Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Approaches in Determining User Story Quality through Requirement Elicitation: A Systematic Literature Review

Angga Hendriana[1], Teguh Raharjo[2], Anita Nur Fitriani[3]
angga.hendriana@gmail.com, teguhr2000@gmail.com, nurfitriani.anita@gmail.com
[1,2,3]Faculty of Computer Science, Master of Information Technology, University of Indonesia, Jakarta

Article Information

Submitted: 27 Dec 2023
Reviewed: 29 Dec 2023
Accepted : 30 Dec 2023

Keywords

User Story Quality,
Requirement Elicitation,
Systematic Literature Review

Abstract

A survey of 108 agile practitioners revealed that user stories are the most widely used method for capturing requirements. However, user stories can be interpreted differently by different stakeholders, leading to potential misunderstandings within the development team. Additionally, the interconnectedness of user stories poses challenges during the requirement elicitation process. A Systematic Literature Review (SLR) of 27 articles about user story elicitation process were selected and these are examined to determine user story quality. This research will provide a comprehensive summary of user story elicitation approaches and their application in addressing user story quality issues. The study will also offer insights into selecting appropriate approaches for resolving challenges in user story requirement elicitation. Finally, most user story elicitation approach primarily focus on addressing the issue of ambiguity.

A. Introduction

In project management, the methodology that is currently popular in software development is agile method. It enables rapid change adaptation to evolving user needs. User stories becomes a common method for capturing user requirements in software development process. They are particularly well-suited for accommodating rapid changes in user needs [1]. Based on survey conducted to 108 agile practitioners, they were asked about the method used in requirement elicitation process. The results showed that 90% respondents of agile practitioners used user stories to capture the requirements [2].

A user story typically follows a specific template to capture essential information about a software requirement in an agile development context. User stories are typically written in a natural language format that follows a specific structure. They often include information about who wants the functionality, what the functionality is, and why it is needed [3].

Despite the widespread adoption of natural language in user stories, empirical research has identified several potential drawbacks associated with this approach. User stories can result in diverse interpretations among stakeholders, potentially leading to misunderstandings within the development team [4]. Furthermore, the interdependency among user stories poses a set of challenges in the requirement elicitation process through user stories [5]. Therefore, further research is necessary to get more understanding of user story issues and approaches to solve them.

The ISO/IEC/IEEE 29148 standard outlines a set of recommended characteristics for effective software requirements [6]. For instance, a software requirement should possess the following characteristics:

- Completeness: It should thoroughly describe the necessary capabilities, constraints, and quality aspects to fulfill the customer's needs.
- Lack of Ambiguity: The requirement should be interpretable in only one way, leaving no room for multiple interpretations.
- Feasibility: It must be achievable within the constraints of the system, considering real-world limitations.
- Verifiability: The requirement should be capable of being proven or verified to the satisfaction of the customer.
- Accuracy: It should accurately represent the customer's needs, avoiding any misrepresentation.
- Consistency: The requirement should not contradict or interfere with other existing requirements, and a consistent vocabulary should be employed throughout the entire document.
- Comprehensibility: The requirement should be clear and easily understood regarding what is expected by the customer.
- Conformance: The requirement must align with an established standard template and writing style for requirements documentation.

In determining the quality of user stories, organizations can take various approaches that align with their organizational conditions. This research aims to summarize what approaches can be used by organizations to enhance the quality of user stories and the problems they aim to address through these approaches in agile software development. Several studies related to quality user stories have been conducted, such as the one carried out by Yanche [7], who conducted a systematic

literature review related to requirement elicitation in user stories. In his research, he only listed the problems that arise in the creation of user stories and did not mention which approaches were used to address each of these problems. Another study related to quality user stories was also conducted by Indra Kharisma through a systematic literature review [1]. In his research, he focused solely on natural language processing approaches, without discussing any other approaches. In contrast to these two studies, this research will comprehensively summarize the existing approaches and how to use the approach in solving the user story quality issue. This study also provides insights into selecting suitable approaches for resolving issues in user stories requirement elicitation.

B. Literature Review

A) User Stories Quality

Based on the quality user stories framework, there are 13 categories should strive to conform by user story writers [2]. Figure 1 shows the quality user stories diagram. The 13 categories are grouped into three main classifications. The first one is syntactic, it checks the grammar of the user story without considering its meaning. The second one is semantic, it ensures that the user story makes sense and the relationships between its components are clear. For the last one is pragmatic, it assesses how the user story is understood by the people who will read it [8].

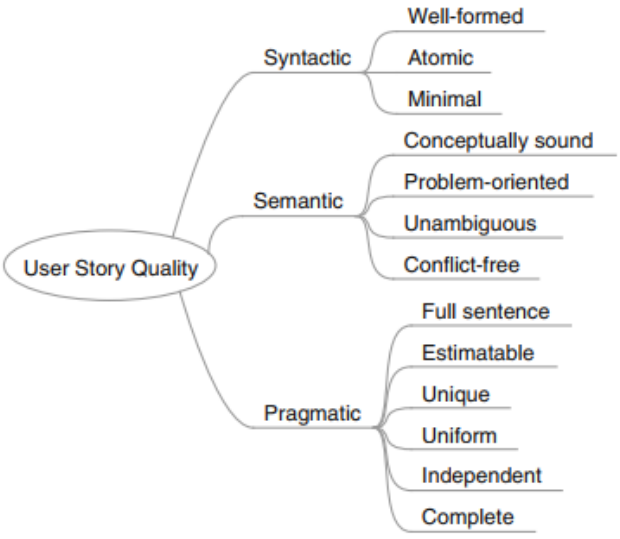

Figure 1. User Story Quality Framework

Explanation of the 13 categories in the user story quality framework is as follows. Well-structured user stories are typically composed of essential components such as the actor (who), the action (what), and the benefit (why) in a clear and organized manner.

