Analyzing User Story Quality: A Systematic Review of Common Issues and Solutions

João Vitor Oliveira[a] and Lisandra Manzoni Fontoura[b]
Programa de Pós-Graduação em Ciência da Computação, Federal University of Santa Maria (UFSM), Brazil

Keywords: User Stories, Issues, Solutions, Systematic Literature Review.

Abstract: User stories are widely adopted in agile development, serving as a fundamental technique for capturing and communicating software requirements. This paper aims to conduct a systematic literature review (SLR) to identify and analyze studies that address issues found in user stories, as well as possible ways to solve them. The primary motivation for this study was the advancement in the use of Large Language Models (LLMs), particularly after the launch of ChatGPT in 2022 by OpenAI. The research identified the main issues and solutions related to user stories between 2020 and 2024, focusing on issues related to user story quality. The results indicate that the most common issue in user stories is quality problems, cited in 15 articles, followed by requirements management and task assignment (12) and the derivation and generation of the conceptual model (8). Estimation is the least mentioned issue, appearing only three times. Regarding solution methods, researchers most frequently used Natural Language Processing, Machine Learning, and other Artificial Intelligence techniques, citing them in 15 articles. This demonstrates the well-established application of AI methods to address these challenges.

1 INTRODUCTION

User stories are widely adopted in agile development as an essential technique for capturing and communicating software requirements. Users or clients typically write these stories to describe their needs for the future software system, using natural language and a limited format. This method has become more popular due to its simplicity and ability to represent the user's perspective in a direct and comprehensible way (Wang et al., 2022).

Approximately 70% of the professionals follow the standardized format: "As a (persona), I want (action) so that (benefit)," which aids communication between stakeholders and development teams (Dalpiaz and Brinkkemper, 2021).

However, despite their widespread adoption, user stories often face challenges regarding clarity, completeness, and accuracy, which can undermine agile development effectiveness and result in software that only partially meets user needs (Wu et al., 2022). This paper conducts a systematic literature review (SLR) to identify and analyze these issues and explore opportunities for improving the use of user stories in agile development.

The remainder of this paper is organized as follows. Section 2 presents the protocol used to conduct the SLR. Section 3 describes and discusses the results, and Section 4 concludes the paper.

2 REVIEW PROTOCOL

We conducted an SLR to identify the main issues in specifications using user stories (US) and possible solutions. SLR is a research methodology that seeks to rigorously synthesize and analyze all available evidence on a given topic of interest. It is widely used in various fields, including software engineering (Kitchenham and Charters, 2007).

The SLR followed the protocol defined by Kitchenham and Charters (2007), which consists of three main phases: planning, conducting, and results reporting. Each phase contains suggested methods and procedures that must be followed to ensure that the study provides acceptable and significant results. Figure 1 provides a summarized view of the process carried out in the SLR.

[a] https://orcid.org/0009-0008-3582-5666
[b] https://orcid.org/0000-0002-4669-1383

Oliveira, J. V. and Fontoura, L. M.
Analyzing User Story Quality: A Systematic Review of Common Issues and Solutions.
DOI: 10.5220/0013218300003929

In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 2*, pages 152-159
ISBN: 978-989-758-749-8; ISSN: 2184-4992

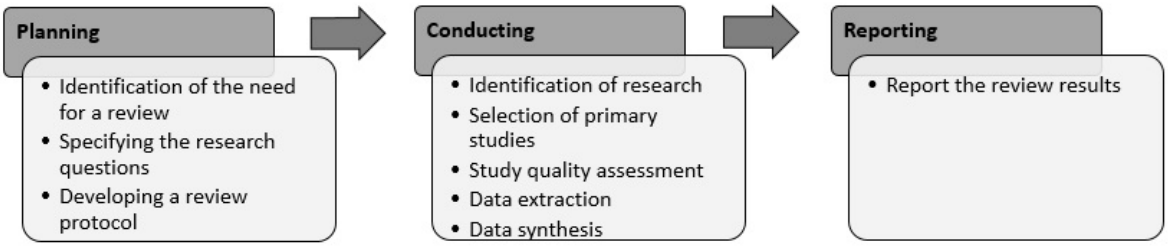

Figure 1: Phases of the process used in the SLR.

