Exploring the Use of ChatGPT for the Generation of User Story Based Test Cases: An Experimental Study

Felipe Sonntag Manzoni[1,2][a], Rávella Rodrigues[1] and Ana Carolina Oran Rocha[1][b]

[1] *Federal University of Amazonas, UFAM - IComp, Manaus, Amazonas, Brazil*

[2] *SiDi Research and Development Institute, Manaus, Amazonas, Brazil*

Keywords: ChatGPT, Software Testing, User Stories, Test Case Generation, Software Engineering, Education, Experimental Study, Software Development.

Abstract: **CONTEXT:** The rapid advancement of Artificial Intelligence (AI) technologies has introduced new tools and methodologies in software engineering, particularly in test case generation. Traditional methods for generating test cases are often time-consuming and rely on manual input, limiting efficiency and coverage. The ChatGPT 3.5 model, developed by OpenAI, represents a novel approach to automating this process, potentially transforming software testing. **OBJECTIVE:** This article aims to explore the application of ChatGPT 3.5 in generating test cases based on user stories from a course in software engineering, evaluating the effectiveness, user acceptance, and challenges associated with its implementation. **METHOD:** The study involved generating test cases using ChatGPT 3.5 and executed by students from the Practice in Software Engineering (PES) course at the Federal University of Amazonas (UFAM) collecting data through surveys and qualitative feedback, focusing on TAM model perceptions and students' self-perceptions. **RESULTS and CONCLUSIONS:** Results indicate a generally positive reception of ChatGPT 3.5 for the objective above, praising it for enhancing several aspects of TC creation, which resulted in high intention of future use and perception of value. However, some challenges have been raised, meaning users should validate and review generated results. Furthermore, results highlight the importance of integrating AI tools while keeping human expertise to maximize their effectiveness.

1 INTRODUCTION

Generating test cases is a critical process in the software development life cycle, ensuring system quality and functionality. Traditionally, this manual process requires in-depth knowledge of system and user requirements (Neto, 2007). With AI advancements, tools like ChatGPT, developed by OpenAI (Brown et al., 2020), have emerged to enhance efficiency and scope in test case generation, significantly supporting the software requirement specification process (Marques et al., 2024).

This study explores ChatGPT 3.5 (Brown et al., 2020) in formulating test cases based on user stories, assessing its influence on productivity, quality, and efficiency while investigating tool acceptance and associated challenges. User stories describe system functionality from the User's perspective and are essential in requirements specification (Cohn, 2004). However, manually generating test cases from them can be challenging. ChatGPT generates contextually relevant responses based on structured prompts, potentially transforming test case development by accelerating the process and enriching test coverage (Shen et al., 2023).

The research was conducted with students from the Software Engineering Practice (PES) discipline at the Federal University of Amazonas (UFAM), who used ChatGPT to generate test cases from pre-defined user stories. The study aimed to evaluate students' acceptance of ChatGPT, identifying its advantages, disadvantages, challenges, and potential applications in software testing. Using the Technology Acceptance Model (TAM) (Davis and Granić, 2024; Marangunić and Granić, 2015), we analyzed users' perceptions of ease of use, usefulness, intention to use, perceived enjoyment, result quality, and demonstrability. This analysis provides insights into integrating ChatGPT into software development and identifying areas for improvement to maximize its impact.

[a] https://orcid.org/0000-0002-2259-6744

[b] https://orcid.org/0000-0002-6446-7510

Manzoni, F. S., Rodrigues, R. and Rocha, A. C. O.
Exploring the Use of ChatGPT for the Generation of User Story Based Test Cases: An Experimental Study.
DOI: 10.5220/0013398700003929

In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 2*, pages 292-299
ISBN: 978-989-758-749-8; ISSN: 2184-4992

2 BACKGROUND

2.1 Software Testing

Software Testing whether software actions and intended features meet expectations through controlled executions. It involves various levels (unit, integration, system, and acceptance testing) and relies on well-defined test cases, test procedures, and test criteria (coverage and adequacy) (Neto, 2007).