1. **Atomic**: focused on a single feature.
2. **Minimal**: containing an actor, a means, and ideally an end.
3. **Conceptually sound**: with a clear distinction between the means and end.
4. **Problem-oriented**: addressing only one issue.
5. **Unambiguous**: easy to understand without extensive interpretation

6. **Conflict-free**: consistent with other user stories.
7. **Full sentences**: user stories should be written as complete sentences, free from typos or grammatical errors.
8. **Estimatable**: each user story should not grow to a size where estimation and planning become exceedingly difficult.
9. **Unique**: not semantically identical to other user stories.
10. **Uniform**: consistent in format with other user stories.
11. **Independent**: non-overlapping in scope.
12. **Complete**: resulting in a feature-complete application when implemented [9].

B) Category Approaches in User Story

Requirements should specify expectations on a software system. In agile software development, requirements can be obtained from user stories [10]. Requirement elicitation approaches in user story can be organized into various categories[11] :

1. **Method**: A systematic procedure in user stories elicitation process.
2. **Taxonomy**: A system for organizing and classifying information to extract requirements from user stories.
3. **Prototype**: An initiate version of a system for gathering requirements from user stories.
4. **Framework**: A structured approach for identifying, gathering, and analyzing user story problem in elicitation process.
5. **Tool**: A software program or application used to gather requirements from user stories.
6. **Technique**: A specific method used to gather requirements from user stories.
7. **Algorithm**: A sequence of steps designed to address the user story identified problem.
8. **Model**: A process used to analyze the purpose of extracting requirements from a user story.
9. **Tutorial**: A step-by-step guidance for eliciting requirements from user story.

C) Systematic Literature Review

Research methodology that involves gathering, evaluating, and analyzing relevant information on particular research is called Systematic Literature Review (SLR). The objective of SLR is to provide a complete understanding of the study, identify areas for further study, and establish a solid foundation for future research. The SLR process begins with defining clear research questions and searching for relevant literature sources. This sources could form journal databases, conferences, and books with specific timeframes and criteria. Then, the identified literature is carefully selected based on predefined criteria, such as the relevance with the study, research methods, and research quality. The chosen literature will get in-depth analysis, including critical reading, data extraction, and information synthesis from each relevant article. Extracted data includes details about the research conducted, methodologies used, key findings, and conclusions drawn. Finally, data from different articles are combined to identify themes or common patterns emerging from the reviewed literature [8].

C. Research Methodology

To identify relevant literature for this research, this study employed a Systematic Literature Review (SLR) approach. Conducting a literature search is necessary to find theories and methods that are relevant to this study. This study adopted the SLR methodology to tackle these research questions by providing an extensive overview of existing research on user story quality evaluation strategies [12]. The systematic literature review (SLR) procedure is segmented into three phases: planning, execution, and reporting. Figure 2 depicts the specifics of these three phases [12].

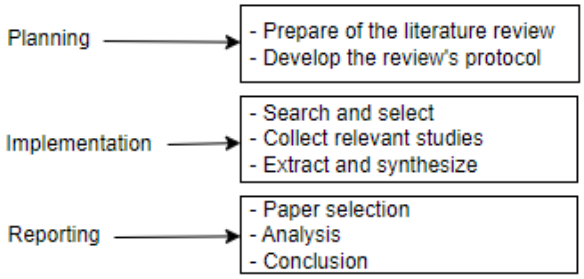

Figure 2. Systematic Literature Review Stages

A) SLR Planning Stage

The first stage in conducting this research using SLR is planning. This step will include the process for developing the review's protocol. This stage is necessary to ensure that the discussions in this study remain focused on the research topic at hand and do not deviate. The research questions to be answered are as follows:

Table 1. Research Question

ID	Questions	Motivation
RQ1	What are the approaches used for handling requirement elicitation problem using user story?	Identify the most used approaches for handling requirement elicitation problem using user story.
RQ2	How to use user story elicitation approach in determining user story quality?	Identify how to use the approache to determine user story quality.

After determining the research questions and objectives, the next step in this study is performing protocol formulations. The keywords used are as follows:

"USER" AND ("STORY" OR "STORIES") AND "QUALITY" AND "REQUIREMENTS" AND ("ENGINEERING" OR "ELICITATION") AND "AGILE" AND ("SOFTWARE" OR "PROJECT") AND "DEVELOPMENT"

In searching for relevant research, the following criteria are applied:

1. The subject area is Computer Science or Information Technology (SI/TI) specific in system or software development.
2. Publication year is within the last 5 years, from 2018 to 2023.
3. Publications are in English language.
4. Full-text publications are available for download.
5. Publications are in the form of journals or conference proceedings.
6. Publications are based on peer-reviewed papers.

