Prompt Engineering Language in The Agile Product Backlog Refinement Process

Pawel Paterek
AGH University of Krakow, Cracow, Poland

pawel.paterek@gmail.com

Abstract: Contemporary advanced business services and products increasingly incorporate artificial intelligence components, such as chatbots and generative AI, across various domains. These are delivered through dedicated, complex, innovative software programs and projects initiated by virtually all industry sectors. Complex project management include challenges such as: the complexity of product backlogs with a number of requirements, highly dynamic changes in customer expectations impacting product backlog quality and requirements engineering, labour shortages, advanced tool adoption and automation, predictability of deliveries, insufficient transparency of processes applied to product backlog management, communication barriers between business and project teams. The primary objective of this paper is to address a key research gap related to the insufficient quality of product backlog management in complex agile software project environments. The paper addresses the research question regarding the potential application of chatbots and generative AI as a methodology to conduct reliable agile product backlog evaluation and subsequently enhance refinement processes through dedicated Prompt Engineering Language (PEL). This paper emphasizes the importance of a structured description of product backlog items and its impact on the overall quality of agile product backlog and delivered software products. Following the literature review, the author's empirical research presents a detailed analysis of the research gap and focuses on applying chatbot and generative AI solutions to evaluate agile product backlog items and to improve related agile refinement processes. Research results demonstrate that agile product backlog refinement processes can be supported by chatbots and generative AI utilizing dedicated Prompt Engineering Language (PEL). These tools are not designed to create business value directly but rather enhance the efficiency and automation of product backlog management processes to respond rapidly to stakeholder expectations within agile environments, ultimately achieving superior project outcomes. Nevertheless, numerous challenges must be addressed, particularly related to AI governance and compliance with numerous policies including data privacy, intellectual property, legal, security and internal ones.

Keywords: Product backlog quality, Product backlog refinement, Prompt engineering language (PEL), Agile software development, Agile project management, Requirement engineering

1. Introduction

Development of advanced products and services raises many challenges in software development projects related to the vast number of changing requirements and complexity of product backlog management (Lucassen et al., 2016; Trkman et al., 2019). Customer demands and competitive market pressures require rapid deliveries, often without extensive project planning, but focusing on modern agile project management approaches. Highly dynamic changes in customer specifications and expectations impact the quality of requirements engineering (Nistala et al., 2022; Lu et al., 2023).

Increasingly, complex projects seek advanced tools support and artificial intelligence, such as chatbots and generative AI (Marczak-Czajka and Cleland-Huang, 2023; Ronanki, Cabrero-Daniel and Berger, 2023). Reliable and high-quality AI chatbot results require prompt creation with sufficiently good structure (White et al., 2023) to support agile product backlog refinement in a predictable manner and therefore to offer a practical tool for software organisations facing delivery pressures, quality issues, and transparency problems. PEL can be understood as one of the methods to create reliable, deterministic and high-quality results important to practitioners seeking actionable AI implementation strategies.

Recent research studies have highlighted the importance of AI tools to address those challenges, but practical and scalable adoption will require addressing non-standard requirements engineering, domain-specific vocabulary, and scalability issues in complex projects with numerous dependencies (Nistala et al., 2022).

The primary goal of the empirical research in this paper is to respond to the research question about the potential application of chatbots and generative AI as a methodology to conduct reliable agile product backlog evaluation and subsequently enhance refinement processes through dedicated prompt (PEL). The aim of the paper is also to expose the importance of structured description of user stories and its impact on the overall quality of agile product backlogs and delivered software products.

The research results showed that agile product backlog refinement processes can be supported by chatbots and generative AI with dedicated and carefully structured prompts. These tools are not designed to create business

value directly or fully automate the process, but to enhance product backlog management to respond rapidly to stakeholder expectations.

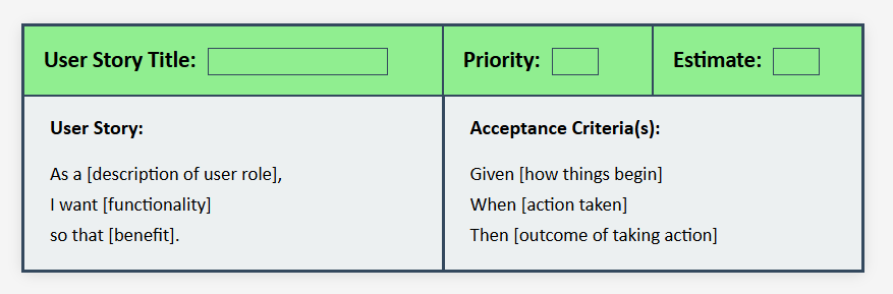
A literature review and prompt engineering using AI chatbots were applied to conduct this research study and to verify research hypotheses. The confusion matrix with a set of classification metrics was applied to evaluate the performance of the final system prompt. The basic limitation of the research study is the limited access to a larger number of production backlog samples. As a future research opportunity, it might be beneficial to repeat the similar analysis with different prompt patterns as well as with other AI chatbots to compare results.

