

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/381798701>

# Mastering User Stories: Cracking the Code of Effective User Story Creation in Agile

Chapter · June 2024

DOI: 10.4018/979-8-3693-3318-1.ch005

---

CITATION

1

---

READS

1,820

2 authors:



[Saurabh Bhattacharya](#)

Chitkara University

9 PUBLICATIONS 9 CITATIONS

SEE PROFILE



[Babita Singla](#)

Chitkara University

47 PUBLICATIONS 153 CITATIONS

SEE PROFILE

## Chapter 5


# Mastering User Stories: Cracking the Code of Effective User Story Creation in Agile

**Saurabh Bhattacharya**

 <https://orcid.org/0000-0002-2729-1835>

*Chitkara Business School, Chitkara University, Punjab, India*

**Babita Singla**

 <https://orcid.org/0000-0002-8861-6859>

*Chitkara Business School, Chitkara University, Punjab, India*

### ABSTRACT

*In the dynamic realm of Agile development, the creation of effective user stories stands as a linchpin for project success. This research article delves into the foundational principles that underpin the craft of user story creation in Agile environments. Recognizing user stories as the nexus between technical teams and end-users, the narrative unfolds through an exploration of key principles, methodologies, and best practices. From empathetic understanding to strategic innovation, the principles outlined serve as a compass for crafting user-centric stories that resonate throughout the Agile development lifecycle. User stories are essential for bridging the gap between technical teams and end users in agile software development. ChatGPT can help Agile teams create user stories by offering preliminary recommendations thanks to its language creation capabilities. The results of software development projects may be greatly improved by using AI, underscoring the need to adopt technical innovations in Agile approaches*

### INTRODUCTION

Agile methodologies, such as Scrum, are gaining popularity due to their lightweight and flexible nature, prioritizing satisfying customer demands through timely delivery of useful software.(Choma et al., 2016). Yet, fast development's awareness of concerns about usability has led to a focus on core functionality and the difficulty of incorporating good User Stories. Another of the main principles of agile development is delivering valuable products to customers on time and continuously. (Guerrero-

DOI: 10.4018/979-8-3693-3318-1.ch005

Ulloa et al., 2023) (*Agile Principles, Patterns, and Practices in C# (Robert C. Martin) | Guide Books*, n.d.) (Séguin et al., 2012) (Magistretti & Trabucchi, 2024). User stories have become an increasingly important tool in agile software creation as a standard way to convey user demands in simple terms. (Schön et al., 2017) (Noel et al., 2018). They follow a structure that outlines WHO needs it, WHAT the platform must do, and WHY the function is required. Although this approach has drawn attention from researchers and developers alike, user stories will inevitably have certain problems due to their natural language composition, including potential confusion, discrepancies, and absent requirements. (Raharjana et al., 2021). Agile approaches use user stories, that are functional components and needs articulated in simple terms with an end user in mind.

The primary subjects of the patterns that were recommended to aid in needs collecting are typically the **WHO, WHAT, and WHY** of the capability. (Wautelet et al., 2014). The evaluation showed that considerably greater and superior agile studies are required. It also gave firms a map of results so that they were able to comprehend how agile might help the company. (Dybå & Dingsøyr, 2008). User Stories (US) are mostly used by agile methodologies as a framework for conveying demands. To say it plainly, the US fails to produce a visual representation of the main functions of the suggested framework. (Trkman et al., 2019). User stories are created at various stages of the procedure, but epics—substantial operational components that need to be delivered—are frequently composed on a separate colored deck.

User Stories (US) are essential components of Agile techniques that utilize minimal abstract and naturally articulate demands. They serve as the primary components of criteria and capability deliverables for enterprises. A common approach for conveying demands in agile projects for software creation is the user story. (Lucassen et al., 2015). Because they support the process of gathering and conveying end-user requirements to determine priority and provide small, useful enhancements throughout every stage of the development phase (Lucassen, Dalpiaz, Werf, et al., 2016), They are thought of as an extremely thorough representation of specifications that developers use to build novel features. (*User Stories Applied | Guide Books*, n.d.). The capacity of an innovation initiative to generate superior user stories is critical to its performance because these narratives impact the efficacy of the entire as a whole, which in turn impacts the final product (Amna & Poels, 2022b). In the end, ensure that production teams are in agreement about what has to be achieved, provide precise guidance for projects, and improve collaboration and communication. (Fathin Najwa Binti Mustaffa et al., 2021). A demand's primary components—who it is for, what the software requires to carry out, and, if desired, why it matters—are captured in user stories. Agile approaches emerged as a reaction to conventional software development methods and are focused on giving consumers only useful capabilities. Agile methods of creation use progressive and continuous methods of software development, which foster close collaboration among the creation crew and the consumer. (Dimitrijević et al., 2015).

Since user stories are a concise means of communicating demands that are increasingly widely used in agile specifications architecture and agile programming, they are quickly replacing additional demands conventions as the method most frequently employed in agile initiatives. (Cao & Ramesh, 2008) (Kassab, 2015) (Wang et al., 2014)

Over the last few decades, there has been a noticeable growth in the utilization of user stories to describe customer demands. (Lucassen, Dalpiaz, van der Werf, et al., 2016) (Rocha Silva et al., 2020) (Lucassen, Dalpiaz, Werf, et al., 2016) (Wautelet et al., 2014). However, user stories might not be sufficient to provide a summary of the entire endeavor. Thus, a direct study of the impact of user stories is necessary. (Lucassen, Dalpiaz, Werf, et al., 2016) (Wautelet, Heng, Hintea, et al., 2016) (Trkman et al., 2019). The Card, a deliberately simplified and blank-slate version of the actual User Story, is a part of

a User Story. The User Story's approval requirements and explanation are understood throughout the review and discourse phases that follow. They are designed in collaboration with customers immediately before the artwork is built. (Wick, 2016).

*User Stories and Responsibilities:* User Stories provide a strong accountability method by providing links among roles or user categories, epics, acceptance criteria, and test cases.

User Stories' "so that" component ensures an obvious linkage to an organization's value and implements accountability. The benefits of the Agile methodology—which are supplied by User Stories and acceptance criteria—are undermined by Use Cases.

Setting higher priorities for consumers enhances the client's worth. User stories are a means of representing the objectives and principles of the customers. Consequently, a user story represents the requirement from the viewpoint of the consumer. There is not much work for the staff to accomplish. A consumer story is less concerned with the functionality that remains to be developed and more concerned with the actual driven-by-goals benefit as seen from the user's point of view. User stories should be advantageous to the customers because they are created from the viewpoint of the end user. The execution strategy for useable software is an especially important component in the achievement of an agile software venture. As such, user stories must be created with the client's purpose in consideration. A user story is used as the measuring and creation component for software functionalities. Each user story consists of a textual definition that acts as an outline and notification, conversations to clarify the narrative's information, and testing for acceptance to determine whether a story is "done". A story or description is meant to be penned on a piece of paper or note card, intentionally concise, and presented in the client's original language. It is simply intended to serve as a starting point for conversations with clients to clarify the details of the narrative and document the demands of the clients for the undertaking via acceptance testing (Mahnič & Hovelja, 2012).

An unstructured, natural language description of any number of software system properties in an abstract concept is what a user story does. It typically has the simple form of "As a, I want so that." Placing the end user's needs and objectives at the forefront, this method simplifies the process of expressing the functional demands from their perspective. Using the narrative idea of a user story, developers, product owners, and users may collaborate and share concepts to generate a shared understanding of the needs. User stories are a useful tool for expressing user needs and stimulating creative thinking. Consideration has to be made during the creation and execution of user stories to ensure that they foster development.

User stories are a method of agile creation used to provide brief, broad overviews of the features needed in the consumer's dialect throughout the early phases of demand collection. (Ecar et al., 2018).

