

Enhancing User Story Generation in Agile Software Development through Open AI and Prompt Engineering

1st Vijayalakshmi Ramasamy
College of Engineering and Computing
Georgia Southern University
Statesboro, USA
vramasamy@georgiasouthern.edu

3rd Gursimran Singh Walia
Department of Computer & Cyber Sciences
Augusta University
Augusta, USA
gwalia@augusta.edu

2nd Suganya Ramamoorthy
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
suganya.ramamoorthy@vit.ac.in

4th Eli Kulpinski, Aaron Antreassian
Computer Science
University of Wisconsin-Parkside
Kenosha, USA
{kulpi001,antre002}@rangers.uwp.edu

Abstract—This innovative practice full paper explores the use of AI technologies in user story generation. With the emergence of agile software development, generating comprehensive user stories that capture all necessary functionalities and perspectives has become crucial for software development. Every computing program in the United States requires a semester- or year-long senior capstone project, which requires student teams to gather and document technical requirements. Effective user story generation is crucial for successfully implementing software projects. However, user stories written in natural language can be prone to inherent defects such as incompleteness and incorrectness, which may creep in during the downstream development activities like software designs, construction, and testing. One of the challenges faced by software engineering educators is to teach students how to elicit and document requirements, which serve as a blueprint for software development. Advanced AI technologies have increased the popularity of large language models (LLMs) trained on large multimodal datasets. Therefore, utilizing LLM-based techniques can assist educators in helping students discover aspects of user stories that may have been overlooked or missed during the manual analysis of requirements from various stakeholders. The main goal of this research study is to investigate the potential application of OpenAI techniques in software development courses at two academic institutions to enhance software design and development processes, aiming to improve innovation and efficiency in team project-based educational settings. The data used for the study constitute student teams generating user stories by traditional methods (control) vs. student teams using OpenAI agents (treatment) such as gpt-4-turbo for generating user stories. The overarching research questions include: RQ-1) What aspects of user stories generated using OpenAI prompt engineering differ significantly from those generated using the traditional method? RQ-2) Can the prompt engineering data provide insights into the efficacy of the questions/prompts that affect the quality and comprehensiveness of user stories created by software development teams? Industry experts evaluated the user stories created and analyzed how prompt engineering affects the overall effectiveness and innovation of user story creation, which provided guidelines for incorporating AI-driven

approaches into software development practices. Overall, this research seeks to contribute to the growing body of knowledge on the application of AI in software engineering education, specifically in user story generation. Investigating the use of AI technologies in user story generation could further enhance the usability of prompt engineering in agile software development environments. We plan to expand the study to investigate the long-term effects of prompt engineering on all phases of software development.

Index Terms—Collaboration network, complex network analysis, structured collaboration network

I. INTRODUCTION

AI technology is designed to enhance human abilities and increase efficiency rather than replace them. According to a Harvard Business Review study of 1,500 companies, the most significant performance improvements come from the collaboration between humans and intelligent machines [2]. This collaboration can be particularly valuable in software development, as it combines algorithmic intelligence with human creativity in tasks related to software engineering, especially during requirements development. Software engineering educators often struggle to teach students how to document requirements, which serve as a blueprint for software development. Agile software development has made generating concise natural language descriptions of functionality, also known as *user stories*, an essential task in requirements documentation [7]. Despite capturing all necessary functionalities and perspectives, user stories can still be prone to inherent defects (e.g., incompleteness, incorrectness) that may be introduced during downstream development activities such as software designs, construction, and testing [8].

In many computing programs nationwide, students must complete a senior capstone project lasting either a semester or a year. This project involves teamwork to gather and document

technical requirements, which are then implemented. Therefore, effective user story generation practices are essential for software projects, as they form the foundation for design, development, and testing. We believe that using LLM-based techniques can be beneficial in assisting pedagogy by helping students uncover potential new user stories that may have been overlooked or left out during manual analysis of requirements gathering from various stakeholders.

The emergence of conversational Artificial Intelligence (AI) technologies introduced by powered tools like Chat GPT-4 (Generative Pre-trained Transformers) [6], Codex (for code generation) [5], DALL-E (for image synthesis) [3], CLIP (for vision-language tasks) [4], and Large Language Models (LLM) have promised advancements and applications by creating new opportunities and challenges in various fields. Such tools have increased productivity in banking, hospitality, and information technology industries. Within the first five days of its launch in November 2022, ChatGPT had gathered one million users, which quickly soared to 100 million in just a few months. Since then, ChatGPT has undergone several updates and has become the fastest-growing service with a significant impact, boasting over 180.5 million users worldwide as of March 2024 [1]. Saka et al. present a chronological summary of important milestones, opportunities, and challenges in developing and releasing GPT models and their evaluation for deployment in the industry [10].