B) SLR Implementation Stage

In searching for relevant research references, the researchers utilized journals from Institute of Electrical and Electronics Engineers (IEEE) Xplore Digital Library, ProQuest, Scopus, and ScienceDirect. The keywords used for this search were *"user AND (story OR stories) AND quality AND requirements AND (Engineering OR Elicitation) AND agile AND (software OR project) AND development"*. The results of the search are as shown in Figure 3.

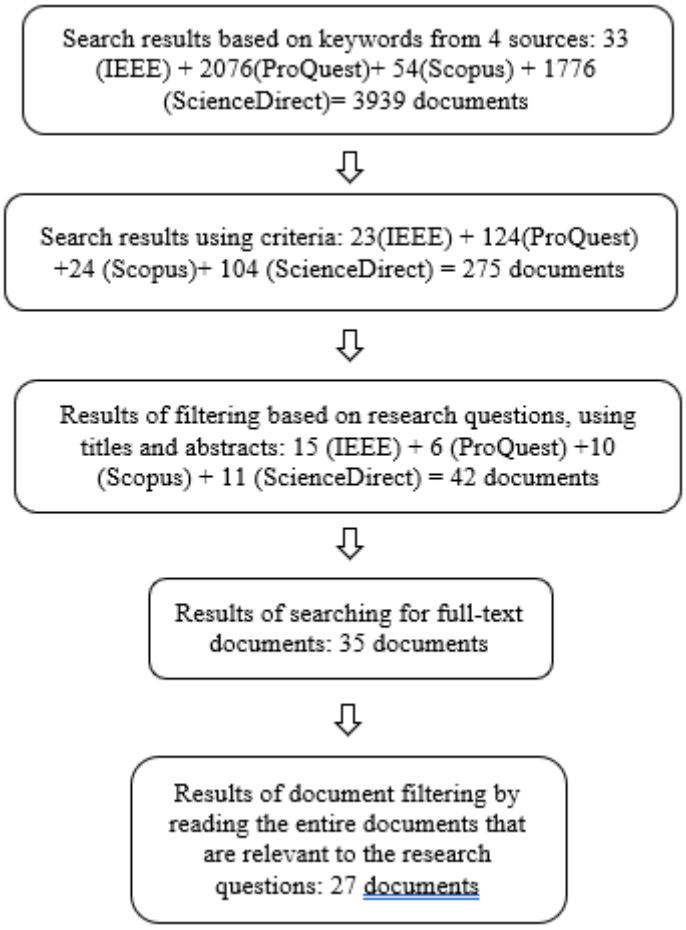

Figure 3. Steps in Systematic Literature Review (SLR) Method

After applying the inclusion and exclusion criteria [13], it was necessary to conduct quality assessment of publications before proceeding with the final selection of studies. To accomplish this, a set of specific questions were formulated in Table 2 to get the quality score used as assessment for the studies.

Table 2. Quality Assessment Question

ID	Questions
QA1	The research discussion is in the scope of requirement elicitation in user story
QA2	The sample size is sufficient to provide meaningful results
QA3	The research employs appropriate methods to address the research questions

Each question has a numerical value ranging from 1 to 3, with 1 indicating low quality, 2 indicating acceptable quality, and 3 indicating high quality. The total possible score for the quality assessment is 9, with a minimum passing quality value set at 4.5. The quality assessment results are summarized in Table 3 below.

Table 3. Quality Assessment Score

Total Score	Reference
9	[14][15][16][17][18][19][20][21][22] [23][24][25][26][27][28][29][30]
8	[31][32][33][34][35][36]
7	[37][38][39][40]
<=6	-

All study received a score higher than 4.5, indicating that these studies are qualified for data extraction and synthesis.

D. Result and Discussion

This stage consists of the results and analysis of selected studies from the SLR. Twenty-seven studies have been chosen to answer the research questions.

A) Approaches in Determining User Story Quality

According to the first research question regarding what requirement engineering methods can be used for managing user stories in Agile Software Project Development.

Similarity Measure Algorithm. This approach is obtained from two literatures [16][18]. The proposed approach effectively utilizes text clustering techniques to identify homogeneous groups of requirements and classify them based on similarity measures. This approach successfully clusters similar user stories [16][18]. The text is enhanced for analysis through the application of multiple preprocessing steps, including stop word removal, tokenization, lemmatization and stemming. These techniques collectively serve to filter and refine the textual requirements.

Recurrent Neural Network Models. This approach is obtained from two literatures [14][39]. This method is employed to identify user stories within vast amounts of data**.** The model used in this method can analyze text from both a grammatical and semantical perspective. It has ability to process not only sequential inputs but also learn its dependencies. After reviewing all the selected paper, this study found several techniques using recurrent neural network including Bi-directional Recurrent Neuronal Networks-Long Short-Term Memory (BRNN-LSTM) [14][39] and Embeddings from Language Models (ELMo) [14].