## **BACKGROUND OF THE PRINCIPLES OF A USER STORY**

The fundamental rules that direct the process of expressing and sharing customer needs inside the Agile framework are known as the principles of successful user story generation in Agile contexts. User stories must adhere to these guidelines to be understandable, applicable, and in line with the demands of the end user as well as the project's objectives. The growth of strong user stories is a fundamental component of the agile software creation process, and it determines the course of the project's achievement. User stories play a role that goes beyond simple functional requirements as businesses increasingly turn to Agile approaches to manage the complexity of modern software development. They become the story's heartbeat that coordinates technical prowess with end-user expectations. The present study

aims to investigate the fundamental ideas that guide the development of effective user stories in Agile work contexts. User stories are elevated to a new level of significance in project progress by Agile's incremental and adaptable mindset, which goes beyond its original purpose as a channel for interaction among developers and end users. Rather, user stories serve as a means of fostering compassion, providing clarity, and facilitating strategic cooperation. To prepare for a thorough examination of the concepts that characterize user-centric narrative in a constantly changing technical environment, this beginning explains the mutually beneficial link between Agile development and the production of user stories. This paper aims to shed light on the route to expertise in navigating the complex web of Agile techniques. It does this by outlining the fundamentals that govern the production of user stories that satisfy technical demands while simultaneously striking a deep chord with a wide range of end users. The set of principles presented here spans the gamut of designing with people in mind, strategically setting goals, conversation clarity, flexibility, and the creative spirit that drives Agile development into an environment where gadgets and human beings easily coexist. Comprehending and implementing these concepts becomes crucial in the current software development age, as the influence on how customers feel is now used to gauge success rather than merely the number of lines of code. The trip starts with an examination of the fundamentals of Agile and the critical function that user stories serve in converting attempts at development into concrete results that are in line with customer needs, goals, and aspirations. This chapter aims to rewrite the tale of agile software creation by using a set of principles. Rather than being a mere functional demand, every user story is a storyboard line that intertwines the subtleties of interpersonal relationships with the complexity of code. The tenets outlined below aim to serve as a compass, steering Agile practitioners regarding an eventuality in which crafting user stories is not just a chore but an artistic endeavor—an endeavor that elevates gadgets to the status of a compassionate, strategically-minded, and inventive force that connects with the wide range of end users. The goal is to cultivate an in-depth knowledge of the complex performance in superior technology and user-centric narratives as we set out on this path through the fundamentals of successful user story development in agile settings. This process establishes the next phase of Agile technology development and the transformative potential it maintains to produce important, significant, and compelling technological innovations for the advantage of consumers internationally. Most agile software creation professionals might not be using user stories effectively (Dimitrijević et al., 2015). Furthermore, backend programmers don't appear to be as focused on user stories as frontend and, perhaps most importantly, full-stack engineers are. (Pokharel & Vaidya, 2020). Professionally constructing and executing user stories may help businesses better convey user needs and foster the development of innovative alternatives. (O'hEocha & Conboy, 2010).

The following are essential guidelines for creating user stories in Agile contexts.

1. User-Centricity:

Justification: Engineers can effectively meet client demands in their solutions by putting the user first, which increases user fulfillment.

2. Empathy:

Justification: Developer teams can better comprehend user demands with the application of sympathetic user stories, which strengthens the bond that exists between the software's intended users and the engineering units.

## ***Mastering User Stories***

### 3. Clarity:

Justification: Well-defined Written user stories help Agile teams communicate and collaborate more effectively by lowering the possibility of misunderstandings.

### 4. Inclusivity:

Justification: By eliminating prejudices and improving convenience, comprehensive user stories guarantee that software products are suitable for a wide variety of consumers.

### 5. Prioritization:

Justification: Setting priorities ensures that development resources are focused on things that will benefit the endeavor and end customers by far.

### 6. Iterative Development:

Justification: Continuous improvement allows teams to modify user stories in response to shifting project specifications and increasing user demands.

### 7. Adaptability:

Justification: Using a flexible methodology guarantees that user stories stay applicable during an Agile project.

### 8. Simplicity:

Justification: Clear and concise user stories aid in improved understanding, which makes it simpler for all participants to comprehend and collaborate on common objectives.

### 9. Strategic Alignment:

Justification: Development efforts are directed toward an important goal when user stories are strategically aligned, which guarantees that they significantly impact the project's achievement.

### 10. Innovation:

Justification: Creative user stories aid in the creation of original and progressive alternatives that distinguish the endeavor.

Figure 1. Key principles for effective user story creation

Aspect	Use Cases	Questions that a User Story should address
User & Goals	The user is an e-commerce entrepreneur seeking to increase website traffic and sales. Their goal is to implement an effective marketing strategy to achieve higher conversion rates and revenue.	Who is the user or persona involved, and what are their specific goals or objectives?
Problem	The user needs to address low website traffic and conversion rates. They aim to enhance their marketing strategy to attract more visitors and convert them into customers.	What problem or need is the user trying to address or fulfill?
Key Features	Targeted advertising, SEO optimization, email marketing, social media campaigns, content creation, conversion rate optimization (CRO), and customer engagement strategies.	What are the essential features or functionalities required to solve the user's problem?
Success Criteria	The user will consider the solution successful if the website experiences a significant increase in traffic, an uptick in conversion rates, and a measurable boost in sales revenue. Additionally, improved customer engagement and positive feedback would indicate the effectiveness of the strategy.	What are the success criteria for the user story, and how will we know if the solution is effective?
Interaction & User Experience	The user will interact with the solution through various marketing channels, such as targeted ads, social media platforms, and email campaigns. A seamless and engaging user experience will be crucial, spanning from the initial interaction with the marketing content to the final purchase process on the website.	How will the user interact with the solution, and what is the envisioned user experience?
Barriers & Challenges	Potential challenges include stiff competition, changing consumer behaviors, evolving digital marketing trends, and limited resources for implementing and monitoring marketing strategies effectively.	What potential obstacles or challenges might the user face in achieving their goals?
Alternative Solutions	The user may explore influencer partnerships, affiliate marketing, referral programs, or exploring new sales channels or market segments. Another approach could involve analyzing and optimizing user experience and website usability to improve conversions.	Are there alternative approaches or solutions that the user may consider?
Project Roadmap	This user story aligns with the larger project's objective of increasing online sales and improving market presence. It contributes to the development roadmap focused on enhancing the digital marketing framework and customer acquisition strategies for the e-commerce platform.	How does this user story fit into the broader project development roadmap?
Technical Requirements	Integration of analytics tools for performance tracking, implementation of SEO best practices, email marketing platforms, social media management tools, and potentially the use of AI for personalized customer targeting and engagement could be considered.	Are there any technical or non-functional requirements that need consideration?
Acceptance Criteria and Definition of Done	Defining the completion of a user story.	What are the specific acceptance criteria that will determine if the user story is satisfactorily completed?

## PREVIOUS STUDIES

User stories are known to include ambiguities. Few comprehensive solutions exist for managing ambiguity across multiple dialect capabilities.(Amna & Poels, 2022a) (Ribeiro & Berry, 2020) (Zahrin et al., 2023). As already mentioned, criteria without a single, clear meaning are referred to as ambiguous. Inconsistency is a big problem in software development. The use of natural language processing (NLP) in user stories comes with benefits and drawbacks. Rigorous evaluation methods and in-depth analysis of NLP techniques are necessary to ensure high-quality investigation.

(Raharjana et al., 2021).

According to (Noel et al., 2018) teams that are less productive and have less professional experience in handling software requirements collaborate less, while teams that make fewer changes produce more user stories. There was no significant productivity variation seen among the participants with the greatest expertise and the rest, even though the persons with less expertise created more user stories.

Another author highlights how crucial it is to include usability considerations in user stories while using agile software creation. It acknowledges the current conversation in the agile and usability groups and aims to contribute by looking at the effects of adding usability recommendations to user stories, even if additional scientific validation is still required.

(Moreno & Yagüe, 2012).

An additional writer investigates the compatibility of lean thinking ideas with the SECI framework for organizational knowledge creation, and user stories—a cornerstone of the XP and Scrum approaches.

It finds that user stories when carefully designed and implemented, may help to the invention of new solutions by promoting user-centric values, wide stakeholder interaction, and avoiding premature solution description. (O'hEocha & Conboy, 2010).

The COSMIC (Common Software Measurement International Consortium) technique is a second-generation technique for operation size measurement, whose estimation success is directly correlated with the level of demand detailed. (Ecar et al., 2018).

In a different investigation with 27 participants, only 45% of agile practitioners employed user stories, and only 14.81% of them knew about them. Compared to backend developers, frontend or full-stack developers were more concentrated. This demonstrates that agile specialists must receive user story training to provide customer value efficiently.

(Pokharel & Vaidya, 2020).