The primary objective of this research study is to explore how OpenAI techniques and tools can be used in team-based software development project-based courses at two academic institutions. In the pilot study, the students working on team-based capstone projects were asked to collect user stories using traditional brainstorming methods with the stakeholders (control group). Then the student teams fine-tune the collected user stories using prompting techniques with OpenAI agents (treatment) such as gpt-3.5-turbo([12], [13], [14], gpt-3.5-turbo-instruct([15], [16]), and speech recognition software whisper([19], [18], [17]) OpenAI models. This primary objective leads to the following overarching research questions.

- **RQ 1:** What aspects of user stories generated using OpenAI prompt engineering differ significantly from those generated using the traditional method?
- **RQ 2:** Can the prompt engineering data provide insights into the efficacy of the questions/prompts that affect the quality and comprehensiveness of user stories created by software development teams?

Experts from the software industry analyzed the user stories created by student teams to study the impact of prompt engineering on the effectiveness and innovation of user story creation processes. They provided recommendations for integrating AI-driven approaches into software development practices. The preliminary study indicated that implementing prompt engineering methods can aid teams in discovering new user stories and enhancing software quality.

The findings and conclusions of this research can offer valuable insights into the effective use of prompt engineering techniques for generating user stories. Furthermore, the results

can assist developers in creating accurate, complete, and error-free user stories.

The rest of the paper is structured as follows. Section II provides an overview of requirements engineering and demonstrates the role, challenges, and significance of OpenAI tools in user story generation in software development. Section III describes a study design that encourages students to use OpenAI tools to generate complete and meaningful user stories. Section IV presents the experimental results and a detailed discussion of the findings. In Section V, we discuss potential threats to validity, and in Section VI, we summarize the findings and outline future research directions.

II. BACKGROUND

This section provides insights into the role of user stories in Agile projects, the shortcomings of traditional user story creation methods, the power of OpenAI GPT in NLP tasks, and the role of GPTs in RE tasks.

A. User Story Generation for Agile Software Development

User stories play a crucial role in requirements engineering (RE), and the principles and best practices for writing user stories used in Agile software development methodologies like Scrum and eXtreme Programming (XP) are widely applied in software engineering [20]–[22]. Software development teams can find it challenging to generate comprehensive user stories that capture all necessary functionalities and perspectives. This can result in incomplete or ineffective software designs, impacting the project's success.

B. OpenAI Generative Pre-trained Transformer Models

The rise of generative AI tools and Large Language Models (LLMs) like ChatGPT has sparked concerns about their potential to overshadow human cognitive functions in various tasks, attracting mass attention due to their accessibility [23]. While ChatGPT demonstrates proficiency in explaining complex concepts such as statistics, its responses may contain inaccuracies inherited from its training data, necessitating critical appraisal by educators [14]. Integrating LLMs such as the GPT series in medical education holds promise for enhancing student learning experiences. Yet, challenges, including bias, plagiarism, and privacy, must be addressed through proper guidelines and training [26]. Despite ChatGPT's ability to generate high-quality essays, linguistic disparities exist between AI-generated and human-written content [25]. Surveys indicate that while students are familiar with AI tools, they are not widely used for academic purposes, suggesting potential untapped benefits [27]. Projects like ASSISTANT aim to advance AI tools for adaptive manufacturing, leveraging machine learning to enhance digital twin models and generative design processes [31]. Integrating AI into collaborative learning endeavors holds promise for revolutionizing STEM education, offering inclusive and engaging learning environments [29]. A framework for integrating generative AI into project-based learning emphasizes understanding algorithmic mechanisms, identifying biases, and utilizing AI tools for effective problem-solving. Prompt engineering emerges as a burgeoning field

focused on designing prompts to guide LLM outputs in various tasks [23].

C. Role of Generative Pre-trained Transformers in Requirements Engineering

Requirements Engineering (RE) is a crucial software development Phase involving elicitation, analysis, specification, and validation of software requirements tasks. However, RE remains challenging due to communication complexities, uncertainty in the early stages, and inadequate automation support. Large-language models (LLMs) have emerged as powerful tools across various domains, including natural language processing (NLP), program understanding, and code generation. OpenAI's GPT models, such as GPT-4, lead the way in this revolution by learning from vast amounts of text data and exhibiting impressive capabilities in understanding context, generating coherent text, and even assisting developers.

LLMs are promising tools for revolutionizing RE in software development. Researchers explore these tools' potential for Agile development in various stages of RE. LLMs can extract requirements from multiple sources and generate relevant prompts to help developers capture essential requirements efficiently. GPT models can analyze requirements documents, identify inconsistencies, and suggest improvements. LLMs can draft clear and concise requirements and validate them by simulating scenarios, performing feasibility checks, and identifying ambiguities. In addition, the authors include a SWOT analysis of using LLMs for RE tasks and report that while LLMs excel in natural language understanding, context awareness, and creative text generation, they present some challenges in fine-tuning, bias mitigation, and handling domain-specific jargon. Therefore, while integrating LLMs into software tools, human judgment can enhance productivity [9].

