

Clear communication is the cornerstone of any successful endeavor. Without it, even the simplest tasks can quickly become complex a scenario where two people are trying to discuss the same thing but in different ways. This case can lead to confusion, misundersta

Now imagine a team of 15 people working on a software development project. Each person brings their unique perspective and unde Each team member may interpret a feature differently without consensus and acceptance criteria user story.

Thus, clear communication between the clients and developers is vital for creating products that meet all user expectations and induc can you clearly and effectively explain to all team members what exactly they need to do and how each feature should work without

Let's say you're building an e-commerce platform and want users to search and explore all the available products. Still, developers ac capabilities without product categories and a hierarchical system. There could be only one conclusion: you must improve communicat project manager explain the task using User Stories. Developers use Users Stories with well-defined acceptance criteria in Agile or Sc development methodology to prevent such situations.

Agile acceptance criteria clearly define what is expected from Scrum user stories, enabling the team to focus on delivering the right s criteria user story capture requirements in a simple, concise format that is easy for everyone to understand. However, user stories can precise evaluation principles, leading to clarity and preventing delays and miscommunication.

At IntelliSoft, we believe in the power of Agile development and the value of user stories acceptance criteria. We widely used this app the **custom fleet management platform** for Ingram Barge. By leveraging **our expertise in software development** and our commitment solutions, we help our clients stay ahead of the curve.

Let's discuss how we may help you make the work on your project more streamlined and problem-free by using user stories acceptar represent the demands of your customers and requirements. In this article, we will also give user story and acceptance criteria exampl create them by yourself.

What Are Acceptance Criteria? What Role Do they Play in Pro

Acceptance criteria and user stories are critical for successful project delivery for any software development project. They are essenti expectations from a product or solution and how the test team can evaluate it for acceptance. Let's look at the definition of acceptan connection to user stories, and their role in project success.

They are pre-established standards or requirements that a product or function must meet to satisfy the users' or stakeholders' expectations. Each system component should work in a given scenario from the end user's perspective and are unique for each case.

Insights ▾

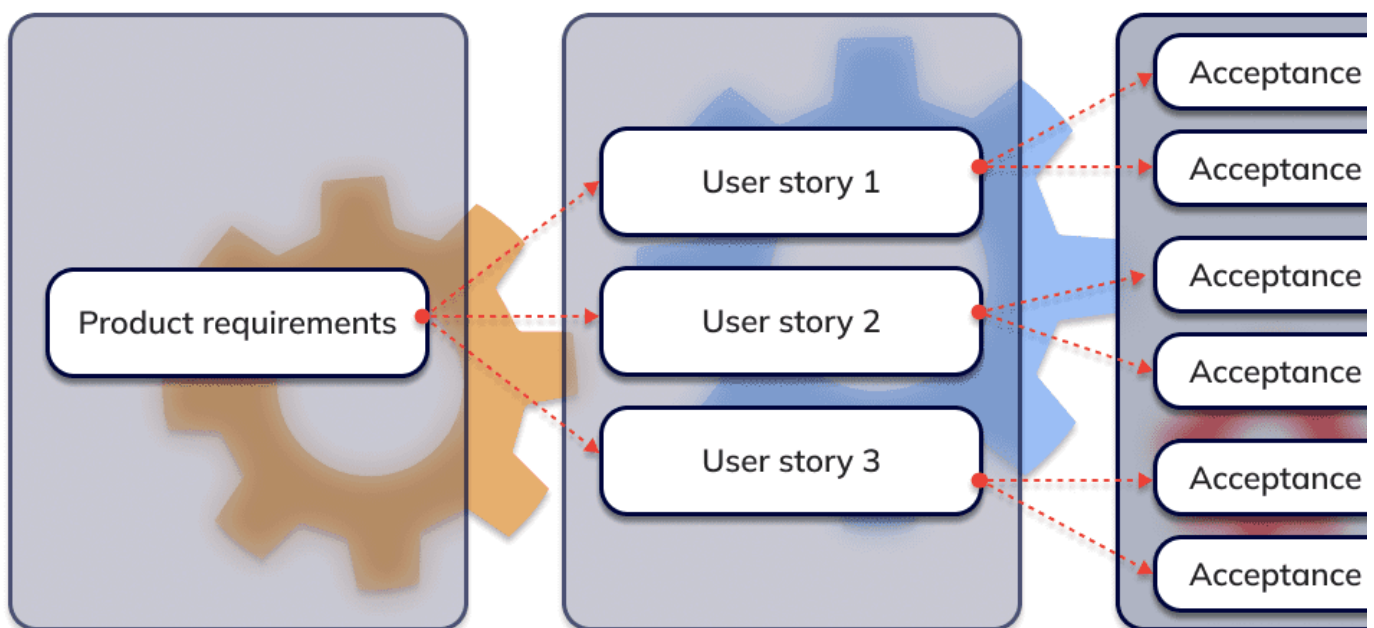
To give you user story and acceptance criteria examples, imagine you're working on a software development project for a client. The solution needs to be able to process 1,000 transactions per minute, have a response time of under two seconds, and have a user-system must meet all these criteria to be considered acceptable. That is a acceptance criteria for user stories.

