

Natural Language Processing for Effective Use of User Stories: Systematic Literature Review

Amal Khalil^{1,*}, Youssef Mellah², Mohammed Saber³, Mohammed Ghaouth Belkasmi⁴

^{1,3,4} Mohammed First University, ENSAO, Oujda, 60020, Morocco

² Novelis, Paris, France

Email : ¹ amal.khalil@ump.ac.ma, ² ysfimell@gmail.com, ³ m.saber@ump.ac.ma, ⁴ m.belkasmi@ump.ac.ma

*Corresponding Author

Abstract— User stories are an essential element of agile software development due to their flexible structure and simplicity. The diverse range of data sources and formats utilized for documenting requirements poses challenges for development teams, who often have to translate client needs into user stories manually, a process that is both time-consuming and error-prone. To address this, interest has grown in automating user story generation. A systematic literature review (SLR) was conducted to assess current methodologies, focusing on the types of corpora used, the natural language processing (NLP) techniques employed and the methods used to evaluate the quality of generated user stories. Our review reveals a significant shortage of publicly available corpora, which limits progress in this field, particularly for machine learning-based approaches. It also identifies a wide range of techniques and highlights the lack of standardized methods for evaluating user story quality. More recent developments involving Large Language Models (LLMs) offer new opportunities for both user story generation and quality evaluation, but also raise concerns regarding hallucination, consistency, semantic fidelity, reproducibility, and the need for human validation. Overall, the findings highlight the need for publicly available datasets, standardized evaluation frameworks, and rigorous empirical studies to assess the effectiveness and reliability of automated and LLM-based user story generation approaches.

Keywords— *Natural Language Processing; User Stories; Systematic Literature Review; Automated User Story Generation; Requirements Engineering; Quality*

I. INTRODUCTION

As software systems become more complex, developers have to consider not only technological issues but also human, organizational and societal considerations. Agile Software Development (ASD) has emerged as a solution to these problems. Iterative planning holds significant importance in ASD, enabling teams to enhance requirements gradually and involve stakeholders effectively. Daily meetings and regular client reviews are employed to oversee team progress and collect feedback on the development of software products [1].

Under these circumstances, ASD employs user stories to depict actual situations faced by end users. User stories are generally created through discussions among stakeholders as

well as team members. They may also be derived from requirements artifacts such as project documentation, standards, communications or business rules. User stories employ a semi-structured format to communicate important features to users, irrespective of their origin [2]: Employing a template enhances productivity and improves consistency and quality.

This study conducts an SLR to investigate recent advances in automated user story generation, considering both specific and elicitation techniques, and examining how actors assess the quality of generated user stories. Three Research Questions (RQ) were derived from our study objectives:

RQ1: What datasets do researchers use, and how do they utilize them, to automatically create user stories?

RQ2: What methods are used to automatically generate user stories?

RQ3: How is the user story quality assessed?

This study presents a compilation of data sets related to user stories, discusses techniques for automatically generating user stories, analyzes criteria for assessing the quality of user stories, and discusses future work. In addition, it explores key concepts regarding agile specifications processing and user story generation, details the study design and answers proposed research questions. Finally, it explores perspectives and opportunities, discusses limitations and mitigation strategies, and concludes.

II. BACKGROUND

Developed by Kent Beck [3] in Extreme Programming, story cards are brief summaries of system behaviour that encourage dialogue within development teams. This idea was improved upon by Mike Cohn [2], who emphasized the value of user-focused functionality and continuous communication between developers and stakeholders. Any team member or stakeholder can write user stories, which usually originate from discussions or other sources of requirements. They improve productivity and deliverable quality by adhering to a semi-structured template that specifies an actor, a goal, and



Received: 23-7-2026

Revised: 23-9-2026

Published: 31-12-2026

the benefit ([4]). Instead of emphasizing too much detail in the stories themselves, the user story template promotes collaboration and a client-focused viewpoint.

According to Lucassen et al. [5], most practitioners use templates to increase efficiency and quality, and user stories are a lightweight form of requirements representation used by developers. By promoting thorough requirements and strengthening stakeholder-developer collaboration, user stories assist development teams in developing products that better meet stakeholder needs and, eventually, averting errors [5]. However, only 45% of agile practitioners are dedicated to using user stories, according to a study by Pokharel et al. [6]. In contrast to full-stack developers, they found that backend developers are less likely to employ user stories. This suggests that in order to increase the adoption of user stories and agile concepts in businesses, there is a need for efficient training on these topics.

III. RELATED WORKS

Through this research, we discovered several relevant secondary studies about user stories generation that informed our investigation. The studies listed here provide insightful information. A rigorous mapping of the literature on user story research was carried out by Amna et al. [7], who examined 186 publications issued between 2001 and 2021. They realized that the majority of research (78%) is solution-oriented (80%) and recent (2015–2021), with an emphasis on algorithms for system design problems during requirements analysis. They pointed out that there aren't many empirically validated solutions for ambiguity in requirements engineering (RE) user stories unclear user stories. In their review of 38 studies on the use of Natural Language Processing (NLP) to define user stories, Raharjana et al. [8] emphasized the utilization of Semantic Role Labelling (SRL) to discover important abstractions and enhance comprehension and development speed. They noted issues like low recall and precision in research and variability in datasets. An SLR was also performed by Dos Santos et al [9] which focused on the state of the art of NLP and ML techniques for handling user stories in the context of RE.