The essay highlights the vital significance that user stories perform in extreme programming and agile software building for the effective construction of software. It also highlights how important meeting consumer requirements are. It examines how different user story-authoring approaches impact requirement validation and offers suggestions for further study in this area.(Chopade & Dhavase, 2017). The techniques that are most frequently addressed include user story development, version control, backlog development, and code refactoring.(Klotins et al., 2021). Different research focused on identifying and fixing problems with the requirements-gathering process for agile software development. It highlighted that management needed to research pertinent elements, with coordination being determined to be the most important group, to accomplish effective demand engineering adoption in an agile framework. The goal of the findings is to provide practitioners and businesses looking to accelerate the agile method of developing software with an invaluable tool(Simhadri & Shameem, 2023).

The primary functional requirements for the control of epics, user stories, high-level release planning, and low-level iteration planning tend to be well addressed, according to an examination of many systems. Nevertheless, there were notable deficiencies in support for personas, help with acceptability testing, and user-role modeling (the latter being adequately handled by a single instrument).(Dimitrijević et al., 2015).

Software engineering organizations sometimes have to deal with specifications that evolve quickly and grow outdated before an effort is completed. Quick alterations to stakeholder choices, developing technological advances, economic worries, and time-to-market needs cause established standards to grow inaccurate.(Cao & Ramesh, 2008). User Stories (US) in large projects often encounter problems including too many US, uneven use, and difficulty arranging iterations and setting a structure for them. Creating an illustration that facilitates the sorting of US pieces, the study of their relationships, and the creation of uniform groups for managing this may be quite beneficial for agile modelers and project leaders. For improved leadership and scheduling efficiency, the intention is to produce a diagram that organizes Epic US with top-level Activity aspects and related pieces grouped around groups of Concepts. (Wautelet, Heng, Kolp, et al., 2016).

The results of another article indicated that the tools under review provided adequate coverage of the functional requirements related to epics, user stories, high-level release planning, and low-level iteration planning. However, they lacked capabilities for persona support, acceptability testing help, and user-role modeling; just one tool provided extensive coverage in this area. The study discovered significant differences in how different systems support agile requirements engineering concepts. Furthermore, practitioners overwhelmingly preferred simple tools—those that are easy to set up, use, understand, and customize—over complex ones, according to a qualitative analysis of user ratings (Dimitrijević et al., 2015).



Getting user stories is one of the main issues with agile development approaches. Theories are used to help identify user stories and enhance understanding of them (Trkman et al., 2019). Representations for user stories and templates for business processes both appear to significantly increase comprehension in specific circumstances when compared to spoken descriptions (Trkman et al., 2016) (Trkman et al., 2019).

Planning poker is often employed in agile software development approaches for estimating user stories, considering the dearth of empirical studies on the topic (Rocha Silva et al., 2020). Experiment with Software Requirements Specs utilize eight standards to determine the level of expectations: correctness, consistency, completeness, stability/importance ranking, verifiability, modifiability, and traceability (Lucassen, Dalpiaz, van der Werf, et al., 2016). The caliber of a user story is crucial since its focus is on textual aspects like phrases, words, and rhymes while addressing simultaneously the content and the structure (Hallmann, 2020).

## **MOTIVATION FOR THIS RESEARCH**

There is currently no scientific proof of user tales' efficacy. This article aims to evaluate practitioners' perceptions of User Stories within the framework of Agile requirements gathering. User stories are the most important aspect of the Agile approach and are a widely used method of capturing the features of the desired software (*User Stories: A Comprehensive Guide* - Justinmind, n.d.). Over and above, User stories enable a collaborative process for determining user needs, which benefits the Agile development methodology. A study on this issue is important because effective user stories are also necessary for efficiently prioritizing and organizing work in Agile situations. (Business, n.d.).

It is possible to generate better, human-centered products and services that align with the end user's perspective—a crucial element of modern development methodologies—by understanding and using the principles of effective US creation. (Obergfell, 2011).

Furthermore, in Agile environments, the successful development of user stories is critical to the accomplishment of any project, making it an important area of research with practical applications. Therefore, studying the foundations of effective user story creation in Agile environments aligns with the tenets of modern software development approaches and has the potential to provide significant tangible benefits. (Pichler, 2016).

## **User Story Formats Used**

User stories are widely used in programming in a range of syntactic forms. Some of the more popular versions are as follows:

1. The typical user story
2. Career Overview
3. Story Based on Roles
4. Event-Related Result
5. Map of User Stories

These variants offer different structural approaches for obtaining user requirements and may be tailored to suit the needs of different projects and stakeholders.

## Mastering User Stories

The basic framework for representing user demands in Agile projects is the traditional User Story format. Generally speaking, it proceeds as follows: “‘As a <role>, I want to <goal>, so that <benefit>’.” (*User Stories Applied | Guide Books*, n.d.) (Wautelet et al., 2014) The major focus areas in this straightforward method are the feature’s goal, the user, and the expected benefits. (Lucassen, Dalpiaz, van der Werf, et al., 2016).

User story frameworks are structured forms used in Agile software development to define the many functions and features that need to be built into an application from the perspective of the end user. These formats are crucial for efficient interaction among stakeholders and engineering teams because they clearly and concisely express the core of user demands.

## INVEST

The INVEST format and the 5-card format are both user storytelling patterns that are often adopted. They are widely used in software creation and are made to ensure that user stories are clear and give the development team the details they need. Below is a quick synopsis of every format:

- **“I” Independent** - Every user story needs to stand alone and not rely on any other to be carried out. A user story is considered independent if it is: 1) fundamentally distinct from other user stories; and 2) cannot rely on other user stories being completed.
- **“N” Negotiable**- A user story is essentially a “call to discussion” whenever a team addresses it that way.
- **“V” Valuable**- User stories ought to benefit the client or end user. When a user story is finished and fulfills a real requirement for the consumer, it is valuable.
- **“E” Estimable**- It must be possible to estimate how much time and effort will be required to implement the User Story. An estimable user story is one whose relative size can be easily estimated by a development team that understands it sufficiently. When used in practice, “relative size” refers to the ability of the engineering team to identify user stories that are around the same size (small, medium, and large T-shirt sizes are frequently useful examples of comparative sizing constructs).
- **“S” Small**: User stories need to be sufficiently brief to be finished in a single sprint or iteration. A brief user story can be finished in a condensed amount of time, such as a few working days. This is significant since teams that regularly work on short user experiences have a far higher completion rate than teams that find it difficult to maintain the length of their stories.
- **“T” Testable**: User stories have to be created in an approach that makes it possible to test them and see whether their implementation is accurate. When the acceptance criteria of a user story can be independently confirmed, it is considered testable.

## 5-CARD

In agile and iterative design, a common method for organizing user stories is the 5-card Format. It offers a structure for gathering crucial data to guarantee that user stories are straightforward and meet end users’ demands.

- **WHO:** This explains the persona or user that the user story might affect. The persona or user who will be affected by the User Story is identified on this card. It identifies the precise person, position, or organization that will gain from the product or capability under development.
- **WHAT:** The particular function or feature that the User Story aims to provide. The particular feature or capability that the User Story aims to give is described in the “What” card. This card gives a detailed explanation of the user’s goal or the issue that the feature is meant to address.
- **WHEN:** The schedule or environment in which the User Story is going to be applied. The user story’s implementation timeframe or scenario is specified on the “When” card. It records the contextual or chronological part of the user story, giving planning and prioritization substantial context.
- **WHY:** The rationale or advantage of putting the user story into practice. This card explains the rationale or advantage of putting the User Story into practice. It makes the product or functionality’s advantages clear by outlining its significance and how it fits into the project’s larger objectives.
- **WHERE:** The place or system that will be affected by the user story. The area or system that the user story will affect is shown on the “Where” card. This card aids in placing the user story in the frame of the wider framework or setting where it will be used.

When user stories are written using the 5-Card Design, they are more likely to be thorough, educational, and in line with end users’ wants and viewpoints. This follows of *MOSCOW* principle prioritizing the ‘*Must Have*’ features and functionality to be delivered on time.