2.1 Planning

The SLR planning consisted of three main steps: identifying the need for the review, defining the research questions, and developing a review protocol.

In the first step, we highlighted gaps in the literature to justify the need for this study. In the second step, we defined two research questions to identify solutions to these issues. In the third step, we created a review protocol that outlined inclusion and exclusion criteria, search methods, and analysis procedures.

The review was necessary due to documented issues with user story clarity, completeness, and accuracy in agile development, coupled with the lack of comprehensive studies addressing these problems. Research questions guide researchers in focusing their studies by addressing specific questions to achieve their objectives. We have formulated two research questions to explore in this work.

RQ01. What issues related to user story specification did the study aim to solve?

RQ02. What solutions to the issues related to user story specification are described in the study?

The inclusion criteria targeted articles published between 2020 and 2024, available in relevant databases, and written in English while excluding articles not focused on user story specification. The review also considered the rise of AI methods after 2022, following the launch of ChatGPT by OpenAI, but adjusted the date range to ensure enough studies were included.

We used the following search string:

((“user story” OR “customer Story” OR “feature description” OR “feature Descriptions” OR “customer stories” OR “user stories”) AND (“issues” OR “defects” OR “difficult” OR “errors” OR “failures”)

2.2 Conducting

We conducted this systematic review using the Parsifal tool, defining inclusion and exclusion criteria to select relevant studies. The search across four databases—ACM, IEEE, Science Direct, and Scopus—yielded 888 articles, and we screened them based on the predefined criteria.

We evaluated the articles using acceptance criteria such as clarity in describing issues, relevance to the research, and, optionally, a description of solutions. After applying these criteria, we selected 40 articles for final analysis, which we considered the most relevant for understanding user story challenges in agile development. Table 1 summarizes the selected works.

Table 1: The total number of articles found in the SLR.

Databases	Initial search	Selected
ACM	329	4
IEE	101	6
Science Direct	386	11
Scopus	72	19
Total	888	40

3 RESULTS AND DISCUSSION

This section describes the main issues found in the systematic review. The issues were grouped based on the work of Kustiawan and Lim (2023) to facilitate understanding and more detailed analysis.

3.1 Research Question 01

To answer the question “RQ01: What issues related to user story specification did the study aim to solve?”, we classified the identified issues into four categories: User Story Quality, covering ambiguity and incomplete requirements (Section 3.1.1); Derivation and Generation of Conceptual Models, dealing with manual creation and conversion to pseudocode (Section 3.1.2); Requirements Management and Task Assignment, focused on managing requirements and legal demands (Section 3.1.3); and Estimation, addressing cost, time, and effort predictions (Section 3.1.4).

3.1.1 User Story Quality

User stories are essential for communicating requirements in agile projects, helping the development team

understand what needs to be delivered. However, as noted by Hallmann (2020), writing user stories can lead to formal errors such as ambiguities, inconsistencies, and lack of detail. These issues affect the quality of the stories and hinder the creation of a shared mental model among team members, increasing the risk of misunderstandings and project failure. Additionally, several studies suggest non-textual alternatives to address issues like unclear requirements and ambiguous user stories.

Furthermore, insufficient stakeholder involvement increases ambiguity, as unclear system descriptions lead to greater uncertainty about its exact requirements. Since the definition of requirements is carried out in an early stage of development, users are often unsure of what they want, as requirements tend to evolve through discussions and interactions among stakeholders and the technical team. Additionally, conventional elicitation methods are limited in terms of stakeholder participation and involvement, leaving room for more ambiguous and incomplete requirements (Da Silva and Savić, 2021), (Gralha et al., 2022), (Bakare et al., 2023).

