From Real-Time Conversation to User Story

Leveraging Agile Requirements through LLM

Ramon Martinez Pereira
Graduate Program in Computer Science (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR)
Curitiba Paraná Brazil
ramon.pereira@pucpr.edu.br

Andreia Malucelli
Graduate Program in Computer Science (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR)
Curitiba Paraná Brazil
malu@ppgia.pucpr.br

Sheila Reinehr
Graduate Program in Computer Science (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR)
Curitiba Paraná Brazil
sheila.reinehr@pucpr.edu.br

ABSTRACT

User stories have become the most widely adopted technique for requirements elicitation and documentation in agile teams. However, ensuring high-quality requirements through user stories remains a significant challenge. Natural Language Processing (NLP) techniques have been proposed to enhance the quality of these artifacts, but technological constraints have limited their effectiveness. More recently, the emergence of Large Language Models (LLMs) has prompted new approaches. Nonetheless, existing proposals remain preliminary and constrained. This research presents a conceptual framework for an LLM-based approach designed to support the generation of high-quality user stories from real-time conversations with end-users, thereby reducing the reliance on traditional requirements documents. The approach integrates context-specific information and incorporates automated and manual validation steps to ensure the quality of LLM-generated content and mitigate hallucinations. A preliminary evaluation was conducted with software engineering experts to evaluate the proposed approach's adherence to real-world problems. The proposal was perceived as easy to use, effective, and capable of producing high-quality outcomes. Furthermore, participants expressed a positive experience with both the introduction to the approach and their involvement in the evaluation. This engagement prompted reflections on novel opportunities for integrating LLMs into their professional practices.

KEYWORDS

Automated user story generation, LLM for agile software requirements, agile requirements with LLM

1 Introduction

Effective communication between the users' needs and the development team is crucial for the success of a software project. Over time, user stories have become the most common notation for capturing requirements in agile development projects [1], [2], [3], [4]. Typically written from the user's perspective, user stories express their expectations using semi-structured natural language, often following standardized templates [5], [6];

Empirical evidence shows that user stories effectively support faster delivery, improve software quality, and enhance agile team productivity [7], [8]. Despite these benefits, the effective communication of requirements through user stories still poses significant challenges. In practice, interpretation difficulties can compromise accuracy and precision [9], and poorly written user stories [3] tend to be ambiguous, incomplete, and open to multiple interpretations, failing to capture complete requirements [10]. Although writing user stories is relatively straightforward, ensuring they are complete, consistent, unambiguous, and testable remains a significant challenge [11].

To mitigate these issues and support both the enhancement of existing user stories [11] and the generation of requirements documents [12], recent advances in Natural Language Processing (NLP), particularly Large Language Models (LLMs) and generative Artificial Intelligence, have been explored. These approaches have demonstrated that LLMs can provide support for high-quality user stories. However, the proposed solutions are still very preliminary and present limitations. Most of them are based upon pre-existing requirements documents (or even user stories), which might add extra effort to the software development process. The heaviest part of the work, capturing the requirements from multiple sources (documents, conversations, interviews, research, etc.), still needs to be done by humans without any support.

The existing proposals also do not count on validation of the LLM's output as a step in the application process, either human or automated. This issue impacts the results by having big and complicated user stories and even making these more prone to hallucinations [13], [14]. Hallucinations could be reinforced due to the existing solutions not considering domain and context-specific confidential information. Since these LLM agents were not trained with company-specific data, the information they are based on could not adhere to the end-user's particular reality.

This research aims to propose an LLM-based approach to enable the creation of high-quality user stories from real-time conversations with users. The remainder of this paper is structured as follows: Section 2 presents the background; Section 3 introduces the proposed approach; Section 4 reports on a preliminary evaluation conducted with software engineering industry experts to validate the approach's applicability to real-world problems; and Section 5 concludes the paper and outlines directions for future work.