The structure of the paper is as follows: section two presents theoretical underpinnings, section three presents the methodology approach, section four presents empirical research results, and finally section five contains a discussion of the results and conclusions.

2. Theoretical Underpinnings

2.1 The User Story as a Functional Requirement in Agile Software Development

User stories are one of the most commonly applied methods for describing the functional requirements of products in agile software development projects (Lucassen et al., 2016; Levy et al., 2022). The user story is described in natural business language to define end-user needs in the most concise way (Levy et al., 2022), to establish priorities for those needs and to facilitate communication and collaboration between end-users, business teams, and development teams (Hendriana, Raharjo and Fitriani, 2023). A user story should unambiguously reflect the needs of end-users in terms of the business value they expect in the final product or service functionality (Dimitrijevic, Jovanovic and Devedzic, 2015).



Source: Made by the author

Figure 1: User Story Template

A typical user story has a clearly defined template (Figure 1): *"As a <user role> I want to achieve <goal/functionality> so that I can <(receive) benefit>"*, which requires specification of the user role, action in the form of functionality or aim for it, and the expected benefit that this action is intended to deliver (Cohn, 2004). Furthermore, user stories are associated with one or more acceptance criteria, namely a list of activities that allow for verification of whether the user story delivered by the development team is ready from the business perspective, and take the form: *"Given – defines initial conditions, When – describes the action performed for the given functionality, Then – specifies the expected result"* (Cohn, 2004). The fundamental properties of a user story are: business value, priority, and effort estimate (Cohn, 2004).

The principal processes associated with the user story lifecycle are: identification and creation, refinement – continuous improvement as further details emerge, and management within agile planning activities (Dimitrijevic, Jovanovic and Devedzic, 2015; Dam et al., 2019). Enhancement of tools and processes for agile product backlog management is the key challenge in software development projects (Dam et al., 2019).

The quality of user stories is the primary focus of numerous research studies related to agile software development (Hendriana, Raharjo and Fitriani, 2023; Xu et al., 2023), as it relates to the nature of the agile approach applied in complex project domains, with frequent and informal face-to-face communication in highly dynamic environments. Ambiguity and differing interpretations amongst key stakeholders represent one of the principal issues related to user story quality (Dalpiaz et al., 2019; Amna and Poels, 2022), due to the challenge of selecting the appropriate approaches or proposing holistic solutions. User stories are not intended to be detailed requirements specifications, design, or architectural solutions, but rather concise and brief functionality goals (Cohn, 2004).

An important research finding related to user story quality is the implementation of high-quality processes for elicitation and refinement throughout the entire lifecycle, from creation, through iteration planning and execution, until final delivery (Levy et al., 2022; Xu et al., 2023). Although the quality of user stories has been investigated extensively, there remain opened future research areas, including insufficient research studies in practical industry applications, lack of domain industry context-awareness, inadequate quality of supporting tools, diverse specification formats, mutual dependencies, and the search for more effective requirements representation than user stories (Lucassen et al., 2016; Trkman et al., 2019).

2.2 AI Tools Support for User Story Creation and Refinement Process

There exist different approaches for the creation and elicitation of new user stories and their subsequent refinement. The most common approaches follow template-based methods to ensure valid syntax and semantics (Cohn, 2004; Dalpiaz et al., 2019), SMART (Specific, Measurable, Achievable, Relevant, Time-bound), INVEST (Independent, Negotiable, Valuable, Estimable, Small, Testable) criteria (Levy et al., 2022), user feedback incorporated by UX designers (Lu et al., 2023), and test-based approaches focused on acceptance criteria of delivered business value (Cohn, 2004). The primary challenges for this strategy are related to ambiguity and unstructured user stories, mixed functional and non-functional requirements, insufficient domain knowledge, subjective prioritisation, creation of atomic user stories with less complex dependencies, communication amongst different teams, and writing effective user stories (Dalpiaz et al., 2019; Lu et al., 2023).