Online News Feature Extraction Method. This approach is obtained from four literatures [24] [26] [31][34]. This method generally consists of seven systematic

stages, such as pre-process, domain targeting, part-of-speech (POS) chunking, identification aspects of who, what and why, and evaluation [26] [31][34]. It is common to use design science research paradigm to conduct this method [11]. This approach is used in requirement elicitation process to get well-defined user stories from natural language using online news articles, as these sources contain relevant information about stakeholders and their requirements [26][31][34].

User Story Clustering Framework. It is achieved from three literatures [27][29][37]. The proposed method primarily involves three steps: feature extraction, data preprocessing, and clustering. For clustering need preprocessing the data for user stories includes removing stop words, stemming words, and removing duplicates. In feature extraction, could be based on the words used, the structure of the sentences, or the sentiment expressed. In the user story, using this method makes the words can be numerically coded, and it will become the input of the following clustering algorithm. Then the user stories will be divided into several group. There are several frameworks found from literature studies, such as CUSNLP (Cluster User Stories based Natural Language Processing)[27] and REP Automation [29].

Crowd-based Elicitation Method. This approach is obtained from two literatures [15][40]. This method fosters active participation from a wide range of stakeholders. The software product stakeholders consist of both current and future users [40]. CREUS (Crowd-based Elicitation via User Stories) offers an approach to gathering requirements that can be used alongside other elicitation methods. This method proposed four stages: preparation, idea generation, refinement, and implementation [15] [40].

TAGRAM Conceptual Model. This study found one literature about TAGRAM from the selected paper in this study [20]**.** Tagram, a framework for tagging and grammar analysis, aids in the preparation of labeled datasets. User stories are the center of this framework and are enhanced with additional details. These user stories will be further enriched with supplementary information. The framework provides functionalities for checking user story quality, managing projects, defining tagging schemes, and assigning tagging algorithms [20].

Ontology Assessment Technique. This approach is obtained from two literatures [17][33]. The ontology assessment is design to help evaluating the user story's quality, and it will guide development teams throughout the user story's creation, analysis, and development phases. This technique consists of the following steps: build the glossary of terms, build the ontology, define the relationships, and make the concepts' dictionary and the ontology's evaluation [17][33].

Natural Language Processing (NLP) technique. This approach is obtained from five literatures [19][28][32] [35][38]. NLP can be employed to process, extract, or examine user story data. In software engineering, NLP can support for managing software requirements [19], identifying actors and actions within requirement documents [28][35], extracting software features [32], and accelerate the software requirements process [38]. One tool that can be utilized in NLP is SpaCy [19][28][32][35].

Artisan Quality User Story Tool (AQUSA). This approach is obtained from one literature [21]**.** Basically, the using of AQUSA is combined with Quality User Story (QUS) Framework. This tool can automatically evaluate the quality of User Stories

[21]. It aims to achieve a near-perfect recall rate. However, to achieve this ambitious goal, not all criteria are considered by the tool. Semantic criteria, which assess the meaning of User Stories, are excluded [21].

User Story Prioritizing Method. This approach is obtained from two literatures [22][30], aims to enhance software quality and maintain its market value. User story prioritization is applicable across various project categories. Drawing upon these two studies, the prioritization process employs a hybrid approach combining elements of both risk prioritization [30] and hybrid prioritization. This hybrid approach incorporates risk assessment to identify and address potential challenges early on, while also considering user value and business objective [22][30]. Several techniques have been identified through literature reviews to facilitate user story prioritization. Among these techniques, the Analytic Hierarchy Process (AHP) [30].

Machine Learning Classifier. This approach is obtained from one literature [23]. The machine learning algorithms are utilized to automate the prediction accuracy of two metrics, Testable and Valuable. The machine learning approach is widely used to predict or assign class labels to user stories, and the quality of user stories can be determined efficiently. There are several machine learning classifier found from literature studies, such as Support Vector Machines (Linear SVC), Multinomial Naive Bayes (N.B), Logistic Regression (LR), Decision Tree (D.T), and K-Nearest Neighbor (KNN) [23].

Prototyping and Enriched User Stories. This approach is obtained from one literature [36]. This is a combination method to improve the quality agile requirements engineering through user story. By employing this method, agile projects can minimize misunderstandings and rework, ensuring the highest level of requirement clarity and quality [36].

Ambiguity Detecting Tool. This approach is obtained from one literature [25]. The tool found from this literature is REVV-Light. REVV-Light is an open-source Web2.0 application that blends information visualization and natural language processing. Its purpose is to assist requirement analysts in identifying instances of terminological ambiguity arising from the use of near-synonyms in user story requirements. [25]. REVV-Light adds explicit support for ambiguity identification and a role-centered organization of the extracted terms.

B) How to Use User Story Elicitation Approach in Determining User Story Quality

Similarity Measure Algorithm. (1) tokenizing the text, removing stop words, and applying either stemming or lemmatization techniques [16]. This will prepare the data for the next steps. (2) This will prepare the data for subsequent steps. Use K-medoids and K-means to group similar user stories together. These algorithms rely on similarity measures to determine the most appropriate clusters[16]. (3) Assess the validity of the user story distribution using the silhouette coefficient (which lies between -1 to +1) [16] [18]. Positive values indicate well-formed clusters, while negative values suggest poorly formed clusters.