According to research by another author, ChatGPT can accurately assess user story functionality in Agile software creation, which is comparable with actual human analysis. It addresses the topic of AI reliability and offers a “best of three” method to increase stable output. It also addresses the implications for non-experts using ChatGPT’s raw data. The paper makes valuable suggestions for further research in this area and significantly advances our knowledge of the reliability and utility of generative AI in user story analysis. (Ronanki et al., 2024) (Ecar et al., 2018). The purpose of another study was to improve the understanding and management of software projects by proposing an automated approach for transforming brief user stories into visual diagrams known as robustness diagrams. The technique produced high-quality illustrations when tested on over 1,400 user stories, however, the caliber of the user stories determines how well the strategy works (Gilson et al., 2020).

The main point of contention in the discussion about usability integration with agile teams is the inclusion of usability suggestions in user stories. This article highlights possible improvements from resolving usability concerns and discusses the consequences for user story structure and the defining process. The results provide a springboard for more study on this subject. Keywords: user stories, agile, and usability (Moreno & Yagüe, 2012).

## RESEARCH METHODOLOGY AND OBJECTIVES

A systematic review can be used to evaluate and comprehend attention phenomena, research challenges, and topics of interest within a certain subject. The aim is to assess the academic literature about a certain study topic using a systematic technique. The recommendations utilized in this study were developed from research requirements that were modified and applied to software development. It is advised that,

## Mastering User Stories

when doing the examination, you look into several digital forms. The Systematic Literature review shortlist consisted of 30 final research publications.

### Search Strategy

The study aims mentioned above guided the development of the following keywords:

(“Agile” OR “User Story” OR “Agile Practices” OR “Requirement Gathering” OR “ User Stories”)

Figure 2. Source database

Methodology	Focus	Testing Level	Key Characteristics
BDD (Behavior-Driven Development)	Behavior from user perspective	User Interface	Scenario-based, Plain language, Tools (e.g., Cucumber)
TDD (Test-Driven Development)	Writing tests before code implementation	Unit	Red-Green-Refactor cycle, Granularity, Iterative
DDD (Domain-Driven Development)	Building around domain model	System	Collaboration with domain experts, Domain complexity, Context mapping

Figure 3. Final shortlisting of the research article

Source	Website
Scopus	<a href="https://www.scopus.com">https://www.scopus.com</a>
IEEE Xplore	<a href="https://ieeexplore.ieee.org">https://ieeexplore.ieee.org</a>
Google Scholar	<a href="https://scholar.google.com">https://scholar.google.com</a>
Springer	<a href="https://www.springer.com">https://www.springer.com</a>
ScienceDirect	<a href="https://www.sciencedirect.com">https://www.sciencedirect.com</a>

### Research Objectives

**RQ 1:** Are agile practitioners capable of properly utilizing user stories to add value for customers?

**RQ 2:** Do user stories created by agile teams fairly reflect one another? Do any gaps exist?

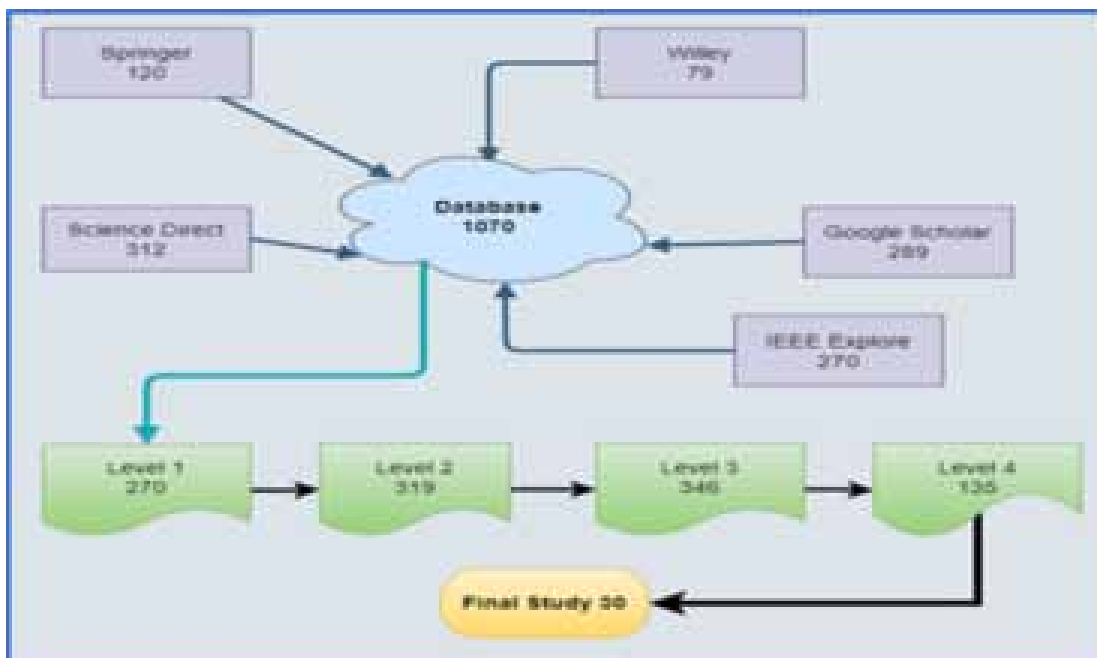
**RQ 3:** To recognize and evaluate the ideas—which go beyond conventional requirements—that support the development of successful user stories in agile contexts

Through an exploration and clarification of the basic concepts and methods that enhance user story production in Agile environments, these aims aim to increase stakeholder satisfaction and project success.

## Inclusion and Exclusion Criteria

The results of this research were located by searching the databases mentioned above. The selection of an article to serve as the primary inquiry was influenced by the body of literature on user stories in Agile Project Management as well as the effectiveness of the methodologies, putting together data from well-selected sources.

Figure 4. Inclusion and exclusion criteria



## THE STORY OF A USER STORY

User stories describe one or more software features from the perspective of the end user in agile software development. User stories, to put it simply, are sets of requirements that express the needs of the user. Usually called a “to-do” list, a user story can have one or more sentences in it. It ensures that the products and procedures created meet the users’ sincere requirements. It’s a wonderful way to describe them in detail.

A well-articulate user story is customer-centric, focuses on providing value, describes the functionality from the user’s point of view, and provides specific, measurable completion conditions.

The user story will express the following precise requirements:

- When can the development team begin and complete work?
- When can the testing team examine the created code to see if it satisfies the real requirements or not?

## Mastering User Stories

- When the Product Owner tests the functionality and certifies that the narrative is complete, it is called “*Done*”.

Through user stories, the user’s goals are expressed in plain language: the user should come first, not the solution. To avoid jumping to conclusions, the product owner must work with consumers to obtain needs. User stories come with a qualifying value statement, and it’s important to know the value that the organization and the user seek from the “so that” portion of the tale. User stories provide acceptance criteria.

A good user story has well-defined Acceptance Criteria as shown in **Figure 5** and **Figure 6**. These are a collection of guidelines that outline the user’s requirements and provide benchmarks that the team’s work must adhere to. They are used in the design of technology and to declare a story to be concluded.

The purpose of writing user stories is:

- To provide a product that the customer requires.
- To explain the product’s capabilities in “plain English.”
- To adhere to the value statement of the EPIC

Figure 5. Principle description of a good user story

Inclusion Criteria	Exclusion Criteria
<ul style="list-style-type: none"><li>• The language of all publications was English only.</li><li>• Researcher questions were addressed by publications depending on their content, keywords, title, and abstract.</li></ul>	<ul style="list-style-type: none"><li>• Published works that are not in the English language</li><li>• Only the most recent, thorough versions were taken into consideration, not earlier iterations of the same article.</li><li>• Those experiments do not address the research objectives.</li><li>• Grey Studies</li></ul>

Figure 6. Aspects of a well-defined user story

Principle of a Good User Story	Principle Description
User-Centricity	User stories are intended to be end-user-focused, conveying their viewpoint, requirements, and objectives.
Empathy	The feelings, difficulties, and objectives of end users are considered while crafting user stories.
Clarity	User stories should be easy for all stakeholders to grasp, succinct, and clear.
Inclusivity	The varied demands and viewpoints of the user base should be taken into consideration in user stories.
Prioritization	Based on their effect and value to the business, rank the user stories.
Iterative Development	Agile development is iterative, enabling user stories to change as time goes on.
Adaptability	It is important for user stories to be flexible in order to accommodate changing project goals and development-stage discoveries.
Simplicity	superfluous complexity should be avoided by keeping user stories clear and concise.
Strategic Alignment	User stories should be strategically aligned with the goals and objectives of the project.
Innovation	Encourage innovative approaches to meeting user demands by fostering an innovative culture inside the user story development process.