Gupta et al. (2022) pointed out that although the agile method has advantages such as flexibility and fast deliveries, it faces challenges in managing requirements, particularly in ensuring clear and accurate communication between stakeholders and developers. The lack of formal and detailed documentation in agile processes can lead to ambiguities, rework, and misalignments with customer needs.

The work of Amna and Poels (2022) describes that ambiguity in natural language-based requirements, especially in user stories, is a problem recognized by the requirements engineering community due to linguistic and cognitive issues. While some studies see this ambiguity as intrinsic, there is a lack of standardization and a limited understanding of the cognitive factors that trigger it. An SLR by Amna (2022) highlighted three research gaps: the need for more studies on cognitive factors causing ambiguity, the low number of studies on ambiguities in related user stories, and the lack of approaches addressing ambiguity across linguistic levels.

Dar (2020) proposed using a gamification tool to address ambiguous and incomplete requirements, enhancing clarity, user engagement, and maintaining their interest.

Putri et al. (2023) highlighted that the semi-natural simplicity of user stories can lead to ambiguity, inconsistencies, and incomplete statements, potentially causing conflicts during development.

Xu et al. (2023a) noted that agile development with user stories often faces incomplete, inconsistent, and imprecise requirements, increasing workload and reducing efficiency in addressing user requirements.

Alhaizaey and Al-Mashari (2023) highlighted that poor definition or neglect of nonfunctional requirements is a common issue in agile projects. Limited research, overly complex proposals, and unclear integration into agile environments hinder their adoption.

Wang et al. (2022) explored user story quality assessment, noting that while studies often address grammatical defects, semantic defect verification remains underexplored. Dalpiaz and Brinkkemper (2021) identified linguistic defects as a quality issue, which, while easily avoidable, are present in 50% of real-world user stories, according to their study.

Thanomwong and Senivongse (2022) highlighted that neglecting risks from low-quality user stories early in development is a key factor in project failure, affecting both development and management.

3.1.2 Derivation and Generation of Conceptual Models

The section on conceptual model derivation explores transforming user stories into structured representations for better requirement understanding.

Bragilovski et al. (2022) identified a research gap in deriving user stories for conceptual design, while Javed and Lin (2021) focus on extracting ER models and business processes from informal requirements.

Gupta et al. (2023) proposed automating conceptual model generation from Behavior-Driven Development scenarios, addressing challenges in converting behavioral descriptions into formal models. Also, Gilson et al. (2020) addressed the problem of manually generating use case scenarios, which can be error-prone and time-consuming in agile development.

Dalpiaz et al. (2021) explored transforming natural language requirements into formal models like class diagrams, while Wu et al. (2022) automates the creation of iStar models from user stories, enabling analysis of objectives and actor dependencies.

Güneş and Aydemir (2020) proposed using NLP for automated goal model extraction from user stories, while Kolhatkar et al. (2023) focus on converting epics and user stories into pseudocode using transformers.

3.1.3 Requirements Management and Task Assignment

Requirements Engineering is experience-driven, manual, and involves various specification documents in different formats, such as business, functional, interface, and customer specifications, capturing critical product knowledge like features,

hierarchy, dependencies, and business context. Requirements specification is crucial for communication within the development team, as each role has distinct informational needs. The requirements engineer must provide relevant information to prevent errors caused by inadequate communication.

Specifying requirements is challenging due to the large number of user stories in modern software projects, complicating task management (Nistala et al., 2022; Oran et al., 2021; Yang et al., 2023). Vera-Rivera et al. (2020) highlight that optimized task assignment is crucial for project success, reducing development time and costs.

Nistala et al. (2022) proposed digitalizing requirements by generating context-sensitive user stories from diverse specifications addressing the challenge of standardizing and integrating various sources, which can lead to inconsistencies and errors.