In their analysis of 86 works on ML in collecting specifications, Cheliger et al. [10] identified 12 data sources and 15 ML tasks. They proved that Naive Bayes and Neural Networks are popular algorithms and observed a trend toward feature embedding. They underlined the necessity of conducting more thorough empirical evaluations of ML techniques. The state of NLP in requirements engineering was mapped by Zhao et al. [11], who came to the conclusion that although technical solutions have advanced, more empirical evaluations in concrete situations are required for future developments. Additionally, they emphasized the necessity for further research on Agile Software Development (ASD) user stories and associated documentation. Kitchenham et al. [13] provided methodological guidelines for conducting evidence-based software engineering and systematic literature reviews. These principles guided the methodological design of the present review. Amna et al. [14] conducted a systematic literature review on ambiguity in user stories, examining the different sources and forms of ambiguity and the approaches proposed

to address this issue. Their work highlights the importance of considering ambiguity when assessing the quality of user stories.

The present review complements these studies by jointly examining three dimensions that are particularly relevant to automated user story generation: (i) the datasets and corpora used to support generation or extraction, (ii) the computational techniques employed to generate or specify user stories, and (iii) the approaches used to assess the quality of the resulting user stories. In addition, the present review considers the transition from traditional NLP and machine learning techniques toward neural and emerging Large Language Model (LLM)-based approaches. This perspective is particularly relevant given the rapid evolution of generative AI and its increasing application to language-intensive Requirements Engineering activities.

Consequently, the objective of this SLR is not to reproduce previous reviews, but to provide an integrated view of the data resources, generation techniques, and evaluation practices associated with automated user story generation, while highlighting the methodological and empirical gaps that remain in this research area.

IV. WORK STRATEGY

To conduct the review, we used three major steps inspired by Pollock et al. [12] involving: review methodology, conducting the review, and reporting the review findings. Our research implemented a composite search strategy that combined database searching with snowballing techniques to identify pertinent literature for a systematic literature review. The database search was conducted using three major digital libraries: Scopus, Web of Science, and the ACM Digital Library. This was followed by an analysis that examined the citations and references of selected studies to uncover more relevant sources.

The process began with the identification of appropriate keywords that were in accordance with the authors' prior knowledge and expertise, as per the recommendations of Kitchenham et al. [13]. The authors refined their search string by testing various combinations of these keywords against a control group of known studies. The search string ultimately included terms related to user stories, and natural language processing: (NLP or "natural language processing") and ("user stories" or "user story"). Scopus was selected for its broad coverage of international publishers, in addition to the Web of Science and ACM (Association for Computing Machinery) digital library databases. The search string was intentionally narrow on NLP and user stories as NLP represents the broader research context examined in this review. Here, machine learning, automated generation and elicitation were considered specific techniques or applications, rather than as different dimensions of search. This approach was used to keep the scope manageable and reproducible.

The final search was conducted in June 2025, for studies published from January 2013, resulting in the retrieval of 199 studies (Scopus: 66 documents, Web of science: 51 documents and ACM: 82 documents). We established specific inclusion and exclusion criteria to ascertain the

relevance of the retrieved documents following the database search:

Inclusion criteria: the study (IC1) is a research publication or a review paper, (IC2) in English, (IC3) released between January 2013 and November 2024, and (IC4) corresponding to the chosen search keywords (related to user stories managed using NLP in the English language).

Exclusion criteria: (EC1) overview of conference keynotes, doctoral symposium papers, editorials, proposals, comments, lecture notes, and tutorials, and (EC2) published in a predatory conference.

The retrieved records were first screened based on their titles, abstracts, and keywords. Potentially relevant studies were subsequently examined in full text. Secondary studies, including systematic literature reviews and systematic mapping studies, were identified separately and were not counted among the primary studies. These secondary studies were used to identify relevant background literature and to support the positioning of the present review.

A. Study selection

The dataset was initially composed of 199 documents. These documents were subsequently cleaned by eliminating conference announcements and irrelevant materials, resulting in a total of 157 documents. The titles, abstracts, and keywords were subsequently subjected to the IC and EC, resulting in identifying 16 potentially relevant studies. This was further refined to 10 studies that were included through a full-text analysis. Seven secondary studies (SLRs and systematic mappings) were identified and categorized separately as related work during the initial analysis ([7]-[11], [13] and [14]).

Consequently, a hybrid search strategy was implemented, which included forward and backward snowballing techniques as described by Wohlin [15]. The snowballing process was conducted over two iterations using the initially selected primary studies and relevant secondary studies as seed sources. Candidate studies identified through snowballing were subjected to the same inclusion and exclusion criteria and full-text assessment.