These test cases detail the conditions and steps for execution and should be clear, reusable, and based on precise requirements (Neto, 2007). However, traceability, change management, and ambiguities in natural language often arise from their definition process (Vogel-Heuser et al., 2015; Aysolmaz et al., 2018; Ellis, 2008). The software testing phase is considered the most expensive, consuming between 40% and 60% percent of project resources. Automation emerges as a solution to optimize resources, reducing costs and time (Shah et al., 2014; Simos et al., 2019).

2.2 Test Case Coverage

Test case coverage ensures that all software functionalities and behaviors are thoroughly tested to detect potential failures (bin Ali et al., 2019). Key coverage methods include:

- **Code Coverage:** verifies the source code as the tests are executed reducing the risk of undetected defects (Wang et al., 2016).
- **Function Coverage:** ensures all functionalities are tested and properly evaluated (Marijan, 2015).
- **Input and Output Coverage:** evaluates input-output combinations, critical for systems with diverse inputs that generate various results (bin Ali et al., 2019).

Despite its importance, test case coverage can present setbacks with **Cost, Time, Defect Detection and Maintenance** as high coverage can be expensive, doesn't guarantee all defects are found and requires continuous adjustments (bin Ali et al., 2019).

2.3 User Stories

One way to better explain the workflow of a requirement to a software engineer is through user stories. This model should be a short and simplified description that represents an important system functionality from the User's point of view (Cohn, 2004). Thus, user stories can be written according to the following model(Wiegers and Beatty, 2013): *As a <type of user> I want to <goal> so that <motivation>.*

Some attributes must be considered when constructing good user stories (Cohn, 2004): **Independent:** avoid dependencies between stories to prevent planning issues. **Negotiable:** stories are adaptable and can be negotiated with stakeholders. **Valuable:** focus on what is meaningful to the user or project stakeholders. **Estimatable:** structure stories so their implementation time can be measured. **Small:** keep stories concise to aid project planning. **Testable:** ensure stories can be validated through testing.

2.4 ChatGPT

ChatGPT, a Large Language Model (LLM) by OpenAI, excels in natural language understanding and generation, enabling human-like interaction and supporting research in bioinformatics (Sima and de Farias, 2023) and problem-solving in mathematics and logic (Frieder et al., 2024). In software engineering, it aids in code generation, requirements specification, and debugging (Marques et al., 2024), yet limitations persist, affecting its performance in handling complex tasks (Borji, 2023). GPT-3 models leverage few-shot learning and prompt engineering for diverse tasks like text completion and translation, but their effectiveness is highly dependent on prompt quality and dataset size, emphasizing the need for improvement (Brown et al., 2020).

2.5 Related Work

Ronanki and others explored ChatGPT's potential in requirements elicitation for Requirements Engineering (RE) using Natural Language Processing (NLP) (Ronanki et al., 2023). Their two-step methodology—synthetic data collection with ChatGPT and expert interviews—evaluated generated requirements across seven quality attributes. Results showed ChatGPT could produce abstract, consistent, and understandable requirements, though it struggled with detail and specificity. In some metrics, its performance matched or surpassed human-created requirements, highlighting its promise in RE.

Alagarsamy and others introduced a novel approach for generating test cases from textual descriptions using a tuned GPT-3.5 model, specifically aimed at Test-Driven Development (TDD) projects (Alagarsamy et al., 2024). Evaluated on five large open-source projects, it generated 7,000 test cases with 78.5% syntactic correctness, 67.09% requirement alignment, and 61.7% of code coverage, outperforming models like basic GPT-3.5, Bloom, and CodeT5.

No studies have specifically explored using Chat-

GPT to generate test cases from user story prompts. This work addresses that gap, using tailored prompt techniques to create effective test cases, demonstrating ChatGPT's efficiency in interpreting texts and generating test cases aiding the software development and test processes.