Connection to User Stories

Acceptance criteria establish a close connection to user stories, which capture requirements in a simple, concise format that is easy to understand. User stories typically follow an "As a [user], I want [functionality], so that [benefit]" format and they help to define the desired outcomes.

Intellisoft

Connection of Acceptance Criteria to User Stories



For user stories examples with acceptance criteria, you need to define clear principles of evaluation, to prevent opening user stories to preventing confusion and delays. Acceptance criteria are the specific conditions that the system must meet to complete the user story. User stories with acceptance criteria define the boundaries of the user story and provide a comprehensive review of what consumers anticipate from a feature or system.

User Story vs Acceptance Criteria: Main Differences

The user story vs. acceptance criteria serves complementary yet distinct roles in Agile and Scrum methodologies. A user story is a brief description of a software feature from the perspective of the end-user or customer. It articulates what the user needs and why, often following a simple format: "As a [user], I want [feature] so that [benefit]." Its purpose is to focus on the value provided to the user, ensuring the development work aligns with and contributes to a valuable outcome.

user satisfaction standpoint

What is acceptance criteria in user story?

Insights ▾

Agile user story acceptance criteria are essential guidelines that define the conditions that must be met for the story to be considered by stakeholders.

The Main Purposes of Acceptance Criteria

Agile user story acceptance criteria serve several essential purposes in software development projects. To put it simply, you need the

Help specify what developers need to achieve

Communicate with stakeholders

Outline boundaries for user stories

Prepare for negative scenarios

Provide a basis for user story acceptance testing

Let's explore the primary functions of acceptance criteria and their importance in ensuring project success.



Specifying Exact Requirements for What Needs to be Developed

One of the primary functions of user story examples with acceptance criteria is to specify the requirements for the development team criteria to outline the product's features, functionalities, and quality attributes, which allows splitting user stories into tasks with clear

Facilitating Communication Between Stakeholders

Another critical function of user story examples with acceptance criteria is facilitating stakeholder communication, ensuring the development team understands the client's vision. Acceptance criteria are a communication tool between project team members, customers, and other stakeholders. By ensuring everyone is on the same page regarding the project's goals, objectives, and deliverables. By clearly understanding the acceptance criteria, stakeholders can provide valuable feedback and make informed decisions, leading to better outcomes.

Establishing Clear Boundaries for User Stories

Acceptance criteria also help to establish clear boundaries for user stories. Through user story example with acceptance criteria, it's easy to distinguish between a product's features and functionalities from the user's perspective. Well-defined criteria outline these features' specific requirements and ensure that the development team delivers the desired outcomes. By establishing clear boundaries for user stories, the project team can prevent scope creep and ensure that the project stays on track.

