Issues Faced by Software Practitioners When Using User Stories: An Empirical Study Based on Stack Exchange Discussions

Nicole Guerreiro
Federal Institute of Ceará
Morada Nova, Ceará, Brazil
nicole.guerreiro09@aluno.ifce.edu.br

Yuri Nunes
Federal Institute of Ceará
Morada Nova, Ceará, Brazil
castro.nunes60@aluno.ifce.edu.br

Denivan Campos
Federal University of North Tocantins
Araguaína, Tocantins, Brazil
denivan.campos@ufnt.edu.br

Edna Dias Canedo
Computer Science Center
Federal University of Pernambuco
Recife, Pernambuco, Brazil
ednacanedo@cin.ufpe.br

Sávio Freire
Federal Institute of Ceará
Morada Nova, Ceará, Brazil
State University of Ceará
Fortaleza, Ceará, Brazil
savio.freire@ifce.edu.br

Abstract

Context. User story are widely used to describe software requirements. Although they reduce vagueness and ambiguity, issues associated with their use have been reported in the technical literature. Social media (e.g., blog posts and question-and-answer) platforms can be explored to understand how professionals share their experiences with user story issues; however, they have not yet been used to investigate such issues. **Goal**. This paper aims to identify the primary issues faced by software practitioners when using user story, their effects, and possible solutions. **Method**. We mined, curated, and selected 42 discussions from three Stack Exchange sites, a question-and-answer platform. We applied content analysis to derive issues, effects, and solutions from these discussions. **Results**. We identified eleven user story issues, of which seven are related to the artifact (e.g., lack of clarity and completeness) and four to process or use (e.g., difficulties in estimation). We also identified thirteen effects (e.g., communication and collaboration issues and misalignment with goals and prioritization), grouped into six categories. For each issue, we identified corresponding solutions (e.g., encouraging clarifying questions or applying quality standards to address the lack of clarity and completeness issue). **Conclusion**. Our findings can support practitioners in identifying user story issues and learning from others' experiences to address them, as well as motivate strategies to mitigate these issues in software projects.

Keywords

User Story, Issues, Agile Software Development, Requirements Engineering, Stack Exchange

1 Introduction

The adoption of agile methodologies has become widespread in the software industry [21]. Among their benefits, improved alignment with requirements contributes directly to increased customer satisfaction [28]. In this context, the Agile Manifesto provides guiding principles that support teams in improving the effectiveness of these methodologies [4].

User stories are a key artifact in agile development, enabling requirements to be expressed in a more precise and structured manner [30]. Typically written in semi-structured natural language, user story often follow templates such as "As a [user], I want [goal] because [reason]" [6]. Despite their widespread use, user stories are prone to several issues. Prior studies have identified problems such as ambiguity, incompleteness, and inconsistency in requirements elicitation [17]. To mitigate these issues, approaches such as quality frameworks [18] and artificial intelligence techniques [22] have been proposed. However, most existing studies investigate user story quality through controlled experiments, surveys, or proposed quality frameworks. Comparatively less attention has been given to understanding how practitioners discuss user story issues in their everyday work and how these issues manifest in real-world software development contexts.

This complementary perspective can be explored through social media, which encompasses diverse experiences and viewpoints, offering a valuable lens to complement traditional empirical methods in software engineering (SE) [10]. Question-and-answer communities are a prominent example, where practitioners share problems and receive solutions from peers. These platforms provide contextualized insights into challenges faced in practice, enabling analyses that are closer to the realities of the software industry [10].

Among these platforms, Stack Exchange[1] stands out due to its structured discussion model and thematic communities. It allows users to post questions and receive answers from a broad community, covering diverse SE topics. In particular, sites focused on software development capture a wide range of issues, from methodological concerns to interpersonal challenges within teams [11]. Previous research has leveraged Stack Exchange discussions to investigate practitioner-reported problems across multiple SE domains [8, 9, 11, 12, 24, 25].

In this paper, we aim to identify the primary issues faced by software practitioners when using user story and their corresponding effects and potential solutions. Then, we collected and curated 42 discussions from three Stack Exchange communities: Software Engineering[2], Project Management[3], and Software Quality Assurance & Testing[4]. We conducted both qualitative and quantitative analyses of these discussions, considering their questions and answers.

[1] https://stackexchange.com/
[2] https://softwareengineering.stackexchange.com/
[3] https://pm.stackexchange.com/
[4] https://sqa.stackexchange.com/

Our results reveal 11 distinct user story issues, with *Process/ workflow issues around user story* and *Lack of clarity and completeness* being the most frequently reported. These issues can be broadly categorized as artifact-related or process-related. We also identified 13 effects, among which *Estimation uncertainty and inaccuracy* and *Reduced delivery of independent and valuable increments* were the most prominent. In addition, we identified 89 possible solutions, which were grouped into categories. Finally, we analyzed the relationships between issues, effects, and solutions using Sankey diagrams [19].

Our findings have implications for both practitioners and researchers. For practitioners, the results indicate that improving user story goes beyond refining their textual quality, requiring improvements in development processes and team coordination. For researchers, the findings suggest that user story issues should be understood as a socio-technical phenomenon, opening opportunities for future studies on how to mitigate their effects and address these challenges in practice.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 describes the research methodology. Section 4 presents the results. Section 5 discusses the findings, including the relationships between issues, effects, and solutions, and compares them with related work. Section 6 addresses threats to validity. Finally, Section 7 concludes the paper and outlines future research directions.

2 Related Work

Lucassen et al. [18] proposed a framework with 13 quality criteria to identify ambiguities in user stories, along with the AQUSA tool for automatic defect detection using Natural Language Processing (NLP). Their empirical evaluation with 1,023 user stories from 18 companies showed that 56% contained defects, highlighting the prevalence of quality issues in practice. The tool achieved high recall (∼93%), demonstrating its potential to support quality assessment.

Bolloju et al. [5] conducted a controlled experiment showing that applying the Work System Snapshot improves the validity of user stories. Similarly, Khan et al. [14] identified key characteristics (such as clarity, independence, estimability, and value) that facilitate effort estimation in agile development.

Several studies have systematically investigated issues in user stories from a requirements engineering perspective. Amna and Poels [3] and Kustiawan and Lim [17] reported recurring problems such as ambiguity, incompleteness, and inconsistency, often associated with requirements elicitation, documentation, and negotiation activities. In addition, Amna and Poels [2] highlighted research gaps related to ambiguity, including the lack of studies addressing human and cognitive factors and the absence of holistic approaches that consider ambiguity across multiple levels and sets of related user stories. Complementarily, Amna [1] proposed a framework grounded in cognitive factors to support the identification of ambiguity issues, extending existing quality models with criteria to detect vagueness, inconsistency, insufficiency, and duplication.

From a solution-oriented perspective, Oliveira and Fontoura [22] found that quality issues remain the most frequently addressed problem, while artificial intelligence techniques (e.g., NLP and machine learning) are commonly used to address them. However, empirical evidence from practice shows persistent challenges: Kuhail and Lauesen [16] reported that user stories often fail to cover a large portion of stakeholder needs, are difficult to verify, and lack traceability.