3 EMPIRICAL STUDY PLANNING

An experimental study was conducted to generate test cases from user stories created during the Practical Software Engineering (PES) course at the Federal University of Amazonas (UFAM). This 7th-semester course (90 hours) provides students with hands-on experience in all phases of software engineering, including project management, requirements elicitation, design, development, testing, and implementation of the software project. Pairs from each project team used a tailored prompt to generate test cases for their user stories.

3.1 ICF

The *Informed Consent Form* (ICF) invited participants to voluntarily contribute to this study using ChatGPT 3.5 for generating software test cases. Participants consented to data analysis from their system interactions and provided feedback via questionnaires, with privacy and anonymity assured. Withdrawal was allowed without penalty to participants.

3.2 Created Prompt

Participants followed detailed instructions on defining the operating scenario, specifying ChatGPT's role, providing system context, identifying input types, selecting user stories, and requesting test case generation from ChatGPT using a pre-developed prompt for ChatGPT 3.5.

A pilot study was conducted to test the prompt on a software project, with data validation performed by two researchers. The prompt was refined based on the project user stories, adjusted to incorporate new information, and calibrated by a researcher with three years of software testing experience. The final prompt along with the structured and step-by-step detailed instructions are detailed as executed by participants in the activity script (Manzoni et al., 2025).

4 EMPIRICAL STUDY EXECUTION

4.1 Prompt Calibration

The prompt script was designed to generate test cases for system testing, ensuring user story requirements are met, and system components work together. ChatGPT 3.5 was instructed to act as a software tester. It was provided with system context details (design motivation, user profiles, benefits, and innovations) and tasked with creating test cases based on user stories.

4.2 Applying the Test Procedure

Throughout the semester, six system projects were developed in the PES discipline and a summary of the projects go as follows:

1. **CONQUEST -** a collaborative mobile app for gamers, offering tips and strategies to unlock game achievements interactively from the community input;
2. **UFAM EXPLORER -** a mobile app for enhancing communication and participation in university life at UFAM, featuring a community feed for events and opportunities;
3. **MERCURY -** a mobile app focused on Sexually Transmitted Infections (STIs) detected by Rapid Testing prevention and testing, with a dedicated community;
4. ***TRUCO* 2 -** a web-based, mobile-optimized card game allowing 2 to 7 players to enjoy playing a turn-based card game that follows the same logic as traditional *truco* online;
5. **COMUNIPLAZA -** a social network to connect individuals with charitable interests, facilitating participation and promotion of philanthropic projects;
6. ***CONSERTA AQUI* -** a web platform linking construction service providers with clients seeking reliable professionals for home or building repairs.

Participants were invited to use ChatGPT 3.5 following the provided instructions (Manzoni et al., 2025). The study was conducted in a computer lab during PES class, with two hours allotted. This included accepting the ICF, recording results, and completing the TAM questionnaire. The results spreadsheet helped identify the equivalence between test cases created by testers and those generated by ChatGPT, evaluate whether the AI-generated cases provided additional system coverage, and assess their relevance for inclusion in the project's testing scope. The study took participants an average of 1 hour and 20 minutes to complete. The course professor and lead researcher were present to assist with any questions.

5 RESULTS AND DISCUSSIONS

The study involved the participation of nine pairs and one solo participant, two pairs for each project. This accounts for a total of 19 participants who took part in the research. The results were divided between the data obtained from the equivalence, coverage, and relevance spreadsheet of the generated tests and the responses to the TAM questionnaire.

5.1 Equivalence

Participants assessed whether ChatGPT 3.5 generated test cases matched those created by the project tester. Each project provided four user stories, and the tool generated three test cases per story, resulting in 12 cases per pair and 120 cases overall. A professional tester with three years of experience reviewed the equivalence analyses and identified unique test cases generated by ChatGPT.

After review, 61 unique test cases were identified (Figure 1). Overlaps occurred as pairs from the same project used identical user stories, leading to similar or identical test cases.