LLMs gained attention after the GPT models like ChatGPT were launched. A recent study shows that GPT models are used in all project phases, with additional value in non-phase-specific areas by leveraging zero-shot learning, few-shot learning, and chain-of-thoughts (COT) learning via prompt designs or integration with existing systems. The findings revealed the use of GPT models throughout the project lifecycle by fine-tuning GPT models with structured data to improve the performance of the models and leverage existing or external databases (RAG) that these models have not accessed. The challenges of leveraging GPT models are highlighted using a use case prototype developed for materials selection and optimization. The research evaluated the application of GPT models in the Architecture, Engineering, and Construction (AEC) industry, aiming to identify opportunities and limitations using a case study. The prototype was tested in three scenarios: zero-shot, few-shot with system prompting, and edge case prompting scenarios [10].

Borg, the principal researcher at CodeScene AB, a company that develops CodeScene (a next-generation code analysis tool), along with the panelists in Feb 2024, predicted a massive increase in NLP and LLMs in software engineering research

due to the rise in RE-oriented research papers. They also noted that since requirements are typically expressed in natural language, this trend will likely continue to advance engineering in various ways. The enthusiasm for the new RE possibilities in product development is attributed to the evolution of user interaction with textual prompts. However, they point out that education is a significant challenge for non-experts and novice developers of LLM technology. Encouraging research on AI engineering for LLM-based systems, including process adaptation, system design, requirements elicitation, and software testing, are essential [24].

III. STUDY DESIGN

The proposed AI-based user story generation system employs an experimental design in which student participants in two academic institutions are randomly assigned to different groups. The research study emphasizes the importance of user stories in agile software development and the role of OpenAI techniques, such as prompt engineering and LLM, in user story generation. It also emphasizes the need for validation techniques, including user feedback surveys and usability testing, to ensure the effectiveness of user stories in meeting user needs and project goals. With this objective, we design a framework for the experimental study as shown in Fig. 1.

The students have been assigned Task 1 of developing applications using traditional brainstorming methods (e.g., role-playing) that enable collaboration, creativity, and teamwork. However, limiting the focus on obvious user needs has restricted the scope and creativity of applications, leading to a lack of consideration for innovative features. Students often struggle to generate creative solutions, resulting in mundane user stories. Often, this method is time-consuming and may not result in enough diverse user stories. To help students generate user stories more efficiently, we propose an automated approach that employs AI-based tools like prompt engineering platforms. To demonstrate the framework, we will use an example throughout. The student team developed the following user stories in a *web application for a charity organization*.

- *Donation Process*: "As a donor, I want to easily make a donation online so that I can support the charity's cause conveniently."
- *Volunteer Sign-Up*: "As a volunteer, I want to easily sign up for volunteer opportunities so that I can contribute my time and skills to the charity."
- *User Engagement*: "As a social media manager, I want to easily share updates and stories about the charity so that I can engage supporters and raise awareness."

In Task 2, students used ChatGPT to improve user stories from Task 1, enhancing the application's usability and effectiveness without using sophisticated prompt engineering. The refined user story for the charity organization web application using ChatGPT is as follows:

- **User prompt**: Refine the user stories generated for a charity donation web application <Task 1 User stories here>.

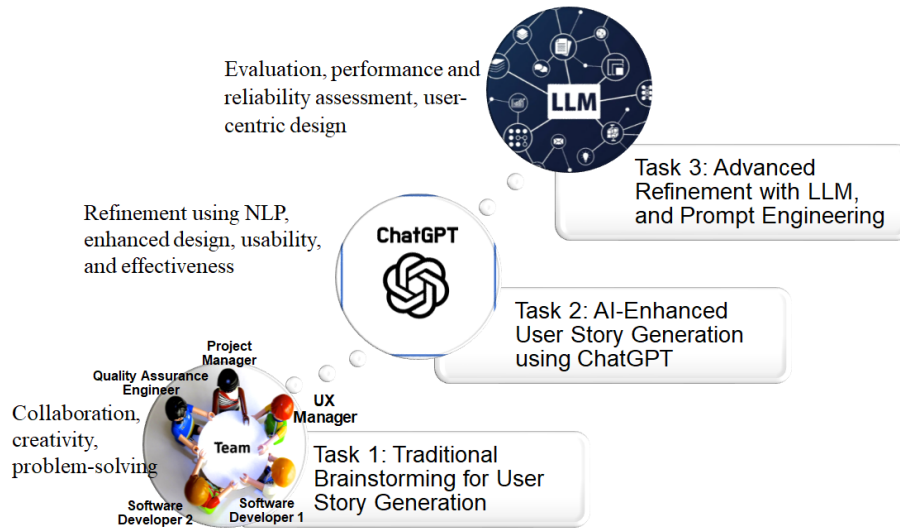


Fig. 1. Automated User Story Generation Framework

• ChatGPT refinement:

- “Enhance the donation process by providing personalized suggestions based on the donor’s interests and previous donation history.”
- “Improve the volunteer sign-up process by offering suggestions for volunteer opportunities based on the volunteer’s skills and availability.”
- “Enhance social media engagement by providing personalized content suggestions for social media posts based on trending topics and supporter interests.”