Recurrent Neural Network Models. (1) understand the existing neural network models [14]. (2) Adapt and customize neural network models to effectively classify user stories, tailoring them to the specific task. (3) select user story dataset to ensure its suitability for testing and validating the models [14]. (4) Enhance the user story dataset by incorporating additional examples, expanding its coverage and

improving its overall quality [39]. (5) Train and evaluate the customized neural network models using the augmented user story dataset, refining their parameters and optimizing their performance. (6) Analyze the results obtained from training and evaluation, drawing insights into the effectiveness of the models and identifying areas for further improvement.[14][39].

Online News Feature Extraction Method. (1) choose a news domain that is relevant to the application being developed [24][26][31][34]. This will help ensure that the extracted features are relevant to the application. (2) label the news articles to identify the aspects of who, what, and why[26][31]. This will create a training dataset for the feature extraction method. (3) clean the news articles to remove HTML tags and other formatting information [35] [26][31]. (4) Tag the words or tokens in the sentences using parts-of-speech (POS) tagging to identify grammatical patterns[26][31]. This will help the feature extraction method to identify important words and phrases. (5) identify the aspect of who, what, and why[26][31]. (6) validate the results [24][26][31][34].

User Story Clustering Framework. (1) accepting user story documents ,contain the detailed descriptions of user needs and requirements, as an input. [27][37]. (2) preprocesses the data by converting it into a list of phrases that serve as keywords [27][29][37]. (3) Feature extraction is performed on the list of phrases to generate numerical vectors using various extraction algorithms [27][37]. These algorithms transform the textual information into a format suitable for clustering. (4) clustering algorithm groups the numerical vectors into distinct clusters based on their similarities [27][29][37].

Crowd-based Elicitation Method. (1) establish a Core Team and Define Objectives [15][40]. The crowd-based elicitation method begins with the formation of a core team responsible for outlining the goals and objectives for the crowd's involvement. (2) The core team select and configure the feedback channel to facilitate communication and participation [15][40]. This channel could be an online forum, a dedicated website, or a social media platform. (3) crowd members are invited to express their ideas (posting or responding to others ides), comments, and suggestions [15][40]. (4) Analyzes the crowd's feedback to determine the level of support and popularity for each idea by the core team [15][40]. (5) Compiles a summary of the crowd's ideas, highlighting the most prominent and well-supported suggestions [15][40]. (6) The core team provides responses to the crowd's ideas, addressing concerns, requesting clarifications, and offering additional insights. (7) Crowd members provide feedback on the core team's responses, continuing the dialogue and ensuring that all perspectives are considered [15][40]. This iterative process allows for refinement and improvement of the ideas. (8) The core team consolidates the crowd's ideas, prioritizing those with strong support and potential for implementation [15][40].

TAGRAM Conceptual Model. (1) Initiate a project, compile a collection of user stories, and establish a corresponding tagging structure [20]. (2) Identify the essential labels and tags to commence the tagging process [20]. (3) Import user stories from text-based files. (4) Evaluate the quality of user stories against predetermined criteria [20].

Ontology Assessment Technique. (1) Define the scope that should be tailored to the specific needs of the project [17][33]. (2) Identify the criteria which is relevant

with the scope of the assessment [17]. (3) Develop rating scale for the criteria that have been identified. Design assessment process to evaluate the user stories[17]. (4) Pilot test the ontology using a small number of user stories and identify areas for improvement. (5) Deploy the ontology for all stakeholders to access and use[17][33].

Natural Language Processing (NLP) technique. (1) understanding user goals, pain points, and desired functionalities [19][28][32]. (2) analyze the extracted user needs to identify patterns, common themes, and relationships between different requirements [19][28]. (3) identifying the user role, desired action, and expected benefit, and translating them into a structured user story format [19][28][32]. (4) analyzing the language, checking for logical flaws, and ensuring user stories are clear, concise, and measurable [19][35][38]. (5) validate user stories by comparing them against user needs and ensuring they accurately reflect the desired functionalities. (6) track changes in user needs and adapt user stories accordingly [35] [38]. (7) analyze user feedback on implemented user stories to identify areas for improvement and potential bugs or usability issues [19][35][38].

Artisan Quality User Story Tool (AQUSA). (1) prepare the project issue reports as dataset to be analyzed [21]. (2) perform pre-processing activities from the dataset. (3) accessing the quality with AQUSA tool. Consequently, a list of defects about the user stories can be obtained. This result related to defects can be exported for further analysis [21].

User Story Prioritizing Method. (1) The initial step in the user story prioritization method involves identifying the key stakeholders who will participate in the prioritization process [22] [30]. These stakeholders may include representatives from various departments, such as product management, development, marketing, and sales. (2) The PO employs fuzzy logic to determine each stakeholder's weight, considering various parameters in the stakeholder analysis. (3) Using the Value Matrix's table, parameters are populated, and "fuzzy logic" assesses the criteria's relationships through union and intersection operations. When multiple stakeholders represent the same value, their designation or position within an organization determines their priority [22][30]. (4) PO identifies user stories involved in each project. (5) A decision matrix is generated with a structure of n rows and m columns [22]. The user stories are listed in the rows, while the prioritization criteria are represented in the columns. (6) estimate the user stories [22][30].