2 Background

Natural Language Processing (NLP) techniques have been widely used to support many different activities in the software engineering domain [15][16][13], [15], [16], [17][13]. This knowledge domain also potentially improves user story quality[18].

A work to be highlighted is the application of NLP to detect knowledge gaps in user stories [21]. This approach optimizes existing but incomplete requirements by suggesting words to the user story creator based on similar user stories. However, the technique relies on an extensive base of pre-existing, well-written, standardized user stories to be effective, which may be challenging.

Lucassen et al. [3] proposed a user story conceptual model and a quality framework, called QUS. They introduced an automated NLP-based tool (AQUSA) to evaluate the compliance of user stories to the proposed quality criteria. However, AQUSA does not ensure the holistic quality suggested by the QUS framework since AQUSA is fully implemented to syntactic (textual structure) criteria only. Authors are not sure that other quality types can be implemented entirely in the tool due to technological limitations.

Despite the narrow scope of application and validation, AQUSA, introduces a machine-aided approach to enhancing the quality of user stories. It incentivized other authors' recent exploration of these concepts by applying more advanced technology. The recent popularization of LLMs like ChatGPT has impacted on scientific research in multiple areas. Some authors have pursued this opportunity within the User Stories field.

Rodriguez et al. explored the use of ChatGPT to recreate the documentation of a healthcare communication platform to promote patient-provider communication and patient engagement in a commercial diabetes prevention program [12]. They generated a shallow scope of content, recreating a few of the theoretical bases, user stories, requirements documents, design diagrams, and code for a subset of the requirements. As a result, a similar work that took 200 human hours to develop was done in 5 hours using ChatGPT on a complex domain like healthcare without having an expert in prompt engineering in the team.

Zhang et al. proposed a conceptual model for LLM agents to support multiple software engineering tasks, like coding or requirements review [11]. The authors propose a multiple-role agent solution, pre-assigning different expectations and functions. In this experiment, multiple iterations between the Product Owner and Requirements Engineer agents enhanced existing user stories from an IT company. Experts from the same IT company considered that the solution improved the user stories by providing clarity, specificity, and business value to the stories. The lower scores evaluated by experts were in simplicity, brevity, and appropriate level of detail, resulting in too big stories, which could increase ambiguity and loss of precision. Their research underscores the indispensable role of human intelligence in monitoring and evaluating LLM-based approaches. The solution also depends on enhancing pre-existing user stories, not creating user stories from scratch.

3 The Proposed Approach

This work proposes a new concept of applying LLMs to the user story generation process that does not require existing documentation or specific prompt engineering knowledge while still guaranteeing high-quality output, as seen in Figure 1.

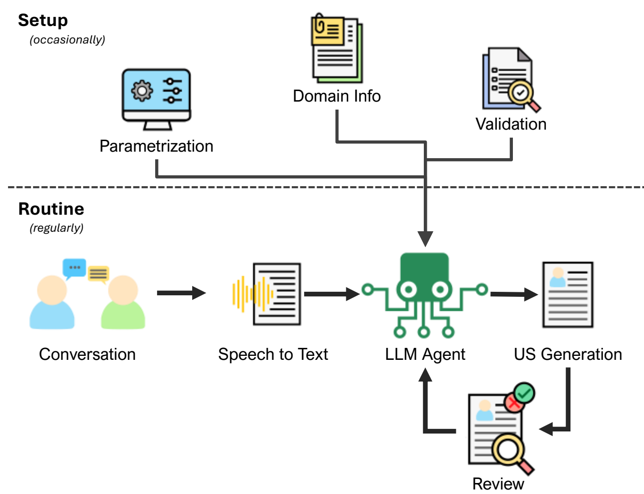

Figure 1: User Story Generation from Real-Time Conversation

The proposed approach is tool-agnostic, meaning it can be implemented using a variety of commercial or open-source LLMs and transcription tools. However, it is still essential to ensure that a private instance of these LLM tools is provisioned, since the proposed approach may involve handling sensitive or confidential information.