Yang et al. (2023) addressed the challenge of classifying and grouping user stories, where manual organization becomes difficult as the number of stories grows. Without automated tools, this can lead to inefficiencies, prioritization issues, and increased complexity. They propose using machine learning (ML) techniques for user story clustering.

Tsilionis et al. (2021a) explored the effectiveness of conceptual modeling versus user story mapping, highlighting the lack of clarity on which approach best addresses communication, clarity, and organization challenges in agile projects.

Ferrara et al. (2024) highlighted the lack of systematic support for fairness requirements engineering, highlighting the challenge of managing fairness in AI systems due to the absence of specific tools in traditional requirements engineering.

Casillo et al. (2022) emphasized the importance of considering privacy attributes early in software development, noting that developers often lack the expertise to integrate legal and social data protection requirements. Villamizar et al. (2020) addressed the challenge of reviewing security in agile requirements for web applications, noting that security issues are often overlooked due to the focus on quick delivery.

Urbieta et al. (2020) argued that agile methods struggle with maintaining agility as requirements accumulate across sprints. Their work proposes to improve requirements traceability by consolidating dispersed information into a structured lexicon.

Siahaan et al. (2023) highlighted that extracting user stories from natural language improves requirements elicitation, but the approach remains limited.

3.1.4 Estimation

Agile models handle requirement changes flexibly, but frequent client requests can increase cost and time. Various cost estimation techniques aim to address inaccuracies, but precise estimates remain challenging (Butt et al., 2022). Factors like coordination, team size, complexity, and daily meeting issues significantly raise costs and time, especially in Scrum projects.

Effort estimation based on user stories is crucial for success, yet user story inconsistencies, technical complexities, and task practices contribute to inaccuracies. Butt et al. (2023) emphasized the need for standardized protocols to improve estimation accuracy.

Butt et al. (2022) proposed a cost estimation technique based on developer expertise in Scrum, while (Butt et al., 2023) focus on ML-based cost prediction. Iqbal et al. (2024) highlighted the use of subjective techniques like planning poker and size metrics for effort estimation.

3.2 RQ 02: What Is the Method for Solving this Problem?

Several articles apply NLP methods to address issues, as suggested by Raharjana et al. (2021), with objectives spanning defect detection to linking models and user stories. NLP studies in user stories aim to discover defects, generate software artifacts, identify core abstractions, and trace connections between the model and user stories.

3.2.1 User Story Quality

Hallmann (2020) linked user story quality to shared mental models and quality assessment criteria among team members. This approach aims to uncover correlations between quality variables, providing insights to enhance agile requirements development and understanding cognitive processes in comprehending user stories. This research can inform design recommendations and AI tools to improve user story quality, fostering better shared mental models and enhancing collaboration and success in agile projects.

To address learning gaps, Amna and Poels (2022) conducted a case study with 41 participants to compare four models for creating and understanding user stories. The results revealed significant differences in effectiveness, speed, and visual effort, but no model emerged as superior.

Dar (2020) addressed requirement ambiguity by developing a gamification tool to elicit clear requirements while ensuring user engagement and interest.

Putri et al. (2023) proposed using K-means and feature selection to improve clustering performance for user stories in agile development, with the Variance Threshold method enhancing results.

Xu et al. (2023a) proposed a quality assessment method for user story-based requirements, including a framework, model, and evaluation criteria. In another study (Xu et al., 2023b), they introduced a method to improve functional requirements quality in agile development through a functional requirements model and evaluation process.

Alhaizaey and Al-Mashari (2023) described a framework using NLP and AI techniques to automate the analysis and prediction of nonfunctional requirements in agile user stories.

Wang et al. (2022) proposed a multidimensional framework to evaluate user story quality, focusing on completeness, testability, and consistency. It combines NLP techniques with iStar modeling-based analysis to identify defects in these areas.

Gupta et al. (2022) focused on using conceptual models to address challenges in agile requirements engineering. These models visually represent system functionalities and rules, enhancing understanding and communication among stakeholders. The authors advocate integrating conceptual models with user stories to improve comprehension.

