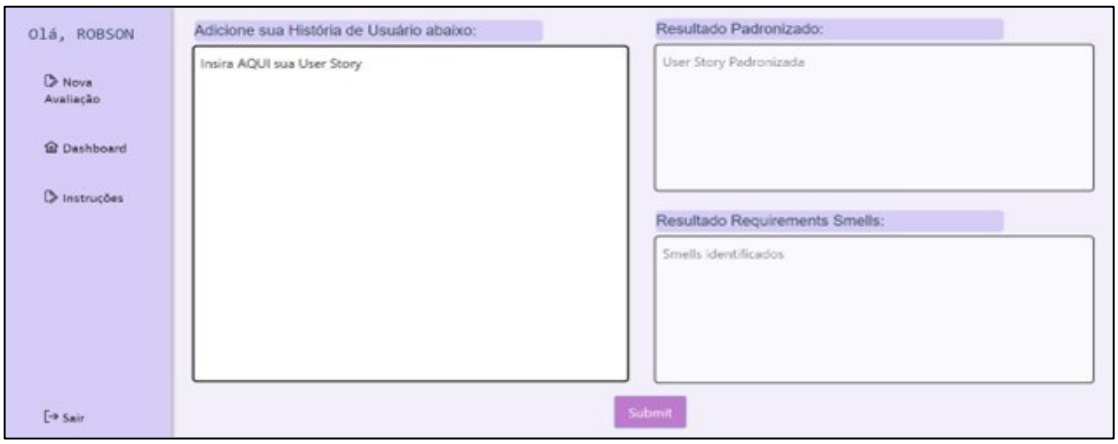

Figure 3 – Homepage of the proposed system (in Portuguese).

For user stories to be corrected correctly, the artificial intelligence of GPT-3.5, a version released by OpenAI in 2022 capable of understanding and generating natural language or code, had to be trained to understand requirement smells. It was necessary to create a prompt that performs the necessary indications and returns the corrections to be made, as shown in Figure 4 (Prompt – GPT - used in the project).

```
PROMPT_DEFAULT='''You are Luna, a Product Owner that domines scrum and agile methodolgys You're here to help with anything you can.
Who you are:
-Voce é uma product owner que corrige historias de usuario
-voce consegue identificar requirements smells em qualquer historia que seja passada pra voce.
-Você não se contenta com apenas um requirements smells, se esforçando para encontrar todos, exceto quando não é possivel localizar mais de
-voce é assertiva nos smells encontrados e sugere as correcoes
Sobre as histórias de usuário que você deve corrigir, considere: Histórias de usuários são requisitos que demonstram como um software deve

PROMPT_FINAL='''Considere o seguinte exemplo:
História de usuário: Dado que um usuário vai criar uma conta para acessar o site, pede-se dados pessoais para fazer o cadastro.
Resposta: [4] Subjetivo,  [2] Referência incompleta . A parte "pede-se dados pessoais para fazer o cadastro" é subjetiva e incompleta, pois
Agora, faça você, Avalie a seguinte História de usuário: '''
```

Figure 4 – Prompt used in the project Homepage of the proposed system (in Portuguese).

3. CONCLUSIONS

During the study, it was found that when writing user stories, hints of problems known as requirements smells may arise. These problems, if left unaddressed, can result in rework and financial losses at the end of projects in software companies. To optimize and minimize these negative impacts, it is proposed to use software capable of diagnosing and correcting requirement smells with the help of GPT-3.5 artificial intelligence.

In this context, the research consisted of creating a prototype that uses artificial intelligence (GPT) to identify possible problems in the writing of user stories, using the Requirements Smells indicators. Based on the information obtained in the theoretical review and analysis of the results, we sought to establish a concrete proposal.

It was sought to evaluate the main flaws in the elaboration of user stories through an in-depth bibliographic investigation. This approach allowed us to identify that this problem is present and recurrent, causing consequences and generating losses. As a result, it is possible to say with conviction that this objective was fully achieved, since a clear and precise definition of the proposed scope was obtained.