Following recent research studies, AI tools support (such as Natural Language Processing) and generative AI solutions (such as AI chatbots) are highlighted to address those challenges; however, their adoption will require resolution of several obstacles including document diversity structures, domain-specific vocabulary, adaptation to practical industry applications, and scaling to complex projects with numerous dependencies (Nistala et al., 2022; Marczak-Czajka and Cleland-Huang, 2023; Ronanki, Cabrero-Daniel and Berger, 2023; Siahaan, Raharjana and Fatichah, 2023). Application of ChatGPT has demonstrated the ability to generate contextual human-valued user stories with well-defined product features, to inspire stakeholders to consider and discuss previously unconsidered features, and to generate user stories across all stakeholder roles and features in a systematic manner; however, it requires human oversight for analysis, prioritisation, and contextual adaptation and should not be intended to fully automate product backlog management processes (Marczak-Czajka and Cleland-Huang, 2023; Ronanki, Cabrero-Daniel and Berger, 2023). Major challenges identified in the application of ChatGPT were related to improving performance and quality metrics, complexity of created solutions (including trustworthiness, transparency and explainability), limited diversity and stakeholder perspectives, insufficient input data about the product itself, and difficulty in articulating benefits in user stories (Marczak-Czajka and Cleland-Huang, 2023). Future research in the area of generative AI application should focus on creating customised prompts to provide consistent results (with more deterministic approaches and fewer extrinsic hallucinations), automated pipeline integration for testing at scale with real-world industry products and domain-specific projects, comparison of different AI models and prompting strategies, and evaluation of effectiveness by exploring quality metrics (Ronanki, Cabrero-Daniel and Berger, 2023).

Logistic regression and discriminant analysis allow the use of numerous user story features (often dichotomous variables) to be classified into different improvement groups (Tillmanns and Krafft, 2017) to improve the user story quality during continuous refinement processes. Creating customised prompts to provide more deterministic results with fewer hallucinations may require precise fact-level validation mechanisms for ensuring the granular factual accuracy that customised prompts demand to achieve the fine-grained hallucination detection and correction capabilities essential for maintaining deterministic output consistency across diverse prompt configurations (Sawczyn et al., 2025). Deployment of automated pipelines with AI tool support and generative AI solutions for scalable industry products may require cloud frameworks for scalable and computational demands (i.e., multimodal analysis of speech and video in agentic solutions) and dynamic resource allocation that can be efficiently met through elastic cloud orchestration systems that provide on-demand access to distributed computing resources whilst maintaining cost-effectiveness and operational flexibility (Miroslaw et al., 2015).

IT democratisation enables business users (Citizen Developers) to create semi-professional solutions (such as numerous RPA, Low-Code or No-Code solutions) without IT department involvement, becoming essential due to high IT resource demands (Sobczak, 2018), which may partially cover needs for user story creation and refinement processes. IT democratisation is part of a broader cultural transformation where university research can facilitate knowledge transfer, similar to how entrepreneurial universities drive innovation by combining

resources and knowledge between science and business sectors to create cultural shifts (Sobczak, 2018; Dameri and Demartini, 2020).

The full functional verification of user stories may not be an achievable goal even with improved AI tools applied; however, following Blikle and Chrząstowski-Wachtel (2018, p. 11-15): "*not verification, but specification is the real bottleneck in functional verification*", therefore improving user stories to ensure the compatibility of product specification with the user expectations. A domain-specific language (DSL) as a specification language with concepts, rules, and vocabulary of a particular domain is more tailored to its unique needs and can be considered to solve the challenges related to the appropriate language for user stories.

The literature review demonstrates a significant impact of AI tool application support for user story elicitation and refinement in improving its quality. This important step revealed a high-level research gap related to the poor quality of agile product backlog, continuous processes of creation, management, and refinement, inherently embedded within the agile approach, as well as the application of AI tools for those processes. For this research gap, one can formulate the following research question:

Is the application of chatbots and generative AI as a methodology to conduct reliable product backlog evaluation and subsequently enhance agile product backlog refinement processes possible through dedicated Prompt Engineering Language (PEL)?

Based on the presented studies, we formulate the following hypotheses:

> ***H1:*** *The quality of written agile product backlog items is inadequate for complex software development projects and it can be evaluated using Prompt Engineering Language as a generative AI chatbot method.*
>
> ***H1.1:*** *User roles in user stories of agile product backlog items are missing in at least 20% of the evaluated product backlog.*
>
> ***H1.2:*** *Functionality goals in user stories of agile product backlog items are missing in at least 20% of the evaluated product backlog.*
>
> ***H1.3:*** *Business value goals in user stories of agile product backlog items are missing in at least 20% of the evaluated product backlog.*
>
> ***H1.4:*** *Product descriptions are provided instead of functionality goals in user stories of agile product backlog items in at least 20% of the evaluated product backlog.*
>
> ***H1.5:*** *Technical language is used instead of business language in user stories of agile product backlog items in at least 20% of the evaluated product backlog.*
>
> ***H2:*** *User story acceptance criteria for agile product backlog items can be generated based on high-quality user stories according to acceptance criteria templates using Prompt Engineering Language as a generative AI chatbot method.*