Machine Learning Classifier. (1) Utilize the One-Vs-Rest approach with Scikit-learn in Python to handle the user story dataset and label independence. This method separates the multilabel dataset into distinct binary labels, namely "Testable" and "Valuable" [23]. (2) perform text pre-processing involves stemming and lemmatization. (3) annotated the user stories as testable or valuable or both. (4) Extract features using the Term Frequency Inverse-Document Frequency (TFIDF) algorithm from Scikit-learn as a configuration for machine learning classifiers. TFIDF transforms the text data into numerical representations that can be effectively processed by machine learning algorithms [23].

Prototyping and Enriched User Stories. (1) The prototyping and enriched user stories process begins with gathering high-level requirements from the client and constructing a business process model that outlines the overall workflow of the

system [36]. This model provides a clear understanding of the system's purpose and functionality. (2) Based on the client's priorities, a specific feature is selected for implementation [36]. This prioritization ensures that the development efforts are aligned with the client's most pressing needs. (3) Building a prototype that may use high-fidelity mockups or a professional tool to accelerate development [36]. (4) The prototype is presented to the stakeholders for validation and verification [36]. This feedback loop ensures that the prototype meets the stakeholders' expectations (5) Validating prototypes with the technical team [36]. (6) An enriched user story is written to provide a comprehensive and detailed description of the user's needs and expectations for the selected feature [36]. (7) The enriched user story is reviewed by the user to ensure that it accurately captures their needs and expectations [36]. (8) adds the finished user story to the backlog [36].

Table 4. List of User Story Problem Using Requirement Elicitation Approach

Problem	Approaches
Ambiguity/vagueness	[16][18][19][14][27][28][29][32] [35][37][38][39]
Incomplete	[24][26][31][34]
Inconsistent/conflict	[17][33]
Incorrectness, untraceable	[21][23]
interdependency	[15][40]
granularity	[22][30]
No relation of user stories, time consuming, user story duplication	[20]

E. Conclusion

In the study conducted by Yanche [7] the most common issue found in user story elicitation is related to ambiguity. Furthermore, in this study, most of the existing approaches are also primarily addressing the issue of ambiguity (Table 4). Therefore, this study aligns with previous research. However, unlike previous studies that lacked detailed explanations of the approaches used to address user story elicitation issues, this study clearly outlines the approaches employed to tackle these challenges.

This study involved a comprehensive review of existing literature on user story elicitation approaches. The main goal was to gain a thorough understanding of these approaches. This goal was achieved by reviewing 27 selected papers. The strength of this study is that this study conducts comprehensive analysis and clearly defines the research questions. Furthermore, this study provides a list of detailed summarized approaches, which has not been conducted by other researchers.

The scope of this study was limited to papers published between 2018 and 2023, and as a result, the result may not fully reflect the entirety of research on user stories elicitation approaches. Additionally, the quality evaluation scores presented in this research were assessed by a single individual, which could potentially introduce some degree of subjectivity.

In future research, the effectiveness of each approach in this study has not yet been measured. The effectiveness of each approach can be studied with the aim of making the selection of approaches more accurate, considering the problems faced by each organization in selecting user stories. In addition, the limitations of each

approach also have not yet been measured. This could also be a research topic for the future work. Researchers can utilize this information to create more efficient methods for gathering requirements through user stories.

F. References

[1] I. K. Raharjana, D. Siahaan, and C. Fatichah, "User Stories and Natural Language Processing: A Systematic Literature Review," *IEEE Access*, vol. 9, pp. 53811–53826, 2021, doi: 10.1109/ACCESS.2021.3070606.

[2] F. Dalpiaz and S. Brinkkemper, "Agile requirements engineering with user stories," in *Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018*, Institute of Electrical and Electronics Engineers Inc., Oct. 2018, pp. 506–507. doi: 10.1109/RE.2018.00075.

[3] M. Kassab, "An empirical study on the requirements engineering practices for agile software development," in *Proceedings - 40th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2014*, Institute of Electrical and Electronics Engineers Inc., Oct. 2014, pp. 254–261. doi: 10.1109/SEAA.2014.77.

[4] T. Rocha Silva, M. Winckler, and C. Bach, "Evaluating the usage of predefined interactive behaviors for writing user stories: an empirical study with potential product owners," *Cognition, Technology and Work*, vol. 22, no. 3, pp. 437–457, Aug. 2020, doi: 10.1007/s10111-019-00566-3.

[5] H. Hibshi, S. T. Jones, and T. D. Breaux, "A Systemic Approach for Natural Language Scenario Elicitation of Security Requirements," *IEEE Trans Dependable Secure Comput*, vol. 19, no. 6, pp. 3579–3591, 2022, doi: 10.1109/TDSC.2021.3103109.

[6] ISO/IEC/ IEEE 29148, "Systems and software engineering-Life cycle processes-Requirements engineering," 2018.

[7] Y. A. Kustiawan and T. Y. Lim, "User Stories in Requirements Elicitation: A Systematic Literature Review," in *2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS)*, IEEE, Aug. 2023, pp. 211–216. doi: 10.1109/ICSECS58457.2023.10256364.