Describing Reaction to Negative Scenarios

Another essential function of user story and acceptance criteria is to describe the reaction to unfavorable scenarios. These criteria outline how the system should behave when things go wrong, ensuring the development team is ready for any eventuality. Invalid user inputs or unusual behavior scenarios. By defining these scenarios and their expected outcomes, the project team can prevent system failures and other issues that could hinder development.

Setting a Basis for User Story Acceptance Testing

Finally, user story and acceptance criteria serve as the basis for user story acceptance testing. Developers use acceptance testing to verify that the product meets the stakeholders' expectations and whether it is ready for release. By defining the acceptance criteria, the project team can ensure that user story acceptance testing is objective and consistent, leading to better outcomes.

NEED SOFTWARE DEVELOPMENT
TEAM?

Contact Us

By understanding these purposes, the development team can avoid ambiguity, ensure the project's success, and deliver products that meet stakeholders' expectations.

Types and Structures of Acceptance Criteria

How to write user stories and acceptance criteria? We'll explore the different types and structures of acceptance criteria and how to use them for success.

Functional and Non-functional Criteria

Functional acceptance criteria refer to the set of conditions that a system or product must meet to fulfill its intended purpose. They focus on functionality, behavior, and ability to perform specific tasks. These acceptance criteria in user story are generally straightforward and

called quality attributes. These criteria focus on performance, security, usability, reliability, and other aspects of the system. Examples of acceptance criteria include response time, load capacity, accessibility, and ease of use. Non-functional criteria often depend on user expectations; therefore, they are more subjective and difficult to measure.

In short, functional acceptance criteria in user story ensure that the system or product performs the intended tasks, while non-functional criteria ensure performance at an acceptable level regarding quality, security, and usability. A project that meets both functional and non-functional criteria is likely to be successful and meet the needs of its stakeholders.

Acceptance Criteria Based on Rules

Acceptance criteria based on rules are meticulously structured criteria that resemble the Product Requirements Document (PRD) in corporate development environments. They define what is acceptable or unacceptable for a particular feature or functionality. Rule-based acceptance criteria are especially beneficial when dealing with complex systems or functionalities requiring specific conditions.

Here's an example of how you can use rule-based acceptance criteria:

Let's say you are working on a system allowing users to create and customize their products. One of the system's features is the ability to choose colors for the product. To ensure that the system works properly, you might create a set of rule-based criteria that looks something like this:

- The color palette should only display colors available to customize a particular product.

- The system should prevent users from choosing the same colors for different product parts.

- The system should prevent users from choosing clashing colors (e.g., light font color on a light background).

- Users should be able to preview the product with their selected colors before finalizing their customization.

- The system should store the chosen colors and apply them correctly to the final product.

In this example, the team based acceptance criteria on rules that ensure the color customization feature works correctly. By defining these rules, they ensure that the feature meets the needs of the stakeholders and that the system works as expected.

Rule-based acceptance criteria can be a powerful tool for ensuring that complex features or functionalities work as expected. It's important to note that the use of rule-based criteria is only sometimes necessary for some features or functionality. You would only want to use them when the system requires specific conditions to work correctly.

Rule-based acceptance criteria can be a powerful tool for ensuring that complex features or functionalities work as expected. By defining these rules, teams can create evaluation principles that accurately reflect the stakeholders' needs and ensure their projects' success.

Acceptance Criteria Based on Scenarios

Scenario-based acceptance criteria take into account specific scenarios or use cases. They help teams understand how users interact with the system in different situations and ensure it works correctly.

Here's an example of how to use scenario-based acceptance criteria:

Let's say you are working on an e-commerce project allowing users to purchase online. One of the system's main features is the ability to add products to their cart and then check out. To ensure that the system works properly, you might create a set of scenario-based acceptance criteria something like this:

Scenario

Insights ▾

Acceptance Criteria



User adds product to cart

The product should be added to the user's cart and the cart should display the correct price and quantity.



