

Implementing Acceptance Criteria with Xray and Jira

- [Overview](#)
 - [Jira and Acceptance Criteria](#)
 - [Xray and Acceptance Criteria](#)
- [Having explicit visibility of Acceptance Criteria and tracking their status with Xray](#)
 - [AC implemented as "sub-requirements" in Xray](#)
 - [Configuring it](#)
 - [Using it](#)
 - [Supported features](#)
- [FAQ](#)
- [References](#)

Overview

In Agile Methodologies, Acceptance Criteria (AC) are used to define when a given user story can be "accepted" by some relevant stakeholder.

The criteria that a component or system must satisfy in order to be accepted by a user, customer, or other authorized entity. - ISTQB Glossary

Acceptance Criteria:

- clarify some aspects of the underlying story that are relevant to someone;
- can be checked and validated by testing; they should be clear enough so that clear accept/don't accept decisions can be taken.

But the following question arises: How can we know whether a certain criterion has been checked or validated by testing?

Before diving into it, we need to be aware of typical patterns teams follow for handling AC.



Please note

"Accepting" the story doesn't mean it's bug-free or that acceptance criteria should describe all possible usage scenarios about the story; it's only a way to clarify some usage rules that are important for some stakeholders and that the user story "can work".

A common misconception is on "done" vs "acceptance criteria". Acceptance criteria are user story specific, while the definition of done applies to all product increments/user stories.

Therefore the fact that acceptance criteria have been met, doesn't mean it can be considered as "done" and thus ready to be shipped.

Jira and Acceptance Criteria

Jira doesn't enforce a way of handling AC.

Usually, teams describe AC as a list in the Description field of the related Story.

As a user, I can login the website to perform restricted operations

 Attach

 Create subtask

 Link issue



 Test Coverage



Description

As a user, I can login the website to perform restricted operations

Acceptance Criteria:

- a specific login page must exist for authenticating
- a user profile has a username and password data

AC can be enumerated using bullets: either numbered bullets or standard ones but with some prefix.

This is an unstructured approach that inhibits full traceability between story<=>AC<=>tests<=>test runs<=>defects.

Some teams may decide to use specific custom fields for this purpose or even to use specific Jira apps/plugins to clearly manage AC. However, these usually don't have unique IDs that can be referred elsewhere.

Xray and Acceptance Criteria

Xray does not provide nor enforce a way of handling AC. But, you can take advantage of some built-in capabilities to have more explicit visibility of it.

While most teams create Tests and associate them to the related requirements/stories (e.g., Story<=>Test(s)), this may not be sufficient if you want to analyze each criterion individually.

If you aim to understand which Tests address a specific acceptance criterion and how it is based on the related testing results, then we need to turn each acceptance criterion into a manageable, identifiable, and coverable entity.

That will allow us to have more detailed traceability, such as **Epic<->Story/ies<->Acceptance Criteria<->Test(s)**, and track the coverage status up to the Acceptance Criteria level.

Having explicit visibility of Acceptance Criteria and tracking their status with Xray

AC implemented as "sub-requirements" in Xray

Xray supports the concept of [requirements and sub-requirements](#).

In this approach, a given story/requirement is decomposed into multiple sub-requirements, which in turn can be covered by tests.

The sub-requirement, in this case, is our acceptance criterion that will be implemented as an issue.

We can define the relation between the Story and related acceptance criteria based on a sub-task or on a specific issue link type, accordingly to what we have defined on [Xray coverage-related settings](#).

Configuring it

1. Create an issue type named "Acceptance Criterion" or equivalent. Go to **Jira Settings > Issues > Issue Types**.

Add Issue Type

Name *

Acceptance Criterion

Description

Type *

☒ Standard Issue Type

☐ Sub-Task Issue Type

Add Cancel

a.

2. Add that issue type to the issue type scheme used by your Jira project
3. Define in [Xray settings](#), the sub-requirement (e.g. the "Acceptance Criterion" issue) to be handled as a coverable issue (i.e. by adding it to the "Requirement Issue Types" column)

Xray Settings

Summary

Miscellaneous

Test Coverage

Defect Mapping

Test Types

Test Environments

Document Generator

Test Step Fields

Test Run Custom Fields

Parameter value lists

Default Column Layouts

Test Coverage

A testable/coverable entity is an issue that you can cover with tests in order to validate

Available Issue Types

- ☒ Task
- ☐ Sub-task
- ☐ Bug
- ☐ Sub Test Execution
- ☐ Precondition
- ☐ Test
- ☐ Test Set
- ☐ Test Plan
- ☐ Test Execution

Coverable Issue Types

- ☒ Acceptance Criterion
- ☐ Story
- ☐ Epic

a.

4. Define in [Xray settings](#), the relation between the parent requirement (e.g. the "Story" issue) and the sub-requirement (e.g. the "Acceptance

Test Coverage Hierarchy

☒ Epic - Issues(Stories) relation
When this option is enabled, Xray will consider all child issues that are associated with Epics. These child issues must still be mapped as Test Coverage issue types.