The proposed approach comprises two main processes: Setup and Routine. This separation is designed to optimize its use by an operator (e.g., Product Owner, Requirements Engineer), that is, the professional responsible for eliciting and documenting requirements in the form of user stories.

The approach also contemplates the user role, referring to the application's end-user, or any stakeholder who holds the system's requirements representing the user's interests. In some organizations, the operator and user roles may be represented by the same person.

The remainder of this section describes each process and step of the conceptual proposal.

3.1 Setup Process

Setup is a process applied to parametrize, configure, train, and validate the content and preparation of the LLM agent. This process should be run as the initial step for interacting with the approach, and anytime the parameters, configurations, and training are tweaked, which is expected to happen occasionally. The setup process steps are described in the upcoming subsections.

3.1.1 Parametrization. The objective of the parametrization is to define the LLM agent's configurations. These parameters can change their behavior, impacting the resulting User Story. The parameters may vary depending on the LLM technology applied (GPT, Llama, Gemini, etc.), but some examples of variables that might be adjusted are listed below.

Base Prompt: the fundamental prompt of the agent. It can be used to express the role the agent will be playing (ex, Product Owner, Requirements Engineer, etc.), the tasks it will execute, the expected tone of the responses, the source of information that should be used for the answers, etc.

Generation Prompt: the prompt used to generate user stories during the routine process. This parameter enables the operator (PO, RE, etc.) to adjust the generation prompt based on their needs and usage/reviews. This prompt can be either simple, like: "Please generate user stories from the following conversation", or a more complex one, adding more details and requirements, or even applying prompt engineering techniques.

Number of tokens in response: How many tokens (words, character sets, or combinations of words and punctuations) should the response contain

Temperature: How creative should the agent be (higher value leads to more creativity but increases the risk of hallucinations).

Model Version: the version of the LLM model to be applied (like GPT 3.5, 4, 4o on OpenAI) to process the prompts. This choice might affect the output's speed and quality.

3.1.2 Domain information. Company or customer-specific information relevant to support the agent to elaborate on the user stories. A few examples are listed below, but the application should not be limited to these options:

Customer Information: the contextual information of the customer, like business segment, time in the market, description of roles that may use the system on that specific customer, business value, vision, cultural aspects, potential risk sensitivity (e.g., stock market listed company, financial sector company, etc.);

Quality Criteria: what the operator, team, or software company considers a good user story. Examples: Cohn's Guidelines [19], INVEST [20], CCC [21], 5Cs [22] QUS Framework [23];

Regulations: regulatory norms or standards the software application should comply with. Examples: GDPR, ISO, etc.;

Template: the desired user story template to write the resulting user stories. Examples: Cohn's [19] template: "I as a (role) want (function) so that (business value)";

Technical Requirements: technical constraints that must be considered when writing user stories. Example: Story: "As a user, I want to upload a profile picture so that my profile can have a personalized image."; Acceptance Criteria: "The user can upload a profile picture up to 5MB in size; The user can upload a profile picture in JPEG or PNG formats".

3.1.3 Validation. A validation step should be taken by the end of the setup process to evaluate the LLM agent's responses to known questions based on the domain information loaded. This step is essential to reduce the risk of hallucinations and can be done manually (by a human) or in an automated way.

Human Validation: the operator should ask the LLM agent questions to ensure the domain information and parameters have been loaded according to expectations. Example: "What is Cohn's user story template? Please provide the reference you used to obtain the answer".

Automated Validation: another agent prepared with prompts, questions, and expected responses to validate the primary agent's outputs. Example: Question: "What is Cohn's user story template? Response: "I as a (role) want (function) so that (business value)". Since the response might not be syntactically like what the verification agent is trained with, but semantically similar, an LLM-based agent is required to provide such flexibility. This agent does not aim to replace the human review but to support it by offering an additional validation layer.

3.2 Routine Process

Routine corresponds to the regular application of the approach. It is the process where the user story generation happens. There are two main methods to apply the approach in practice: real-time or post-processed. For real-time mode, the routine process would run cyclically, in parallel with the operator and user conversation. For the post-processed application method, the operator would record the conversation and apply the routine process.