Task 2 introduces some level of personalization, but current systems cannot often tailor user experiences based on individual preferences and history. The user stories in both tasks focus on static functionalities, but in today’s dynamic world, systems need to be adaptive and responsive to changing user needs and trends. Therefore, the proposed User Story Generation tool overcomes the above problems by providing advanced personalization by analyzing user behavior and preferences to suggest tailored content and features. It can also automate repetitive tasks, such as scheduling volunteer shifts or tracking donations, to improve efficiency and reduce manual effort. The LLM tool can analyze real-time data and trends to suggest dynamic content updates, keeping the application relevant and engaging. The user stories generated by the tool for the above tasks are as follows.

- “As a donor, I want to receive personalized donation suggestions based on my interests and previous donation history so that I can make more informed and meaningful contributions.”
- “As a volunteer, I want the volunteer sign-up process to suggest opportunities based on my skills and availability so that I can find relevant and fulfilling volunteer experiences.”
- “As a social media manager, I want the social media content suggestions to be personalized based on supporter

interests and trending topics so that I can effectively engage supporters and raise awareness.”

Thus, the proposed LLM tool would significantly enhance user stories by providing advanced personalization, automation, and dynamic content updates, addressing the gaps in today’s practical world. Efficient validation techniques, such as user feedback surveys, usability testing, and A/B testing, can be used to justify the improvements in user stories and validate the effectiveness of the LLM tool in enhancing user experiences.

By incorporating these strategies, students can create a more thorough and original set of user stories for their capstone project. This will help them better comprehend users’ requirements, improve their creativity and analytical thinking abilities, and deliver a more significant final product for the charitable organization.

A. Data Collection

The data used in this study consisted mainly of user stories created by university students with prior knowledge of agile processes. The user stories were collected from various sources on the internet. A total of 43 students, organized into 19 teams as part of a capstone software engineering course, participated in the study.

The hand-crafted user stories serve as the study’s first task, giving a baseline for what complete user stories should look like. This allowed us to use OpenAI’s ChatGPT-4 model and create user stories so we could compare the generated user stories to the quality hand-crafted ones. This analysis leads us to task 2, which involves using prompt engineering to enhance the generated user stories.

The third and final task of the study involves training an LLM on quality user story data. This is done to give the model a baseline for proper responses given a project summary from the user. Our finished model can take varying project summary lengths to create quality user stories while expanding

on the summaries to include certain specificity requirements, potential stakeholders, and other important information for a project. The third step of training the LLM model to optimize further the user stories generated

IV. RESULTS AND DISCUSSION

Task 2 involved an activity AI-enhanced user story generation using ChatGPT across various applications. Each team utilized its own user prompt to procure user stories generated by ChatGPT tailored to their specific application needs, which were subsequently submitted for evaluation. Following the submission, we conducted a post-test student survey comprising a set of questions aimed at gauging the efficacy and impact of this approach. We collected feedback from 18 teams, each consisting of four members, spanning diverse application domains.

In Task 2, students used ChatGPT to improve user stories from Task 1, enhancing the application's usability and effectiveness without using sophisticated prompt engineering. The refined user story for the charity organization web application using ChatGPT is as follows: Prompt by ChatGPT: Design a user story for a charity donation web application from the perspective of a donor.

"As a donor, I want to easily contribute funds to a charitable cause, So that I can support organizations making a positive impact". "As a volunteer, I want to search for diverse volunteer opportunities, so that I can find causes that align with my skills and interests." "As a volunteer, I want to easily register on the platform, so that I can access and apply for volunteer opportunities." "As a volunteer, I want to manage my profile and preferences, so that I can keep my information up-to-date and receive relevant volunteer opportunities." "As a volunteer, I want to track the status of my volunteer applications, so that I can stay informed about the progress of my contributions." "As a volunteer, I want to provide feedback on my volunteering experiences, so that organizations can improve their volunteer programs and better engage volunteers." "As a user, I want to receive regular updates on the impact of my donations, so that I can feel connected to the causes I support and understand the difference my contributions make."