Figure 7. Components of a good user story

Aspect	User Stories Must Always Have a User
Start of the user story	Begin by identifying the user asking for the value to be delivered, structured as 'As a [USER NAME]...'
Focus of the user story	Focus on what the user wants to achieve, not on providing a solution.
Qualifying value statement	Include the 'so that' part to show what value the user and the business want.
Acceptance criteria	Contain a checklist of requirements that flesh out the user need and provide specific measures for the team's output to be acceptable.
Size and complexity of user stories	Stories should be small and simple, broken down into achievable chunks to complete within a sprint cycle.
Additional Information	It is essential for the Product Owner to work with users to avoid jumping to solutions when capturing their requirements.

## The Best Ways to Write Effective User Stories

**Persona-driven Stories:** Persona-driven stories provide a richer context for engineers to better grasp the user's motivations and limits. Before assigning the team to generate user stories, it's a good idea to establish personas. Personas are fictional personas created by actual consumers. (Jansen et al., 2021b) (Jansen et al., 2021a) (Nielsen et al., 2021) (Misztal-Radecka & Indurkha, 2020). Their names, visu-

als, characteristics, goals, and behaviors all allude to the things that consumers want to achieve or the problems they think the product will solve. Persona characteristics are considered while crafting user stories (Gilson et al., 2020). Personas are an excellent tool for documenting your user and consumer research. Personas are fictional characters created from real people in the intended audience. Along with an aim, they frequently contain a name, an image, and relevant characteristics, attitudes, and behaviors. The goal is the benefit the persona aspires to or the problem the character believes the product will assist in solving. User stories are not meant to be specifications, but rather a collaborative tool that facilitates rapid development using a simple methodology. Stories shouldn't be handed off to a development team; instead, they should be incorporated into a conversation between the team and the product owner. By just gathering the most crucial information, this will expedite delivery. Essentially using the team's creativity and experience throughout the product backlog refinement process may result in better user stories through collaborative story-crafting. If the development team is unable to take part in the user story work, use cases—a more formal way of recording product functionality—may need to be taken into consideration. (*A Full Guide to Backlog Refinement Meetings*, n.d.)

**Give an example using Storytelling:** The team may better understand customer wants and personalize the issue by crafting an in-depth user story. When a feature is designed with the requirements of the user in mind, it is easier to see how valuable it will be.

**Backward Stories:** By flipping the traditional user story framework, the team is better able to focus on the end goal and develop solutions that practically work. A backward story aids the team in devising the most effective strategy to reach the objective.

**Use the User's Language:** If user stories are written in a separate user language, the engineering team will find them more relevant and comprehensible. Using terms that people are acquainted with allows for the creation of a more relevant and useful product.

**The Goldilocks Principle:** To avoid overwhelming the team, it's important to balance the quantity of information in user stories. It should provide just enough background material to help the team understand the value of the narrative and the needs of the customer without restricting creative thinking.

**Maintaining the Relevance of User Stories:** User stories are guaranteed to correctly represent the needs and priorities of today's users when they are regularly reviewed and revised. Maintaining relevance in user stories makes it easier to create products that eventually meet user needs.

**Cooperation and Communication:** The team's ability to communicate clearly and succinctly ensures that user stories are understood and aligned with the overall product vision. Refinement of user stories and effective delivery of solutions that satisfy user requests are made possible by teamwork.

**Feedback-Driven Improvement:** By refining user stories in response to user input, the product is kept up to date with consumers' expectations and is continuously improved.

Incorporating user feedback into user stories makes it easier to create products that actively adjust to the shifting needs of their users.

**Prioritize User Outcomes:** User stories should focus on the desired outcomes rather than specific features or outputs. Encouraging the development team to try out different approaches might lead to more innovative and practical solutions.

**Encourage the Engineering and Scrum Team:** Well-crafted user stories need to provide the group room to think and act. This approach might lead to more customized and user-centered solutions.

**Distinguish Between User Roles:** It's critical to identify and distinguish between distinct user roles when crafting user stories. Differentiating between user roles can help create more productive and targeted user stories.



**Quantitative Outcomes:** By using quantitative results in user stories, the development team may establish precise, measurable goals. Establishing measurable objectives enables more accurate monitoring of the work's impact and strategy modification based on real data.

Analyse user desires in further detail: Understanding the underlying motivations behind users' actions is crucial to meeting their deeper needs and objectives. Developing a better understanding of what customers want might motivate the team to develop new ideas.

**Consider Non-User Stories:** "System stories" are requirements that may not directly involve user contact, but they still need to be considered. Non-user stories help make sure that backend requirements that are crucial to the system's integrity and operation are met.

**Edge Cases:** Atypical scenarios that deviate from standard user experiences yet are crucial to both user satisfaction and system resilience are called edge cases. In exceptional circumstances, user stories that consider edge scenarios offer a smooth user experience.

**Try Different story formats:** Experimenting with different story formats may yield new insights into the needs and preferences of consumers. Trying out various formats, including "How may we...", might lead to varying opinions of what the user needs.

**Usage Scenarios:** Usage scenarios humanize user stories by presenting real-world experiences with the product. They also improve the clarity of the user's needs by offering a realistic depiction of how the user would engage with the product.

**Preconditions and Postconditions:** Including preconditions and postconditions in user stories makes them more comprehensive and clearly stated. describes the requirements that serve as acceptance standards and must be fulfilled both before and after the user story is executed.

**Emotional User Mapping:** Consider the user's intended emotions when using the product to add an emotional element to the user story. leads to a design approach that prioritizes user emotions and is more human-centered.

**Unexpected Twist:** By adding unexpected challenges or turns to user stories, creative ways of problem-solving are stimulated. forces the group to think outside the box and come up with innovative solutions.

**User Stories For Future:** Helps the team create user stories that align with the successful future vision of the team. Reverse engineering the road to a successful outcome stimulates creativity and yields product-consistent user stories.

**Simplifying User Stories:** Identify the user, action, and benefit associated with each of the key components of the narrative. Ensure that user stories are coherent and aligned with user needs.

**Multidimensional User Stories:** Consider several aspects, such as the where, when, why, how, and who of the user's chosen action. Segment complex user stories into many, more in-depth user stories. Taking Styled Notes at Work:

User stories describe how a product is used from the perspective of the customer or user. They help record specific activities, including bookings and product searches.

## **How to Develop a Good User Story**

Write the user story from the user's point of view. You may create the ideal user story scenario with the aid of personas. Maintain collaboration when creating a user story. Storytelling should be straightforward. Normally the Start is with the Capability, then to Feature, and then epics. EPIC is further broken down into User stories. The stories should be improved till they are ready. Add the requirements for acceptance. Write user stories down on paper or index cards. Make sure others can see and access your stories.

## Mastering User Stories

The link between the user, the story, and the functioning of the product is depicted in the following **Figure 8**.

*Figure 8. Link between user story and the user*

Component	Description
Actor	The user or stakeholder who will be interacting with the feature or function.
Goal/Task	The specific action or objective that the user wants to accomplish with the feature or function.
Acceptance Criteria	The conditions that must be met for the feature or function to be deemed complete and acceptable.
Priority	(Additional Element) The relative importance or urgency assigned to the user story.
Complexity	(Additional Element) The level of effort or difficulty involved in implementing the user story.

Three essential elements of a well-written user story are:

**Who** needs it, **What** needs to be done, and **Why?** This provides context for the user story.

User stories ensure each story has the required functionality and help the testing and engineering teams in their work on unit and QA test cases.