Following this process, eight additional primary studies were identified and included. Therefore, the final set consisted of 18 primary studies, while the identified secondary studies were retained separately as related work.

The complete selection process is summarized in Figure 1.

B. Quality assessment

To evaluate the methodological quality of the selected studies, we applied a quality assessment approach. We were motivated by the quality assessment strategy used by Schon et al. [16]. Table 1 displays the checklist used to assess the quality of the included studies.

Citation count was considered separately as a descriptive indicator of the visibility of the selected studies. Since citation frequency may be influenced by factors such as publication age and research community size, it was not included in the methodological quality assessment score.

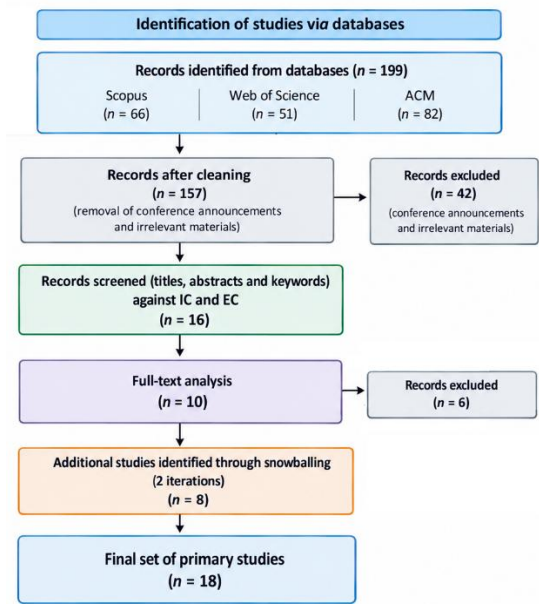


Fig. 1. PRISMA-like flow diagram illustrating the study selection process, from database identification and screening to full-text assessment and snowballing

Table 1. Quality assessment criteria

Quality assessment	Evaluation criteria	Score	Description
QA1	Was the purpose of the study clearly stated?	-1	No, there was no description of the aims.
		0	The goals were inadequately defined and lacked clarity.
		1	Yes, the goals were clearly stated.
QA2	Does the research provide a thorough explanation of the suggested resolution?	-1	No, details were lacking.
		0	Partially, you must read the references if you want to use the strategy or solution.
		1	Yes, the method can be applied with the provided description.
QA3	Has the suggested solution or method been confirmed?	-1	No, it was not confirmed.
		0	It received partial validation in a laboratory setting, so only certain aspects of the proposal were confirmed.
		1	Yes, through a case study.
Citation count (descriptive)	Has the research been referenced in other scientific writings?	-1	No, no one referenced the study.
		0	Slightly. The work was referenced in one to five scientific articles.
		1	Yes, the work was referenced in more than five scientific papers.

Citation data are therefore reported only to provide additional context on the academic visibility of the primary studies and do not affect their inclusion or quality rating.

V. REVIEW FINDINGS

A. Summary of studies

A total of 18 primary studies were included in the review, comprising 28% (5 studies) journal articles, 72% (13 studies) conference papers. The datasets analyzed include short papers, with 44% of studies (8 papers) being full papers and 55% (10 papers) being short papers. The distribution of publication venues and study types indicates that research on automated user story generation is still relatively emerging, with a substantial proportion of the identified studies published as short conference papers. Advancements in machine learning, language models, and NLP techniques could automate user story writing in software development projects. Evaluation of proposed approaches is needed to validate the topic's potential.

Regarding quality assessment, all studies (18 papers) were evaluated and retained as included. Each study's purpose is evaluated in the first question (QA1). In 94.4% of the investigations, the answer to this question was yes (17 papers). QA2 assessed whether the proposed method was described in sufficient detail. In 72.2% of the studies, this question had a positive response (13 papers). QA3 assessed the validation of the proposed method. Suitable validation methods were used in just 83.3% of the studies (15 papers). The last indicator (Citation count) looks for how many citations each study has received. As a result, 88.8% of research received more than five citations from other studies (16 papers). The relevant studies quality assessment results are displayed in Figure2.

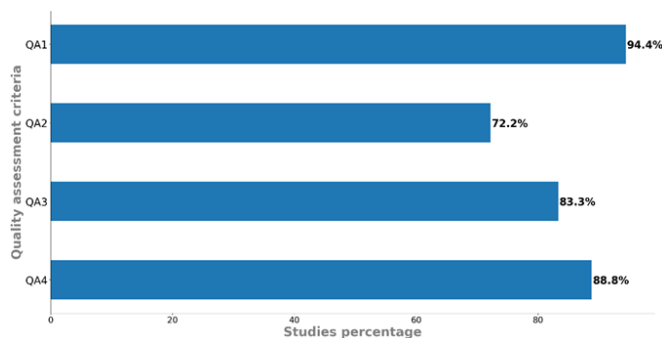


Fig. 2. Quality assessment percentage scores

B. What datasets do researchers use, and how do they utilize them, to automatically create user stories? (RQ1)

The success of machine learning approaches is contingent upon the selection of valuable datasets, which facilitates reproducibility and comparison across various studies. In a review of primary studies, 16 software requirements datasets were identified, with only eight of them being publicly accessible. The proprietary nature of software engineering practices, which frequently maintain the confidentiality of their development processes, may be the reason for the limited availability of public datasets [17]. Furthermore, two lexical dictionaries, Wordnet [18] and Bloomsoft [19], were incorporated to aid in the development of a dictionary of user story terms. In a notable example, Rodeghero et al. [20]