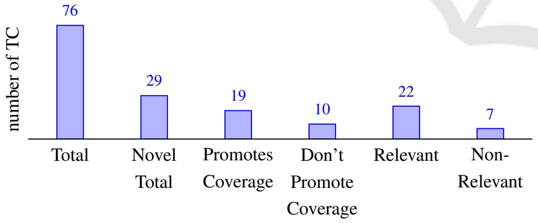

Figure 1: Participant's opinion about Coverage and Relevance of the ChatGPT generated Test Cases.

Of the 61 unique TC, participants determined that 47.5% (29 cases) were novel to the project test plan. This result was likely influenced by the selection of user stories from early sprints, which are easily exhausted on test procedures.

5.2 Coverage and Relevance

Participants assessed whether the generated TC provided enhanced system coverage. Among the 29 unique and novel TC (47.5% of total unique TC), 65.5% (19) were deemed to offer greater coverage of the system features/user stories highlighting the following good and bad contributions to coverage:

- **ConQuest.** A test case for favoriting a researched game was noted for enhancing usability as described by D1.
- **UFAM Explorer.** Test cases focused on confirming user actions to prevent negative experiences and testing publication ordering by upvotes were seen by D3 as vital for consistency.
- ***Truco* 2.** Identified gaps included testing the "password" field, absent in existing cases, and verifying move updates by other players, critical for game functionality as stated by D6. Although they also noted that a TC for a card not being played was unnecessary since such a scenario is impossible.
- **Comuniplaza.** D7 emphasized as crucial for user security when interacting with trusted institutions the CPF validation, however, redundant tests were identified, such as overlapping navigation flows and unrelated functionalities like profile creation and viewing.
- **Conserta Aqui.** Redirection checks were noted as key by D10 to ensuring seamless navigation.

Ultimately, 75.86% (22 of 29) of the unique test cases generated were deemed relevant for inclusion in the system testing scope. These results demonstrate the importance of focusing on critical, high-coverage test cases to ensure system coverage, quality and usability while avoiding redundancies.

5.3 TAM Results

Each study participant answered the Technology Acceptance Model (TAM) individually, resulting in 19 completed TAM forms (P1 to P19).

5.3.1 Perceived Ease of Use

Participants' opinions on perceived ease of use were evaluated through four premises (E1 to E4), with results shown in Figure 2.

Agreement levels on **E1** (68.4% total) indicate that most participants found the interaction with ChatGPT clear and understandable. Results for **E2** (63.2% SA+TA) suggest users perceived the tool as easy to use, with low mental demand, supporting its efficiency and intuitiveness. Similar findings on **E3** (68.5% SA+TA) reinforce this perception of accessibility. Agreement levels on **E4** (68.4%) indicate that users believe ChatGPT enhances the TC generation process, improving coverage and overall satisfaction.

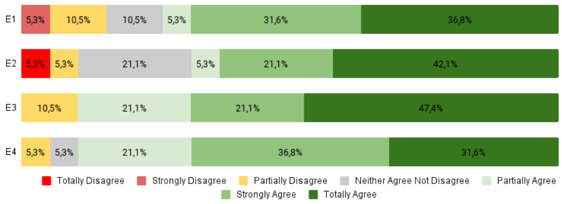

Figure 2: User's opinion on Ease of Use perception.

5.3.2 Perceived Utility

Figure 3 summarizes results for the perceived utility considering assertions U1 to U4.

About 68.4% of participants agreed from **U1** that using ChatGPT improved performance in generating test cases, perceiving efficiency gains in their work when using ChatGPT. Similarly, 68.4% agreed through **U2** that ChatGPT enhances productivity by accelerating test case generation, recognizing its positive impact on productivity. Through **U3** 57.9% of users agreed that users see improvements in quality, speed and comprehensiveness of the test cases using ChatGPT-3.5. Finally, through **U4** 68.4% of participants agreed that ChatGPT is useful for generating test cases, acknowledging its practical benefits in software development. In summary, the findings indicate that ChatGPT is widely perceived as a valuable tool for enhancing performance, productivity, effectiveness, and usefulness in generating test cases transforming traditional testing practices into faster, more efficient processes.