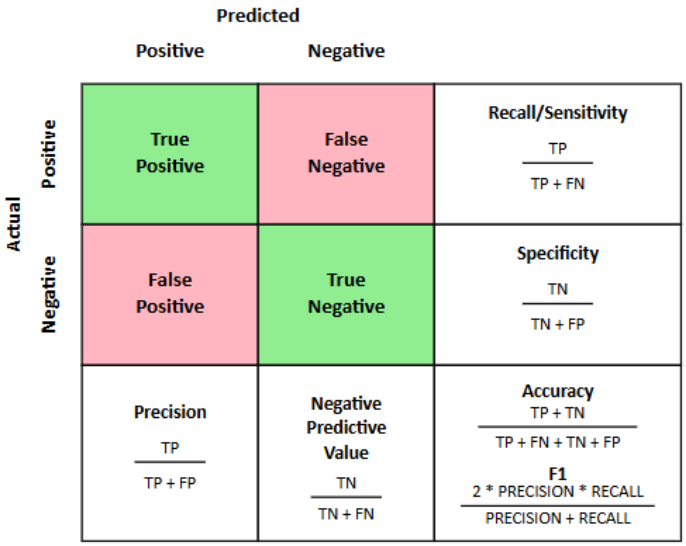
3. Methodology

A prompt is a set of natural language instructions given to a large language model (LLM) such as ChatGPT to guide its behaviour, customise its output, and program it to perform specific tasks. Prompt engineering is the systematic approach of designing and structuring these prompts to effectively communicate with LLMs and achieve desired outcomes (White et al., 2023; White et al., 2024).

Effective prompts should begin with conversational scoping statements such as "*from now on*" or "*act as X*" to establish clear context, followed by fundamental contextual statements that convey the core ideas and constraints. These should include conditional statements that specify when particular rules or behaviours should be applied (White et al., 2023). The most effective approach involves combining multiple patterns for enhanced results, starting with broad prompts and then incrementally narrowing them down with additional constraints as needed (White et al., 2024). The outcome depends on providing concrete examples when defining custom properties, using well-known terminology that the LLM was likely trained on, and iteratively refining prompts based on the quality of received outputs (White et al., 2023). Customised prompts can direct LLMs to perform requirements-related tasks such as analysing stakeholder needs, generating requirement specifications, or validating requirement completeness without requiring full model retraining, but with an LLM fine-tuning process on earlier pre-trained LLMs or with application of different mechanisms for reliable hallucination detection (Sawczyn et al., 2025).

There exist groups of patterns that fall into several major categories, addressing different aspects of LLM interaction. **Input Semantics** control group patterns are aimed at understanding input, **Output Customisation** group patterns focus on types, formats, structure of the generated output, **Error Identification** categories focus on identifying and resolving errors in the output, **Prompt Improvement** categories focus on improving the quality and accuracy of the input and output by requesting verifiable facts, explanations of reasoning, and better question formulations, **Interaction** group patterns change the conversation dynamics through interactions with users, letting the LLM ask questions, assigning specific roles, or breaking complex problems into manageable sub-questions, and **Context Control** groups focus on controlling the contextual information (White et al., 2023). For software development engineering specifically, one may be interested in pattern groups such as **Requirements Elicitation**, System Design and Simulation, Code Quality or Refactoring (White et al., 2024).

The confusion matrix is a methodology to evaluate the performance of logistic regression and classification machine learning models (Tillmanns and Krafft, 2017) as well as advanced neural networks such as those applied in AI chatbots. The confusion matrix indicates the classifier's success by providing a simple matrix table with four combinations of predicted and actual values that helps visualise whether the model is "confused" in distinguishing between two classes (Gusarova, 2023). The summary and formulas of key metrics based on the confusion matrix for classification machine learning models are presented in Figure 2. The precision metric is most appropriate when minimising false positives is critical, for example, in loan approvals where one wants to be confident about predicted positive cases, even if some true positives are missed (Gusarova, 2023). Recall is most appropriate when finding all positive cases is the priority and some false positives can be tolerated, such as medical diagnosis for serious diseases where missing a true positive case is more costly than having false alarms (Gusarova, 2023). F1 Score is used when there is a need to balance both precision and recall, especially in cases where the dataset is imbalanced or when both false positives and false negatives have significant costs that need to be optimised together (Gusarova, 2023).