3.2.1 Conversation. Represents the verbal communication between the operator and the user to define the expected software outcome. This conversation could happen either in person or remotely. If the conversation is in person, a means to digitally capture the audio is required. It could be done through any audio recorder for post-processed operations or a live stream for real-time application of the approach. The operator should ask as many questions as needed to support the user in detailing their expectations and capture as many details as possible. In case the operator and the user roles are represented by the same person, they could explain verbally the context, the users' requests, the biggest pains, and expectations of changes on the existing software.

3.2.2 Speech-to-text. The conversation between the operator and user is converted in real-time from voice (speech) to text. A wide variety of tools in the market could support this job. Some video conference tools already have this functionality embedded. This step aims to capture a textual version of the conversation, that is, a transcript, to serve as the basis for the upcoming steps.

3.2.3 User Stories (US) generation. The trained LLM agent processes the conversation in text and is requested with the prompt defined in the parameterization step (3.1.1) to generate the user stories. User stories are generated based on the conversation and the setup's domain information and presented to the operator.

3.2.4 Review. The generated user story is reviewed to ensure it meets the customer's requirements. It is highly recommended to have a human review in this step, where the operator analyses the stories generated and evaluates if they meet the expected quality criteria, regulations, technical limitations, etc., to avoid potential hallucinations, missing requirements, and other mistakes the LLM agent can make.

An additional automatic review can be added to support the human review. Another trained LLM agent could validate the expected requirements from the domain information and the customer conversation. This agent may not be the same as the verification agent. Still, using an agent different from the one used for the stories' generation is recommended to specify different tasks and expectations.

This step could also be an opportunity for the operator to review the resulting user stories with the user, providing an additional layer of validation and reliability. In a real-time application, the user could see the process happening live.

4 Evaluation

An evaluation was conducted to assess the applicability of the proposed approach in real-world software development contexts. It took place in April 2025 and involved two software industry experts in separate sessions, each lasting approximately 60 minutes. An electronic form was used as support material, to guide participants through the evaluation process, formalize their consent to participate in the study, and collect feedback on the use of the proposed approach.

4.1 Evaluation Design

The evaluation was designed to familiarize the participants with the proposed approach, as well as the LLM and speech-to-text technologies available within their organization; explore how these technologies relate to and support the proposed approach, apply the approach to their specific context, and evaluate the experience. The evaluation was organized into three main steps, as described below.

Step 1 - Introduction: The researcher summarized the conceptual proposal of the approach through an instruction booklet containing the approach's concepts and definitions, but no theoretical background or related works. This material was made available to the participant in a section of the electronic form.

Another booklet provided orientation on instantiating the conceptual approach using tools available within the participants' company – Google Gemini Advanced as the private LLM agent and Google Meet as the speech-to-text tool. The booklet correlated each conceptual step of the approach to the controls of the tools. The Google Gemini and Meet Application Instructions Booklet was also made available to the participants through the electronic form. Both booklets are publicly available, and a link is provided in the section "Artifacts Availability".

Step 2 - Application: The participants were requested to apply the approach to a real-world case according to the orientations received in the first step. They could use the support of the instruction booklets. The participants should do the setup with all domain-specific information they would find relevant, explaining the changes they would like to have in the software in a Google Meet transcript, generating and reviewing the corresponding user stories on Gemini Advanced (model 2.0 flash). To simplify the application of the approach on this preliminary study, the operator (who is responsible for generating user stories from the requirements) and the user (who holds and describes the requirements) were the same person. Only human verification and review were performed for the setup and routine processes.