employed a manually annotated dataset to train an ML classifier for the recognition of functional and rationale data by extracting user story information from requirements-gathering meeting transcripts. Additionally, Veitia et al. [21] provided two datasets that were employed to recognize user stories in software issue records, and rather than explicitly generating user stories, they employ NLP and ML to extract information regarding a user story from software issue trackers. Kumar et al. [22] implemented a Python script that utilizes linguistic heuristics to divide intricate user stories into simpler ones, notably for CRUD operations. A dataset that was generated from a project at Duke University libraries that concentrated on digital collections was shared by Resketi et al. [23]. The dataset was designed to summarize user experiences.

Heng et al. [24] contributed a dataset that was organized through an ontology to improve the quality of user stories. The dataset included user stories along with Behavior-Driven Development (BDD) cases. Siahaan et al. [25] employed the Public Requirements Documents Dataset (PURE) and lexical dictionaries such as Wordnet and Bloomsoft to develop a user stories terms dictionary. This dictionary has been previously implemented in a variety of requirements engineering (RE) methodologies. Herwanto et al. [26] used their Named Entity Recognition (NER) dataset [27] in which the existence of features that are important to privacy was carefully labeled in each user story.

Nevertheless, a number of datasets that are cited in the literature are not publicly available as a result of industrial confidentiality or access restrictions. For example, Raharjana [28] and Siahaan [25] generated online news datasets but refrained from disclosing them, while Thamrongchote [29] validated their findings using historical project data without sharing it. These studies emphasize the potential of employing news corpora in RE to enhance domain comprehension. Mateus et al. [30] proposed a method for generating user stories from Business Process Model and Notation (BPMN) graphs; however, they did not employ a specific dataset for evaluation. Furthermore, Henriksson et al. [31] and three brief papers ([29] and [32]) presented concepts without dataset validation, with an emphasis on conceptual frameworks for automated requirements elicitation.

C. What methods are used to automatically generate user stories? (RQ2)

The current study investigates the automatic identification and specification of user stories in software engineering through NLP techniques. It highlights the integration of methods to achieve optimal outcomes, noting the diverse methodologies that have been implemented in recent research.

1. Ontologies: Thamrongchote [29], Murtazina [33], and Heng [24] employed ontologies to systematically arrange user story details, improve utilization across projects, evaluate quality, and validate datasets. Their research emphasizes the significance of structured data acquisition in the development of user stories.

2. Chatbots: Lam et al. [34] and Dwitama et al. [35] implemented chatbot applications to facilitate requirements

elicitation. Lam's study used IBM Watson chatbot, while Dwitama's employed AIML (Artificial Intelligence Markup Language) to analyze Indonesian texts. Both studies demonstrate the potential of AI in requirements engineering, despite the fact that they encounter obstacles associated with context adaptability.

3. Named Entity Recognition (NER): Raharjana [28], Herwanto [26] and Henriksson [31] employed NER techniques to extract user story data. Raharjana integrated NER with Part-of-Speech (POS) labeling, Herwanto employed NER to pinpoint privacy requirements within an agile software development setting, while Henriksson suggested a data model for organizing requirements from a variety of sources.

4. Language Heuristics: In order to generate user stories from architectural modules and business process models, respectively, numerous studies, such as those conducted by Santos et al. [36] and Mateus et al. [30], implemented language heuristics.

5. Machine Learning: Rodeghero [20] and Veitia [21] explored the use of ML classification techniques to identify user story content in conversations and issue trackers. They achieved promising results using Logistic Regression and Neural Networks.

6. Additional Methods: A variety of additional methods were studied, such as the Requirements-Collector tool by Panichella et al. [37], which incorporates multiple ML algorithms for requirements specification, and the Bag of Words (BoW) by Resketi et al. [23] for keyword extraction.

Generally, the research suggests that the user story elicitation process is significantly improved by the use of linguistic heuristics, regex algorithms, and a variety of ML techniques. Although NLP approaches remain the most commonly adopted, they are not the only effective solution. The results indicate that combining different methodologies can lead to improved software requirements engineering outcomes.

D. How is the user story quality assessed? (RQ3)

Several methodologies are employed by researchers to assess the quality of automatically generated user stories. The adoption of requirements engineering quality guidelines for evaluating user stories is relatively minimal, which is a significant discovery. Of the 18 studies that were analyzed, 39% (7 studies) did not include any quality evaluation, predominantly focusing on presenting their methodologies without validation ([24], [29], [30], [31], [32], [37], [36]).