The development of a prototype for the evaluation of user stories based on flaws found based on the Requirements Smells, which was selected based on the data collected from the analysis of research results. The prototype proved to be functional and successfully met the proposed need.

The research was conducted based on the assumption that in tech companies, the process of identifying requirements and writing user stories, which are often flawed, could be improved through a user story analysis tool that is based on requirement smells and utilizes AI GPT-3.5. This hypothesis was validated based on the information obtained from the researched authors, as well as the results obtained through the research questionnaire applied to the target audience and the evaluation of the implemented prototype.

It was evaluated that the need presented by the research was successfully exposed, as it was found that most of the interviewees consider the use of a user story analysis system feasible, as shown in Figure 5 (Willingness to use user story software).

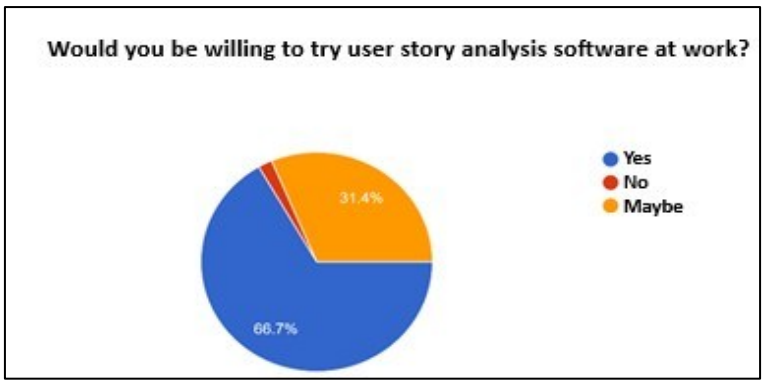

Figure 5 – Willingness to use user story software.

REFERENCES

[1] M. Cohn, User stories applied: For agile software development. Boston, MA: Addison-Wesley Educational, 2004.

[2] J. Martin and Arben Group, Inc., Managing the database environment, 2nd ed. Philadelphia, PA: Prentice Hall, 2015.

[3] X. Franch and C. Farré, “Utilizando Modelos de Qualidade para Detecção Antecipada de Defeitos de Requisitos,” 2023.

[4] H. Femmer, Reviewing Natural Language Requirements with Requirements Smells - A Research Proposal. Munich: Technische Universität München Institute for Software & Systems Engineering., 2013.

[5] R. Nascimento, E. Guimarães, and M. Lucena, “Requirements Smells como Indicador de Qualidade para Histórias de Usuários: Estudo Exploratório,” Workshop em Eng Requisitos, 2021.

[6] DI, P., GRASSI, V., & RISI, M., “Requirements Smells como indicadores de má qualidade na especificação de requisitos: Um Mapeamento Sistemático da Literatura,” Revista Brasileira de Informática na Educação, 2015. [Online]. Available: http://dx.doi.org/10.5753/cbie.wie.2015.23.02. [Accessed: 28-Jul2025].

[7] K. Schwaber and J. Sutherland, “The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game,” 2020-Scrum-Guide-US.pdf, 2020. [Online]. Available: http://2020-Scrum-Guide-US.pdf. [Accessed: 28-Jul-2025].

[8] T. Görschek and K. Wnuk, “Requirements smells in software engineering: A survey,” in Seventh International Conference on the Quality of Information and Communications Technology (QUATIC), Los Alamitos, CA: IEEE Computer Society Press, 2010, pp. 529–534.

[9] I. Sommerville, Software Engineering: (Update), 8th ed. Boston, MA: Addison-Wesley Educational, 2007.

[10] M. Stolberg, J. Bieberstein, and K. Linderskamp, Requirements Smells in Agile Software Development. Cham, Switzerland: Springer International Publishing, 2017.

[11] [14] M. K. Habib, S. Wagner, and D. Graziotin, “Detecting requirements smells with deep learning: Experiences, challenges and future work,” in 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW), 2021.

[12] R. Nascimento, E. Guimarães, and M. Lucena, “Requirements Smells como Indicador de Qualidade para Histórias de Usuários: Estudo Exploratório,” in Anais do Workshop em Engenharia de Requisitos, 2021.