Finally, dos Santos et al. [7] analyzed approaches for the automatic generation of user story, identifying a wide range of techniques, particularly based on NLP and machine learning, while noting that the area is still in an early stage of development. Although previous studies have investigated the quality, ambiguity, and automatic analysis of user stories, they have predominantly relied on quality frameworks, controlled experiments, literature reviews, and NLP-based approaches [15, 26]. Comparatively little attention has been paid to understanding how practitioners discuss user story issues in real-world software development settings, including the effects of these issues and the solutions adopted in practice. Question-and-answer communities provide an opportunity to investigate these aspects from practitioners' perspectives.

3 Research Methodology

This section describes the research questions (RQs) defined in this study, as well as the data collection and data analysis procedures. These procedures were based on previous studies that investigated distinct SE phenomena in Stack Exchange [8, 9, 11, 12, 24, 25].

3.1 Research Questions

We aimed to identify the primary issues faced by software practitioners when using user stories, their effects, and possible solutions. To achieve this goal, we defined the following RQs:

- **RQ_1 What issues are reported by software development practitioners when using user story in their activities?** The objective of this question is to identify the main issues related to the use of user story in software development processes, as reported by practitioners in the field. The issues analyzed will provide us with a deep understanding of the factors that contribute to their occurrence.
- **RQ_2 What are the primary effects experienced by software development practitioners due to user story issues?** This question aims to identify the effects associated with user story issues, as discussed by practitioners on Stack Exchange. Understanding these effects helps reveal how such issues impact practitioners' day-to-day work.
- **RQ_3 What are the main solutions suggested to address the reported issues?** This question aims to identify the solutions proposed by software development practitioners to deal with user story issues. Understanding these solutions can help other practitioners learn from shared experiences and select approaches that fit their context.

3.2 Data Collection

For this study, we selected three Stack Exchange communities: Project Management, Software Engineering, and Software Quality Assurance and Testing. These communities are structured around a main question, followed by answers and comments that form discussion threads. Additionally, in the communities, discussions are categorized using tags provided by the question author and receive user-based scores to indicate their relevance.

We selected these communities because: (i) Project Management Stack Exchange gathers Product Owners, Scrum Masters, and agile managers, representing the environment where user stories are created, refined, and prioritized; (ii) Software Engineering Stack Exchange includes developers and architects responsible for interpreting and implementing user stories to meet customer needs; and (iii) Software Quality Assurance and Testing Stack Exchange brings together testing and quality analysts, where user stories are used as acceptance criteria to validate whether implemented solutions meet requirements.

To collect discussions on user story issues, we used the Stack Exchange Data Explorer (SEDE)[5], a web-based tool that allows users to run SQL queries on publicly available Stack Exchange data. Through SEDE, we accessed posts from users reporting issues related to user stories in their software development processes.

We retrieved discussions containing terms such as "user story," and "user stories" combined with problem-related keywords such as "problem," "issue," "difficulty," "difficulties," "challenge," "risk," "drawback," "disadvantage," "obstacle," "shortcoming," "weakness," "complication," and "deficiency." A key challenge was ensuring that discussions genuinely addressed problems with user story rather than merely mentioning them in neutral or positive contexts. Therefore, we retained a broad set of keywords to capture diverse manifestations of practitioner-reported challenges.

The initial SQL query[6] returned 1,214 discussions. The dataset was then refined using the following criteria to improve quality and relevance:

- Criterion 1: Restricted the dataset to questions only and excluded entries without an associated user, ensuring posts were associated with identifiable users (924 discussions excluded). Through this criterion, we retained only posts associated with identifiable users to restrict the dataset to discussions created by participants with observable activity on Stack Exchange. Although authorship was not directly used in the analysis, this decision improved dataset consistency but may have excluded relevant discussions whose authors were no longer associated with an identifiable account.
- Criterion 2: Filtered out low-scoring discussions (Score > 0), retaining only posts with at least one positive vote (45 discussions excluded).
- Criterion 3: Ensured minimum engagement (CommentCount > 1), requiring at least two comments (123 discussions excluded).
- Criterion 4: Excluded posts from users with very low reputation (Reputation > 0), ensuring minimal credibility (no exclusions).
- Criterion 5: Removed incomplete discussions, defined as those lacking answers from users other than the original author (6 discussions excluded).
- Criterion 6: Removed duplicate discussions based on normalized titles (lowercase comparison), retaining only one instance per topic (no exclusions).

After applying these criteria, we obtained 52 discussions from three Stack Exchange communities.

[5]https://data.stackexchange.com/

[6]The query is available in our replication package [13].

3.3 Data Selection

To verify whether the discussions addressed user story issues, we defined the following inclusion (ICs) and exclusion (ECs) criteria: IC: Discussion that addresses issues related to user stories; EC1: Discussion that does NOT address problems with user story; and EC2: Discussion in a context outside the industry (company, team, project for industry).

First, a pilot study was conducted with nine discussions (three from each community). Two researchers independently applied these criteria to the dataset. A consensus meeting was then held with the participation of an experienced researcher in qualitative analysis of social media (e.g., Stack Exchange and Dev.to). During this meeting, disagreements were discussed to align the interpretation of the criteria. After the pilot, the researchers proceeded to the full analysis and subsequently met to compare their results. The agreement between the authors was assessed using Cohen's Kappa coefficient, yielding a value of 0.51, which indicates a moderate level of agreement according to commonly accepted thresholds. Disagreements were resolved by a third senior researcher, who also reviewed all classifications to ensure consistency.

At the end of this process, 22 discussions were retained from Project Management, 19 from Software Engineering, and 1 from Software Quality Assurance and Testing, totaling 42 discussions. To illustrate the selection process, one excluded discussion[7] described a developer who had been promoted to architect and became responsible for leading a team with limited experience and poor adherence to project planning. This discussion was excluded because it did not focus on concrete problems related to the use of user stories.

3.4 Data Analysis

For the analysis of the 42 discussions, we applied a qualitative content analysis approach [20] to identify textual segments related to user story issues, their corresponding effects, and possible solutions. Additionally, we extracted contextual information, such as the agile methodology, involved roles, and associated ceremonies.

Initially, we created a spreadsheet to store and organize the data. A calibration pilot study was then conducted with two researchers (the same researchers who applied the ICs and ECs). They independently analyzed ten randomly selected discussions to identify textual segments representing issues, agile methodologies, roles, ceremonies, effects (derived from questions), and solutions (derived from answers). Subsequently, a consensus meeting was held with an experienced qualitative researcher to compare the results, discuss discrepancies, and align decisions regarding textual segment extraction.

Following the pilot, the researchers performed the full analysis in batches of 10–12 discussions, conducting consensus meetings after each batch. At the end of this process, we identified 42 textual segments for issues, 16 for agile methodologies, 22 for ceremonies, 8 for roles, 42 for effects, and 262 for solutions.