It is apparent from the above user stories generated by traditional brainstorming methods might cover the basic functionalities and requirements of the charity donation web application, such as browsing charitable causes, making donations, and receiving confirmation. However, ChatGPT can help expand upon these initial ideas by suggesting additional features or enhancements that stakeholders may not have considered initially. For example, it could suggest incorporating a feature that allows donors to set up recurring donations or provide personalized recommendations based on past donation history.

Using the examples given above, we utilize OpenAI and prompt engineering to enhance these stories. We aim to provide more detailed requirements and user stories based on careful prompt creation to create more complete project outlines that will need less revision once the project development starts. After taking our examples and using OpenAI

to enhance the user stories, not only were the original stories expanded upon, but the requirements and stakeholders were also brought into scope. This closely followed our goal to extend and enhance stories given to our model.

A. Analysis of Post-test Student Survey

The student survey was conducted to gather student feedback and identify areas for improvement. The post-test survey used a Likert scale to assess students' reflections on working on team-based projects with the OpenAI tool Chat GPT. Following are the prompts of the questions, observations, and inferences of the student survey results presented in Fig. 2 addressing RQ 1 and RQ 2.

Through detailed analysis of the survey results and observation, we noted several significant improvements in student learning outcomes.

(a) Improve students' problem-solving skills: Most student teams found that using Chat GPT to generate user stories resulted in better solutions than traditional brainstorming methods. Additionally, they felt that the accuracy of the resulting user stories improved significantly when the correct prompts were used. On the other hand, participants who provided irrelevant prompts, especially those unfamiliar with basic prompt engineering and its usage, obtained poor results.

(b) Prepare students for real-world software development environments: While around 70% of students reported positive results, indicating a beneficial impact on their preparedness for real-world scenarios, we analyzed the reasons behind the remaining 30% who provided negative feedback. For the majority who reported positive outcomes, it suggests that the experience of working with OpenAI tools for user story generation significantly contributed to their readiness for real-world software development environments. These students likely found the task helpful in honing their skills, gaining practical experience, and familiarizing themselves with tools and techniques commonly used in industry settings. Their positive feedback underscores the relevance and utility of incorporating OpenAI tools into educational curricula to better align with industry standards and expectations. On the other hand, the feedback from the remaining 30% of students who expressed negative sentiments was closely analyzed. Understanding their perspectives and reasons for dissatisfaction is crucial for identifying potential areas of improvement in implementing OpenAI tools for user story generation. Possible factors contributing to their negative feedback could include difficulties in understanding or utilizing the tools effectively, challenges in translating generated user stories into actionable development tasks, or discrepancies between the generated stories and their expectations or project requirements. In addressing these concerns, educators and curriculum designers can implement targeted interventions such as additional training sessions, providing clearer guidelines or prompts for utilizing OpenAI tools, and offering personalized support to students struggling with the task. By addressing these challenges and optimizing the user experience of working with OpenAI tools, educators can strive to enhance the overall effectiveness and satisfaction