Step 3 - Evaluation: The participants were requested to provide feedback on applying the proposed approach to their context. A questionnaire has been adapted from the Technology Acceptance Model (TAM3) [24]. The objective was to evaluate the approach's Ease of Use (E), Usefulness (U), Output Quality (Q), and Anticipated Enjoyment (F). A Likert scale objective response was requested with 5 levels, varying from Strongly Disagree to Strongly Agree for E and U. For Q and F, the scale was from Extremely Unlikely to Extremely Likely. For each subsection of the evaluation form, the participant was also requested to provide their confidence level (C) in the ratings made, within 5 responses varying from Not at all confident to Completely Confident. An open-ended question was added at the end of the form for any additional comments or suggestions.

4.2 Evaluation Application

Two industry experts participated in the evaluation. The two participants work for a large global enterprise that develops solutions across a wide range of sectors, including construction, transportation, forestry, agriculture, military, geospatial and others. Both professionals are currently allocated to the transportation segment, which focuses on developing solutions to support logistics activities, especially those involving trucks.

Participant A currently holds the job title of Product Owner (PO), which, in the context of its company, is usually responsible for the interface between the internal product team (product manager, user experience designers, engineering management, etc.) and the development team. The PO is responsible for eliciting and documenting requirements (product discovery), prioritizing the delivery backlog, and creating user stories. This expert is based in Brazil, has eight years of experience in software engineering, and has been working with agile methods and user stories for the past four years. The real-world scenario applied in Participant A's evaluation involved an existing Internet-of-Things (IoT) application. The goal was to improve its usability for both web and mobile users.

Participant B currently works as a Product Manager (PM). In their current company, they are responsible for the external interface of a product, that is, other products, end-users, customers, marketing, business, sales, design, support, and other areas. In this company, these are the professionals responsible for gathering and defining the requirements, and consequently, the direction to which the software product will evolve through a roadmap. On larger teams in this organization (approximately 25 members), it is not usual for the PM to interact directly with the development team, manage the backlog, or create user stories. However, for smaller products, the PM is also the PO, responsible for eliciting and documenting requirements, managing the backlog, and creating user stories. The latter is the case of Participant B. This professional is based in the USA and has 16 years of experience in software engineering, with 13 years of experience working with agile methods, consuming user stories, and 9 years of experience writing these. Participant B applied the approach to increment an existing system with a notification functionality in a web portal.

The participants were requested to apply the approach to a problem in their context. Participant A has configured a Gemini agent with a tailor-made base prompt, mentioning the agent's role (PO), the company's product for which the agent will work on, the task it will be responsible for, in this case to generate user stories, and describing the persona of the end user for which the stories would be created for. The prompt will not be shared as part of this research because it involves company proprietary and confidential information. For the domain information, the participant included a broader description of the persona who will use the feature, a copy of the product's public website, and a document with notes taken from conversations with stakeholders. The requests in the verification step were: "Can you read the documents attached?", "Tell me more about the persona", "Tell me about the product".

Participant A created a Google Meet conference with transcription enabled, describing the changes they would like to have on their systems. The transcripts were added to the Gemini agent, with the generation prompt: "Please generate the user stories of this new feature, ready to be shared with the development team, with the goal to estimate and develop it". Five user stories were generated. The participant provided feedback to the agent in the review step: to split one of the stories, providing more specific details on how the change should be implemented, and to remove a story, because it was a feature the system already had but was not provided as part of the agent's setup.

Participant B has used a base prompt adapted from Zhang et al. [11] provided as an example in the Gemini and Meet instructions booklet. The corresponding product and team's knowledge base (wiki) was provided as domain information to the agent, as well as the personas that may be interacting with the product. A US-specific regulation for truck drivers was also provided as context. For the verification step, the participant has asked two questions: "How many hours of driving does a driver have in a day?" and "Who is the manager for the product in context?". The agent has passed the verification, responding to the questions correctly.

Participant B went through the same process of speaking about the requirements in Google Meet and collecting the transcription. The generation prompt was "Generate user stories based upon this conversation about a new feature". Four user stories were generated. However, when the participant requested changes and added more business information, the agent crashed and provided random responses (hallucinations). The team's wiki was removed from the agent's knowledge base, and stories were generated once again. This time, seven stories were created.