User removes product from cart

The product should be removed from the user's cart and the cart should display the correct price and quantity.



User modifies quantity of product in cart

The cart should update to display the correct price and quantity on the user's modifications.



User checks out

The system should prompt the user for their billing and shipping information, display the correct total price, and confirm the order.



User cancels checkout

The system should cancel the checkout process and return the user to their cart with the correct products and quantities.

This user story acceptance criteria example describes specific scenarios users might encounter while using the system. By defining the corresponding acceptance criteria, teams can ensure that the system works correctly in each situation and meets the needs of the stakeholders. It's important to note that scenario-based acceptance criteria should be based on realistic scenarios and use cases. The user story acceptance criteria should reflect the stakeholders' needs and the users' expectations.

How to Choose the Right Approach

It is essential to choose the right approach to creating user story acceptance criteria format. Here are some factors to consider when choosing the structure of acceptance criteria:

The complexity of the feature or functionality: Rule-based acceptance criteria can be helpful when working on complex components. Scenario-based acceptance criteria may be more appropriate for simple systems or functionalities.

Needs of the stakeholders: It is crucial to understand the needs of the stakeholders and ensure that the acceptance criteria reflect their needs. If the stakeholders are concerned about specific scenarios or use cases, scenario-based acceptance criteria may be the best approach.

Expectations of the users: Acceptance criteria should reflect the users' expectations. If users have specific requirements for a feature, it's important to add those requirements to the list.

Project timeline: The timeline for the project can also impact the choice of approach. If there is limited time to create acceptance criteria, a straightforward process may be necessary.

Development methodology: The type of development methodology can also impact the choice of approach. For example, scenario-based acceptance criteria may be more appropriate if the team uses an Agile methodology.

Choosing the right approach to creating acceptance criteria depends on many factors. Consider them carefully and select a method that best fits the stakeholders' needs and the users' expectations.

General Rules on How to Write Acceptance Criteria

Use positive language instead of negative language when writing acceptance criteria. Avoid “not” and focus on what the product or example, instead of saying, “The product should not crash,” say, “The product should function without any crashes.”

Use active voice

Use active voice when writing acceptance criteria. It makes it easier to understand and helps to identify who is responsible for each c instead of saying, “The product should be tested,” say, “The testing team should test the product.”

Avoid using pronouns

Use specific nouns instead of pronouns to avoid confusion. For example, instead of saying, “They should test the product,” say, “The t the product.” It is crucial when multiple people or teams are involved in the project.

Avoid conjunctions

Avoid using conjunctions like “and” or “or” in acceptance criteria. It can make the requirements ambiguous and hard to understand. Ir statement into separate parts. For example, instead of saying, “The product should be fast and easy to use,” say, “The product shoul should be easy to use.”

Avoid unattainable absolutes

Avoid using “always,” “never,” or “100%” in acceptance criteria. These absolutes are often impossible to achieve or test, leading to cor Instead, focus on what is achievable and measurable. For example, instead of saying, “The product should always be perfect,” say, “1 function without any errors.”

Look at user story acceptance criteria template:

The login page should allow users to enter their email addresses and passwords.

The search function should return relevant results based on the user's query.

The checkout process should enable users to enter their payment information and complete the transaction without errors.

The product should load within 3 seconds on a 4G network.

Once users register, they should receive an email notification to their s email address.

Ready-to-use Acceptance Criteria Templates

Writing acceptance criteria can be time-consuming and challenging, especially if you're new to the process. Available templates can process and ensure that you cover all the necessary bases. But first things first: we need to create user stories.

NEED HELP WITH PROJECT'S MVP?

Learn More →

Creating User Stories Templates

Here are a couple of user story examples written using this template:

Insights ▾

As a customer, I want to search for products by keyword to find the products that interest me quickly.

As a mobile app user, I want to be able to save my search filters so that I don't have to re-enter them every time I search.