3.4.1 Qualitative coding. Considering the extracted textual segments, one researcher performed manual open coding [27] by assigning a label to each segment, i.e., a word or phrase that captures

[7]https://softwareengineering.stackexchange.com/questions/106966/in-scrum-why-shouldnt-the-product-owner-and-scrummaster-roles-be-combined

the meaning of each segment. To illustrate this process, consider the following excerpts regarding effects: "This immediately raised the total points for the product of 20 points![...] those 20 points feel so awfully unrealistic!" and "it seems to me that if sprint points and task hours don't correlate well, then we must be doing a bad job at estimating either one or both of these." From these excerpts, we derived the initial codes *Inflated estimation* and *Poor estimation*.

We observed that several codes were semantically similar, enabling their unification. For instance, *Inflated estimation* and *Poor estimation* were merged into *Estimation uncertainty and inaccuracy*, as they represent effects related to inaccurate estimation. The extracted codes were submitted to ChatGPT for an initial semantic grouping using the following zero-shot prompt designed to identify semantically similar codes:

> *"You will act as a qualitative researcher supporting code analysis. I will provide a set of codes related to [problems / effects / solutions] associated with user stories. Your task is to perform an initial semantic grouping of the codes, bringing together those that represent similar or related concepts."*

During refinement, we identified three recurring issues: 1) inappropriate category assignments, e.g., *Unit testing increasing effort and causing Sprint failures* was moved from *Rework and delivery issues caused by user story problems* to *Process/workflow issues around user stories*; 2) duplicate assignments, e.g., *UX stories with no direct value for the user* appeared in both *UX/UI integration challenges around user stories* and *User stories lacking user value orientation*, and was consolidated into the latter; and 3) omitted codes, e.g., *Difficulty in delivery* was later assigned to *Workflow disruption and delivery blockage*. The resulting groups were iteratively reviewed and manually refined by an experienced researcher until conceptual consistency was achieved.

By analyzing the unified codes within each subset, we identified relationships that enabled grouping into categories. For issues, we distinguished two types: *user story artifact issues* (e.g., *Lack of clarity and completeness in user story*) and *user story process issues* (e.g., *Process/workflow issues around user story*). These types distinguish whether an issue is related to how user stories are written, structured, and documented, or to how workflows influence user story and vice versa. For effects, we grouped them into six categories; for example, *Delays and work blockages* and *Quality assurance and validation issues* effects were grouped under the *Quality* category. For solutions, we organized them into ten categories; for instance, *Process adaptation* includes 15 solutions, such as *Adapting and applying Agile/Scrum practices appropriately* and *Structuring and refining the product backlog*. Again, the grouping process was supported by ChatGPT and all results are checked by an experienced researcher.

3.4.2 Relationship between issues, effects, and solutions. Since issues and effects were extracted from the same questions, we assumed a direct relationship between them based on their co-occurrence within the same discussion context. Similarly, issues and solutions were linked through the question–answer structure of Stack Exchange discussions. These relationships were visualized using Sankey diagrams [19] and heap maps.

4 Results

In total, we analyzed 42 discussions on user stories published between 2011 and 2022. They were collected from three Stack Exchange communities: Project Management (22 discussions), Software Engineering (19 discussions), and Software Quality Assurance & Testing (1 discussion). The number of answers per discussion ranged from 1 to 8, totaling 107 answers (average = 2.5 per discussion). The number of comments ranged from 2 to 36, totaling 385 comments (average = 9.2 per discussion). Discussion scores ranged from 1 to 26, with a total score of 283 (average = 6.7 per discussion).

Regarding agile methodologies, Scrum was the most frequently reported, appearing in 50% of the discussions. Kanban, XP, Lean Startup, and Behavior-Driven Development (BDD) were each mentioned in 2.4% of the cases, while 38% of the discussions did not specify the agile methodology adopted. Concerning ceremonies, the discussions were associated with Sprint Planning (31%), Sprint Execution (12%), Release Planning (2%), and Backlog Refinement (2%), whereas 52% did not specify the ceremony. About roles, 12% of the discussions referred to the development team, 5% to the Scrum Master, and 2% to the Product Owner, while 81% did not specify any role.

The following subsections present the results for each RQ.

4.1 RQ1: Issues related to User Story

We identified 11 distinct issues related to user stories. Table 1 presents these issues by type (*User Story Artifact Issues* and *User Story Process Issues*), along with their number of mentions and percentage of total mentions.

Table 1: User story related issues

Issue type	Issue	#I	%I
User Story Artifact Issues	Lack of clarity and completeness in user story	5	12%
	User story lacking user value orientation	5	12%
	Poor decomposition and granularity of user story	3	7.1%
	Poorly defined or missing acceptance criteria	3	7.1%
	User story framed as solutions or technical tasks	3	7.1%
	Lack of product/domain knowledge	1	2.4%
	User story writing improvement need	1	2.4%
	Total	21	50%
User Story Process Issues	Process/workflow issues around user story	11	26.2%
	Difficulties in estimating user story	7	16.7%
	UX/UI integration challenges around user story	2	4.8%
	Rework and delivery issues caused by user Story problems	1	2.4%
	Total	21	50%

Legend:
#I: Total number of mentions.
%I: Percentage of #I in relation to the mentions total (42).

The distribution shows that practitioners discuss issues both at the artifact level, i.e., how user stories are written, structured, and documented; and at the process level, reflecting how workflows

influence user stories and vice versa. Although both types have the same number of mentions, a greater variety of issues is associated with user story artifacts than with process-related aspects.

Regarding individual issues, *Process/workflow issues around user story* is the most frequently mentioned, followed by *Difficulties in estimating user story*, *Lack of clarity and completeness in user story*, and *user story lacking user value orientation*.

In this study, *Process/workflow issues around user story* refer to difficulties in organizing, applying, and adapting agile practices involving user stories within the workflow, as illustrated in "Here it can get really difficult to relate tasks and user story. Are there techniques to overcome this issue or is it just something we have to accept and make the best of it?" *Difficulties in estimating user story* concerns challenges in estimating effort, cost, and time, as shown in "I noticed that the correlation between story points and task hours is not very good". *Lack of clarity and completeness in user story* captures insufficient or poorly specified requirements, for example "there are more ways to write bad/unusable user story than writing really good ones. And it's easy to get to the wrong tracks even if you're careful." Finally, *user story lacking user value orientation* refers to interdependent stories or those lacking clear user-facing value, as illustrated in "Is it ok to define a user story that has no real business value without another story?"

Finding #1: When software practitioners discuss user story, they raise concerns about their structure and their interaction with development processes. The most discussed issues are *Process/workflow issues around user story*, *Difficulties in estimating user story*, and *Lack of clarity and completeness in user story*.

4.2 RQ2: Effects related to User Story Issues

In total, we found 13 effects discussed by software practitioners due to user story issues. Table 2 shows the effects divided into categories, along with their number of mentions and percentage of total mentions. The most frequently mentioned effects are *Estimation uncertainty and inaccuracy* (10 mentions; 24.4%), *Reduced delivery of independent and valuable increments* (6; 14.6%), and *Organizational and backlog management* (4; 9.8%).