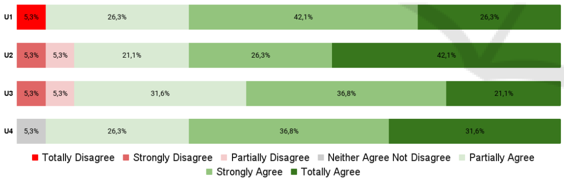

Figure 3: User's opinion on Utility perception.

5.3.3 Intention of Use

Participants' intention to use ChatGPT was assessed through three assertions I1 to I3, with results shown in Figure 4.

In response to whether they would use ChatGPT if given enough time for development on **I1**, 68.4% positively agreed, suggesting a strong willingness among participants to use the tool in their development activities. **I2** asked if they would choose ChatGPT over other tools given their expertise, 68.4% showed confidence in ChatGPT's value in software development. **I3** regards the intention to use ChatGPT in the coming months where 89.5% positive agreed to it. This highlights a strong desire to continue using the tool based on current positive experiences.

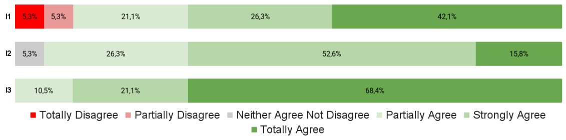

Figure 4: User's opinion on Intention of Use perception.

5.3.4 Perceived Pleasure

Participants' responses regarding perceived pleasure were gathered through three assertions (ENJ1 to ENJ3), with results presented in Figure 5.

ENJ1 assessed whether users find ChatGPT enjoyable, with 73.7% giving positive responses. **ENJ2** see whether the process of using ChatGPT is enjoyable and 68.4% positively agreed. **ENJ3** explored whether users have fun using ChatGPT, with varied responses with 36.9% positive responses and 15.8% negative responses indicating that not everyone finds the tool fun.

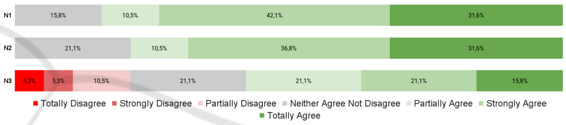

Figure 5: User's opinion on Pleasure perception.

5.3.5 Quality of the Results

Three assertions (QR1 to QR3) captured participants' responses regarding the quality of results, as shown in Figure 6.

Evaluating the quality of results generated by ChatGPT through **QR1**, 68.4% of participants classified the results as favorable. Assertion **QR2** assessed whether users encountered issues with the quality of the results, finding mixed responses, indicating that a portion still experiences problems with result quality. Participants classified the results as excellent for generating test cases on **QR3**, yielding 61.1% positive responses.

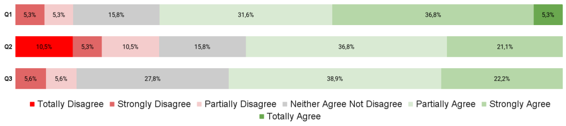

Figure 6: User's opinion on the Quality of Results perception.

5.3.6 Demonstrability of Results

Regarding the demonstrability of results, four assertions (BI1 to BI4) were used to obtain participants' responses, the results of which are presented in Figure 7. The data reveal a positive trend among participants, with the majority indicating that they can easily communicate and understand the results of using ChatGPT.

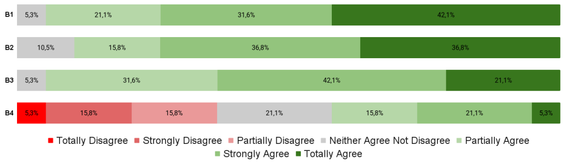

Figure 7: User’s opinion on the Demonstrability of Results perception.