## Types of Methodology Used to Break Down the User Story

The scenarios of a User Story are often described in one of three ways., they are Behavior Driven Development, Test-Driven Development, and Domain-Driven-Development

### Behavior Driven Development

User stories and the BDD (Behavior Driven Development) approach are closely connected. BDD entails utilizing natural language to describe the system's behavior from the viewpoint of the user. This makes it easier to make sure the system is user-friendly and built to suit the demands of the user. BDD places more importance on the system's behavior than the specifics of its technical implementation. This can facilitate non-technical stakeholders' comprehension of the system and enable them to offer design-related input. To ensure stakeholder agreement on specifications, Anaplan's sparse computation engine's calculating algorithms are validated using Behavior-Driven Specification (BDS) in conjunction with actionable acceptance criteria (EAC).(Allred et al., 2024). Behavior-driven development (BDD) is a testing approach that Agile teams are steadily implementing. Using test scenarios, BDD fosters cooperation and common comprehension(Natarajan & Pichai, 2024)

## Test-Driven Development (TDD)

The Test-Driven Development (TDD) methodology prioritizes writing tests before building software. This suggests that after developing tests that define the expected behavior of the software, programmers build the code to put the tests into action. TDD aims to improve code quality, reduce errors, and save time and money by verifying that the program meets its requirements and functions as planned. Testing comes before writing code when building software utilizing the Test-Driven Development (TDD) methodology. Before implementing all of its features, testing the program to ensure that it operates as intended helps to maintain focus on requirements, reduce errors, and save resources. Test-driven development (TDD) provides several advantages, such as early bug identification and enhanced code efficiency via a modular approach. Programming and testing procedures become more efficient as a result of these advantages. (Ramzan et al., 2024)

## Domain-Driven-Development (DDD)

“Domain-driven development” (DDD) is a software engineering approach that creates software by using a domain model that represents the major concepts and business processes of an enterprise or organization, together with their connections. With DDD, developers may effectively work with domain experts to deconstruct complex domains into simpler parts and construct models, links, and business processes. In essence, DDD facilitates the development of software that closely conforms to the domain requirements and expert knowledge present in the actual world.

Figure 9. BDD, TDD, and DDD components



Figure 10. BDD, TDD, and DDD characteristics comparison

Method	Concepts	Advantages	Disadvantages	Format	Example
BDD	Coverage of User	User-oriented, transparent, collaborative	Requires significant collaboration	Normal language	As a user, I want to...
	Clarity of Features	Clear and understandable scenarios	Overhead		
	Automation of Test Scenarios	Automation potential, low overhead	Initial setup overhead, maintenance		
	Code Fit as Frameworks	Flexibility of code for testing	High learning curve		
	Parameterization in Features	Flexibility in test scenarios	Time-consuming for complex features		
	Continuous Integration	Easy integration with CI/CD pipelines	Execution speed		
TDD	Test First Development	Immediate feedback on code quality	Time-consuming for complex features	Test first structure	Writing tests before writing the code
	Incremental Development	Gradual building of functionality	Initial learning curve		
	Flexibility in Code Design	Promotes simple and modular code	High initial overhead		
	Reduced Debugging Efforts	Errors detected in design and fix	High developer overhead		
	Better Code Coverage	Comprehensive testing of code	Can lead to over-testing		
DDD	Ubiquitous Language	Shared understanding between developers	Complex to implement from scratch	Domain specific	Consistency in naming conventions and code
	Explicit Contexts	Clearly defined application contexts	Increased complexity		
	Boundary Contexts	Well-defined contexts for better understanding	High initial setup		
	Domain Events	Clear representations of domain interactions	Increased architecture complexity		
	Aggregates	Encapsulation of domain models	Can lead to over-engineering		
	Entities and Value Objects	Clear representations of domain entities	Can be challenging to implement		

## Never Rely on User Stories Alone

When creating an excellent user experience, other techniques that should be employed in addition to user stories include story maps, workflow diagrams, storyboards, sketches, and prototypes. The reason for this is that user journeys and visual design may not be sufficiently reflected in user stories. User stories also fall short when it comes to technical needs; modeling languages like UML or technical stories are more appropriate for this kind of work. Furthermore, while generating fast prototypes or mockups to test concepts, the focus may need to shift from writing tales to co-creating the prototype, even if user stories are crucial for the creation of software. Remember that the purpose of user stories is to expedite software development without superfluous complexity; they are not intended to document requirements.

## DISCUSSION

One strategy to address potential quality issues is to improve user story-identification creation using modeling techniques. (*User Stories Applied | Guide Books*, n.d.). Focus groups, questionnaires, story-writing classes, and interviews are some of these methods. (Garg et al., 2015) (Hadar et al., 2014)—which are usually suggested as a successful strategy (Maiden & Rugg, 1996). Sometimes, business process models—which are publicly available documents that are already in place inside the company—supplement these strategies.

User needs surveys that are not full or accurate lead to high chances of software program failure (Bhattacharya, 2023) (Simhadri & Shameem, 2023). The challenges that have been identified include the lack of stakeholders, unclear demands, frequent fluctuations in demand, and a lack of analytical skills (Bhattacharya, 2023). The outcome brings out the importance of requirement elicitation in software development and the potential for system failures when elicitation methods are not effective. It highlights that a system could have several demands, thus it's important to use elicitation techniques

to identify and rank them. A requirement is defined as a necessity or request. The research study focuses on these approaches and their real-time applications to validate a comprehensive understanding of the requirements for system development and to comprehend stakeholder needs (Garg et al., 2015). Requirements elicitation in software development is the process of gathering stakeholder demands through techniques like interviewing. Domain knowledge may be beneficial in this process as well, as it helps improve communication and comprehension, even though it can also have some negative effects. An empirical study was conducted to determine how prior domain knowledge affects needs elicitation through interviews. Specialization affects one's ability to comprehend and communicate with customers. The elicitation process and overall team chemistry may be enhanced by focusing analysts with varying levels of topic expertise to form requirements analysis individuals that are more successful (Hadar et al., 2014). The research on user involvement in the development of usable systems aims to clarify the nature of user engagement, outline its expected benefits, and evaluate various user involvement approaches. The study pays especially close attention to the early phases of the process of development.

## **RESULT AND ANALYSIS**

The review of the literature indicates that user interaction usually has positive effects, particularly on user satisfaction. Additionally, there is evidence that gathering needs successfully involves using consumers as your primary source of information. However, the study also highlights the need to pay close attention to users' responsibilities, especially in real-world product development situations, and the need for more cost-effective ways to ascertain users' implicit needs and expectations (Kujala, 2003). The research ends by discussing the overall advantages of including users in the development of useful systems, particularly in terms of improving user satisfaction and enabling more effective requirements capture. It also brings to light the necessity of cautiously incorporating user feedback and the need for more efficient methods of gathering implicit demands from users in real-world development scenarios. User stories will likely be impacted in the future by several noteworthy advancements and trends in the software development industry. Several potential improvements might impact the way that user stories are created in the present Agile Environment.

## **Integration of AI and Machine Learning**

It could be feasible to include these technologies in the user story creation process as they advance. Artificial intelligence (AI) systems have the potential to generate user stories automatically by identifying trends in large datasets of user interactions, feedback, and needs. According to (Xu et al., 2024), this might ensure that user requests are accurately documented and significantly speed up the user story-building process. The intersection of artificial intelligence (AI) and user experience (UX) design, whereby AI capabilities are used to enhance the UX design process overall (*Enhancing User Experience Design Workflow with Artificial Intelligence Tools*, n.d.).

In the capacity of a business analyst, software development may profit from the utilization of generative AI, particularly OpenAI's Chat GPT. AI has a significant impact on streamlining the building of user stories and other vital elements, and it can also generate user requirements using natural language (Sr. Director - Enterprise Architecture, Fortune Brands Home & Security, USA et al., 2023). Artificial Intelligence is crucial since it leverages machine learning to evaluate user behavior and provide person-

alized, real-time product recommendations. By providing customers with a seamless and personalized e-commerce experience, this approach demonstrates how Agile UX and AI can be effectively combined to accommodate shifting market demands (Vasantham et al., 2023).

### **Context-Aware and Personalization User Stories**

User stories may eventually need to include context—the environment in which users interact with the system—in addition to functional demands, as user experience and personalization become more important. As a result, development may become more targeted and effective by producing context-aware user stories that are tailored to match specific user profiles and use cases. Emotional and moral concerns: Besides technical needs, future user stories may cover ethical and emotional concerns as well. Understanding how users feel about specific features or how different functionality may affect their well-being may be a crucial aspect of crafting user stories. It could be necessary to incorporate emotional and ethical personas into the user story creation process to ensure that the software not only meets functional requirements but also adheres to users' expectations in terms of ethics and emotions.