Thanomwong and Senivongse (2022) proposed a model for prioritizing risks in user stories to enhance risk awareness and mitigation. The model can be combined with methods such as the Analytical Hierarchy Process or weighted scoring to rank and prioritize risks associated with user stories.

Dalpiaz and Brinkkemper (2021) proposed tool-supported methods to improve user story creation, detailing the Quality User Story Framework and the AQUSA tool in a tutorial format.

3.2.2 Conceptual Model Derivation and Generation

Bragilovski et al. (2022) proposed deriving a holistic view of structural and interaction aspects from user stories, represented through class and use case diagrams. They conducted a controlled experiment with 77 undergraduate students using examples derived from user stories.

Javed and Lin (2021) proposed iMER (Iterative process of Entity Relationship and Business Process Model Extraction), an iterative process that automatically extracts ER and business process models from requirements using NLP and semantic analysis. The approach refines models as more information is processed, improving efficiency and accuracy while easing the workload of analysts and developers.

Gupta et al. (2023) proposed an approach to automatically generate conceptual models (e.g., class, use case, and activity diagrams) from BDD scenarios using NLP. The method extracts key information and converts it into visual representations, improving consistency, efficiency, and quality.

Gilson et al. (2020) described an automated approach to generate use case scenarios from user stories using transformation rules. Validated through experiments, the method enhances productivity and consistency by converting informal user stories into formal use-case scenarios.

Dalpiaz et al. (2021) evaluated techniques for deriving conceptual models from user requirements through experiments with professionals and students. The study compared automated and manual methods, assessing accuracy, completeness, and quality. Results showed that while automated approaches help, challenges remain in interpreting natural language requirements, with model quality varying by technique.

Wu et al. (2022) proposed an iStar model generation approach using NLP to streamline the extraction of relationships between user stories. The method involves extracting nodes from user stories and applying a BERT model to measure similarities between these nodes, facilitating the identification of relationships. This approach aims to reduce the time required for development teams to manually extract and map concepts from diverse user stories, ultimately enhancing efficiency and improving the structuring of requirements in agile projects.

Güneş and Aydemir (2020) described an automated method using NLP to extract goal models from user stories. The approach analyzes text to identify key elements like goals and objectives, converting them into formal models. Validated through experiments, it demonstrates time savings and improved accuracy in goal model creation, aiding system analysis and development in agile projects.

Kolhatkar et al. (2023) proposed a methodology to convert English problem descriptions into pseudocode using NLP. The process includes text-to-code and code-to-pseudocode conversion. The study found that the CodeT5 model achieved the highest BLEU score when trained separately on these tasks, with BLEU measuring the similarity between machine-translated and reference translations.

3.2.3 Requirements Management and Task Assignment

Nistala et al. (2022) proposed an automated approach using NLP and semantic analysis to digitize requirements and generate context-sensitive user stories, improving efficiency, accuracy, and consistency. Yang

et al. (2023) introduced a ML-based framework for clustering user stories, automating feature classification into similarity-based groups, and addressing the limitations of traditional clustering techniques.

Tsilionis et al. (2021a) compared conceptual modeling and user story mapping using a Rationale Tree to assess comprehensibility, requirement detection, and adaptability. Also, Tsilionis et al. (2021b) analyzed how rationale trees and user story maps improve software issue organization, clarity, and traceability, finding both techniques valuable, with strengths depending on project context and needs.

Oran et al. (2021) introduced the ReComP framework to address issues in requirement communication artifacts, specifying informational needs for each development role and suggesting improvements. Validated through independent studies, ReComP reduced user story issues by 77% and eliminated all issues in use cases identified by testers.

Özcelikkan et al. (2022) developed a multi-objective Scrum planning model that optimizes sprint capacity, prioritizes user stories, and clusters related stories using NSGA-II and SPEA2 heuristics. The model improves planning, estimation, and management by optimizing business value, time, cost, and quality. Ferrara et al. (2024) introduced ReFAIR, a contextual recommender that uses NLP to identify and incorporate fairness requirements into software projects, proving its effectiveness in experiments.