BI1 looked if they can easily share results and findings, while **BI2** was regarding the ability to convey the consequences of using ChatGPT, and **BI3** whether results were clear or not, on these 3 questions the overall results were positive almost in total. For **BI4**, regarding the difficulty of explaining ChatGPT’s benefits or drawbacks, 26.4% gave positive responses while 15.8% partially disagreeing and 15.8% strongly disagreeing, indicating some challenges in articulating its advantages or disadvantages.

5.3.7 Participants’ Self-Perception Assessment Regarding the Use of ChatGPT 3.5

Advantages and Disadvantages. To understand the effectiveness of using ChatGPT 3.5 in the task of formulating test cases, we asked participants to describe the advantages and disadvantages of this activity. Opinions reflect a balanced view of the use of ChatGPT 3.5 in formulating TC and generating new ideas.

As for the advantages, 21% of participants highlighted that the tool helps speed up and make the process more productive as stated by **P3** and **P4**, with 16% stating that it allows for the quick and practical formulation of test cases, including providing new ideas ("*The advantage is a different perspective and consequently new test cases for the system*" - **P7**). In addition, 10% found that the tool was seen as helpful in developing more detailed and complete cases ("*Can go in-depth into user stories that are difficult to generate test cases for and greatly improves tests already done*" - **P11 and P19**). Another positive point mentioned by 15% of them was the practicality of use and the ability to quickly validate and explore ideas ("*It helps to be more creative and better explore the possibilities of tests that could be done*" - **P9 and P18**).

On the other hand, participants also pointed out significant disadvantages of using ChatGPT 3.5. One of the main criticisms from 26% of participants was the possibility of generating generic or vague answers as stated by **P10 and P14**. In addition, there were mentions of the need to provide very detailed prompts to avoid out-of-scope or irrelevant results by 21% of them ("*You have to detail what you want very well, making the requested points clear*" - **P16 and P18**). The tool was also criticized by 5% of them for occasionally making the process so easy that it can make the User too dependent on the tool, potentially making them "lazy" and more susceptible to errors if ChatGPT also makes mistakes as seen by **P1**.

How the Tool Helped. Seeking to understand how ChatGPT helped in the formulation of test cases, participants identified several important contributions of the tool highlighting strengths in generating ideas and saving time.

26.3% of participants mentioned the ChatGPT’s ability to generate varied and specific test cases, highlighting that the tool helped to create test cases that had not been previously considered, offering new ideas and insights ("*It helped to create new ways of testing*" - **P5 and P17**). In addition, 15.8% of them found that the tool stood out for its effectiveness in saving time and in the rapid formulation of test cases ("*It helped me to formulate answers quickly*" - **P4 and P18**). 21.1% of them evaluated the validation and refinement of ideas ("*New ideas and refinement of test cases*" - **P11**), with the tool helping to structure and refine tests more effectively.

On the other hand, some responses (5.3%) highlighted the need for good knowledge of the project to fully leverage ChatGPT’s capabilities ("*Have a good understanding of the project*" - **P9**). Furthermore, although the tool was useful in creating complex cases, this was not the main focus of the responses analyzed, indicating that ChatGPT’s help is more evident in general aspects and initial ideation than in highly specific details.

Difficulties Encountered. Participants described the difficulties they encountered when using ChatGPT 3.5 to formulate test cases. Responses highlighted the need for specificity in commands and the challenge of interpreting and adapting the generated responses.

Despite the many advantages mentioned in using ChatGPT to formulate test cases, participants also identified several difficulties associated with the tool. A common difficulty for 17.6% of participants was creating accurate and contextually appropriate prompts to obtain valuable results as stated by **P5 and P12**.

Another issue faced was the occurrence of out-of-context answers or repetitive information. Several participants noted that ChatGPT sometimes generated answers that were not directly applicable or that contained repetitions (23.5% of participants) ("*The chat delivers a lot of repetitive information*" - **P4**). Furthermore, the tool had difficulties in filtering and structuring test cases appropriately, which can lead to responses inconsistent with the scope of the project (29.4% of participants) ("*Filtering the cases can be a bit confusing*" - **P8**; "*Some responses came back different than expected*" - **P9**). Furthermore, in some cases, ChatGPT provided information when, in fact, the command was for the tool to wait for further instructions (11.8% of participants) ("*When giving the initial command asking to wait for instructions, it gave a text explaining what we had entered*" - **P17**).