Estimation uncertainty and inaccuracy refers to issues such as bugs, unforeseen complex tasks, and inaccurate or inflated estimations, as illustrated by: "This immediately raised up the total points for the product of 20 points!" *Reduced delivery of independent and valuable increments* captures the absence of standalone value and dependencies among user story, as shown in: "Releasing the product with only US1 would provide the capability to design card views that are useless without US2." and "If we build just story 1 it seems not to be valuable. If we build just story 2 [...] provides no value." Lastly, *Organizational and backlog management* relates to difficulties with disorganized backlogs, loss of traceability, and task duplication, for example: "Then comes the chaos. [...] sub-tasks add an extra complexity."

Table 2: Effects related to user story issues

Category	Effect	#E	%E
Planning & Estimation	Estimation uncertainty and inaccuracy	10	24.4%
Product & Value	Reduced delivery of independent and valuable increments	6	14.6%
	Misalignment with product goals and priorities	2	4.9%
Process & Execution	Sprint instability and unreliable planning	3	7.3%
	Workflow disruptions and delivery blockages	3	7.3%
	Delays and work blockages	1	2.4%
	Inconsistent adherence to development practices	1	2.4%
Quality	Delays and work blockages	3	7.3%
	Quality assurance and validation	3	7.3%
Human & Social	Breakdowns in team communication and collaboration	3	7.3%
	Increased rework and time pressure	2	4.9%
	Reduced autonomy and professional motivation	1	2.4%
Organizational	Organizational and backlog management	4	9.8%

Legend:
#E: Total number of mentions.
%E: Percentage of #E in relation to the mentions total (42).

Finding #2: Due to user story issues, software practitioners report impacts mainly on task estimation, value delivery, and managerial aspects.

As shown in Table 2, we grouped the effects into the following categories:

- **Planning & Estimation**: Includes *Estimation uncertainty and inaccuracy*, reflecting difficulties in estimating bugs and unforeseen complex tasks.
- **Product & Value**: Encompasses *Misalignment with product goals and priorities* and *Reduced delivery of independent and valuable increments*, indicating misalignment with objectives and limited value generation.
- **Process & Execution**: Includes *Workflow disruptions and delivery blockages*, *Sprint instability and unreliable planning*, *Inconsistent adherence to development practices*, and *Delays and work blockages*, capturing impacts on workflow and agile execution.
- **Quality**: Groups *Reduced quality assurance and validation effectiveness* and *Poor task decomposition and structuring*, reflecting challenges in quality assurance and user story breakdown.
- **Human & Social**: Includes *Increased rework and time pressure*, *Reduced autonomy and professional motivation*, and *Breakdowns in team communication and collaboration*, highlighting team-level impacts.
- **Organizational**: Contains *Inefficient backlog organization and management*, affecting organizational and managerial aspects.

Finding #3: User story issues lead to various effects perceived by software practitioners. Although these effects are primarily related to *Planning & estimation*, *Product & value*, and *Process Execution*, we also identified effects in the *Human & social* dimension.

4.2.1 Relationship between User Story Issues and Effects. Figure 1 presents a Sankey diagram relating the four most frequently mentioned user story issues to their corresponding effects. *Process/workflow issues around user story* is associated with the largest number of observed effects. While most effects are shared across multiple issues, some are uniquely linked to a single issue, including *Workflow disruption and delivery blockage*, *Increased rework and time pressure*, *Misalignment with product goals and priorities*, *Delays and work blockages*, *Poor task decomposition and structuring*, *Reduced quality assurance and validation effectiveness*, and *Inconsistent adherence to development practices*.

An illustrative example of the relationship between *Lack of clarity and completeness in user stories* and *Misalignment with product goals and priorities* emerged within the same discussion thread. One practitioner noted that "there are more ways to write bad/unusable user stories than writing really good ones. And it's easy to get to the wrong tracks even if you're careful" and further explained that "If you don't have lots of experience with user stories that may hinder your product because they distract you from the actual goal." Taken together, these statements indicate how ambiguity in requirements artifacts can hinder proper understanding of product objectives.

Figure 1: Relationship between the four most mentioned user story issues and their corresponding effects

We also analyzed the relationship between user story issue types and their effects (Figure 2). Most effects are associated with both issue types. However, *Inconsistent adherence to development practices*, *Misalignment with product goals and priorities*, and *Reduced autonomy and professional motivation* are linked only to user story artifact issues, while *Delays and work blockages* and *Sprint instability and unreliable planning* are associated with user story process issues.

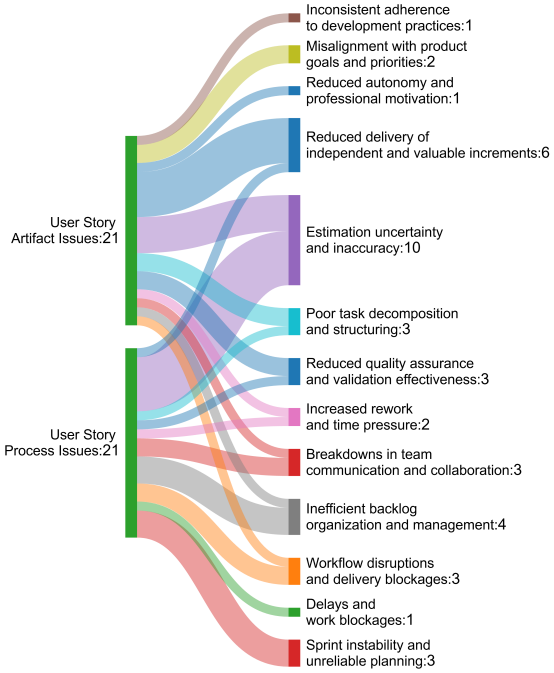

Figure 2: Relationship between user story issue types and effects related to user story

Lastly, we related user story issue types to effect categories. Figure 3 presents the Sankey diagram, showing that although both issue types are associated with all effect categories, *user story process issues* are more strongly linked to *organizational* and *process* effects, while *user story artifact issues* are more associated with *product & value* and *quality* effects.