Of the remaining studies, 28% (5 studies) did not conduct controlled experiments or adhere to RE guidelines and instead relied solely on quantitative measures ([20], [21], [22], [25], [35]). In contrast, qualitative methods were implemented in 28% of the studies (5 studies) to judge the characteristics of the user stories that were generated ([23], [26], [33], [34], [38]). Notably, one study (5%) performed an empirical evaluation based on the authors' RE experience [28]. Specific methodologies included the utilization of standard machine learning metrics (precision, recall, etc.) to assess data quality against manually annotated datasets ([20],

[21], [26]) and guided interviews with software specialists to validate ontologies for user story generation [24]. Other research employed statistical evaluations, including the comparison of silhouette values to assess user story cohesion [22] and the measurement of chatbot performance using a variety of metrics [35].

Some researchers utilized domain ontologies to validate user stories, guaranteeing that critical elements such as role, action, and benefit were present [33]. In one study, the AQUASA (Automatic Quality User Story Artisan) tool was employed to preprocess and statistically evaluate user stories [23], while in another, expert feedback was sought to evaluate the accuracy of the generated outputs [38]. Additionally, the survey, based on ISO 9126, was conducted to assess the completeness and accuracy of user stories in controlled experiments [34]. In conclusion, the overall reliance on established RE guidelines is limited, as many studies concentrate on either quantitative measures or qualitative assessments without comprehensive validation, despite the existence of a variety of techniques for evaluating user story quality.

VI. DISCUSSION

The complex structure of the development process is evident in the various sources and formats that contribute to the formulation of software requirements [39]. Through seminars, surveys, and interviews, all those with an interest in the project, such as clients, users, and domain experts, actively engage in the process of identifying both functional and non-functional needs. Developers favor user stories due to their simplicity and expediency [8]. Nevertheless, manually generated user stories frequently experience quality and consistency issues as a result of factors such as time constraints and inexperience ([40], [41]). Clear communication among development teams and stakeholders is contingent upon the presence of high-quality user experiences.

The automated generation of user stories has garnered increasing attention in recent research, as evidenced by this SLR. The potential benefits of automated user story generation for software development are significant, despite its ongoing development. Initially, the requirements gathering process could be streamlined through the use of automated techniques, which would subsequently reduce development time. Secondly, the quality and consistency of user stories could be improved if the automation methods are precise. Finally, this method would enable teams to devote more time to development rather than refinement meetings, resulting in faster stakeholder feedback and a greater degree of agility in the development process.

A. Accessibility of user story datasets

This paper explores the obstacles to the automatic generation of user stories at the confluence of Artificial Intelligence (AI) and RE. It emphasizes the scarcity of publicly accessible datasets, which is essential for the validation of research and the improvement of the quality of the results. The majority of the studies that were reviewed are brief and lack comprehensive validation, with some experiments or essential data being absent. The necessity of

larger datasets is underscored, as small datasets can result in overfitting and impede model generalization, thereby impeding progress and adoption within the RE community.

That implies that these challenges could be resolved through a transdisciplinary approach that integrates AI and RE. There is an increasing interest in AI in the software development industry, and companies have a wealth of data from project monitoring tools that could be used to train machine learning models. This has the potential to provide RE practitioners with improved insights and tools for their work, which could be significantly beneficial.

B. Investigated techniques

The investigation of methods for automated user story processing reveals a wide range of approaches, which are categorized into six primary groups: Tool, Supervised Learning, Unsupervised Learning, Neural Networks, Knowledge Representation et NLP.

1. Tool: User story information is collected through conversational interfaces by chatbots, which are frequently powered by NLP and LLMs. The automatic generation of user stories is facilitated by tools such as IBM Watson chatbot [34] and Docx4j parser [33], which enable the extraction and manipulation of requirements from a variety of formats.

2. Supervised Learning: This method involves learning from labeled examples to classify user story information. Research has employed a variety of algorithms, such as sentiment analysis and NER, to improve the identification of user stories ([20], [31], [37]).

3. Unsupervised Learning: Algorithms such as K-means clustering are employed to identify patterns in unlabeled data, which is beneficial for the validation of user stories and the verification of consistency [22].

4. Neural Networks: In the identification of user stories, neural networks, mainly LSTM (Long Short-Term Memory) models [21], ELMo (Embeddings from Language Models) [21], BERT (Bidirectional Encoder Representations from Transformers) [42] and LLMs (Large Language Models) such as GPT [43], have demonstrated potential. Recent research indicates that LLMs can considerably enhance the generation of user stories and requirement summaries.

5. Knowledge Representation: User story data is organized using ontologies, which define the terminology and relationships within a domain ([24], [29], [33]). Nevertheless, they encounter obstacles such as the necessity of domain specialists and inadequate lexical coverage. Semantic Role Labeling (SRL) is used to find the "who, what, and why" in user stories, thereby improving the understanding of their key elements ([28], [25]).

6. NLP: User stories were identified or generated in all of the studies that were reviewed at some point using natural language processing. Main techniques consist of:

- **Tokenization:** Necessary for NLP and ML, tokenization involves breaking text into smaller units.
- **Part-of-Speech (POS) Tagging:** The process of assigning grammatical categories to words or tokens.