Source: made by the author based on M. Gusarova (2023)

Figure 2: Confusion matrix with classification metrics

The following research study analysed business requirements originating from two datasets of user stories. The first dataset comprised 100 user stories derived from an actual IT project, representing a random sample from a larger product backlog managed in Atlassian JIRA software. The second dataset consisted of 100 user stories originating from workshops conducted with students during postgraduate studies classes, describing the functionality of fictional products devised for exercises in creating business requirements in the form of user stories. Anthropic Claude Sonnet 4 AI chat was used as a generative AI tool to conduct this research study with an example system prompt created by the author. Finally, results gathered with the chat were compared with manual assessment (Ground-Truth) by using a confusion matrix with classification metrics.

4. Results

4.1 User Story Analysis System Prompt

The iterative empirical research process combined with experiments on recommended prompt engineering language patterns (White et al., 2023; White et al., 2024) allowed for the creation of the system prompt executed in Anthropic Claude Sonnet 4 AI chat to conduct product backlog evaluation and refinement for two explored datasets of user stories. The system prompt applied for user story evaluation and refinement:

You are an experienced Agile Product Owner with 20 years of experience in Agile Project Management. A model user story should be a clear and concise sentence following the User Story Standard Format and should not be too technical a product description but rather an expected functionality to reach user goal/benefit and possibly understood with business language by both Development Team and Product Owner and Stakeholders.

Analyse each user story against the standard user story format and provide a summary table.

User Story Standard Format: "As a [persona], I [want to], [so that]."

Where:

- Persona: Who we're building this for (specific user type/role)
- Want to: User's intent/goal (implementation-free description)
- So that: Overall benefit/bigger picture they're trying to achieve

Examples:

- As Max, I want to invite my friends, so we can enjoy this service together.
- As a manager, I want to understand my colleagues' progress, so I can better report our success and failures.

Task: Review each user story and indicate (Y/N) if it contains each required component. Then refine each user story to follow the standard format while keeping the idea of task as close as possible to be compliant with user story standard template and using a business language focused on expected functionality and goal by users rather than product description. Create acceptance criteria using Given/When/Then format for each user story. Present results in a table format with columns: User Story | User Role | Functionality | Business Value | Product Description | Business Language | User Story Refined | Acceptance Criteria

Where:

- Product Description indicates: Y = too much product description, N = focused on functionality to achieve rather than product description.
- Business Language indicates: Y = uses business language as expected, N = uses more technical than business language.
- User Story Refined: Provide refined version following standard format. User stories requiring refinement are those with at least one N in columns 2, 3, 4, 6 or Y in column 5. Do not create artificial information - use placeholders like <user role> for missing information that Product Owner needs to complete.
- Acceptance Criteria: Create using Given/When/Then format following template: "Scenario: (explain scenario). Given (how things begin), when (action taken), then (outcome of taking action)." Focus on testable conditions and user experience rather than technical implementation.

The results of the user story analysis are presented in Table 1, which contains the percentage of requirements fulfilling the examined criterion related to our research hypotheses, separately for each dataset (IT Project and Workshops). The results are provided separately for manual human analysis (Ground-Truth) and for analysis using Anthropic Chat with our self-developed system prompt.

Table 1: Summary results of user stories analysis (CHT – Chat, GT – Ground-Truth)

Dataset	Method	Dataset Size	User Role [H1.1]	Functionality [H1.2]	Business Value [H1.3]	Product Description [H1.4]	Business Language [H1.5]
IT Project	GT	100	73%	82%	32%	49%	70%
IT Project	CHT	100	73%	69%	13%	11%	20%
Workshop	GT	100	100%	100%	65%	0%	100%
Workshop	CHT	100	89%	100%	61%	4%	100%

Source: Made by the author

In order to verify the research hypotheses, the focus was placed on real IT project results (refer to rows 1 and 2 in Table 1), whilst the workshop results serve more to confirm the possibility of improving user story quality once appropriate training is established within on-the-job processes. User roles were missing in 27% of user stories (**confirmed H1.1**) and this is relatively easy to improve based on workshop results. Functionality was missing in 18% (**H1.2 was rejected**); however, it was close to the 20% threshold and, based on workshop results, also exposed room for easy improvement. Business value was missing in 68% of the IT project and in 35% of the workshops, which definitely **confirmed H1.3** and exposed the challenge of formulating it. In addition to the three mandatory template items of user stories, excessive product description compared to expected functional requirements was identified in more than half the user stories (51%) – **confirmed H1.4** – and overly technical language instead of business language was identified in 30% of user stories – **confirmed H1.5**; however, both appear easy to refine with proper training or tool support based on workshop results. In summary, although H1.2 is rejected, the overall quality of user stories can be improved, either with human review or dedicated tool support (**confirmed H1**).