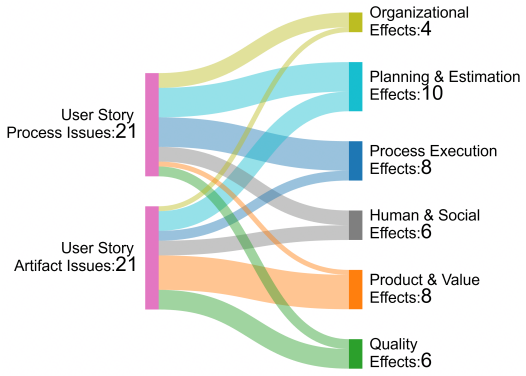

Figure 3: Relationship between user story issues types and effect categories

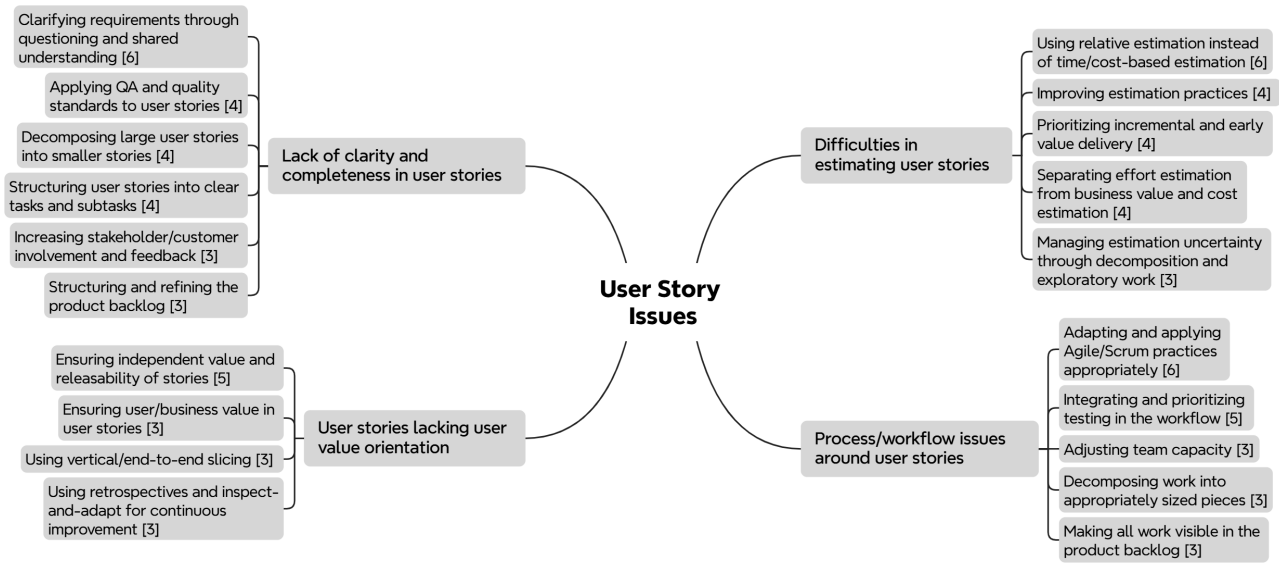

Figure 4: Relationship between the four most frequently reported user story issues and their corresponding solutions with at least three occurrences. The number in brackets indicates the number of occurrences.

Finding #4: User story issues types lead to distinct effects: artifact-related issues primarily compromise product quality and value, whereas process-related issues mainly affect workflow and organizational aspects.

Finding #5: The relationship between user story issues and their potential solutions shows that effective mitigation depends on understanding the specific characteristics of each issue rather than applying generic approaches.

4.3 RQ3: Possible Solutions for User Story Issues

We identified 89 possible solutions to address user story issues reported by practitioners, all of which are available in our replication package [13]. Figure 4 presents a mind map illustrating the relationship between the four most frequently reported user story issues and their corresponding solutions, considering only those with at least three occurrences.

A closer inspection of Figure 4, we observe that addressing *Lack of clarity and completeness in user story* involves practices such as *Clarifying requirements through questioning and shared understanding*, as illustrated in "Clarify requirements through questioning and rapid prototyping" and "Developers must ask clarifying questions to understand requirements before implementation." Similarly, to address *user story lacking value orientation*, practitioners emphasize *Ensuring independent value and releasability of stories*, as reflected in "each story should be valuable independently" and "keeping dependent stories separate and delivering potentially shippable product."

For *Difficulties in estimating user story*, practitioners apply *Using relative estimation instead of time/cost-based estimation*, as shown in "Use story points as relative estimates, not as time or cost equivalents" and "point value determined by environmental factors and team, not time or cost." Lastly, to address *Process/workflow issues around user story*, they emphasize *Adapting and applying Agile/Scrum practices appropriately*, for example "scrum enabling self-directed team with no universal right way to write stories" and "figuring out what works best for your team, product, and environment."

We grouped the possible solutions into the following categories:

- **Process adaptation**: it groups 15 solutions related to adapting the development process to better align with the project context and delivery constraints. Examples of solutions are *Adapting and applying Agile/Scrum practices appropriately* and *Structuring and refining the product backlog*.
- **Effort estimation & planning**: it includes 13 solutions to support effort estimation and improve development planning, such as *Using relative estimation instead of time/cost-based estimation* and *Managing estimation uncertainty through decomposition and exploratory work*.
- **Work structuring**: it has 11 solutions to organize work and enable incremental development and clearer ownership, for example, *Structuring user story into clear tasks and sub-tasks* and *Decomposing work into appropriately sized pieces*.
- **Requirements understanding**: it groups 11 solutions focused on understanding requirements, user needs, and problem context. Examples include *Clarifying requirements through questioning and shared understanding* and *Using supporting artifacts to clarify and derive user story*.
- **Collaboration & alignment**: it includes 9 solutions to foster communication, shared understanding, and alignment with stakeholders, such as *Improving team collaboration, ownership, and resilience* and *Increasing stakeholder/ customer involvement and feedback*.

- **Design–development alignment**: it encompasses 8 solutions to synchronize design with development activities, for instance *Managing UX/UI work within or across sprints* and *Splitting functionality and design into separate stories.*
- **Value orientation**: it includes 7 solutions to guide development decisions toward increasing user and business value, for example *Ensuring user/business value in user story* and *Aligning delivery and release decisions with business value.*
- **Quality assurance**: it groups 6 solutions to ensure that user stories are complete, correct, and testable. *Integrating and prioritizing testing in the workflow* and *Applying QA and quality standards to user story* are examples of these solutions.
- **Continuous learning**: it has 4 solutions that allow teams to iteratively learn through feedback and adaptation, such as *Using creative problem-solving techniques* and *Using retrospectives and inspect-and-adapt for continuous improvement.*
- **Technical foundations**: it includes 4 solutions related to technical practices for implementing and delivering user story, for example *Reducing upfront design overhead* and *Managing technical debt explicitly.*

We also related the user story issue types with solution categories (See Figure 5). We can see that *Work structuring* and *Requirements understanding* are more related to *user story artifact issues*, while *Technical foundations* is only related to *user story artifact issues.*

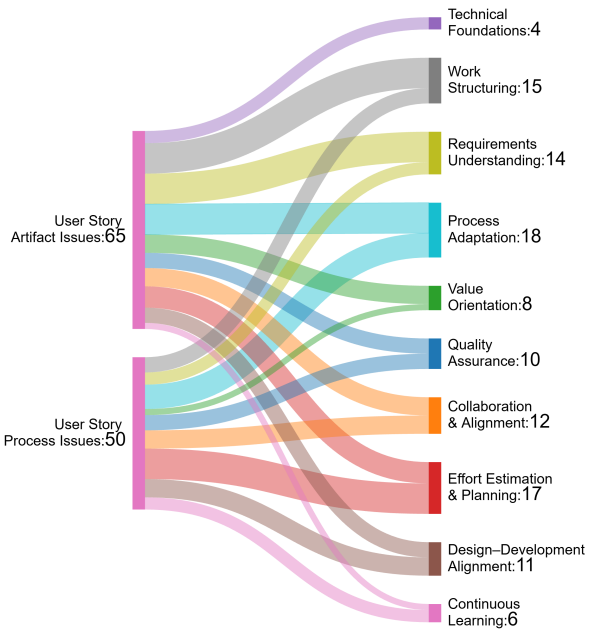

Figure 5: Relationship between user story issues types and solution categories

Finding #6: *Process adaptation* and *Effort estimation & planning* play a central role in addressing user story issues, as they are shared across both issue types.