☒ Issue - Sub-tasks relation
When this option is enabled, Xray will consider all sub-task issues. Either the Issue Type and the Sub-task issue type must be mapped as Test Coverage issue types.

Issue Links relation

Satisfy

Issue Link Type Direction

☒ Outward - satisfies

☐ Inward - is satisfied by

Xray will use this issue link type configuration for the hierarchy between issues. Any issue type using this link type must also be mapped as Test Coverage issue type.

Criterion" issue)

- a. Note: if you want to define this relation based on a custom issue link type (e.g., you could create a "Satisfy: satisfies/is satisfied by"), then you need to create it first

Issues

Issue linking is currently ON.

To deactivate issue linking, simply click below.

Deactivate

 For the users you wish to be able to link issues, ensure that they have the **Link Issues** permission for that particular project.

Name	Outward Description	Inward Description
Blocks	blocks	is blocked by
Cloners	clones	is cloned by
Defect	created	created by
Duplicate	duplicates	is duplicated by
Relates	relates to	relates to
Test	tests	is tested by

Add New Link Type

Add a new link type

Name

Satisfy

(eg "Duplicate")

Outward Link Description

satisfies

(eg "duplicates")

Inward Link Description

is satisfied by

(eg "is duplicated by")

Add

i.



Please note

In Xray, sub-requirements can be identified and related to parent requirements using:

- a specific, configured, issue link (e.g., you could create a "Satisfy: satisfies/is satisfied by")
- sub-tasks

Using sub-tasks may allow further control over the progress of the parent issue (e.g. the Story), such as giving the ability to restrict closing it or not. However, the usage of "sub-tasks" namely for this purpose, is something that has to be evaluated in your specific context.

To use sub-tasks:

1. Please make sure they are enabled. Go to **Jira Settings > Issues > Issue types > Sub-tasks**.
2. Create the sub-requirement/acceptance criterion as a sub-task issue type. Go to **Jira Settings > Issues > Issue Types**.

Add Issue Type

Name *

Acceptance Criterion

Description

Type *

- ☐ Standard Issue Type
- ☒ Sub-Task Issue Type

Add

Cancel

a.

3. In [Xray settings](#), make sure the Issue-Sub-tasks relationship is defined for test coverage hierarchy

Test Coverage Hierarchy

☒ Epic - Issues(Stories) relation
When this option is enabled, Xray will consider all child issues that are associated with Epics. These child issues must still be mapped as Test Coverage issue types.

☒ Issue - Sub-tasks relation
When this option is enabled, Xray will consider all sub-task issues. Either the Issue Type and the Sub-task issue type must be mapped as Test Coverage issue types.

Issue Links relation

Satisfy

Issue Link Type Direction

☒ Outward - satisfies

☐ Inward - is satisfied by

Xray will use this issue link type configuration for the hierarchy between issues. Any issue type using this link type must also be mapped as Test Coverage issue type.

a.

Using it

Create the Story as usual. The story may be part of an Epic or not.

- If using sub-tasks as a way to relate the AC to the Story, then proceed as follows:

Projects / Calculator / CALC-1216 / CALC-1303


As a user, I can login the website to perform restricted operations

[Attach](#) [Create subtask](#) [Link issue](#) [Test Coverage](#) [...](#)



Description


As a user, I can login the website to perform restricted operations

Subtasks

 CALC-1304

a specific login must exist for authenticating

  TO DO

 Acceptance Criterion

a user profile has a username and password data

Create

Cancel

From the Acceptance Criterion issue create Tests and execute them as usual, reporting the final status (e.g., PASSED/FAILED)

a specific login page must exist for authenticating

Attach

Link issue

Test Coverage

...

Description

Add a description...

Test Coverage



UNCOVERED

No Tests are associated with this issue.

→

Create Test

Test Execution for CALC-1304

Attach

Create subtask

Link issue

Tests

...

Description

Add a description...

Tests

Add Tests

Overall Execution Status



1 PASSED

TOTAL TESTS: 1

		Filters			100	Columns		
	Rank	Key	Summary	Test Type	TestRun Assignee	Status	Actions	
<input type="checkbox"/>	1	CALC-1306	Test a specific login page must exist for authenticating	Manual	Sérgio Freire	<div>PASSED</div>	<div></div>	...

You can then check coverage directly on the Story, which is based on all related testing results (i.e., results from Tests covering all the related Acceptance Criterion issues).

As a user, I can login the website to perform restricted operations

- Attach
- Create subtask
- Link issue
- Test Coverage
-

Description

As a user, I can login the website to perform restricted operations

Subtasks

0% Done

CALC-1304	a specific login page must exist for authenticating	TO DO
CALC-1305	a user profile has a username and password data	TO DO

Test Coverage

- Create Test
- Execute

Analysis & Scope

Scope: Latest Final Status

NOK

Status	Key	Summary	Test Status
TO DO	CALC-1306	Test a specific login page must exist for authenticating	PASSED
TO DO	CALC-1308	Test a user profile has a username and password data	FAILED

You can also go to each Acceptance Criterion issue to understand if it's being satisfied or not, based on the coverage information, which in turn is based on the related testing results. In this example, one of the AC (CALC-1305) is NOK.

a specific login page must exist for authenticating

Attach

Link issue

▼

Test Coverage

...