[8] 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.

[9] S. N. Fathin Najwa Binti Mustaffa, J. Bin Sallim, and R. Binti Mohamed, "Enhancing High-Quality User Stories with AQUSA: An Overview Study of Data Cleaning Process," in *Proceedings - 2021 International Conference on Software Engineering and Computer Systems and 4th International Conference on Computational Science and Information Management, ICSECS-ICOCSIM 2021*, Institute of Electrical and Electronics Engineers Inc., Aug. 2021, pp. 295–300. doi: 10.1109/ICSECS52883.2021.00060.

[10] F. Dalpiaz and S. Brinkkemper, "Agile Requirements Engineering: From User Stories to Software Architectures," in *Proceedings of the IEEE International Conference on Requirements Engineering*, IEEE Computer Society, 2021, pp. 504–505. doi: 10.1109/RE51729.2021.00076.

[11] Z. A. Barmi, A. H. Ebrahimi, and R. Feldt, "Alignment of requirements specification and testing: A systematic mapping study," in *Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation Workshops, ICSTW 2011*, 2011, pp. 476–485. doi: 10.1109/ICSTW.2011.58.

[12] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - A systematic literature review," *Information and Software Technology*, vol. 51, no. 1. pp. 7–15, Jan. 2009. doi: 10.1016/j.infsof.2008.09.009.

[13] B. Kitchenham, "Procedures for Performing Systematic Reviews," 2004.

[14] F. J. Peña Veitía, L. Roldán, and M. Vegetti, "User Stories identification in software's issues records using natural language processing," in *2020 IEEE Congreso Bienal de Argentina, ARGENCON 2020 - 2020 IEEE Biennial Congress of Argentina, ARGENCON 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020. doi: 10.1109/ARGENCON49523.2020.9505355.

[15] A. Menkveld, S. Brinkkemper, and F. Dalpiaz, "User story writing in crowd requirements engineering: The case of a web application for sports tournament planning," in *Proceedings - 2019 IEEE 27th International Requirements Engineering Conference Workshops, REW 2019*, Institute of Electrical and Electronics Engineers Inc., Sep. 2019, pp. 174–179. doi: 10.1109/REW.2019.00037.

[16] B. Kumar, U. Tiwari, and D. C. Dobhal, "User Story Splitting in Agile Software Development using Machine Learning Approach," in *PDGC 2022 - 2022 7th International Conference on Parallel, Distributed and Grid Computing*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 167–171. doi: 10.1109/PDGC56933.2022.10053226.

[17] C. Tona, R. Juarez-Ramirez, S. Jimenez, and R. G. P. Lopez, "Toward Developing an Ontology for Assessing Quality of User Stories in Scrum Framework," in *Proceedings - 2022 10th International Conference in Software Engineering Research and Innovation, CONISOFT 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 39–48. doi: 10.1109/CONISOFT55708.2022.00016.

[18] B. Kumar, U. K. Tiwari, D. C. Dobhal, and H. S. Negi, "User Story Clustering using K-Means Algorithm in Agile Requirement Engineering," in *Proceedings of International Conference on Computational Intelligence and Sustainable Engineering Solution, CISES 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1–5. doi: 10.1109/CISES54857.2022.9844390.

[19] F. Gilson and C. Irwin, "From user stories to use case scenarios towards a generative approach," in *Proceedings - 25th Australasian Software Engineering Conference, ASWEC 2018*, Institute of Electrical and Electronics Engineers Inc., Dec. 2018, pp. 61–65. doi: 10.1109/ASWEC.2018.00016.

[20] M. Bragilovski, S. Erez, C. Mordehai, S. Rahamim, N. Shpack, and A. Sturm, "TAGRAM: A Framework for Tagging User Stories," in *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*, IEEE, Sep. 2023, pp. 62–66. doi: 10.1109/REW57809.2023.00017.

[21] S. N. Fathin Najwa Binti Mustaffa, J. Bin Sallim, and R. Binti Mohamed, "Enhancing High-Quality User Stories with AQUSA: An Overview Study of Data Cleaning Process," in *Proceedings - 2021 International Conference on Software*

Engineering and Computer Systems and 4th International Conference on Computational Science and Information Management, ICSECS-ICOCSIM 2021, Institute of Electrical and Electronics Engineers Inc., Aug. 2021, pp. 295–300. doi: 10.1109/ICSECS52883.2021.00060.

[22] N. H. Borhan, H. Zulzalil, S. Hassan, and N. M. Ali, "A Hybrid Prioritization Approach by integrating non-Functional and Functional User Stories in Agile-Scrum Software Development (i-USPA):A preliminary study," in *2022 IEEE International Conference on Computing, ICOCO 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 276–282. doi: 10.1109/ICOCO56118.2022.10031863.

[23] I. M. Subedi, M. Singh, V. Ramasamy, and G. S. Walia, "Classification of Testable and Valuable User Stories by using Supervised Machine Learning Classifiers," in *Proceedings - 2021 IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 409–414. doi: 10.1109/ISSREW53611.2021.00111.