Table 2 provides two selected examples of original user stories, one from each dataset, together with the chat output results with refined text of the user story and proposed generated acceptance criteria that allow verification of the second hypothesis. It is important to note that the chat was asked directly in the prompt not to imagine those missing parts, but to highlight them with placeholders such as <user role> or <take action> that the one responsible for product backlog management needs to complete.

Table 2: Examples of refined user stories and generated acceptance criteria

Dataset	Original User Story	Refined User Story	Acceptance Criteria [H2]
IT Project	As a PO need to Validate status change for tickets to be able to resolve the tickets	As a PO, I want to Validate status change for tickets to be able to resolve the tickets, so that **<business value>**	Scenario: ***User achieves the desired functionality***. Given PO needs to **<perform action>**, when they **<take action>**, then **<expected outcome>**
Workshop	As a user of an online bookstore I want to have the possibility to browse titles from a given category	As a user I want to browse titles from a given category so that **<business value>**	Scenario: User browses category titles. Given the user is on the bookstore, when they select a category, then they can view all titles in that category

Source: Made by the author

Examples of refined user stories and generated acceptance criteria (Table 2) allow hypothesis **H2 to be confirmed**, as even a simple AI chat prompt with human language (PEL) can be used to trigger at least the review of those items for potential improvement within the product backlog refinement process.

4.2 User Story Analysis Evaluation Metrics

The summary of classification metrics evaluation is presented in Table 3. All five classification metrics (Accuracy, Precision, Recall, Specificity and F1) for both datasets (IT Project and Workshop) were calculated using the exact formulas as shown in Figure 2, for each of the five evaluated user story properties (User Role, Functionality, Business Value, Product Description and Business Language). Confusion Matrix elements (TP – True Positives, FP – False Positives, TN – True Negatives and FN – False Negatives) used in the classification metrics formulas were calculated based on Predicted Values from Chat (CHT) and Actual Values from Ground-Truth (GT).

Table 3: Summary results of evaluation metrics

Dataset	Metric	User Role [H1.1]	Functionality [H1.2]	Business Value [H1.3]	Product Description [H1.4]	Business Language [H1.5]
IT Project	TP	73	65	12	8	17
IT Project	FP	0	4	1	3	3
IT Project	TN	27	14	67	48	27
IT Project	FN	0	17	20	41	53
IT Project	Accuracy	1.000	0.790	0.790	0.560	0.440
IT Project	Precision	1.000 (+)	0.942 (+)	0.923 (+)	0.727 (+)	0.850 (+)
IT Project	Recall	1.000 (+)	0.793 (+)	0.375 (-)	0.163 (-)	0.243 (-)
IT Project	Specificity	1.000	0.778	0.985	0.941	0.900
IT Project	F1	1.000 (+)	0.861 (+)	0.533 (-)	0.267 (-)	0.378 (-)
Workshop	TP	89	100	61	0	100
Workshop	FP	0	0	0	4	0
Workshop	TN	0	0	35	96	0
Workshop	FN	11	0	4	0	0
Workshop	Accuracy	0.890	1.000	0.960	0.960	1.000
Workshop	Precision	1.000 (+)	1.000 (+)	1.000 (+)	0.000	1.000 (+)
Workshop	Recall	0.890 (+)	1.000 (+)	0.938 (+)	N/A	1.000 (+)
Workshop	Specificity	N/A	N/A	1.000	0.960	N/A
Workshop	F1	0.942 (+)	1.000 (+)	0.968 (+)	N/A	1.000 (+)

Source: Made by the author

Values above 0.7 were taken as reasonably fair and acceptable for this type of research study example, aimed for further improvements in practical implementation. For our imbalanced datasets, accuracy can be misleading and evaluation focus was placed more on precision, recall, and F1. Although most results are good or excellent (0.8-0.9 or higher), the key is to establish benchmarks based on specific use cases, business requirements, and domain standards rather than relying solely on general value thresholds. In the case of this research study, the most important conclusion is that **one can assess in a predictable, measurable and reliable way and compare the different prompts (PEL) and chatbot results**.

Similarly to the previous section, the focus is on real IT project results (refer to Precision – row 6, Recall – row 7 and F1 – row 8 in Table 3), whilst the workshop results serve more to confirm the possibility of improving user story quality with additional training. Very good precision results (above 0.9) provide confidence that it is feasible to identify exact elements from the user story template (User Role, Functionality or Business Value) that require improvement – **confirmed verification of hypotheses H1.1 – H1.5**. Recall and F1 metrics revealed that evaluation of Business Value, Product Description and Business Language of user stories might be more challenging, resulting in more false negatives – this means the production solution should be fine-tuned with a more precise prompt (PEL) to specify how to detect those elements and finally achieve better confidence levels.