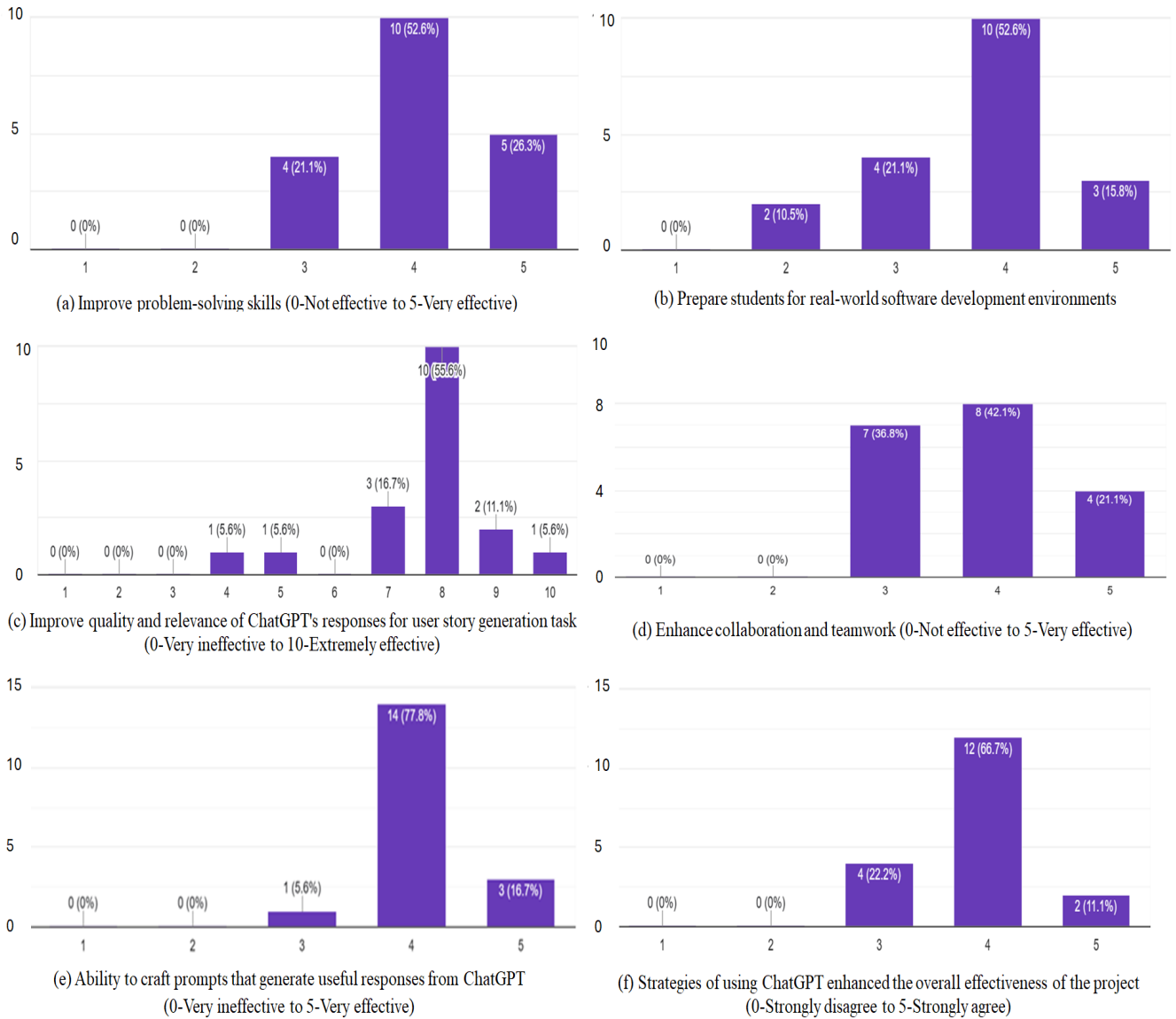


Fig. 2. Survey Results after activity-2: Generating user stories with OpenAI tools improve problem-solving skills and preparedness for real-world environments

of students engaging in user story generation tasks, ultimately better preparing them for real-world software development environments.

(c) Quality and relevance of ChatGPT's responses for user story generation task: The analysis of Fig. ?? results provided the following insights. We explored how students rated the quality and relevance of ChatGPT's responses when provided with specific prompts. Out of the total students surveyed, approximately 55.6% provided positive feedback, suggesting satisfaction with the quality and relevance of ChatGPT's responses to the prompt input. Additionally, a notable subset comprising 17% of the students rated ChatGPT's responses as top-tier, indicating exceptionally high levels of quality and relevance in response to the provided prompts. However, it's important to note that a small proportion of students

expressed dissatisfaction with the relevance of ChatGPT's responses to the prompt input, as indicated by the mention of "very few students" providing less relevant feedback. While this constitutes a minority viewpoint within the sample, it underscores the need for further investigation into the factors influencing the perceived relevance of ChatGPT's responses. Possible considerations may include the clarity of prompts, the specificity of task requirements, and the overall effectiveness of communication between students and the AI tool. Overall, the analysis suggests a predominantly positive assessment of ChatGPT's responses to prompt input for the user story generation task, with most students expressing satisfaction with the quality and relevance of the AI-generated content. This underscores the potential of ChatGPT as a valuable tool in facilitating user story generation processes, albeit with

some variability in perceived relevance that warrants further exploration and refinement. (d) Enhance collaboration and teamwork: Many student teams reported improved collaboration and teamwork while using OpenAI tools, surpassing the effectiveness of traditional brainstorming techniques. Firstly, using OpenAI tools facilitated smoother communication and idea exchange among team members. The user-friendly nature of these tools allowed students to express their thoughts more effectively, leading to a more cohesive and coordinated team effort. Additionally, OpenAI tools provided students with innovative solutions and perspectives, stimulating creativity within the team. These collaborative efforts enabled them to address challenges from various perspectives, develop more robust and functional user stories, and learn more project development strategies.

(e) Improve ability to craft prompts for useful ChatGPT responses: The survey results reveal a generally positive outlook among students regarding their confidence in crafting prompts that elicit useful responses from ChatGPT. Out of the 19 student teams surveyed, approximately 14 indicated a higher confidence level, with 3 of them selecting the highest rating of "extremely effective." This indicates a strong belief among these students in their ability to formulate prompts that yield valuable responses from ChatGPT effectively. Additionally, only one student provided a neutral response, suggesting a relatively minor proportion of uncertainty or ambivalence regarding prompt crafting ability. Overall, most students exhibit confidence in their proficiency to generate prompts conducive to productive interactions with ChatGPT, reflecting a favorable perception of their capabilities in leveraging this AI tool for project-related tasks.