A single review was requested to have the third story broken into smaller stories, resulting in 8 user stories. Unfortunately, none of the resulting generated user stories can be shared in this paper since they involve company specific private content. Both participants were requested to evaluate their experience using the electronic form.

4.3 Preliminary Results

The preliminary evaluation results are shown in Table 1. No strong disagreement responses were obtained for Ease of Use and Usefulness, nor were they unlikely or extremely unlikely for the Output Quality and Enjoyment. On the confidence level questions, all ratings provided were "Very Confident" or above.

Both participants agreed that the approach is easy to learn to operate, to get the approach to do what they needed to do and strongly agreed that the interaction between them and the approach would be clear and understandable. There was no consensus on flexibility: Participant A agreed with the approach being flexible to interact with, while Participant B was neutral. Both participants strongly agreed they found the approach easy to use.

From the perspective of Usefulness, both agreed that the approach would enable them to accomplish tasks more quickly, using the approach would increase their productivity, and make it easier to have their jobs done. Participant B provided a single response of disagreement throughout the whole form on enhancing their effectiveness on the job. Participant A provided a neutral response to this question. An opposite response was collected when the subjects were asked if using the approach would improve their job performance: Participant A was still neutral while Participant B agreed. In general, both participants agreed that the approach would be helpful in their jobs, with Participant A providing a "Strongly Agree" response.

Participants A and B considered that the Output Quality was likely high. Regarding whether the user stories they would make using the approach would be high quality, Participant A responded as "Neither", while Participant B considered it "Extremely likely". Both participants mentioned they find the approach enjoyable to use.

Regarding open feedback, both participants have reinforced the novelty of the approach to their professional context. They mention that the approach felt familiar since they had used LLM-powered tools before but had never trained an agent with domain-specific information or generated user stories directly from conversations. They mentioned that participating in the evaluation has revealed a new approach to performing this and other jobs in their professional routine, like preliminary estimation or breaking down stories into smaller and more technically detailed tasks.

It is important to highlight that neither of the participants had great expertise working with LLMs, reinforced by the simple prompts used by them.

The participants successfully applied the approach to their context, generating user stories that they consider of good quality. They mentioned that the initial stories created using the approach were very similar to the ones they would create manually. In this evaluation, no significant adjustments were made to the stories during the review processes, only punctual tweaks such as adding more context to the agent or splitting a story into two.

Question	A	B
E1. Learning to operate the approach would be easy for me.	Agree	Agree
E2. I would find it easy to get the approach to do what I want it to do.	Agree	Agree
E3. My interaction with the approach would be clear and understandable.	Strongly Agree	Strongly Agree
E4. I would find the approach to be flexible to interact with.	Agree	Neutral
E5. It would be easy for me to become skillful at using the approach.	Strongly Agree	Strongly Agree
E6. I would find the approach easy to use.	Strongly Agree	Strongly Agree
C1. How confident are you in the ratings you have made on this page? (Ease of Use)	Very Confident	Very Confident
U1. Using the approach in my job would enable me to accomplish tasks more quickly.	Agree	Agree
U2. Using this approach would improve my job performance.	Neutral	Agree
U3. Using the approach in my job would increase my productivity.	Agree	Agree
U4. Using this approach would enhance my effectiveness on the job.	Neutral	Disagree
U5. Using this approach would make it easier to do my job.	Agree	Agree
U6. I would find this approach useful in my job.	Strongly Agree	Agree
C2. How confident are you in the ratings you have made on this page? (Usefulness)	Very Confident	Very Confident
Q1. Assuming I were to use the approach, the quality of the output I would get would be high.	Quite Likely	Quite Likely
Q2. The user stories I would make using the approach would be of high quality.	Neither	Extremely Likely
F1. I would find using the approach to be enjoyable.	Quite Likely	Extremely Likely
C3. How confident are you in the ratings you have made on this page?	Very Confident	Completely Confident

Table 1: Evaluation Questions and Responses.