5. Discussion and Conclusions

The empirical research results as well as the literature research review confirmed the possibility of AI tool application support for user story elicitation and refinement to improve its quality. The quality of agile product backlogs, as well as continuous processes of creation, management, and refinement within the agile approach, can be enhanced with the application of AI tools, chatbots and generative AI with dedicated Prompt Engineering Language (PEL) allowing reliable results to be received as confirmed by measurable metrics in the presented research results, therefore filling the identified research gap (Marczak-Czajka and Cleland-Huang, 2023; Ronanki, Cabrero-Daniel and Berger, 2023). Both user stories and acceptance criteria can be generated with a high level of quality; however, the final version of PEL should be tuned to the production purpose of a given industry, domain context and scalability requirements (Nistala et al., 2022). Taking these into consideration, it requires

human attention to results review and adaptation as it is not intended to blindly or fully automate product backlog management processes (Siahaan, Raharjana and Fatichah, 2023).

The research findings indicate a noticeable lack of accuracy in user stories derived from the real IT project. This discrepancy is acknowledged with both Ground-Truth data results as well as AI Chat results, exposing two major items that drive good quality user stories with AI tool support – tuning the PEL prompt itself, as mentioned, and what is equally important, user input at a sufficient level of information. In agile software development, the common underlying root causes of missing or insufficient input may be related to the agile approach itself, as it is primarily applied in complex project domains with frequent and informal face-to-face communication, where changes are provided iteratively throughout its lifecycle. Thus, user story creation and refinement are also based on input fed from the product owner and stakeholders in a frequently updated feedback loop (Cohn, 2004; Hendriana, Raharjo and Fitriani, 2023; Xu et al., 2023). This confirms that strong support is required for incorporating AI tools as well as human review within the agile feedback loop process.

Good quality of user stories and engineering requirements is crucial in the entire SDLC (Software Development Life Cycle) process as an input to AI-supported software code generation and AI-supported software code testing, verification and validation (possible future research), because ultimately incorrect stakeholder specifications and requirements will result in a working product or service that is mismatched with expected business value. Here, another future research may evaluate creating domain specification languages with concepts, rules, and vocabulary to fit particular domains and unique contexts of given industry domains to ensure a priori correctness before undertaking subsequent and costly SDLC steps (Miroslaw et al., 2015; Blikle and Chrząstowski-Wachtel, 2018). Once the input is proven, it can be further supported with advanced AI solutions such as machine learning, multimodal generative AI with agentic solutions, as well as IT democratisation approaches with low-code or no-code solutions such as RPA or agentic approaches to allow higher automation levels of software development processes (Tillmanns and Krafft, 2017; Sobczak, 2018). An important future research gap is related to enhancing the customised language prompts (PEL) to provide more deterministic and replicable results with fewer hallucinations (Sawczyn et al., 2025); however, at the same time being simple enough to be understood and developed by domain industry experts, without necessarily being software engineering professionals, IT experts or scientific researchers.

Ethics declaration: Ethical clearance was not required for the research.

AI declaration: Anthropic Claude Sonnet 4 AI chat was used as an AI Generative tool aimed to conduct the research study with the system prompt created by an author (see Appendix). Additionally the same chat was used for initial English proof reading and to create images. AI chat was not used to create the research concept or text of this paper.

References

Amna, A.R. and Poels, G., 2022. Ambiguity in user stories: A systematic literature review. *Information and Software Technology*, 145 (5), pp. 1-14.

Blikle, A. and Chrząstowski-Wachtel, P., 2018. *A denotational engineering of programming languages to make software systems reliable and user manuals clear, complete and unambiguous* [Online]. Research Gate. Available from: http://dx.doi.org/10.13140/RG.2.2.27499.39201/3 [Accessed 6 July 2025].

Cohn, M., 2004. *User stories applied: For agile software development*. Boston: Addison-Wesley.

Dalpiaz, F., van der Schalk, I., Brinkkemper, S., Aydemir, F.B. and Lucassen, G., 2019. Detecting terminological ambiguity in user stories: Tool and experimentation. *Information and Software Technology*, 110 (6), pp. 3-16.

Dam, H.K., Tran, T., Grundy, J., Ghose, A. and Kamei, Y., 2019. Towards effective AI-powered agile project management. *IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results*, Montreal, Canada: ACM, 27-May-2019, pp. 41-44.