(f) Strategies used for prompting ChatGPT to improve the overall effectiveness of the project: The results of the survey indicate a predominantly positive perception among students regarding the effectiveness of their strategies for using ChatGPT in enhancing software engineering projects. Out of the 19 teams surveyed, approximately 77% expressed a belief that their strategies involving ChatGPT contributed to enhancing the overall effectiveness of their projects, suggesting the value of leveraging ChatGPT as a tool to augment their project development processes. However, it's notable that around 22.2% of students provided neutral responses, indicating a degree of uncertainty regarding the impact of their ChatGPT utilization strategies on project effectiveness. While this constitutes a minority viewpoint within the sample, it highlights the need for further exploration into the factors influencing students' perceptions of the effectiveness of ChatGPT integration into project workflows. Possible considerations for addressing this neutrality may include evaluating the students' specific prompts when utilizing ChatGPT, assessing the alignment of these strategies with project objectives, and identifying any challenges or barriers encountered during implementation. Additionally, providing opportunities for students to share their experiences and insights related to best prompting practices and ChatGPT usage could facilitate a deeper understanding of its potential benefits and limitations in the context of software

engineering projects.

The results will be valuable for refining the LLM program and addressing experimental errors. The outcomes of this study are expected to contribute to potential improvements in software design and development processes, ultimately enhancing innovation and efficiency in project-based educational settings. Overall, this research seeks to contribute to the growing body of knowledge on the application of artificial intelligence in software engineering education, specifically in user story generation. Investigating the use of AI technologies in user story generation could further enhance the effectiveness of prompt engineering in agile software development environments. We plan to expand the study to investigate the long-term effects of prompt engineering on all phases of software development.

V. CONCLUSION AND FUTURE WORK

This research study investigated the potential application of OpenAI techniques in software development courses at two academic institutions. This exploration aims to enhance software design and development processes, with the ultimate goal of improving innovation and efficiency in team project-based educational settings. The study utilizes data from student teams, comparing those who generate user stories using traditional methods (control group) with those who use OpenAI agents (treatment group), such as gpt-4-turbo, for generating user stories. The students created user stories for software projects using traditional techniques and then using OpenAI's Chat GPT. The student reflections survey demonstrates the significant outcomes in real-world settings using OpenAI prompt engineering for user story generation. The expert evaluation of user stories demonstrated the overall effectiveness and innovation of user story creation, which provided guidelines for incorporating AI-driven approaches into software development practices. In summary, this study contributes to potential improvements in software design and development processes, ultimately enhancing innovation and efficiency in project-based educational settings. Overall, this research seeks to contribute to the growing body of knowledge on the application of artificial intelligence in software engineering education, specifically in user story generation. Investigating the use of AI technologies in user story generation could further enhance the effectiveness of prompt engineering in agile software development environments. Our plan is to expand the study to investigate the long-term effects of prompt engineering on all phases of software development by proposing RQ 3: How can the results from RQ1 and RQ2 be used to train students in effectively generating user stories using prompt engineering? Industry experts evaluated the user stories created and analyzed how prompt engineering affects the overall effectiveness and innovation of user story creation, which provided guidelines for incorporating AI-driven approaches into software development practices.

REFERENCES

- [1] Shewale R. "CHATGPT Statistics - Users, Revenue." DemandSage, 19 Mar. 2024, www.demandsage.com/chatgpt-statistics/. Accessed 20 Mar. 2024.