Casillo et al. (2022) proposed an approach combining NLP, linguistic features, and deep learning (DL) to identify privacy aspects in user stories. NLP extracted semantic and syntactic information, which was processed by a pre-trained convolutional neural network using transfer learning. The approach was evaluated with 1,680 user stories.

Vera-Rivera et al. (2020) aimed to automate and optimize user story assignments based on team members' experience (junior, senior, or novice). The proposed Java algorithm was validated in a real-world case study.

Urbieta et al. (2020) proposed a traceability approach using an index structure to access user stories. The method extracts and organizes scattered information into symbols within the Language Extended Lexicon (LEL).

Villamizar et al. (2020) proposed an approach to review security aspects in agile web application requirements by linking user stories to security properties using NLP.

3.2.4 Estimation

Butt et al. (2023) proposed a cost estimation technique based on user story complexity and developer experience, validated through two projects of varying sizes. Butt et al. (2022) described a framework to control cost overruns and schedule deviations, tested in several software industry case studies. Their results showed that the estimation technique improved project accuracy by mitigating development issues.

3.2.5 Summary

Based on the findings presented above, Table 2 summarizes the issues and solutions identified in the SLR.

The results from the SLR indicate that the most frequent issue in user stories is related to quality, with 15 works. This is followed by challenges in requirements management and task assignment (12 works) and derivation and generation of conceptual model (8). Estimation was the least frequently mentioned issue, appearing only three times, suggesting a potential gap that warrants further research.

Regarding solution methods, NLP, ML and AI were the most commonly employed, cited in 15 works, highlighting the significant role of AI in addressing these challenges.

It is important to note that some studies only describe problems without presenting solution methods. On the other hand, other studies only present solutions without describing the problems.

Table 2: Issues and solutions identified in the SLR.

Issues	Works	Solutions
User Story Quality	15	Systematic method (4) Case study (1) Gamification (1) NLP, ML, or AI (3) Conceptual models (2)
Derivation/ Generation of CM	8	Case Study (2) NLP, ML or AI (6)
Requirements Management and Task Assignment	12	NLP, ML or AI (6) Empirical experimentation (2) Framework (1) Multi-objective model (1) Algorithm (1) Traceability approach (1)
Estimation	3	Framework (2)

4 CONCLUSION

In this work, we conducted a Systematic Literature Review (SLR) from 2020 to 2024 to identify key problems and solutions in user stories. While the time

range may limit the analysis, it allowed us to focus on recent, relevant issues.

We found a balance between three main problem groups: User Story Quality, Derivation and Generation of Conceptual Models, and Requirements Management and Task Assignment, with a slight emphasis on quality-related issues. This highlights the growing importance of user story quality in practice.

Natural Language Processing (NLP) and AI techniques were frequently used to address these challenges. However, the review also revealed a range of alternative strategies, including systematic methods and frameworks, particularly for estimation problems. These methods complement AI solutions by emphasizing structured practices and frameworks for more precise requirement definition.

The main limitation of this study is the temporal scope, which may exclude earlier or more recent studies, affecting the comprehensiveness of the findings. Future work will use this knowledge to develop solutions for issues in user story specification.

ACKNOWLEDGEMENTS

We thank the Brazilian Army and its Army Strategic Program ASTROS for the financial support through the SIS-ASTROS GMF project (898347/2020).

REFERENCES

Alhaizaey, A. and Al-Mashari, M. (2023). A framework for reviewing and improving non-functional requirements in agile-based requirements. In *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–7. IEEE.

Amna, A. (2022). Identifying ambiguity problems in user stories: A proposed framework. In *CEUR Workshop Proceedings*, Belgium. CEUR Workshop Proceedings.

Amna, A. R. and Poels, G. (2022). Ambiguity in user stories: A systematic literature review. *Information and Software Technology*, 145:106824.