Description

Add a description...

Linked issues

is tested by

CALC-1306

Test a specific login page must exist for authenticating

⌵

TO DO ▼

Test Coverage

Create Test

Execute ▼

Analysis & Scope

Scope: Latest Final Status ▼

→

OK

Status	Key	Summary	Test Status
⌵ TO DO	CALC-1306	Test a specific login page must exist for authenticating	PASSED

Prev 1 Next

a user profile has a username and password data

Attach

Link issue

▼

Test Coverage

...

Description

Add a description...

Linked issues

is tested by

CALC-1308

Test a user profile has a username and password data

⌵

TO DO ▼

Test Coverage

Create Test

Execute ▼

Analysis & Scope

Scope: Latest Final Status ▼

NOK

Status	Key	Summary	Test Status
⌵ TO DO	CALC-1308	Test a user profile has a username and password data	FAILED

Prev 1 Next

In the Traceability Report, it's possible to see full traceability between Epic<=>Stories<=>Acceptance Criteria<=>Tests<=>Test Runs<=>Defects. The following screenshot clearly shows this hierarchy:

Traceability Report ?

[Share](#) [Export](#)

Analysis & Scope Filters

Scope: Latest Final Status ▼ Project: Calculator Contains: CALC-1216 Issue Type: Epic, Story, Acceptan... ▼ ✔ Show Test Runs

QUICK FILTERS: OK (0) NOK (1) NOTRUN (0) UNKNOWN (0) UNCOVERED (0)

REQUIREMENTS	TESTS	TESTRUNS	DEFECTS
<p>▼ CALC-1216 TO DO</p> <p>authentication</p> <p>NOK</p>			
<p>▼ CALC-1303 TO DO</p> <p>As a user, I can login the website to perform restricted operations</p> <p>NOK</p>			
<p>CALC-1304 TO DO</p> <p>a specific login page must exist for authenticating</p> <p>OK</p>	<p>CALC-1306 TO DO</p> <p>Test a specific login page must exist for authenticating</p> <p>PASSED</p>	<p>CALC-1307 View Details</p> <p>Fix Version/s: - Finished On: Today 04:33 PM Executed By: Sérgio Freire Test Environments: -</p> <p>PASSED</p>	
<p>CALC-1305 TO DO</p> <p>a user profile has a username and password data</p> <p>NOK</p>	<p>CALC-1308 TO DO</p> <p>Test a user profile has a username and password data</p> <p>FAILED</p>	<p>CALC-1309 View Details</p> <p>Fix Version/s: - Finished On: Today 04:38 PM Executed By: Sérgio Freire Test Environments: -</p> <p>FAILED</p>	<p>CALC-1310 TO DO</p> <p>username field is missing</p>

Supported features

	Supported?
create one or more AC per Story/requirement	X
uniquely identify each AC	X
ability to create linkable references to AC (e.g ability to mention AC from somewhere else)	X
AC as issues	X
see AC on the Story/requirement issue screen along with their status	X
link Tests to each AC	X
"ignore" AC without Tests for the calculation of coverage	X
full traceability story/requirement<=>AC<=>tests<=>test runs<=>defects	X

FAQ

- Do we have to implement AC as issues?
 - No. Only if you wish to track coverage status (i.e., their acceptance information) on a per acceptance criterion basis.
- Can a Test be considered a form of AC?
 - A test is a concrete example of a certain usage of the requirement/story, so to an extent it can be seen as a sort of acceptance criteria. However, usually tests are one level below AC and contain fine details.
- How can we implement Acceptance Criteria whenever adopting Gherkin (e.g. "Cucumber") to support our BDD endeavour?
 - Recent versions of Cucumber have support for the Rule entity, which is an acceptance criterion. The Rule sits between the Feature and the Scenarios. Each Scenario represents an example (and thus Scenario have been renamed to Example) of a given Rule. However, even if you're not adopting a recent version of Gherkin supporting the Rule keyword, your Scenario/Scenario Outline can represent AC; it all depends on how you use them. Xray doesn't yet support the Rule keyword.
- Are there other ways of implementing AC in Jira? How? Can they be used together with Xray?

- You can use your own way to implement AC, such as having a dedicated custom field on the Story. You can also decide to use a Jira app/plugin (e.g., Checklist for Jira, Smart Checklist for Jira) for that purpose. All those are valid approaches but keep in mind that if you want to track the coverage status of each acceptance criterion defined that way, as these are not entities that Xray has visibility of, then you won't be able to do so.

References

- <https://callumakehurstriansblog.wordpress.com/2021/08/25/why-checking-an-ac-isnt-enough/amp/>
- <https://cucumber.io/blog/example-mapping-introduction/>