- [2] Wilson J. and Daugherty P. (2018). "Collaboration between humans and intelligent machines: Combining algorithmic intelligence with human creativity", *Harvard Business Review*, 96(4), 54-62.
- [3] "DALL-E: Creating images from text." OpenAI, 2023, openai.com/research/dall-e.
- [4] "CLIP: Connecting text and images." OpenAI, 2023, openai.com/research/clip.
- [5] "Codex: AI-powered code generation." OpenAI, 2023, openai.com/research/codex.
- [6] "ChatGPT, GPT-4." OpenAI, 30 Jun. 2023, chat.openai.com/chat.
- [7] Cohn M. "User Stories Applied: For Agile Software Development." Addison Wesley, 2004.
- [8] Abrahamsson A., Fronza I., Moser R., Vlasenko J., and Pedrycz W. "Predicting Development Effort from User Stories," in *Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement*. IEEE Computer Society, pp. 400–403, 2011.
- [9] Arora, C., Grundy, J., and Abdelrazek, M. (2023). Advancing Requirements Engineering through Generative AI: Assessing the Role of LLMs. *ArXiv*. /abs/2310.13976
- [10] Saka, A., Taiwo, R., Saka, N., Salami, B. A., Ajayi, S., Akande, K., and Kazemi, H. (2024). GPT models in construction industry: Opportunities, limitations, and a use case validation. *Developments in the Built Environment*, 17, 100300. <https://doi.org/10.1016/j.dibe.2023.100300>
- [11] Borg, M. (2024). Requirements Engineering and Large Language Models: Insights From a Panel. *IEEE Software*, 41(2), 6-. <https://doi.org/10.1109/MS.2023.3339934>
- [12] OpenAI. (n.d.). Models - OpenAI API. <https://platform.openai.com/docs/models>
- [13] OpenAI. (n.d.). API Reference - OpenAI API. <https://platform.openai.com/docs/api-reference>
- [14] Ellis, A. R., and Slade, E. (2023). A New Era of Learning: Considerations for ChatGPT as a Learning Companion. *Journal of Statistics and Education*, 12(3), 128-133. [2]
- [15] OpenAI. (n.d.). API Reference - OpenAI API. <https://platform.openai.com/docs/api-reference>
- [16] OpenAI. (n.d.). Use API with model "gpt-3.5-turbo-instruct"? <https://community.openai.com/t/use-api-with-model-gpt-3-5-turbo-instruct/496733>
- [17] OpenAI. (2022). Introducing Whisper. <https://openai.com/research/whisper>
- [18] Hugging Face. (2023). [openai/whisper-large](https://huggingface.co/openai/whisper-large). <https://huggingface.co/openai/whisper-large>
- [19] GitHub. (2022). Is there a documentation or Reference for the Whisper API. <https://github.com/openai/whisper/discussions/525>
- [20] Cockburn, A., and Highsmith, J. (2001). Agile software development: The people factor. *Computer*, 34(11), 131-133.
- [21] Martin, R. C. (2003). *Agile software development: principles, patterns, and practices*. Prentice Hall PTR.
- [22] Chau, T., Maurer, F., and Melnik, G. (2003). Knowledge sharing: agile methods vs. Tayloristic methods. *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2003., 302-307.
- [23] Bertalan Mesko. (2023). Prompt Engineering as an Important Emerging Skill for Medical Professionals: Tutorial. *Journal of Medical Internet Research*, vol. 25. <https://doi.org/10.2196/50638>
- [24] Amanda R. Ellis and Emily Slade. (2023). A New Era of Learning: Considerations for ChatGPT as a Tool to Enhance Statistics and Data Science Education. *Journal of Statistics and Data Science Education*, 31:2, 128-133, <https://doi.org/10.1080/26939169.2023.2223609>
- [25] Stefan Herbold, Annette Hautli-Janisz, Ute Heuer, Zlata Kikteva and Alexander Trautsch. A large-scale comparison of human-written versus ChatGPT-generated essays. *Scientific Reports*.
- [26] Alaa Abd-alrazaq1, Rawan AlSaadi1, Dari Alhuwail, Arfan Ahmed1, Pdraig Mark Healy, Syed Latifi, Sarah Aziz1, Rafat Damseh, Sadam Alabed Alrazak, Javaid Sheikh (2023). Large Language Models in Medical Education: Opportunities, Challenges, and Future Directions. *JMIR Medical Education*, 9:e48291.
- [27] Harpreet Singh, Mohammad-Hassan Tayarani-Najaran and Muhammad Yaqoob (2023). Exploring Computer Science Students' Perception of ChatGPT in Higher Education: *Education Science*. vol.13, 924. <https://doi.org/10.3390>.
- [28] G. Castañé, A. Dolgui, N. Kousi, B. Meyers, Simon Thevenin, E. Vyhmeister, P.-O. Östberg (2023). The ASSISTANT project: AI for high level decisions in manufacturing: HAL Open Science.
- [29] Marco Antonio Rodrigues Vasconcelos, Renato P. dos Santos (2023). Enhancing STEM learning with ChatGPT and Bing Chat as objects to think with: A case study: *EURASIA Journal of Mathematics, Science and Technology Education*, 2023, 19(7), em2296 ISSN:1305-8223.
- [30] Yinqiu Liu, Hongyang Du, Dusit Niyato, Jiawen Kang, Shuguang Cui, Xuemin (Sherman) Shen and Ping Zhang (2023). Optimizing Mobile-Edge AI-Generated Everything (AIGX) Services by Prompt Engineering: Fundamental, Framework, and Case Study: *IEEE Networks*, 2023, <https://doi.org/10.1109/MNET.2023.3335255>.
- [31] Castañé, G., Dolgui, A., Kousi, N., Meyers, B., Thevenin, S., Vyhmeister E., and Östberg, P.-O. The ASSISTANT project: AI for high level decisions in manufacturing. *International Journal of Production Research*, 61(7), 2288-2306, 2023.