### **Collaborative and Interactive User Story Development**

In future generations, user story development techniques might become more participatory and team-based. Using virtual reality or mixed reality environments to enable stakeholders to interact with and provide feedback on user story prototypes may be necessary to create a more engaging and captivating user story development approach.

**Real-Time Iterative User Stories:** Future iterative and agile development methodologies may incorporate real-time iteration and adaptation of user stories. The idea of user stories changing continuously following real-time user input and use data may enable a more flexible and adaptive development strategy.

**Inclusion of Multimodal User Stories:** With the growing popularity of voice connections, augmented reality, and other non-traditional interaction paradigms, future user stories may need to consider multimodal interactions. It could be necessary to record user requirements in many modalities, such as expression, gesture, and touch, to ensure that software systems are designed to accommodate a range of engagement approaches.

### **Blockchain-Based Transparency and Traceability**

User story traceability and transparency throughout the software development lifecycle may be possible using blockchain technology. Providing a tamper-proof and auditable record of how user stories evolve, are prioritized, and are ultimately performed, might enhance accountability and transparency in the product development process.

## **CONCLUSION**

These potential developments suggest that user stories will become more personalized, dynamic, and aware of the needs of users as a whole in the future. As technology develops, user stories will most likely adapt to better represent and communicate user needs in a way that maximizes the value that software

systems offer. The article's main ideas are summarised in the conclusion, which highlights how they all contribute to the development of successful user stories in Agile settings. It highlights how following these guidelines may strengthen the user-centric narrative and promote project outcomes that are effective as well as positive end-user experiences. The fundamentals of creating successful user stories in Agile settings offer a thorough framework for managing the complex interactions between user-centricity and technical excellence. Agile teams can create stories that go beyond conventional functional requirements by prioritizing users and adhering to values like inclusion, empathy, clarity, and flexibility. User stories are guaranteed to make a significant contribution to project success through strategic alignment, iterative development, and prioritization based on business value. As a guiding principle, simplicity fosters improved communication and lowers the possibility of misunderstandings. Together, the tenets promote an innovative culture that rewards original thinking in meeting user demands. These values act as a direction for developers who are navigating the ever-changing landscape of Agile methodologies, pointing them in the direction of a future in which technology and human experience effortlessly combine to create solutions that are not only technically viable but also deeply meaningful to a wide range of end users.

## REFERENCES

- Allred, J. D., Fraser, S., & Pezzoni, A. (2024). Behavior-Driven Specification in Practice: An Experience Report. In D. Mendez & A. Moreira (Eds.), *Requirements Engineering: Foundation for Software Quality* (pp. 330–343). Springer Nature Switzerland. doi:10.1007/978-3-031-57327-9\_21
- Amna, A. R., & Poels, G. (2022a). Ambiguity in user stories: A systematic literature review. *Information and Software Technology*, 145, 106824. doi:10.1016/j.infsof.2022.106824
- Amna, A. R., & Poels, G. (2022b). Systematic literature mapping of user story research. *IEEE Access : Practical Innovations, Open Solutions*, 10, 51723–51746. doi:10.1109/ACCESS.2022.3173745
- Bhattacharya, S. (2023). The Software Principles of Design for Data Modeling: Gathering Requirements—A Detailed Guide. In *The Software Principles of Design for Data Modeling* (pp. 226–247). IGI Global.
- Cao, L., & Ramesh, B. (2008). Agile Requirements Engineering Practices: An Empirical Study. *IEEE Software*, 25(1), 60–67. doi:10.1109/MS.2008.1
- Choma, J., Zaina, L. A. M., & Beraldo, D. (2016). UserX Story: Incorporating UX Aspects into User Stories Elaboration. In M. Kurosu (Ed.), *Human-Computer Interaction. Theory, Design, Development and Practice* (pp. 131–140). Springer International Publishing. doi:10.1007/978-3-319-39510-4\_13
- Chopade, M. R. M., & Dhavase, N. S. (2017). Agile software development: Positive and negative user stories. *2017 2nd International Conference for Convergence in Technology (I2CT)*, (pp. 297–299). 10.1109/I2CT.2017.8226139
- Dimitrijević, S., Jovanović, J., & Devedžić, V. (2015). A comparative study of software tools for user story management. *Information and Software Technology*, 57, 352–368. doi:10.1016/j.infsof.2014.05.012
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9–10), 833–859. doi:10.1016/j.infsof.2008.01.006

Ecar, M., Kepler, F., & da Silva, J. P. S. (2018). Cosmic User Story Standard. In J. Garbajosa, X. Wang, & A. Aguiar (Eds.), *Agile Processes in Software Engineering and Extreme Programming* (pp. 3–18). Springer International Publishing., doi:10.1007/978-3-319-91602-6\_1

*Enhancing User Experience Design workflow with Artificial Intelligence Tools*. (n.d.). Retrieved March 10, 2024, from <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1800706&dswid=-8150>

Fathin Najwa Binti Mustaffa, S. N., Sallim, J. B., & Binti Mohamed, R. (2021). Enhancing High-Quality User Stories with AQUA: An Overview Study of Data Cleaning Process. *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*, (pp. 295–300). IEEE. 10.1109/ICSECS52883.2021.00060

Garg, N., Agarwal, P., & Khan, S. (2015). Recent advancements in requirement elicitation and prioritization techniques. *2015 International Conference on Advances in Computer Engineering and Applications*, (pp. 237–240). IEEE. 10.1109/ICACEA.2015.7164702

Gilson, F., Galster, M., & Georis, F. (2020). Generating Use Case Scenarios from User Stories. *Proceedings of the International Conference on Software and System Processes*, (pp. 31–40). IEEE. 10.1145/3379177.3388895

Guerrero-Ulloa, G., Rodríguez-Domínguez, C., & Hornos, M. J. (2023). Agile Methodologies Applied to the Development of Internet of Things (IoT)-Based Systems: A Review. *Sensors (Basel)*, 23(2), 2. doi:10.3390/s23020790 PMID:36679594

Hadar, I., Soffer, P., & Kenzi, K. (2014). The role of domain knowledge in requirements elicitation via interviews: An exploratory study. *Requirements Engineering*, 19(2), 143–159. doi:10.1007/s00766-012-0163-2

Hallmann, D. (2020). “I Don’t Understand!”: Toward a Model to Evaluate the Role of User Story Quality. In V. Stray, R. Hoda, M. Paasivaara, & P. Kruchten (Eds.), *Agile Processes in Software Engineering and Extreme Programming* (pp. 103–112). Springer International Publishing., doi:10.1007/978-3-030-49392-9\_7

Jansen, B. J., Salminen, J., Jung, S., & Guan, K. (2021a). Creating Data-Driven Personas. In B. J. Jansen, J. Salminen, S. Jung, & K. Guan (Eds.), *Data-Driven Personas* (pp. 93–118). Springer International Publishing. doi:10.1007/978-3-031-02231-9\_4

Jansen, B. J., Salminen, J., Jung, S., & Guan, K. (2021b). *Data-Driven Personas*. Springer International Publishing. doi:10.1007/978-3-031-02231-9

Kassab, M. (2015). The changing landscape of requirements engineering practices over the past decade. *2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*, (pp. 1–8). IEEE. 10.1109/EmpiRE.2015.7431299

Klotins, E., Unterkalmsteiner, M., Chatzipetrou, P., Gorschek, T., Prikladnicki, R., Tripathi, N., & Pompermaier, L. B. (2021). Use of Agile Practices in Start-ups. *E-Informatica Software Engineering Journal*, 15(1). doi:10.37190/e-Inf210103