- **Text Parsing:** Utilizing constituency and dependency parsing to analyze sentence structure, thereby facilitating the identification of user story elements.

- **Named Entity Recognition (NER):** The process of classifying entities in text to improve the identification of roles in user stories.

- **Bag-of-Words (BoW):** A method of representing text by word frequency that is useful for identifying themes but fails to capture the meaning of a sentence. In order to extract user story information effectively, these methods frequently necessitate the integration of language heuristics.

C. Adoption of user story quality norms

The user story quality analysis reveals a lack of reliance on qualitative standards in studies, which is frequently attributed to the prevalence of short papers that omit qualitative evaluations. Software practitioners frequently assess the integrity of user stories [5], it is therefore imperative that authors utilize frameworks or guidelines to assess the quality of user stories, as inadequate requirement descriptions impede the development of software products that are effective.

Although quantitative measurements are employed in certain studies ([20], [21], [22], [35]), they should be used in conjunction with qualitative evaluations rather than as the sole method. Controlled experiments may mitigate evaluation bias, and guidelines and technical approaches [44] are available for qualitative assessment. [34] are notable for their use of ISO 9126 to evaluate completeness, accuracy, and reliability, while [23] employ the AQUA tool, which is founded on the QUS (Quality User Story) framework, to evaluate user stories. Even if empirical evaluations are valuable, they may not be sufficient to evaluate automated user story generation, indicating the necessity of additional controlled experiments. The utilization of ontologies for quality evaluation, as illustrated by Murtazina et al. [33], offers an additional potential enhancement.

In summary, the automatic generation of user stories has the potential to improve the quality of software requirements. Statistical evaluations alone may not sufficiently capture the importance of qualitative guidelines. Therefore, incorporating these guidelines is essential to better reflect their critical role throughout the software development lifecycle.

D. Emergence of large language models in user story generation and evaluation

The evolution from traditional NLP and machine learning techniques toward Large Language Models represents an important development in Requirements Engineering. Unlike traditional approaches based on linguistic rules, feature engineering or task-specific models, LLMs are trained on data and offer flexible language understanding and generation capabilities. They support many requirements-related tasks such as requirements elicitation, analysis, specification, and user story generation [45].

LLMs can also support the evaluation of user stories. For example, Ronanki et al. [43] investigated the use of ChatGPT for user story quality assessment and reported substantial agreement with the human evaluations while highlighting issues with output stability and trustworthiness. However, user stories generated by LLMs may contain omissions, unsupported information, ambiguities or inconsistencies, and their results can be sensitive to prompts and generation settings [46].

Therefore, a comprehensive assessment of LLMs for user story generation is necessary. Prompting strategies, including zero-shot and few-shot approaches, should be systematically evaluated. Automatic metrics should be supplemented with expert evaluation to address semantic fidelity, completeness, and hallucinations. Thus, LLMs can complement the existing NLP and machine learning methods, while also introducing new challenges in terms of reliability, reproducibility, and human oversight.

E. Evidence base and limitations

The evidence base of this review consists of 18 primary studies. Although relatively limited, this number reflects the specific scope of the review and the emerging nature of research on automated user story generation. The studies were selected according to predefined eligibility criteria, complemented by database searching and backward and forward snowballing. The selected studies also show considerable heterogeneity in terms of datasets, techniques, experimental settings, and evaluation criteria, which limits direct quantitative comparison.

Nevertheless, the selected studies provide sufficient evidence to identify recurring trends and research gaps, particularly the limited availability of public datasets and the lack of standardized evaluation practices. However, the small and heterogeneous evidence base may limit the generalizability of these findings across domains and industrial settings. Future reviews should therefore revisit these findings as the field evolves and more empirical studies, datasets, and LLM-based approaches become available.

VII. CONCLUSION

This systematic literature review examines existing research on automatic user story generation in software development, emphasizing the importance of user stories in the agile environment for requirements gathering. The review analyzed 18 papers, and explored the types of requirements corpora used, the natural language processing and machine learning approaches used, and the quality of the generated user stories. We found a lack of high-quality, publicly available corpora, which limits innovation and reproducibility. The reviewed studies employ diverse techniques, although many rely on established NLP methods. The review indicates opportunities for improving guidelines for validating user story quality. Overall, the results show that automatic user story generation is an emerging research topic with many future research opportunities. Future work should address publicly available and representative datasets, reproducible experimental protocols, standardized evaluation frameworks, and a thorough investigation of LLM-based generation and evaluation methods.