Bakare, A., Ciancarini, P., Farina, M., Kruglov, A., Okonicha, O., Smirnova, M., and Succi, G. (2023). Learning from west african storytellers. *Frontiers in Computer Science*, 5:1183602.

Bragilovski, M., Dalpiaz, F., and Sturm, A. (2022). Guided derivation of conceptual models from user stories: A controlled experiment. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 131–147, Cham. Springer International Publishing.

Butt, S. A., Ercan, T., Binsawad, M., Ariza-Colpas, P. P., Diaz-Martinez, J., Pineres-Espitia, G., et al. (2023). Prediction based cost estimation technique in agile development. *Advances in Engineering Software*, 175:103329.

Butt, S. A., Khalid, A., Ercan, T., Ariza-Colpas, P. P., Melisa, A. C., Piñeres-Espitia, G., and Ortega, R. M. (2022). A software-based cost estimation technique in scrum using a developer's expertise. *Advances in Engineering Software*, 171:103159.

Casillo, F., Deufemia, V., and Gravino, C. (2022). Detecting privacy requirements from user stories with nlp transfer learning models. *Information and Software Technology*, 146:106853.

Da Silva, A. R. and Savić, D. (2021). Linguistic patterns and linguistic styles for requirements specification: Focus on data entities. *Applied Sciences*, 11(9):4119.

Dalpiaz, F. and Brinkkemper, S. (2021). Agile requirements engineering: from user stories to software architectures. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pages 504–505. IEEE.

Dalpiaz, F., Gieske, P., and Sturm, A. (2021). On deriving conceptual models from user requirements: An empirical study. *Information and Software Technology*, 131:106484.

Dar, H. S. (2020). Reducing ambiguity in requirements elicitation via gamification. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 440–444. IEEE.

Ferrara, C., Casillo, F., Gravino, C., De Lucia, A., and Palomba, F. (2024). Refair: Toward a context-aware recommender for fairness requirements engineering. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pages 1–12. IEEE/ACM.

Gilson, F., Galster, M., and Georis, F. (2020). Generating use case scenarios from user stories. In *Proceedings of the International Conference on Software and System Processes*, pages 31–40. ACM.

Gralha, C., Pereira, R., Goulão, M., and Araujo, J. (2022). Assessing user stories: The influence of template differences and gender-related problem-solving styles. *Requirements Engineering*, 27(4):521–544.

Güneş, T. and Aydemir, F. B. (2020). Automated goal model extraction from user stories using nlp. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 382–387. IEEE.

Gupta, A., Poels, G., and Bera, P. (2022). Using conceptual models in agile software development: A possible solution to requirements engineering challenges in agile projects. *IEEE Access*, 10:119745–119766.

Gupta, A., Poels, G., and Bera, P. (2023). Generating multiple conceptual models from behavior-driven development scenarios. *Data & Knowledge Engineering*, 145:102141.

Hallmann, D. (2020). "i don't understand!": Toward a model to evaluate the role of user story quality. In *Agile Processes in Software Engineering and Extreme Programming: 21st International Conference on Agile Software Development, XP 2020, Copenhagen, Denmark, June 8–12, 2020, Proceedings*, pages 103–112. Springer International Publishing.

Iqbal, M., Ijaz, M., Mazhar, T., Shahzad, T., Abbas, Q., Ghadi, Y., et al. (2024). Exploring issues of story-based effort estimation in agile software development (asd). *Science of Computer Programming*, 236:103114.

Javed, M. and Lin, Y. (2021). imer: Iterative process of entity relationship and business process model extraction from the requirements. *Information and Software Technology*, 135:106558.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.

Kolhatkar, G., Madan, A., Kowtal, N., Roy, S., and Sonawane, S. (2023). Converting epics/stories into pseudocode using transformers. In *2023 IEEE 20th India Council International Conference (INDICON)*, pages 37–42. IEEE.