- Kujala, S. (2003). User involvement: A review of the benefits and challenges. *Behaviour & Information Technology*, 22(1), 1–16. doi:10.1080/01449290301782
- Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M., & Brinkkemper, S. (2015). Forging high-quality User Stories: Towards a discipline for Agile Requirements. *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, (pp. 126–135). IEEE. 10.1109/RE.2015.7320415
- Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M., & Brinkkemper, S. (2016). Improving agile requirements: The Quality User Story framework and tool. *Requirements Engineering*, 21(3), 383–403. doi:10.1007/s00766-016-0250-x
- Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M., & Brinkkemper, S. (2016). The Use and Effectiveness of User Stories in Practice. In M. Daneva & O. Pastor (Eds.), *Requirements Engineering: Foundation for Software Quality* (pp. 205–222). Springer International Publishing., doi:10.1007/978-3-319-30282-9\_14
- Magistretti, S., & Trabucchi, D. (2024). Agile-as-a-tool and agile-as-a-culture: A comprehensive review of agile approaches adopting contingency and configuration theories. *Review of Managerial Science*. doi:10.1007/s11846-024-00745-1
- Mahnič, V., & Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software*, 85(9), 2086–2095. doi:10.1016/j.jss.2012.04.005
- Maiden, N. M., & Rugg, G. (1996). ACRE: Selecting methods for requirements acquisition. *Software Engineering Journal*, 11(3), 183–192. doi:10.1049/sej.1996.0024
- Misztal-Radecka, J., & Indurkha, B. (2020). Persona Prototypes for Improving the Qualitative Evaluation of Recommendation Systems. *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, (pp. 206–212). ACM. 10.1145/3386392.3399297
- Moreno, A. M., & Yagüe, A. (2012). Agile User Stories Enriched with Usability. In C. Wohlin (Ed.), *Agile Processes in Software Engineering and Extreme Programming* (pp. 168–176). Springer. doi:10.1007/978-3-642-30350-0\_12
- Natarajan, T., & Pichai, S. (2024). Behaviour-driven development and metrics framework for enhanced agile practices in scrum teams. *Information and Software Technology*, 170, 107435. doi:10.1016/j.infsof.2024.107435
- Nielsen, L., Larusdottir, M., & Larsen, L. B. (2021). Understanding Users Through Three Types of Personas. In C. Ardito, R. Lanzilotti, A. Malizia, H. Petrie, A. Piccinno, G. Desolda, & K. Inkpen (Eds.), *Human-Computer Interaction – INTERACT 2021* (pp. 330–348). Springer International Publishing. doi:10.1007/978-3-030-85616-8\_20
- Noel, R., Riquelme, F., Lean, R. M., Merino, E., Cechinel, C., Barcelos, T. S., Villarroel, R., & Munoz, R. (2018). Exploring Collaborative Writing of User Stories With Multimodal Learning Analytics: A Case Study on a Software Engineering Course. *IEEE Access : Practical Innovations, Open Solutions*, 6, 67783–67798. doi:10.1109/ACCESS.2018.2876801

- O’Heocha, C., & Conboy, K. (2010). The Role of the User Story Agile Practice in Innovation. In P. Abrahamsson & N. Oza (Eds.), *Lean Enterprise Software and Systems* (pp. 20–30). Springer. doi:10.1007/978-3-642-16416-3\_3
- Obergfell, Y. (2011, December 1). *Scrum User Story—The Best Guide for Scrum User Stories*. International Scrum Institute™. [https://www.scrum-institute.org/Scrum\\_User\\_Stories.php](https://www.scrum-institute.org/Scrum_User_Stories.php)
- Pichler, R. (2016, March 24). 10 Tips for Writing Good User Stories. *Roman Pichler*. <https://www.romanpichler.com/blog/10-tips-writing-good-user-stories/>
- Pokharel, P., & Vaidya, P. (2020). A Study of User Story in Practice. *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, (pp. 1–5). IEEE. 10.1109/ICDABI51230.2020.9325670
- Raharjana, I. K., Siahaan, D., & Fatichah, C. (2021). User Stories and Natural Language Processing: A Systematic Literature Review. *IEEE Access : Practical Innovations, Open Solutions*, 9, 53811–53826. doi:10.1109/ACCESS.2021.3070606
- Ramzan, H. A., Ramzan, S., & Kalsum, T. (2024). Test-Driven Development (TDD) in Small Software Development Teams: Advantages and Challenges. *2024 5th International Conference on Advancements in Computational Sciences (ICACS)*, (pp. 1–5). IEEE. 10.1109/ICACS60934.2024.10473291
- Ribeiro, C., & Berry, D. (2020). The prevalence and severity of persistent ambiguity in software requirements specifications: Is a special effort needed to find them? *Science of Computer Programming*, 195, 102472. doi:10.1016/j.scico.2020.102472
- Rocha Silva, T., Winckler, M., & Bach, C. (2020). Evaluating the usage of predefined interactive behaviors for writing user stories: An empirical study with potential product owners. *Cognition Technology and Work*, 22(3), 437–457. doi:10.1007/s10111-019-00566-3
- Ronanki, K., Cabrero-Daniel, B., & Berger, C. (2024). ChatGPT as a Tool for User Story Quality Evaluation: Trustworthy Out of the Box? In P. Kruchten & P. Gregory (Eds.), *Agile Processes in Software Engineering and Extreme Programming – Workshops* (pp. 173–181). Springer Nature Switzerland. doi:10.1007/978-3-031-48550-3\_17
- Schön, E.-M., Thomaschewski, J., & Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces*, 49, 79–91. doi:10.1016/j.csi.2016.08.011
- Séguin, N., Tremblay, G., & Bagane, H. (2012). Agile Principles as Software Engineering Principles: An Analysis. In C. Wohlin (Ed.), *Agile Processes in Software Engineering and Extreme Programming* (pp. 1–15). Springer. doi:10.1007/978-3-642-30350-0\_1
- Simhadri, R. S., & Shameem, M. (2023). Challenges in Requirements Gathering for Agile Software Development. *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, (pp. 406–413). IEEE. 10.1145/3593434.3594237
- Trkman, M., Mendling, J., & Krisper, M. (2016). Using business process models to better understand the dependencies among user stories. *Information and Software Technology*, 71, 58–76. doi:10.1016/j.infsof.2015.10.006

- Trkman, M., Mendling, J., Trkman, P., & Krisper, M. (2019). Impact of the conceptual model's representation format on identifying and understanding user stories. *Information and Software Technology*, 116, 106169. doi:10.1016/j.infsof.2019.08.001
- Vasantham, V., Preetham, K., Kumar, G. P., Krishna, L., & Sandeep, K. (2023). Combination of Scrum Lean-UX-based AI UX Design. *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, (pp. 1372–1378). IEEE. 10.1109/ICESC57686.2023.10193361
- Wang, X., Zhao, L., Wang, Y., & Sun, J. (2014). The Role of Requirements Engineering Practices in Agile Development: An Empirical Study. In D. Zowghi & Z. Jin (Eds.), *Requirements Engineering* (pp. 195–209). Springer. doi:10.1007/978-3-662-43610-3\_15
- Wautelet, Y., Heng, S., Hintea, D., Kolp, M., & Poelmans, S. (2016). Bridging User Story Sets with the Use Case Model. In S. Link & J. C. Trujillo (Eds.), *Advances in Conceptual Modeling* (pp. 127–138). Springer International Publishing. doi:10.1007/978-3-319-47717-6\_11
- Wautelet, Y., Heng, S., Kolp, M., & Mirbel, I. (2014). Unifying and Extending User Story Models. In M. Jarke, J. Mylopoulos, C. Quix, C. Rolland, Y. Manolopoulos, H. Mouratidis, & J. Horkoff (Eds.), *Advanced Information Systems Engineering* (pp. 211–225). Springer International Publishing. doi:10.1007/978-3-319-07881-6\_15
- Wautelet, Y., Heng, S., Kolp, M., Mirbel, I., & Poelmans, S. (2016). Building a rationale diagram for evaluating user story sets. *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, (pp. 1–12). IEEE. 10.1109/RCIS.2016.7549299
- Wick, A. (2016, April 11). *From The Archives: User Stories And Use Cases - Don't Use Both!* Business Analyst Articles, Webinars, Templates, Jobs. <https://www.batimes.com/articles/from-the-archives-user-stories-and-use-cases-don-t-use-both-d45/>
- Xu, X., Xu, Z., Ling, Z., Jin, Z., & Du, S. (2024). *Emerging Synergies Between Large Language Models and Machine Learning in Ecommerce Recommendations* (arXiv:2403.02760). arXiv. <https://doi.org/arXiv.2403.02760> doi:10.48550
- Zahrin, M. F., Osman, M. H., Hassan, S., Haron, A., & Halin, A. A. (2023). Ambiguity Detection and Improvement for Malay Requirements Specification: A Systematic Review. *International Journal on Advanced Science, Engineering and Information Technology*, 13(6), 6. doi:10.18517/ijaseit.13.6.18535