Potential Use of the Tool in Generating Test Cases. Seeking to understand the potential use of ChatGPT 3.5 in generating test cases for systems, we asked participants to describe this potential from their perspective as future software testers.

The analysis of participants' opinions reveals a largely positive view of the potential use of ChatGPT 3.5 in test case generation, with some reservations regarding its practical application. Participants believe that ChatGPT can significantly improve the testing process, especially in terms of agility and support (31.6% of participants) ("*It can greatly improve the process, but it still needs adjustments*" - **P1**). The tool's ability to accelerate development and explore a more significant number of scenarios is seen as an essential advantage (15.8% of participants) ("*I believe it will make things much easier, faster*" - **P10**).

Participants highlighted ChatGPT as a valuable support tool that can assist in generating test cases, helping to make the process more efficient and detailed (21.1% of participants) ("*Very important to speed up the reasoning process, mainly*" - **P3**). However, the need for caution and supervision is a common concern, as ChatGPT can have limitations and go beyond the desired scope (26.3% of participants) ("*It can be extremely useful, but it must be used with care and attention so as not to hinder more than help*" - **P6**). Some participants also mentioned that, although ChatGPT is helpful, it should not replace the work of the human tester. The tool is a complement that can provide valuable insights and save time. Still, it is crucial that the tester maintains in-depth knowledge and performs adequate validation of the information generated (21.1% of participants) ("*I believe that using ChatGPT can greatly help the tester, but not replace him*" - **P11**).

In summary, the general perspective is that ChatGPT 3.5 can be a powerful tool for generating test cases, offering support and efficiency, but always in conjunction with the supervision and specialized knowledge of the testers.

6 CONCLUSION

The use of ChatGPT 3.5 in the formulation of test cases based on user stories has proven to be promising, although it presents significant limitations that must be addressed. In the experimental study carried out in the Practical in Software Engineering (PES) discipline at the Federal University of Amazonas (UFAM), it was observed that many participants showed a clear intention to continue using the tool, motivated mainly by the perception of increased productivity and the generation of new ideas. The quality of the results generated by ChatGPT was evaluated positively by most participants, who highlighted the speed and practicality of formulating test cases. However, some participants reported problems with the quality of the results, mentioning that the answers were often generic and not very specific.

The demonstrability of the results was another highlight, with most participants stating that they can easily communicate and understand the results obtained using ChatGPT. This indicates that the tool can be integrated into collaborative software development processes, facilitating communication between team members. Participants identified significant difficulties, such as creating precise prompts to obtain valuable answers. Furthermore, they mentioned that ChatGPT generated responses that were out of context or repetitive. These difficulties highlight the importance of the careful and well-structured use of ChatGPT, highlighting the need for command specificity. The results of this study may have been influenced by the use of user stories that were not very detailed and did not have clear acceptance criteria, especially those from sprints 1 and 2, linked to simple functionalities such as registration and login. This limitation may have affected the quality and specificity of the test cases generated by ChatGPT.

For future studies, improving the instructions and prompts will be essential. It will be necessary to define more rigorous criteria for selecting user stories and develop strategies for formulating more precise and contextualized prompts. Training participants on how to create and adjust prompts effectively may also be essential to maximize the tool's benefits. In addition, including more detailed and complex user stories may provide a more robust assessment of the effectiveness of ChatGPT in generating test cases.

In summary, ChatGPT 3.5 has significant potential as a tool to support test case generation, providing increased productivity and innovation. However, its practical use requires a balanced approach that combines the automation offered by the tool with the knowledge and supervision of human testers. The findings of this study provide valuable guidelines for the practical application of ChatGPT in software development projects, pointing to future improvements and refinements.