5 Discussion

In this section, we discuss the main findings of our study with respect to the RQs, as well, compare the identified issues with related work and highlight the unique contributions of our study.

5.1 Revisiting the Research Questions

In the following, we revisit the RQs and discuss the broader implications of our findings.

RQ$_1$: What user story issues are reported by practitioners? Our findings suggest that practitioners perceive user story issues as both artifact-related and process-related concerns. The two issue types appeared with the same frequency (See Table 1), indicating that practitioners do not view user story problems solely as deficiencies in requirements documentation. While previous research has predominantly focused on improving user story quality through better writing, completeness, and consistency, the discussions analyzed in this study reveal that many challenges emerge from planning, coordination, estimation, backlog management, and workflow practices. This suggests that improving user stories requires attention not only to the artifact itself but also to the organizational and process context in which it is created and used.

RQ$_2$: What are the effects of user story issues? The identified effects demonstrate that user story issues have consequences that extend beyond requirements quality. Practitioners frequently associated these issues with estimation uncertainty, reduced value delivery, and organizational challenges, indicating that deficiencies in user stories may propagate throughout the development lifecycle. Moreover, the presence of effects in the *Human & Social* category suggests that user story issues can influence collaboration, communication, and motivation within teams. These findings reinforce the view that user stories play a central role in coordinating development activities and that their deficiencies may affect both technical and organizational dimensions of software projects.

RQ$_3$: What solutions are proposed to address user story issues? The proposed solutions reveal that practitioners rarely rely on a single strategy to mitigate user story issues. Instead, they combine practices related to process adaptation, requirements understanding, collaboration, work structuring, and value orientation. This diversity of solutions suggests that user story issues are context-dependent and cannot be addressed exclusively through artifact-focused interventions. Rather, effective mitigation requires a combination of technical, organizational, and communication-oriented practices that consider both the user story artifact and the surrounding development process.

5.2 Relating User Story Issues, Effects, and Solutions

Figure 6 presents two heat maps relating effect categories and solution categories to each type of user story issue. The maps show that solutions are not applied uniformly across effect categories but instead follow structured patterns shaped by both the type of effect and the nature of the underlying issue. Process-related user story issues are primarily associated with *Process & Workflow* effects, whereas artifact-related issues are more frequently linked to *Product & Value* effects. However, deficiencies in user story artifacts

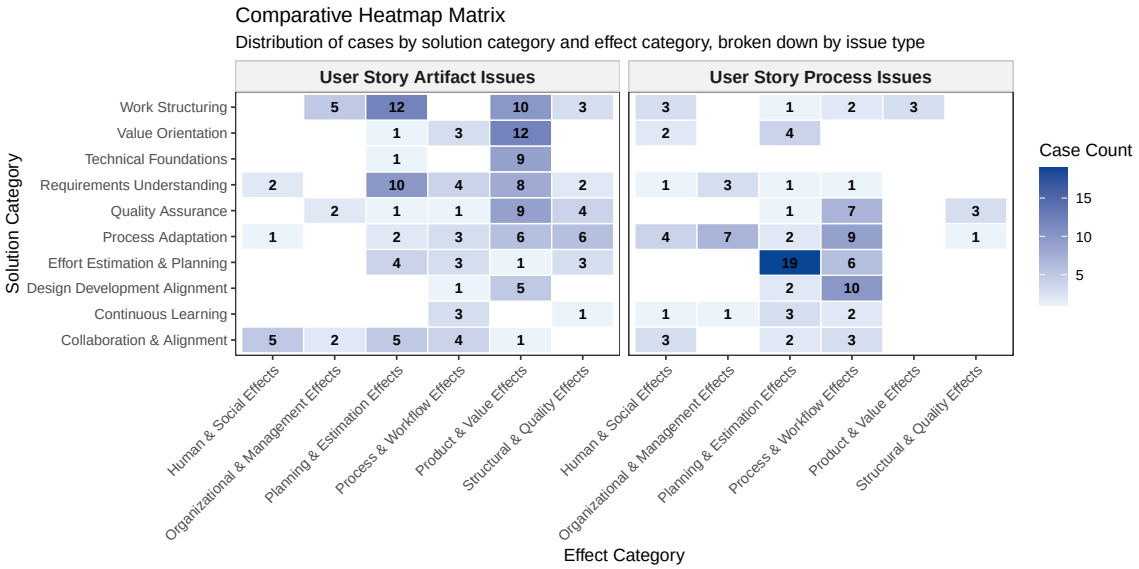

Figure 6: Comparative heap map relating user story issue types and their corresponding effect categories and solution categories

can propagate to process-level activities, amplifying their effects throughout the software development lifecycle.

The maps also show that user story artifact issues are addressed primarily through *Work Structuring* and *Requirements Understanding*, whereas solutions for user story process issues are concentrated in *Effort Estimation & Planning* and *Process Adaptation*. More broadly, the diversity of solution categories suggests that practitioners address user story issues through multiple complementary types of intervention rather than relying on a single approach. These patterns reinforce the view that user story issues are inherently sociotechnical, involving both artifact quality and the organizational context in which user stories are created, refined, and implemented.

5.3 Comparison with Related Work

Related work has addressed user story through systematic literature reviews [1–3, 14, 17], empirical studies [5], and proposed solutions [5, 14, 18]. In contrast, our work adopts a qualitative approach grounded in practitioner-generated data from Stack Exchange, capturing problems that may not emerge in controlled or formal studies, as well as the solutions proposed by practitioners. Furthermore, we explicitly relate issues, effects, and solutions by leveraging the question–answer structure, where solutions (answers) are naturally connected to specific problems (questions).

Regarding user story issues, prior studies have shown that user stories are often difficult to verify and fail to cover a substantial portion of project requirements [16]. Kustiawan and Lim [17] identified a broad set of issues, including ambiguity, incompleteness, inconsistency, duplication, maintainability challenges, and difficulties in estimation and negotiation. Similarly, Amna and Poels [3] highlighted issues such as inconsistency, insufficient requirement description, and duplication of functionality while Amna and Poels [2] reported ambiguity at different linguistic levels and Amna [1] focused on user story quality. In addition, Khan et al. [14] focused on characteristics of user story that affect effort estimation. Overall, these studies predominantly focus on the user story artifact itself. In contrast, our findings reveal that issues also emerge from the development process, introducing the notion of user story process issues. Only Amna and Poels [2] emphasized effects of user story issues, reporting communication, estimates, and value delivery effects.