5 Conclusion

This research presents an approach to generating high-quality user stories using LLMs. The proposal differs from [11], [12], [25] in three main aspects. It eliminates the need for pre-existing user stories or requirements documents, as it generates user stories directly from conversations, the most common source of requirements during the elicitation process. This contributes to reducing the manual effort required for documenting and summarizing requirements, saving time for practitioners.

Second, the approach introduces both manual and automated verification and review steps to supervise the outputs generated by the LLM agent. Third, it enables the inclusion of domain-specific knowledge into the process. These aspects enhance the precision of the AI assistant and help mitigate hallucinations [13], [14].

The proposed approach contributes to the quality of user stories in multiple ways. By leveraging LLM to generate user stories directly from user conversations, it reduces dependency on human effort, allowing the operator to focus on optimizing the details rather than writing the story from scratch. Additionally, it supports the application and evaluation of quality criteria throughout the requirements engineering process.

The preliminary evaluation conducted in this study reveals the effectiveness of the approach in industrial settings. The participating experts provided highly positive feedback. They evaluated the approach as easy to use, useful, enjoyable, and capable of generating high-quality user stories with significant confidence.

One limitation encountered during the evaluation involved slowness and hallucinations when using the Gemini 2.0 Flash model to process a large file. This issue points to limitations of the specific LLM version used and suggests that future work should explore alternative versions of Gemini or other commercial and open-source LLMs for broader evaluation.

It is also worth noting that only the fundamental user story generation agent was employed during the preliminary evaluation for the sake of simplicity. Future research will explore multi-agent interactions to support not only generation, but also validation and review activities within the proposed approach.

Despite the promising preliminary results, the approach has not yet undergone extensive validation in real-world applications. The small number of participants limits the generalizability of the findings. Further empirical studies are necessary to refine the approach and evaluate its applicability across diverse contexts, teams, and domains. Ongoing research aims to address these limitations and strengthen the overall robustness of the solution.

ARTIFACT AVAILABILITY

The approach instructions and Google Gemini / Meet Instructions booklets mentioned in this paper are publicly available at: https://doi.org/10.6084/m9.figshare.29118686.

REFERENCES

[1] L. Cao and B. Ramesh, "Agile requirements engineering practices: An empirical study," IEEE Softw, vol. 25, no. 1, 2008, doi: 10.1109/MS.2008.1.

[2] M. Kassab, "The changing landscape of requirements engineering practices over the past decade," in 5th International Workshop on Empirical Requirements Engineering, EmpiRE 2015 - Proceedings, 2016. doi: 10.1109/EmpiRE.2015.7431299.

[3] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Improving agile requirements: the Quality User Story framework and tool," Requir Eng, vol. 21, no. 3, pp. 383–403, Sep. 2016, doi: 10.1007/s00766-016-0250-x.

[4] X. Wang, L. Zhao, Y. Wang, and J. Sun, "The Role of Requirements Engineering Practices in Agile Development: An Empirical Study," in Communications in Computer and Information Science, 2014. doi: 10.1007/978-3-662-43610-3_15.

[5] A. M. Moreno and A. Yagüe, "Agile user stories enriched with usability," in Lecture Notes in Business Information Processing, 2012. doi: 10.1007/978-3-642-30350-0_12.

[6] J. Choma, L. A. M. Zaina, and D. Beraldo, "UserX story: Incorporating UX aspects into user stories elaboration," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016. doi: 10.1007/978-3-319-39510-4_13.

[7] H. Ordóñez, A. F. E. Villada, D. L. V. Vanegas, C. Cobos, A. Ordóñez, and R. Segovia, "An impact study of business process models for requirements elicitation in XP," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015. doi: 10.1007/978-3-319-21404-7_22.

[8] M. Trkman, J. Mendling, and M. Krisper, "Using business process models to better understand the dependencies among user stories," Inf Softw Technol, vol. 71, 2016, doi: 10.1016/j.infsof.2015.10.006.