[24] P. Harth, O. Jähde, S. Schneider, N. Horn, and R. Buchkremer, "From Data to Human-Readable Requirements: Advancing Requirements Elicitation through Language-Transformer-Enhanced Opportunity Mining," *Algorithms*, vol. 16, no. 9, p. 403, Aug. 2023, doi: 10.3390/a16090403.

[25] F. Dalpiaz, I. van der Schalk, S. Brinkkemper, F. B. Aydemir, and G. Lucassen, "Detecting terminological ambiguity in user stories: Tool and experimentation," *Inf Softw Technol*, vol. 110, pp. 3–16, Jun. 2019, doi: 10.1016/j.infsof.2018.12.007.

[26] D. Siahaan, I. K. Raharjana, and C. Fatichah, "User story extraction from natural language for requirements elicitation: Identify software-related information from online news," *Inf Softw Technol*, vol. 158, Jun. 2023, doi: 10.1016/j.infsof.2023.107195.

[27] B. Yang, X. Ma, C. Wang, H. Guo, H. Liu, and Z. Jin, "User story clustering in agile development: a framework and an empirical study," *Front Comput Sci*, vol. 17, no. 6, Dec. 2023, doi: 10.1007/s11704-022-8262-9.

[28] M. Ahmed, S. U. R. Khan, and K. A. Alam, "An NLP-based quality attributes extraction and prioritization framework in Agile-driven software development," *Automated Software Engineering*, vol. 30, no. 1, May 2023, doi: 10.1007/s10515-022-00371-9.

[29] L. K. Lam, C. A. L. Hurtado, and L. W. Portillo, "Framework for automating requirement elicitation using a chatbot," in *Proceedings of the 2022 IEEE Engineering International Research Conference, EIRCON 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/EIRCON56026.2022.9934823.

[30] P. Thanomwong and T. Senivongse, "User Story Risk Prioritization Model for Agile Software Development," in *Proceedings of 2022 International Conference on Data and Software Engineering, ICoDSE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 161–166. doi: 10.1109/ICoDSE56892.2022.9972041.

[31] N. Ngaliah, D. Siahaan, and I. K. Raharjana, "User Story Extraction from Online News with FeatureBased and Maximum Entropy Method for Software

Requirements Elicitation," *IPTEK The Journal for Technology and Science*, vol. 32, no. 3, p. 125, Jan. 2022, doi: 10.12962/j20882033.v32i3.11625.

[32] F. Casillo, V. Deufemia, and C. Gravino, "Detecting privacy requirements from User Stories with NLP transfer learning models," *Inf Softw Technol*, vol. 146, Jun. 2022, doi: 10.1016/j.infsof.2022.106853.

[33] S. Heng, K. Tsilionis, and Y. Wautelet, "Building User Stories and Behavior Driven Development Scenarios with a Strict Set of Concepts: Ontology, Benefits and Primary Validation," in *Proceedings of the ACM Symposium on Applied Computing*, Association for Computing Machinery, Mar. 2023, pp. 1422–1429. doi: 10.1145/3555776.3577696.

[34] M. Rahmi Dewi, I. Kharisma Raharjana, D. Siahaan, and C. Fatichah, "Software Requirement-Related Information Extraction from Online News using Domain Specificity for Requirements Elicitation: How the system analyst can get software requirements without constrained by time and stakeholder availability," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Feb. 2021, pp. 81–87. doi: 10.1145/3457784.3457796.

[35] T. Gunes and F. B. Aydemir, "Automated Goal Model Extraction from User Stories Using NLP," in *Proceedings of the IEEE International Conference on Requirements Engineering*, IEEE Computer Society, Aug. 2020, pp. 382–387. doi: 10.1109/RE48521.2020.00052.

[36] N. Keshk, M. El-Ramly, and A. Salah, "A Proposal for Enhancing Agile Requirements Engineering with Prototyping and Enriched User Stories," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jun. 2022, pp. 59–63. doi: 10.1145/3531056.3542773.

[37] B. Yang, H. Guo, and H. Liu, "Evaluation and assessment of machine learning based user story grouping: A framework and empirical studies," *Sci Comput Program*, vol. 227, Apr. 2023, doi: 10.1016/j.scico.2023.102943.

[38] M. Elallaoui, K. Nafil, and R. Touahni, "Automatic Transformation of User Stories into UML Use Case Diagrams using NLP Techniques," in *Procedia Computer Science*, Elsevier B.V., 2018, pp. 42–49. doi: 10.1016/j.procs.2018.04.010.

[39] H. Guo, O. Kafali, and M. Singh, "Extraction of Natural Language Requirements from Breach Reports Using Event Inference," in *Proceedings - 2018 5th International Workshop on Artificial Intelligence for Requirements Engineering, AIRE 2018*, Institute of Electrical and Electronics Engineers Inc., Oct. 2018, pp. 22–28. doi: 10.1109/AIRE.2018.00009.

[40] J. Wouters, A. Menkveld, S. Brinkkemper, and F. Dalpiaz, "Crowd-based requirements elicitation via pull feedback: method and case studies," *Requir Eng*, vol. 27, no. 4, pp. 429–455, Dec. 2022, doi: 10.1007/s00766-022-00384-6.