Concerning solutions, prior work has largely focused on improving the quality of user story writing. For example, Kuhail and Lauesen [16] emphasized the importance of expressing user problems rather than predefined solutions. Kustiawan and Lim [17] identified solution approaches such as models, tools, techniques, frameworks, and taxonomies. Amna and Poels [3] reported solutions including ontologies, controlled languages, templates, and methods to reduce ambiguity and improve requirement quality. Additionally, Bolloju et al. [5] demonstrated that using a work system snapshot before defining user story improves their validity. While these solutions are primarily artifact-oriented, our findings suggest that addressing user story issues requires a broader socio-technical perspective that encompasses both artifact and process dimensions.

A key contribution of this study is the shift from an artifact-centered perspective to a broader sociotechnical view of user story issues. While prior research has focused mainly on clarity, completeness, consistency, and estimability, practitioners frequently discussed these issues in relation to planning, backlog management, stakeholder communication, workflow coordination, and estimation. Thus, user story issues arise not only from artifact deficiencies but also from their interaction with the development process.

In summary, our findings confirm several issues previously reported in the literature, such as ambiguity, incompleteness, and estimation challenges, while extending current knowledge in three ways. First, we reveal that practitioners discuss process-related issues as frequently as artifact-related issues. Second, we show how these issues are associated with specific effects perceived during

software development activities. Third, we provide an integrated perspective that connects issues, effects, and practitioner-proposed solutions. Together, these contributions offer a more comprehensive understanding of user story challenges in software projects.

5.4 Implications for Practitioners and Researchers

This study has implications for both practitioners and researchers. For practitioners, the results indicate that improving user stories goes beyond refining their writing; it requires investment in process adaptation and team coordination. Moreover, different types of issues demand different interventions. For example, artifact-related issues call for requirements clarification and quality assurance, whereas process-related issues require improved workflow management and estimation practices. The relationships between issues, effects, and solutions provide a useful starting point for practitioners to learn from others' experiences. Although the specific contexts in which these solutions were discussed cannot be fully determined, practitioners can analyze them and select those most suitable for their projects.

For researchers, the findings suggest that user story issues should be understood as a socio-technical phenomenon, as they involve both artifact- and process-related dimensions. From this perspective, future studies can propose integrated approaches that consider issues, effects, and solutions, rather than focusing solely on the quality of user story. Our findings complement the INVEST framework (Independent, Negotiable, Valuable, Estimable, Small, and Testable) [29], indicating that it is necessary to consider the broader context surrounding user story, not only their content but also their surrounding processes. Thus, our results can serve as a starting point for investigating user story issues while incorporating human and organizational factors.

6 Trustworthiness of the Study

We follow the guidelines of the Empirical Standards for SE [23] to ensure the rigor and reliability of this study. To establish **credibility**, we strengthened the study's findings by applying a well-defined search string to retrieve discussions from Stack Exchange communities (Software Engineering, Project Management, and Software Quality Assurance and Testing). Although many retrieved discussions did not address user story issues in practical settings, we applied consistent inclusion and exclusion criteria to select those aligned with the study's goal. Interpretation bias was mitigated through researcher triangulation, including a pilot study, independent paired analysis, and consensus meetings. We also used ChatGPT as a support tool to assist in the grouping process; potential hallucination-related threats were mitigated by using a fixed prompt and manually validating the results. This validation was conducted by an experienced researcher in qualitative studies in SE. As we limited the dataset to users with observable Stack Exchange activity, this approach may have excluded relevant discussions from unidentifiable accounts.

Confirmability was reinforced by adopting an open coding process in which codes were derived directly from the data, and by incorporating peer review to ensure that interpretations were consistently scrutinized and validated.

Regarding **transferability**, we acknowledge that the findings may not be directly generalizable to all SE contexts. The use of Stack Exchange introduces inherent limitations in accurately characterizing participants. This limitation became evident when attempting to report the agile methodologies, roles, and ceremonies involved in the discussions. Although this information was not available for all discussions, we observed that most cases referred to Scrum. Therefore, the findings should be interpreted as a characterization of community-validated user story issues rather than universally applicable behaviors across diverse organizational contexts. Finally, **dependability** was established through a transparent and well-documented research protocol (Section 3) and the availability of a replication package [13], which enables independent auditing of the research process.

7 Conclusion

This paper investigates the issues reported by software practitioners when using user stories in software development, examining not only the issues themselves, but also their associated effects and the solutions proposed in practice. By analyzing real-world discussions from Stack Exchange communities, we identify a diverse set of issues and reveal that they extend beyond artifact-level concerns to include process-related challenges embedded in development workflows.Our findings show that user story issues are not isolated problems, but part of a broader socio-technical phenomenon, where deficiencies in artifacts, processes, and communication practices interact and reinforce one another. In particular, the relationships identified between issues, effects, and solutions highlight that effective mitigation requires a contextual understanding of how these elements are interconnected, rather than the application of isolated best practices.

By structuring and relating issues, effects, and solutions, this study contributes to bridging the gap between academic research and practitioners' experiences, providing a more comprehensive perspective on user story challenges in real-world settings. These insights can support practitioners in reflecting on their practices and guide researchers in developing more effective approaches to improve requirements quality in agile environments. As future work, we intend to (i) collect additional evidence of user story issues from other social media platforms, such as Dev.to and Medium; (ii) conduct interviews with software practitioners to better understand how the identified solutions are applied in practice; and (iii) perform an industrial case study to investigate the contextual factors that influence the emergence of user story issues.

Artifact Availability

All artifacts related to this study are available at https://zenodo.org/records/21400048. The artifacts will remain publicly available after publication.

Acknowledgments

This study was financed in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). We used Large Language Models for language refinement, but all analysis and interpretation remain our sole responsibility.

References

[1] AR Amna. 2022. Identifying ambiguity problems in user stories: a proposed framework. In *Proceedings of the doctoral consortium papers presented at the 34th international conference on advanced information systems engineering (CAiSE 2022). CEUR Workshop Proceedings*.

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

[3] Anis R Amna and Geert Poels. 2022. Systematic literature mapping of user story research. *IEEE access* 10 (2022), 51723–51746.

[4] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. 2001. Manifesto for agile software development. (2001).

[5] Narasimha Bolloju, Steven Alter, Anupriya Gupta, Simran Gupta, and Sanyam Jain. 2017. Improving scrum user stories and product backlog using work system snapshots. (2017).

[6] Joelma Choma, Luciana AM Zaina, and Daniela Beraldo. 2016. UserX story: Incorporating UX aspects into user stories elaboration. In *International Conference on Human-Computer Interaction*. Springer, 131–140.

[7] Carlos Alberto dos Santos, Kevin Bouchard, and Bianca Minetto Napoleão. 2025. Automatic user story generation: a comprehensive systematic literature review. *International Journal of Data Science and Analytics* 20, 1 (2025), 1–24.