Given-When-Then Acceptance Criteria Template

The Given-When-Then (GWT) template is a popular and effective way of writing scenario-based acceptance criteria. It's a simple for criteria into three parts:

Given: This describes the initial conditions or state of the system before the user action.

When: This describes the user action that triggers the system to perform a function.

Then: This describes the expected outcome or result of the user action.

Here's an example of acceptance criteria written using the GWT template:

Given that the user logs into the system,

When the user clicks the "Add to Cart" button,

Then the system should add the item to the user's cart.

You can see the similarities of this approach with behavior-driven development (BDD). It uses natural language to describe the expected system. You can add the "And" element to the account for other conditions or outcomes in more complex scenarios. For example:

Given a registered user is on the product page,

When the user adds the product to the cart,

Then the product should be added to the cart,

And the cart should display the updated total.

Comprehensive Acceptance Criteria Template

You can use a template that covers all aspects of acceptance criteria. It includes the following additional elements:

User story:

It describes the feature or functionality from the user's perspective.

Scenario:

It describes the situation and user behavior.

Here's an example of comprehensive acceptance criteria:

User story: *As a registered user, I want to be able to view my order history.*

Scenario: *Trying to view previous orders.*

Given a registered user logs in,

When the user enters their profile page,

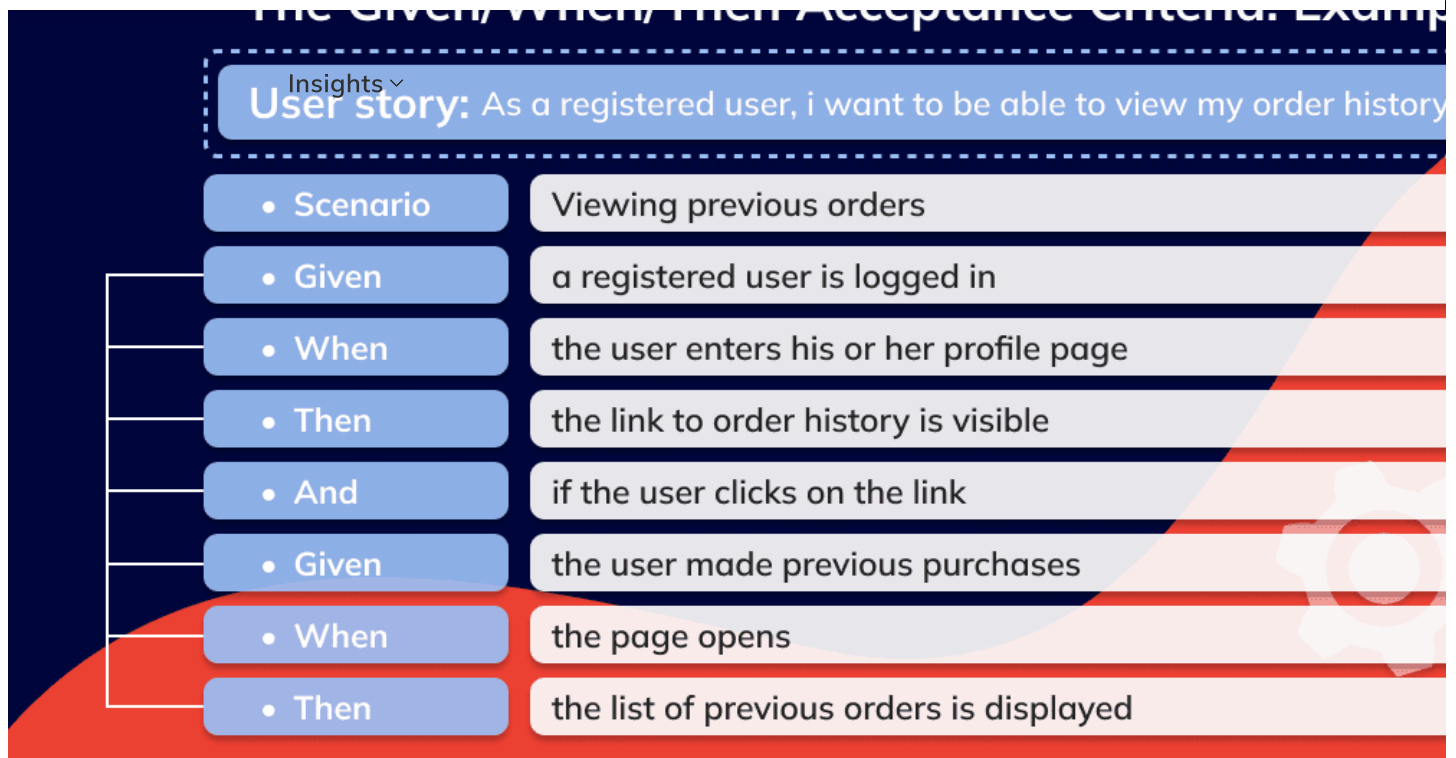
Then the link to order history is visible,