[9] Y. Wautelet, S. Heng, M. Kolp, and I. Mirbel, "Unifying and Extending User Story Models," in Lecture Notes in Computer Science, 2014, pp. 211–225. doi: 10.1007/978-3-319-07881-6_15.

[10] A. R. Amna and G. Poels, "Systematic Literature Mapping of User Story Research," IEEE Access, vol. 10, pp. 51723–51746, 2022, doi: 10.1109/ACCESS.2022.3173745.

[11] Z. Zhang, M. Rayhan, T. Herda, M. Goisauf, and P. Abrahamsson, "LLM-based agents for automating the enhancement of user story quality: An early report," in International Conference on Agile Software Development, 2024, pp. 117–126.

[12] D. V. Rodriguez et al., "Leveraging Generative AI Tools to Support the Development of Digital Solutions in Health Care Research: Case Study," JMIR Hum Factors, vol. 11, no. 1, 2024, doi: 10.2196/52885.

[13] C. N. Coelho Jr et al., "Effort and Size Estimation in Software Projects with Large Language Model-based Intelligent Interfaces," 2024. [Online]. Available: https://arxiv.org/abs/2402.07158

[14] K. Ronanki, C. Berger, and J. Horkoff, "Investigating ChatGPT's Potential to Assist in Requirements Elicitation Processes," in Proceedings - 2023 49th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2023, 2023. doi: 10.1109/SEAA60479.2023.00061.

[15] T. Johann, C. Stanik, A. M. B. Alizadeh, and W. Maalej, "SAFE: A Simple Approach for Feature Extraction from App Descriptions and App Reviews," in Proceedings - 2017 IEEE 25th International Requirements Engineering Conference, RE 2017, 2017. doi: 10.1109/RE.2017.71.

[16] M. Arias, A. Buccella, and A. Cechich, "A Framework for Managing Requirements of Software Product Lines," Electron Notes Theor Comput Sci, vol. 339, 2018, doi: 10.1016/j.entcs.2018.06.002.

[17] N. H. Bakar, Z. M. Kasirun, N. Salleh, and H. A. Jalab, "Extracting features from online software reviews to aid requirements reuse," Applied Soft Computing Journal, vol. 49, 2016, doi: 10.1016/j.asoc.2016.07.048.

[18] I. K. Raharjana, D. Siahaan, and C. Fatichah, "User Stories and Natural Language Processing: A Systematic Literature Review," IEEE Access, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3070606.

[19] M. Cohn, User Stories Applied: For Agile Software Development (Addison Wesley Signature Series), vol. 1, no. 0. 2004.

[20] B. Wake, "INVEST in Good Stories, and SMART Tasks," XP123: Exploring Extreme Programming. Accessed: Oct. 09, 2024. [Online]. Available: https://xp123. com/article-s/invest-in-good-stories-andsmart-tasks

[21] R. Jeffries, "Essential XP: Card, conversation, confirmation," Ronjeffries.Com. Accessed: Oct. 09, 2024. [Online]. Available: https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/

[22] J. Patton, User Story Mapping: Discover the Whole Story, Build the Right Product, 1st edition. O'Reilly Media, 2014.

[23] G. Lucassen, F. Dalpiaz, J. M. E. M. Van Der Werf, and S. Brinkkemper, "Forging high-quality User Stories: Towards a discipline for Agile Requirements," in 2015 IEEE 23rd International Requirements Engineering Conference, RE 2015 - Proceedings, 2015. doi: 10.1109/RE.2015.7320415.

[24] F. D. Davis, "A technology acceptance model for empirically testing new end-user information systems: Theory and results," Management, vol. Ph.D., 1985, doi: oclc/56932490.

[25] T. Rahman and Y. Zhu, "Automated User Story Generation with Test Case Specification Using Large Language Model," Apr. 2024, Accessed: Sep. 30, 2024. [Online]. Available: https://arxiv.org/abs/2404.01558v1