[8] Sávio Freire, Felipe Gomes, Larissa Barbosa, Thiago Souto Mendes, Galdir Reges, Rita S. P. Maciel, Manoel Mendonça, and Rodrigo Spínola. 2023. Requirements Engineering Issues Experienced by Software Practitioners: A Study on Stack Exchange. In *Requirements Engineering: Foundation for Software Quality: 29th International Working Conference, REFSQ 2023, Barcelona, Spain, April 17–20, 2023, Proceedings* (Barcelona, Spain). Springer-Verlag, Berlin, Heidelberg, 3–20. doi:10.1007/978-3-031-29786-1_1

[9] Eliakim Gama, Sávio Freire, Manoel Mendonça, Rodrigo O. Spínola, Matheus Paixao, and Mariela I. Cortés. 2020. Using Stack Overflow to Assess Technical Debt Identification on Software Projects. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering* (Natal, Brazil) *(SBES '20)*. Association for Computing Machinery, New York, NY, USA, 730–739. doi:10.1145/3422392.3422429

[10] Vahid Garousi, Michael Felderer, and Mika V Mäntylä. 2019. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and software technology* 106 (2019), 101–121.

[11] Felipe Gomes, Eder Santos, Sávio Freire, Thiago Souto Mendes, Manoel Mendonça, and Rodrigo Spínola. 2023. Investigating the Point of View of Project Management Practitioners on Technical Debt-A Study on Stack Exchange. *Journal of Software Engineering Research and Development* 11, 1 (2023), 12–1.

[12] Felipe Gomes, Eder Pereira dos Santos, Sávio Freire, Manoel Mendonça, Thiago Souto Mendes, and Rodrigo Spínola. 2022. Investigating the Point of View of Project Management Practitioners on Technical Debt: A Preliminary Study on Stack Exchange. In *Proceedings of the International Conference on Technical Debt* (Pittsburgh, Pennsylvania) *(TechDebt '22)*. Association for Computing Machinery, New York, NY, USA, 31–40. doi:10.1145/3524843.3528095

[13] Nicole Guerreiro, Yuri Nunes, Denivan Campos, Edna Dias Canedo, and Sávio Freire. 2026. *Complementary Material of Issues Faced by Software Practitioners When Using User Stories: An Empirical Study Based on Stack Exchange Discussions*. doi:10.5281/zenodo.21400048

[14] Muhammad Ijaz Khan, Zia Ud Din, Muhammad Ali Abid, and Tariq Naeem. 2019. User Story Characteristics Affecting Software Cost in Agile Software Development: A Systematic Literature Review. *Int. J. Comput. Sci. Netw. Secur.* 19, 12 (2019), 13–18.

[15] Barbara Kitchenham, Lech Madeyski, and David Budgen. 2023. How Should Software Engineering Secondary Studies Include Grey Material? *IEEE Transactions on Software Engineering* 49, 2 (2023), 872–882. doi:10.1109/TSE.2022.3165938

[16] Mohammad Amin Kuhail and Soren Lauesen. 2022. User story quality in practice: A case study. *Software* 1, 3 (2022), 223–243.

[17] Yanche Ari Kustiawan and Tek Yong Lim. 2023. User stories in requirements elicitation: A systematic literature review. In *2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS)*. IEEE, 10.1109/ICSECS58457.2023.10256364, 211–216.

[18] Garm Lucassen, Fabiano Dalpiaz, Jan Martijn EM van der Werf, and Sjaak Brinkkemper. 2016. Improving agile requirements: the quality user story framework and tool. *Requirements engineering* 21, 3 (2016), 383–403.

[19] Richard C Lupton and Julian M Allwood. 2017. Hybrid Sankey diagrams: Visual analysis of multidimensional data for understanding resource use. *Resources, Conservation and Recycling* 124 (2017), 141–151.

[20] Philipp Mayring. 2000. Qualitative Content Analysis. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* 1, 2 (Jun. 2000). doi:10.17169/fqs-1.2.1089

[21] Jan-Niklas Meckenstock. 2024. Shedding light on the dark side–A systematic literature review of the issues in agile software development methodology use. *Journal of Systems and Software* 211 (2024), 111966.

[22] João Vitor Oliveira and Lisandra Manzoni Fontoura. 2025. Analyzing User Story Quality: A Systematic Review of Common Issues and Solutions. *ICEIS (2)* (2025), 152–159.

[23] Paul Ralph, Nauman bin Ali, Sebastian Baltes, Domenico Bianculli, Jessica Diaz, Yvonne Dittrich, Neil Ernst, Michael Felderer, Robert Feldt, Antonio Filieri, Breno Bernard Nicolau de França, Carlo Alberto Furia, Greg Gay, Nicolas Gold, Daniel Graziotin, Pinjia He, Rashina Hoda, Natalia Juristo, Barbara Kitchenham, Valentina Lenarduzzi, Jorge Martínez, Jorge Melegati, Daniel Mendez, Tim Menzies, Jefferson Molleri, Dietmar Pfahl, Romain Robbes, Daniel Russo, Nyyti Saarimäki, Federica Sarro, Davide Taibi, Janet Siegmund, Diomidis Spinellis, Miroslaw Staron, Klaas Stol, Margaret-Anne Storey, Davide Taibi, Damian Tamburri, Marco Torchiano, Christoph Treude, Burak Turhan, Xiaofeng Wang, and Sira Vegas. 2020. Empirical Standards for Software Engineering Research. arXiv:2010.03525 [cs.SE] https://arxiv.org/abs/2010.03525

[24] Beatriz Santana, Sávio Freire, José Amancio Macedo Santos, and Manoel Mendonça. 2024. Psychological Safety in the Software Work Environment. *IEEE Softw.* 41, 4 (July 2024), 86–94. doi:10.1109/MS.2024.3386532

[25] Beatriz Silva De Santana, Sávio Freire, Leandro Cruz, Lidivânio Monte, Manoel Mendonca, and José Amancio Macedo Santos. 2023. Exploring Psychological Safety in Software Engineering: Insights from Stack Exchange. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering* (Campo Grande, Brazil) *(SBES '23)*. Association for Computing Machinery, New York, NY, USA, 503–513. doi:10.1145/3613372.3613411

[26] Joachim Schöpfel and Dominic J Farace. 2010. Grey literature. *Encyclopedia of library and information sciences* 3 (2010), 2029–2039.

[27] Anselm Strauss and Juliet Corbin. 1998. *Basics of qualitative research techniques*. Thousand oaks, CA: Sage publications.

[28] Richard Vidgen and Xiaofeng Wang. 2009. Coevolving systems and the organization of agile software development. *Information Systems Research* 20, 3 (2009), 355–376.

[29] Bill Wake. 2003. INVEST in Good Stories, and SMART Tasks. http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/. Accessed: 2026-05-04.

[30] Xinyu Wang, Liping Zhao, Ye Wang, and Jie Sun. 2014. The role of requirements engineering practices in agile development: an empirical study. In *Requirements Engineering: First Asia Pacific Requirements Engineering Symposium, APRES 2014, Auckland, New Zealand, April 28-29, 2014. Proceedings*. Springer, https://doi.org/10.1007/978-3-662-43610-3_15, 195–209.