REFERENCES

Alagarsamy, S., Tantithamthavorn, C. K., Arora, C., and Aleti, A. (2024). Enhancing large language models for text-to-testcase generation. *ArXiv*, abs/2402.11910.

Aysolmaz, B., Leopold, H., Reijers, H. A., and Demirors, O. (2018). A semi-automated approach for generating natural language requirements documents based on business process models. *Information and Software Technology*, 93:14–29.

bin Ali, N., Engström, E., Taromirad, M., Mousavi, M. R., Minhas, N. M., Helgesson, D., Kunze, S., and Varhosaz, M. (2019). On the search for industry-relevant regression testing research. *Empirical Software Engineering*, (24):2020–2055.

Borji, A. (2023). A categorical archive of chatgpt failures. *ArXiv*, abs/2302.03494.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., 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. M., 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 *In Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages Article 159, 25 pages. Curran Associates Inc.

Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional, 13th edition.

Davis, F. D. and Granić, A. (March, 2024). *The Technology Acceptance Model: 30 Years of TAM*. Springer Cham, 1 edition.

Ellis, K. (2008). Business analysis benchmark: The impact of business requirements on the success of technology projects. *IAG Consulting*.

Frieder, S., Pinchetti, L., Griffiths, R.-R., Salvatori, T., Lukasiewicz, T., Petersen, P., and Berner, J. (2024). Mathematical capabilities of chatgpt. *Advances in neural information processing systems*, 36.

Manzoni, F. S., Rodrigues, R., and Rocha, A. C. O. (2025). Support documentation for study execution. Technical report, Federal University of Amazonas, IComp - UFAM, dx.doi.org/10.6084/m9.figshare.28053713.

Marangunić, N. and Granić, A. (2015). Technology acceptance model: A literature review from 1986 to 2013. *Universal Access in the Information Society*, 14:81–95.

Marijan, D. (2015). Multi-perspective regression test prioritization for time-constrained environments. In *2015 IEEE International Conference on Software Quality, Reliability and Security*, pages 157–162. IEEE.

Marques, N., Silva, R. R., and Bernardino, J. (2024). Using chatgpt in software requirements engineering: A comprehensive review. *Future Internet*, 16(6).

Neto, A. C. D. (2007). *Intrduction to Software Testing*. Magazine of Software Engineering, year 1, n. 1, 54 edition.

Ronanki, K., Berger, C., and Horkoff, J. (2023). Investigating chatgpt's potential to assist in requirements elicitation processes. *49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 354–361.

Shah, H. B., Harrold, M. J., and Sinha, S. (2014). Global software testing under deadline pressure: Vendor-side experiences. *Information and Software Technology*, 56(1):6–19.

Shen, Y., Heacock, L., Elias, J., Hentel, K. D., Reig, B., Shih, G., and Moy, L. (2023). Chatgpt and other large language models are double-edged swords. *Radiology*, 307(2):230163.

Sima, A. C. and de Farias, T. M. (2023). On the potential of artificial intelligence chatbots for data exploration of federated bioinformatics knowledge graphs. *ArXiv*, abs/2304.10427.

Simos, D. E., Bozic, J., Garn, B., Leithner, M., Duan, F., Kleine, K., Lei, Y., and Wotawa, F. (2019). Testing tls using planning-based combinatorial methods and execution framework. *Software Quality Journal*, 27:703 – 729.

Vogel-Heuser, B., Fay, A., Schaefer, I., and Tichy, M. (2015). Evolution of software in automated production systems: Challenges and research directions. *Journal of Systems and Software*, 110:54–84.

Wang, S., Ali, S., Yue, T., Bakkeli, Ø., and Liaaen, M. (2016). Enhancing test case prioritization in an industrial setting with resource awareness and multi-objective search. In *Proceedings of the 38th International Conference on Software Engineering Companion*, volume Companion of *ICSE '16*, pages 182–191, Austin, TX, USA. Association for Computing Machinery.

Wiegers, K. E. and Beatty, J. (2013). *Software requirements*. Microsoft Press, 3rd edition. Cited on page 37.