And if the user clicks on the link,

Given the user made previous purchases,

When the page opens,

Then the list of prior orders is displayed.



These templates can be beneficial for teams that are new to writing acceptance criteria. However, it's important to note that template projects or teams. Feel free to modify the templates or create your own to fit your needs.

LAUNCH DISCOVERY PHASE FOR YOUR PROJECT

[Launch Now](#)

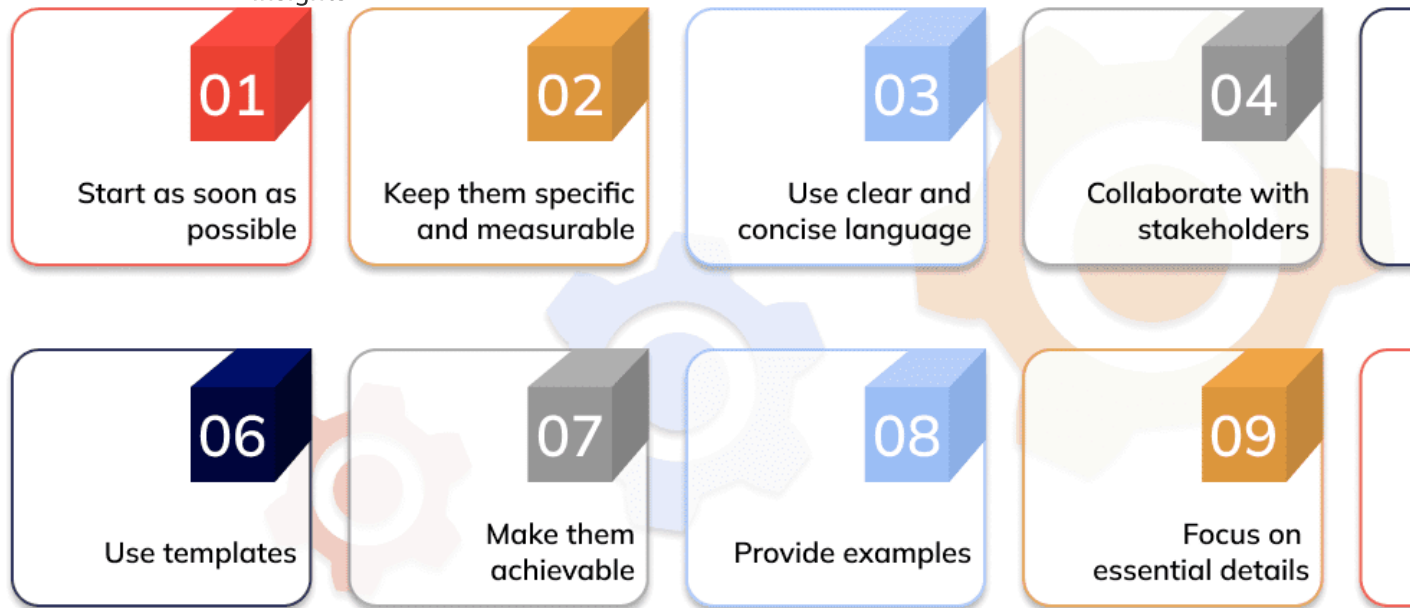
Best Practices for Writing Acceptance Criteria

The product owner or project manager usually writes acceptance criteria, but the process can involve the development team's input. There are several best practices to ensure that the requirements accurately reflect the stakeholders' needs and the users' expectations.