Dameri, R.P. and Demartini, P., 2020. Knowledge transfer and translation in cultural ecosystems. *Management Decision*, 58 (9), pp. 1885-1907.

Dimitrijevic, S., Jovanovic, J. and Devedzic, V., 2015. A comparative study of software tools for user story management. *Information and Software Technology*, 57 (1), pp. 352-368.

Gusarova, M., 2023. *Introduction to data science with python: A complete guide to learn data science and machine learning step by step based on a Fintech end-to-end project*. Los Angeles: Independently published.

Hendriana, A., Raharjo, T. and Fitriani, A.N., 2023. Approaches in determining user story quality through requirement elicitation: A systematic literature review. *Indonesian Journal of Computer Science*, 12 (6), pp. 3599-3614.

Levy, Y., Stern, R., Sturm, A., Mordoch, A. and Bitan, Y., 2022. An impact-driven approach to predict user stories instability. *Requirements Engineering*, 27 (2), pp. 1-18.

Lu, D., Yang, Q., Bi, Y. and Tian, P., 2023. Analysis of the importance of user stories in agile development. *11th IPMA Research Resonating with Project Practices*, Nanjing, China: IPMA, 22-23-April-2023, pp. 116-129.

Lucassen, G., Dalpiaz, F., van der Werf, J.M.E.M. and Brinkkemper, S., 2016. Improving agile requirements: the quality user story framework and tool. *Requirements Engineering*, 21 (4), pp. 383-403.

Marczak-Czajka, A. and Cleland-Huang, J., 2023. Using ChatGPT to generate human-value user stories as inspirational triggers. *IEEE 31st International Requirements Engineering Conference Workshops (REW),* Hannover, Germany: IEEE, 04-05-September-2023, pp. 52-61.

Miroslaw, L., Baros, V., Pantic, M. and Nordborg, H., 2015. Unified cloud orchestration framework for elastic high performance computing on Microsoft Azure. *2015 NAFEMS World Congress*, San Diego, US: NAFEMS, 21-24-June-2015, pp. 291-298.

Nistala, P.V., Rajbhoj, A., Kulkarni, V., Soni, S., Nori, K.V. and Reddy, R., 2022. Towards digitalization of requirements: generating context-sensitive user stories from diverse specifications. *Automated Software Engineering*, 29 (1), pp. 26.

Ronanki, K., Cabrero-Daniel, B. and Berger, Ch., 2023. ChatGPT as a tool for user story quality evaluation: Trustworthy out of the box? *Agile Processes in Software Engineering and Extreme Programming – XP 2022 2023 Workshops*, Amsterdam, The Netherlands: Springer, 13-16-June-2023, pp. 173-181.

Sawczyn, A., Binkowski, J., Janiak, D., Gabrys, B. and Kajdanowicz, T., 2025. FactSelfCheck: Fact-Level black-box hallucination detection for LLMs. [Online], *arXiv preprint.* Available from: https://arxiv.org/pdf/2503.17229 [Accessed 6 July 2025].

Siahaan, D., Raharjana, I.K. and Fatichah, C., 2023. User story extraction from natural language for requirements elicitation identify software-related information from online news. *Information and Software Technology*, 158 (6), pp. 17.

Sobczak, A., 2018. Robotic process automation - current state and future directions. *Przegląd Organizacji*, (10), pp. 52-61.

Tillmanns, S. and Krafft, M., 2017. Logistic regression and discriminant analysis. In: C. Homburg, M. Klarmann, and A. Vomberg, eds. *Handbook of market research*. Cham: Springer, pp. 1-39.

Trkman, M., Mendling, J., Trkman, P. and Krisper, M., 2019. Impact of the conceptual model's representation format on identifying and understanding user stories. *Information and Software Technology*, 116 (12), pp. 106-169.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J. and Schmidt, D.C., 2023. A prompt pattern catalog to enhance prompt engineering with ChatGPT. *30th Conference on Pattern Languages of Programs,* Monticello, IL, US: ACM, 22-25-October-2023, pp. 1-31.

White, J., Hays, S., Fu, Q., Spencer-Smith, J. and Schmidt, D.C., 2024. ChatGPT prompt patterns for improving code quality, refactoring, requirements elicitation, and software design. In: A. Nguyen-Duc, P. Abrahamsson, and F. Khomh, eds. *Generative AI for effective software development*. Cham: Springer, pp. 1-14.

Xu, X., Dou, Y., Qian, L., Jiang, J., Yang, K. and Tan, Y., 2023. Quality improvement method for high-end equipment's functional requirements based on user stories. *Advanced Engineering Informatics*, 56 (4), pp. 1-22.