WIKIFREE

THE STORY STUDIO 38

How to Write User Stories and Acceptance Criteria



wikifree.com

Start with a user need

A story helps the team discuss a valuable change, not merely fill in a sentence template.

Define the purpose

A user story describes a need from the perspective of someone who benefits. It supports a conversation and agreed examples of acceptable behaviour.

Class-booking case

An attendee needs to cancel a future booking so another person can use the place. The team must understand timing, ownership, capacity and confirmation.

Use the studio

Develop the story and its criteria together, then split, review and verify the work. User stories are a useful technique, not a mandatory format in Scrum.

TAKE IT TO YOUR PROJECT

Write the user and the problem before choosing the interface.

Identify the real user and outcome

A role label should point to a recognisable need.

Ask useful questions

Who experiences the problem, what are they trying to achieve, and why does the change matter? Use observation or representative examples to challenge assumptions.

Booking case

The attendee wants to release a place when plans change. Reception staff want reliable capacity. Both perspectives can reveal conditions the team should discuss.

Avoid a hidden solution

"Add a red button" names a design. Explore the cancellation outcome before deciding which interaction serves it best.

TAKE IT TO YOUR PROJECT

Describe the result the user needs without naming a screen control.

Write the first story

Use a concise form that leaves room for the right conversation.

A common format

"As a class attendee, I want to cancel my future booking so that someone else can use the place." The wording connects a role, capability and reason.

Check the value

The story should describe a useful change the team can demonstrate. A technical task can support it, but a database-only task may not deliver the user outcome on its own.

Clarify the boundary

The example concerns the attendee's own eligible booking. Discuss administrator actions, past classes and other rules rather than assuming they are included.

TAKE IT TO YOUR PROJECT

Write one story and underline its intended value.

Have the conversation

The short card is a reminder to clarify the behaviour and its limits.

Explore the rules

Discuss ordinary use, exceptions, data, timing, permissions, dependencies and quality expectations with the people who understand them.

Booking case

Can an attendee cancel after a cutoff? What happens when the class is already cancelled? How is the capacity updated, and what evidence confirms the change?

Capture decisions

Record the agreed rule, examples and unresolved questions in a form the team can use. A polished sentence cannot replace this shared understanding.

TAKE IT TO YOUR PROJECT

Ask three questions that could change your story's scope.

Write acceptance criteria

Criteria make the relevant conditions observable and testable.

Use clear outcomes

State what must hold for this story to meet the need. Avoid vague words without an agreed measure, and separate important business rules from implementation preferences.

Booking case

Successful cancellation changes the booking status, restores the appropriate capacity and creates the required confirmation under the agreed rules.

Keep them focused

Criteria belong to the story's behaviour and boundaries. Shared quality requirements may also apply through the team's Definition of Done.

TAKE IT TO YOUR PROJECT

Write three observable criteria for your story.

The All-in-One PM Templates Bundle

940+ Premium Templates • One Payment • Lifetime Access

$49

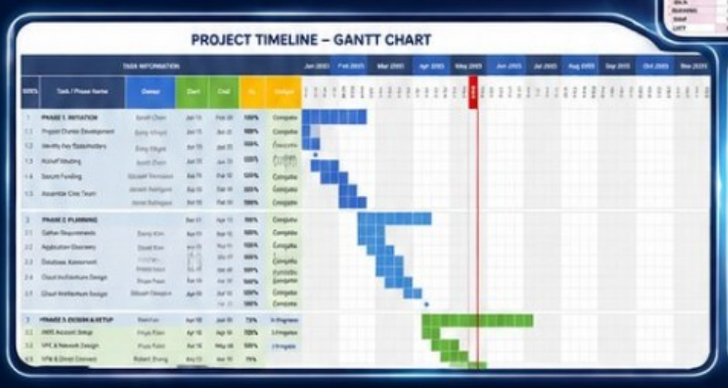
SAVE 94%

Worth $3,300+











EXCEL & GOOGLE SHEETS | PROFESSIONAL & RELIABLE | AUTOMATED DASHBOARDS | SAVE TIME & STAY ON TRACK | INSTANT ACCESS DOWNLOAD

GET LIFETIME ACCESS

Use Given-When-Then where useful

A scenario should connect context, action and observable consequence.

Understand the structure

Given establishes the relevant state. When identifies the event or action. Then describes the result that can be observed. Use the format when it improves clarity.

Booking example

Given an attendee owns an eligible future booking and the class has four available places, when the attendee cancels once, then the booking is cancelled and availability becomes five.

Check the logic

Use specific enough data to expose ambiguity. Do not hide several unrelated behaviours in one enormous scenario.

TAKE IT TO YOUR PROJECT

Write a single scenario with a measurable result.

A complete happy-path example

Make the first scenario concrete before adding exceptions.

Context

Member M1 owns confirmed booking B123 for a future class. Cancellation is allowed under the agreed timing rule. Available capacity is 4.

Action and result

When M1 confirms cancellation, B123 becomes Cancelled, availability becomes 5 and the required confirmation is queued. The screen shows the agreed successful result.

Evidence

Verify the visible result and relevant stored state. The example assumes one place per booking and no separate refund rule; define such conditions if the actual service needs them.

TAKE IT TO YOUR PROJECT

Check each expected result against the stated starting condition.

Explore exceptions deliberately

The ordinary path does not define every important boundary.