Tips for Writing Acceptance Criteria

Tips for Writing Acceptance Criteria

Insights ▾



Start as soon as possible

Before the software development process starts, you should define all acceptance criteria so your team can know their goals.

Keep them specific and measurable

Acceptance criteria should be specific enough that anyone can understand them and measurable so that you can test them.

Use clear and concise language

Use language that is easy to understand and avoid ambiguity, confusion, or misinterpretation.

Collaborate with stakeholders

Please work with the stakeholders to ensure the acceptance criteria accurately reflect their expectations. They should be involved in providing feedback to ensure it can fully satisfy their needs.

Keep them up-to-date

As the project progresses, the acceptance criteria may need to be updated to reflect changes in the system or stakeholder needs.

Use templates

Use templates to help guide the process and ensure the inclusion of all necessary elements.

Focus on essential details

Acceptance criteria should focus on essential details necessary for the software to meet the users' needs. Avoid including unnecessary clarity in determining if the software meets the acceptance criteria.

Make them achievable

Additional details and unnecessary functionality can clutter the backlog and become a developer problem. Define the most crucial features to make meeting the project's deadlines easier.

Provide examples

Providing examples of how the software should function can be helpful when defining acceptance criteria. It can help stakeholders understand how the software should behave in different situations.

HOW MUCH WILL YOUR PROJECT COST?

[Estimate Now](#)

By following these best practices, you can ensure that the software has high quality and meets the project's requirements.

Challenges in Writing Acceptance Criteria

While writing acceptance criteria is essential to the software development process, it can also present several challenges. Here are some challenges that teams may encounter when writing these criteria:

Ambiguity and vagueness

Acceptance criteria can be challenging to write because they must be specific and unambiguous. However, stakeholders may use vague language to provide enough detail, leading to ambiguity and making it difficult to determine if the software meets the criteria.

Overcomplication

One of the main challenges in writing acceptance criteria is balancing scope and specifics. Writing overly complicated acceptance criteria can make it difficult to determine if the software meets user expectations. It is important to focus on the essential details and avoid overly complex requirements.

Defining acceptance criteria during development

It would be hard for your team of developers to know what they should do if the acceptance criteria of features and functionality they are developing are not clearly defined. The solution: start early.

Misalignment with customer requirements

Acceptance criteria should reflect the stakeholders' needs and the users' expectations. However, misalignment with customer requirements can occur if stakeholders fail to provide enough detail or if there is a lack of communication between the development team and stakeholders.

Inadequate testing and validation

You should test and validate acceptance criteria to ensure the software meets the requirements. However, inadequate testing and validation of the testing process needs to be more thorough or if the acceptance criteria need to be adequately validated.

To overcome these challenges, you can establish clear communication with stakeholders and ensure that the acceptance criteria accurately reflect their needs and expectations. Focusing on the main details and keeping the wording simple is also essential. Testing and validation should be thorough to ensure the software meets the requirements and that there are no issues or gaps. By addressing these challenges, you can create acceptance criteria that reflect the stakeholders' needs and the users' expectations, leading to a successful software development project.

Conclusion

Acceptance criteria provide a clear understanding of the needs and expectations of stakeholders and users, which in turn helps develop software that meets those needs and expectations.