REFERENCES

- [1] Koch, A. Agile Software Development. Norwood, MA: Artech House, 2004.
- [2] Cohn, Mike. User Stories Applied: For Agile Software Development. Boston: Addison-Wesley Professional, 2004.
- [3] Beck, Kent. Extreme Programming Explained: Embrace Change. Boston: Addison-Wesley Professional, 2000.
- [4] Dalpiaz, Fabiano, Iris van der Schalk, Sjaak Brinkkemper, Fatma Başak Aydemir, and Garm Lucassen. "Detecting Terminological Ambiguity in User Stories: Tool and Experimentation." *Information and Software Technology* 110 (2019): 3–16.
- [5] Lucassen, Garm, Fabiano Dalpiaz, Jan Martijn E. M. van der Werf, and Sjaak Brinkkemper. "The Use and Effectiveness of User Stories in Practice." In *Requirements Engineering: Foundation for Software Quality: 22nd International Working Conference, REFSQ 2016, Gothenburg, Sweden, March 14–17, 2016, Proceedings*, 205–222. Cham: Springer, 2016.
- [6] Pokharel, Pradip, and Prashant Vaidya. "A Study of User Story in Practice." In *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, 1–5. Piscataway, NJ: IEEE, 2020.
- [7] Amna, A. R., and Geert Poels. "Systematic Literature Mapping of User Story Research." *IEEE Access* 10 (2022): 51723–51746.
- [8] Raharjana, I. K., Daniel Siahaan, and Chastine Fatichah. "User Stories and Natural Language Processing: A Systematic Literature Review." *IEEE Access* 9 (2021): 53811–53826.
- [9] Santos, C. A., Karine Bouchard, and Bruno Minetto Napoleão. "Automatic User Story Generation: A Comprehensive Systematic Literature Review." *International Journal of Data Science and Analytics* (2024): 1–24.
- [10] Cheliger, C., Jing Huang, Guohua Wu, Nadia Bhuiyan, Yan Xu, and Yong Zeng. "Machine Learning in Requirements Elicitation: A Literature Review." *AI EDAM* 36 (2022): 32.
- [11] Zhao, Lin, Waleed Alhoshan, Alessio Ferrari, Kabelo J. Letsholo, Mofolusho A. Ajagbe, Elena-Viorica Chioasca, and Riza Theresa Batista-Navarro. "Natural Language Processing for Requirements Engineering: A Systematic Mapping Study." *ACM Computing Surveys* 54, no. 3 (2021): 1–41.
- [12] Pollock, Alex, and Eivind Berge. "How to Do a Systematic Review." *International Journal of Stroke* 13, no. 2 (2018): 138–156.
- [13] Kitchenham, Barbara A., David Budgen, and Pearl Brereton. *Evidence-Based Software Engineering and Systematic Reviews*. Boca Raton, FL: CRC Press, 2015.
- [14] Amna, A. R., and Geert Poels. "Ambiguity in User Stories: A Systematic Literature Review." *Information and Software Technology* 145 (2022): 106824.
- [15] Wohlin, Claes. "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering." In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, 1–10. New York: ACM, 2014.
- [16] Schön, Eva-Maria, Jörg Thomaschewski, and María José Escalona. "Agile Requirements Engineering: A Systematic Literature Review." *Computer Standards & Interfaces* 49 (2017): 79–91.
- [17] Ferrari, Alessio, Felice Dell'Orletta, Andrea Esuli, Vincenzo Gervasi, and Stefania Gnesi. "Natural Language Requirements Processing: A 4D Vision." *IEEE Software* 34, no. 6 (2017): 28–35.
- [18] Fellbaum, Christiane, ed. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.
- [19] Castillo-Barrera, F. E., M. Amador-García, H. G. Pérez-González, F. E. Martínez-Pérez, and F. J. Torres-Reyes. "Adapting Bloom's Taxonomy for an Agile Classification of the Complexity of the User Stories in Scrum." In *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)*, 139–145. Piscataway, NJ: IEEE, 2018.
- [20] Rodeghero, Paige, Shaonan Jiang, Asher Armaly, and Collin McMillan. "Detecting User Story Information in Developer-Client Conversations to Generate Extractive Summaries." In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, 49–59. Piscataway, NJ: IEEE, 2017.