Choose representative cases

Consider ineligible timing, missing records, invalid inputs, unavailable dependencies and the state already changed. Select cases based on the real need and risk.

Booking case

An attempted cancellation after the agreed cutoff follows the authorised rule and does not silently restore capacity. A missing booking produces an understandable outcome under the service's policy.

Keep decisions explicit

If the team does not know the rule, record a question and obtain the appropriate decision. Do not let a developer or tester invent a business policy by accident.

TAKE IT TO YOUR PROJECT

Choose an exception that would materially affect the user.

Check permissions and repeated actions

A result should remain correct under relevant ownership and repetition conditions.

Ownership scenario

A different member attempts to change MI's booking. Define the permitted behaviour and ensure protected information or actions remain within the agreed access rules.

Repeated action

When a cancellation is submitted again after B123 is already cancelled, capacity must not rise from 5 to 6 merely because the action was repeated.

Use the evidence

The exact user message and technical handling can be designed appropriately, but the agreed state and outcome must remain consistent.

TAKE IT TO YOUR PROJECT

Write a scenario that proves a repeated action does not duplicate its effect.

Include relevant quality conditions

Some requirements cut across the immediate happy path.

Discuss the need

Consider accessibility, security, performance, reliability and operating support where relevant. State measurable conditions or point to the agreed shared standard.

Booking case

The cancellation flow should work with the agreed supported input methods and accessibility checks. A user unable to activate the confirmation control cannot achieve the story's value.

Connect the records

Keep story-specific criteria and shared Definition of Done understandable. Neither should be used to hide the other's required evidence.

TAKE IT TO YOUR PROJECT

Identify a cross-cutting condition affecting the story.

Review story quality

A useful story is small, valuable and understandable enough to work on.

Use a checklist

The familiar INVEST lens considers independence, negotiability, value, estimability, smallness and testability. Use it to improve the conversation rather than as an absolute scoring ritual.

Booking case

A story covering bookings, payments, refunds and event sales is probably too broad for one small delivery. Examine which useful outcome can be separated.

Check the context

Dependencies may remain even after splitting. Make them visible and avoid forcing artificial independence that harms the product's coherence.

TAKE IT TO YOUR PROJECT

Choose one weakness in a story and propose a better boundary.

Split into usable slices

A small story should still support a meaningful outcome.

Choose a suitable split

Consider business rules, user cases, data categories or workflow variations while preserving value. Keep technical tasks linked to the user-facing result.

Booking case

Start with eligible single-place cancellations, then handle a separate group-booking case if needed. Each slice needs complete behaviour and relevant quality.

Avoid a horizontal handoff

A screen without its required state change is unlikely to satisfy the cancellation need. Plan the cross-functional work for the slice.

TAKE IT TO YOUR PROJECT

Split one large story into two demonstrable results.

Connect stories with delivery and tests

Traceability helps the team preserve the need while the solution evolves.

Use the thread

Link the source need, story, agreed rules, implementation work, verification and relevant acceptance. Keep identifiers and versions proportionate to the environment.

Booking case

US-04 links to the cancellation rule, B123 scenario and repeated-action check. When the cutoff changes, the team can find affected guidance and tests.

Check the owner

A need or rule change requires the appropriate product and governance decisions. Updating a test alone does not establish an authorised policy change.

TAKE IT TO YOUR PROJECT

Trace one criterion from its source to its verification evidence.

Refine with the right people

Readiness for work comes from understanding, not only a checklist status.

Review the scope

Discuss value, examples, dependencies, uncertainty and delivery feasibility with relevant product, delivery, testing and user expertise.

Booking case

The team learns that confirmation must be queued rather than guaranteed delivered immediately through an external email service. It revises the criterion to match the agreed commitment.

Use optional working agreements

Teams may use a readiness checklist, but Scrum does not mandate a separate Definition of Ready. Keep the approach helpful and open to learning.

TAKE IT TO YOUR PROJECT

Identify the unanswered question that most affects the next delivery decision.

Work a change in the rule

A small wording change can alter several behaviours.

Scenario

The cancellation cutoff moves earlier, and a new group-booking need appears. The original story only covers one eligible place.

Reason through it

Confirm the authorised rule, assess affected scenarios, user guidance and dependencies, and decide whether group cancellation should be a separate slice. Keep the current commitment clear.

Verify the result

Recheck normal, ineligible and repeated-action scenarios against the revised rule. Record the evidence and any unresolved decisions.

TAKE IT TO YOUR PROJECT

List the records affected by one changed acceptance rule.



Learn from 350+ top universities and companies

ILLINOIS

Duke UNIVERSITY

Google

UNIVERSITY OF MICHIGAN

IBM

VANDERBILT UNIVERSITY

JOHNS HOPKINS UNIVERSITY

Invest in your career

Explore new skills

Access 10,000+ courses in AI, business, technology, and more.

Your story-and-criteria desk check

The team should be able to explain, demonstrate and verify the intended change.

Check the need

Is the user recognisable, the value meaningful and the boundary small enough? Which rule or dependency still needs a decision?

Check the examples

Do criteria show observable results with clear starting conditions, relevant exceptions and shared quality expectations? Are the numbers and state changes consistent?

Use the booking case

One eligible cancellation changes B123 once and capacity from 4 to 5, with the agreed confirmation and access behaviour. Apply that precision to your own story.

TAKE IT TO YOUR PROJECT

Ask a teammate to demonstrate the story from its criteria before implementation.