Kustiawan, Y. A. and Lim, T. Y. (2023). User stories in requirements elicitation: A systematic literature review. In *2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS)*, pages 211–216. IEEE.

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

Oran, A. C., Santos, G., Gadelha, B., and Conte, T. (2021). A framework for evaluating and improving requirements specifications based on the developers and testers perspective. *Requirements Engineering*, 26(4):481–508.

Özcelikkan, N., Tuzkaya, G., Alabas-Uslu, C., and Sennaroglu, B. (2022). A multi-objective agile project planning model and a comparative meta-heuristic approach. *Information and Software Technology*, 151:107023.

Putri, R. A., Yuhana, U. L., and Amri, T. C. (2023). K-means and feature selection mechanism to improve performance of clustering user stories in agile development. In *2023 International Conference on Modeling & E-Information Research, Artificial Learning and Digital Applications (ICMERALDA)*, pages 39–43. IEEE.

Raharjana, I. K., Siahaan, D., and Fatichah, C. (2021). User stories and natural language processing: A systematic literature review. *IEEE Access*, 9:53811–53826.

Siahaan, D., Raharjana, I. K., and Fatichah, C. (2023). User story extraction from natural language for requirements elicitation: Identify software-related information from online news. *Information and Software Technology*, 158:107195.

Thanomwong, P. and Senivongse, T. (2022). User story risk prioritization model for agile software development. In *2022 International Conference on Data and Software Engineering (ICoDSE)*, pages 161–166. IEEE.

Tsilionis, K., Maene, J., Heng, S., Wautelet, Y., and Poelmans, S. (2021a). Conceptual modeling versus user story mapping: Which is the best approach to agile requirements engineering? In *International Conference on Research Challenges in Information Science*, pages 356–373. Springer International Publishing.

Tsilionis, K., Maene, J., Heng, S., Wautelet, Y., and Poelmans, S. (2021b). Evaluating the software problem representation on the basis of rationale trees and user story maps: premises of an experiment. In *Software Business: 11th International Conference, ICSOB 2020, Karlskrona, Sweden, November 16–18, 2020, Proceedings 11*, pages 219–227. Springer International Publishing.

Urbieta, M., Antonelli, L., Rossi, G., and do Prado Leite, J. C. S. (2020). The impact of using a domain language for an agile requirements management. *Information and Software Technology*, 127:106375.

Vera-Rivera, F. H., Barbosa Mora, J. L., and Gaona, C. (2020). Generación automática de la planificación de la entrega en desarrollo de software agil, asignación de historias de usuario a los desarrolladores usando algoritmos genéticos. *AiBi Revista de Investigación, Administración e Ingeniería*, 8(2):29–38.

Villamizar, H., Kalinowski, M., Garcia, A., and Mendez, D. (2020). An efficient approach for reviewing security-related aspects in agile requirements specifications of web applications. *Requirements Engineering*, 25(4):439–468.

Wang, T., Wang, C., Li, T., Liu, Z., and Zhai, Y. (2022). User story quality assessment based on multi-dimensional perspective: A preliminary framework. In *iStar*, pages 7–13.

Wu, C., Wang, C., Li, T., and Zhai, Y. (2022). A node-merging based approach for generating istar models from user stories. In *Software Engineering and Knowledge Engineering*, pages 257–262.

Xu, X., Dou, Y., Qian, L., Jiang, J., Yang, K., and Tan, Y. (2023a). Quality improvement method for high-end equipment's functional requirements based on user stories. *Advanced Engineering Informatics*, 56:102017.

Xu, X., Dou, Y., Qian, L., Zhang, Z., Ma, Y., and Tan, Y. (2023b). A requirement quality assessment method based on user stories. *Electronics*, 12(10):2155.

Yang, B., Guo, H., and Liu, H. (2023). Evaluation and assessment of machine learning based user story grouping: A framework and empirical studies. *Science of Computer Programming*, 227:102943.