- [21] Veitía, F. J. P., L. Roldán, and M. Vegetti. "User Stories Identification in Software's Issues Records Using Natural Language Processing." In 2020 IEEE Congreso Biental de Argentina (ARGENCON), 1–7. Piscataway, NJ: IEEE, 2020.
- [22] Kumar, Bhaskar, Udit Tiwari, and Dinesh Chandra Dobhal. "User Story Splitting in Agile Software Development Using Machine Learning Approach." In 2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC), 167–171. Piscataway, NJ: IEEE, 2022.
- [23] Resketi, M. R., H. Motameni, H. Nematzadeh, and E. Akbari. "Automatic Summarising of User Stories in Order to Be Reused in Future Similar Projects." *IET Software* 14, no. 6 (2020): 711–723.
- [24] Heng, S., Konstantinos Tsilonis, and Yves Wautelet. "Building User Stories and Behavior Driven Development Scenarios with a Strict Set of Concepts: Ontology, Benefits and Primary Validation." In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, 1422–1429. New York: ACM, 2023.
- [25] Siahaan, Daniel, I. K. Raharjana, and Chastine Fatichah. "User Story Extraction from Natural Language for Requirements Elicitation: Identify Software-Related Information from Online News." *Information and Software Technology* 158 (2023): 107195.
- [26] Herwanto, G. B., G. Quirchmayr, and A. M. Tjoa. "Leveraging NLP Techniques for Privacy Requirements Engineering in User Stories." *IEEE Access* 12 (2024): 22167–22189.
- [27] Herwanto, G. B., G. Quirchmayr, and A. M. Tjoa. "A Named Entity Recognition Based Approach for Privacy Requirements Engineering." In 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW), 406–411. Piscataway, NJ: IEEE, 2021.
- [28] Raharjana, I. K., Daniel Siahaan, and Chastine Fatichah. "User Story Extraction from Online News for Software Requirements Elicitation: A Conceptual Model." In 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), 342–347. Piscataway, NJ: IEEE, 2019.
- [29] Thamrongchote, C., and Wanida Vatanawood. "Business Process Ontology for Defining User Story." In 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), 1–4. Piscataway, NJ: IEEE, 2016.
- [30] Mateus, D., Daniel S. Silveira, and João Araújo. "A Systematic Approach to Derive User Stories and Gherkin Scenarios from BPMN Models." In *International Symposium on Business Modeling and Software Design*, 235–244. Cham: Springer, 2023.
- [31] Henriksson, Andreas, and Jelena Zdravkovic. "A Data-Driven Framework for Automated Requirements Elicitation from Heterogeneous Digital Sources." In *IFIP Working Conference on The Practice of Enterprise Modeling*, 351–365. Cham: Springer, 2020.
- [32] Li, Y., H. Shibata, and Y. Takama. "Chatbot-Mediated Personal Daily Context Modeling upon User Story Graph." In 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), 1–6. Piscataway, NJ: IEEE, 2019.
- [33] Murtazina, M., and Tatiana Avdeenko. "An Ontology-Based Approach to the Agile Requirements Engineering." In *International Andrei Ershov Memorial Conference on Perspectives of System Informatics*, 205–213. Cham: Springer, 2019.
- [34] Lam, L. K., C. A. L. Hurtado, and L. W. Portillo. "Framework for Automating Requirement Elicitation Using a Chatbot." In 2022 IEEE Engineering International Research Conference (EIRCON), 1–4. Piscataway, NJ: IEEE, 2022.
- [35] Dwitama, F., and Albert Rusli. "User Stories Collection via Interactive Chatbot to Support Requirements Gathering." *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 18, no. 2 (2020): 890–898.
- [36] Santos, N., João Pereira, F. Morais, J. Barros, N. Ferreira, and Ricardo J. Machado. "Deriving User Stories for Distributed Scrum Teams from Iterative Refinement of Architectural Models." In *Proceedings of the 19th International Conference on Agile Software Development: Companion*, 1–4. New York: ACM, 2018.
- [37] Panichella, Sebastiano, and Marcelo Ruiz. "Requirements-Collector: Automating Requirements Specification from Elicitation Sessions and User Feedback." In 2020 IEEE 28th International Requirements Engineering Conference (RE), 404–407. Piscataway, NJ: IEEE, 2020.
- [38] Nistala, P. V., A. Rajbhoj, V. Kulkarni, S. Soni, K. V. Nori, and R. Reddy. "Towards Digitalization of Requirements: Generating Context-Sensitive User Stories from Diverse Specifications." *Automated Software Engineering* 29, no. 1 (2022): 26.
- [39] Aranda, G. N., Aurora Vizcaino, and Mario Piattini. "A Framework to Improve Communication During the Requirements Elicitation Process in GSD Projects." *Requirements Engineering* 15 (2010): 397–417.
- [40] Wagner, Stefan, Daniel Méndez Fernández, Michael Felderer, Antonio Vetrò, Marcos Kalinowski, Roel Wieringa, Dietmar Pfahl, Tayana Conte, Marie-Therese Christiansson, Des Greer, et al. "Status Quo in Requirements Engineering: A Theory and a Global Family of Surveys." *ACM Transactions on Software Engineering and Methodology* 28, no. 2 (2019): 1–48.
- [41] Kustiawan, Y. A., 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), 211–216. Piscataway, NJ: IEEE, 2023.
- [42] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, MN: Association for Computational Linguistics, 2019.
- [43] Ronanki, K., B. Cabrero-Daniel, and C. Berger. "ChatGPT as a Tool for User Story Quality Evaluation: Trustworthy Out of the Box?" In *International Conference on Agile Software Development*, 173–181. Cham: Springer, 2023.
- [44] Lucassen, Garm, Fabiano Dalpiaz, Jan Martijn E. van der Werf, and Sjaak Brinkkemper. "Improving Agile Requirements: The Quality User Story Framework and Tool." *Requirements Engineering* 21, no. 3 (2016): 383–403.
- [45] Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., & Wang, H. (2024). Large Language Models for Software Engineering: A Systematic Literature Review. *ACM Transactions on Software Engineering and Methodology*, 33(8), Article 220.
- [46] J. J. Norheim, E. Rebentisch, D. Xiao, L. Draeger, A. Kerbrat, and O. L. de Weck, "Challenges in applying large language models to requirements engineering tasks," *Design Sci.*, vol. 10, Art. no. e16, 2024, doi: 10.1017/dsj.2024.8