To ensure that acceptance criteria are effective, you need to adhere to the best practices:

Collaborate with stakeholders

Provide examples

Review and test

Insights ▾

Despite the challenges that may arise when writing acceptance criteria, you can overcome them with clear communication and plan. If your team needs help writing effective acceptance criteria, consider hiring a team of IntelliSoft experts who can provide guidance as your acceptance criteria are clear, concise, and compelling.

Remember, effective user story acceptance criteria are essential for creating successful products. By following best practices and advice, your team can create software that meets the needs and expectations of stakeholders and users, leading to a successful and satisfying product.

Related readings:[What Is Software Testing and When Your Product Needs It?](#)[Migrating Legacy Systems: Essential Stages and Tips from Pros](#)[Key Trends Shaping Digital Transformation You Shouldn't Miss in 2022-2023](#)**About Kosta Mitrofanskiy**

I have 25 years of hands-on experience in the IT and software development industry. During this period, I have helped many companies to gain a technological edge across different industries. I can help you with dedicated teams, hire developers, developing a product design and MVP for your healthcare, logistics, or IoT projects. If you have any questions concerning our cooperation or need an NDA to sign, contact info@intellisoftware.net.

[in](#)

TABLE OF CONTENTS

1. What Are Acceptance Criteria? What Role Do they Play in Projects?[2. The Main Purposes of Acceptance Criteria](#)[3. Types and Structures of Acceptance Criteria](#)[4. General Rules on How to Write Acceptance Criteria](#)[5. Ready-to-use Acceptance Criteria Templates](#)[6. Best Practices for Writing Acceptance Criteria](#)

Intellisoft

Services ▾

Pricing

Hire Developers ▾

Industries ▾

Contact us

Related Posts

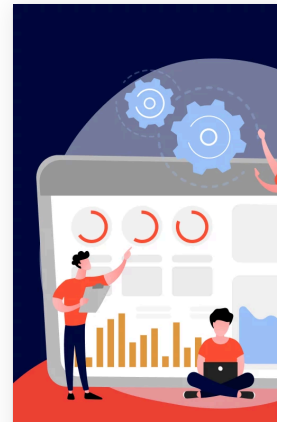
Insights ▾



**Restaurant Management Software:
How to Develop the Solution**



**Delphi Programming Language: A
Beginner's Guide**



**Software Develop
Tools: Supercharg**

Intellisoft

REVIEWED ON

5 REVIEWS

USA:

5348 Vegas Dr., Las Vegas, NV 89108, USA

Switzerland:

Bernstrasse 15, 8952 Schlieren, Switzerland

Ukraine:

29 Obolonska str, Kyiv 04071, Ukraine

Bulgaria:

Bulevard "Bulgaria" 73, 1680 Manastirski Livadi, Sofia

Categories


Team extension

Industries

Our cases

About us

Blog



Services ▾

Pricing

Hire Developers ▾

Industries ▾

Contact us

Solutions for Start-ups

Web Development Services

Product Design Services

Mobile Development Services

Quality Assurance and Testing Services

DevOps Services

Machine Learning

Data Analytics Services

Managed IT Services & Support

Cloud Development

API Integration

Data Backup and Recovery

IT Consulting

Database Management Company

Software Integration Services

Digital Transformation Services

Hire developers

Hire React.js Developers

Hire Remote Software Developers

Hire .NET Developers

Hire Rust Developers

Hire JavaScript Developer

Hire PHP Developers

Hire Node.js Developers

Hire iOS Developers

Hire Golang Developers

Hire Angular Developers

Hire Android App Developers

Hire Vue.js Developers

Hire Laravel Developers

Hire Python Developers

Hire Xamarin App Developers



Services ▾

Pricing

Hire Developers ▾

Industries ▾

Contact us

Software development company

Austin Development Company **Insights** ▾

Chicago Development Company

Los Angeles Development Company

London Development Company

New York Development Company

Nashville Development Company

Houston Development Company

Copyright © 2024 IntelliSoft Confidential. Distribution is prohibited without IntelliSoft written approval | [Privacy Policy](#)