# 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.

Projects / 🛃 ET Workshop / 🔲 ET-1

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



#### 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.

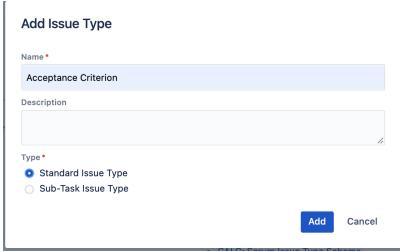
In this approach, a given story/requirement is decomposed into multiple sub-requirements, which in turn can be covered by tests.

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.

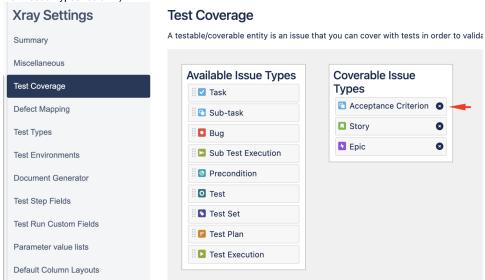
#### Configuring it

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



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)



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



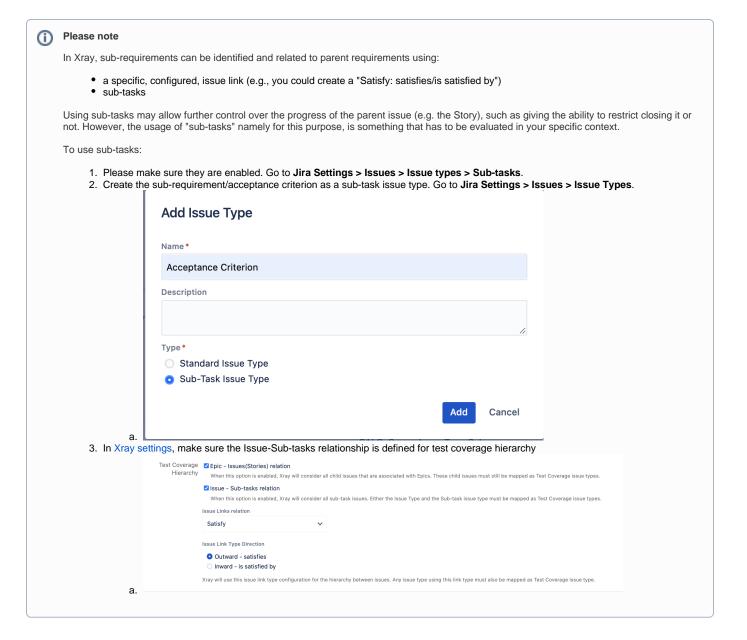
Criterion" issue)

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

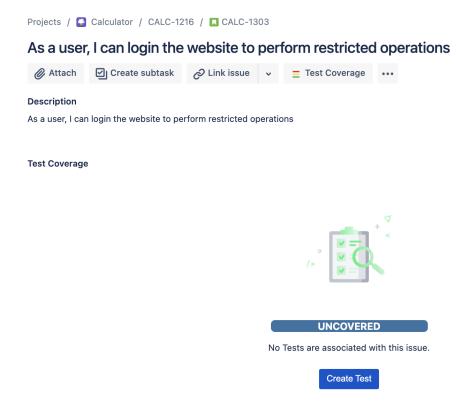
#### Issue linking is currently ON. To deactivate issue linking, simply click below. Deactivate To the users you wish to be able to link issues, ensure that they have the Link Issues permission for that particular project. Outward Description Inward Description Blocks blocks is blocked by Cloners clones 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

①



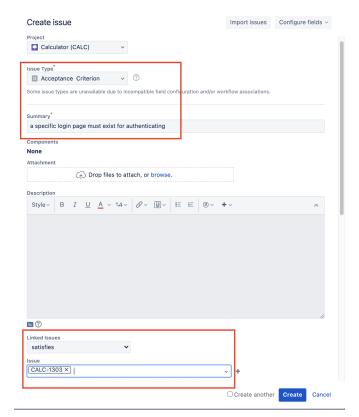
#### **Using it**

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



Then create the Acceptance Criterion issues. The exact details depend on how you decided to implement the AC and relate them to the Story issues:

If using a specific issue link type to relate the AC to the Story, then:
 Create the Acceptance Criterion issue and link the previous issue to the Story using the configured issue link type:



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

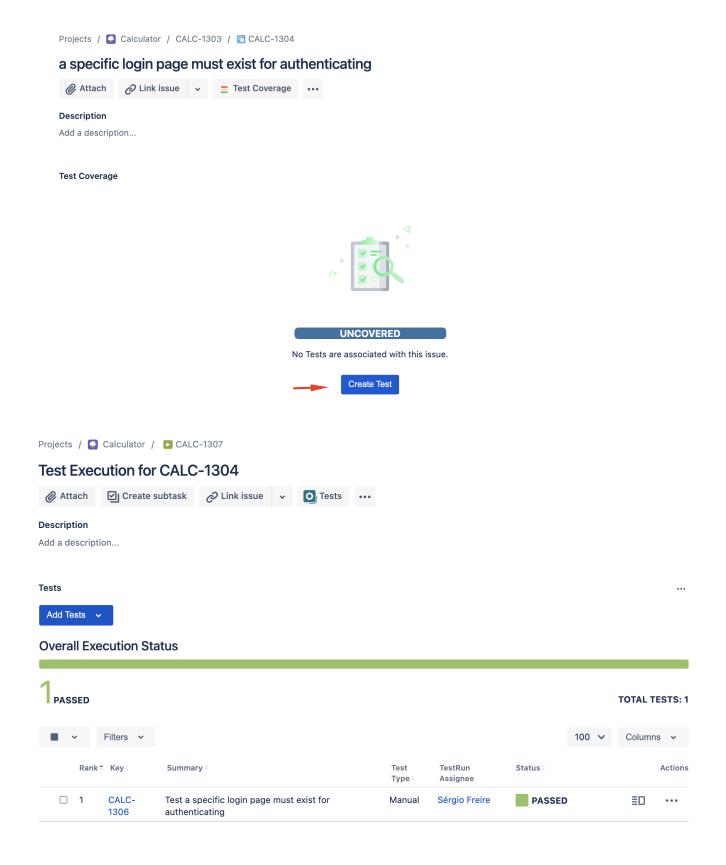
Projects / Calculator / CALC-1216 / CALC-1303

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

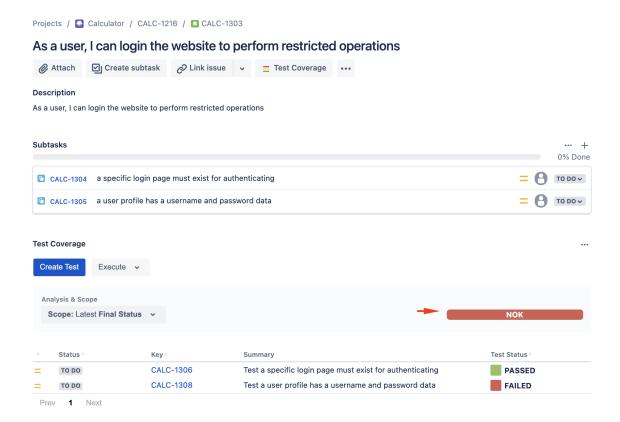




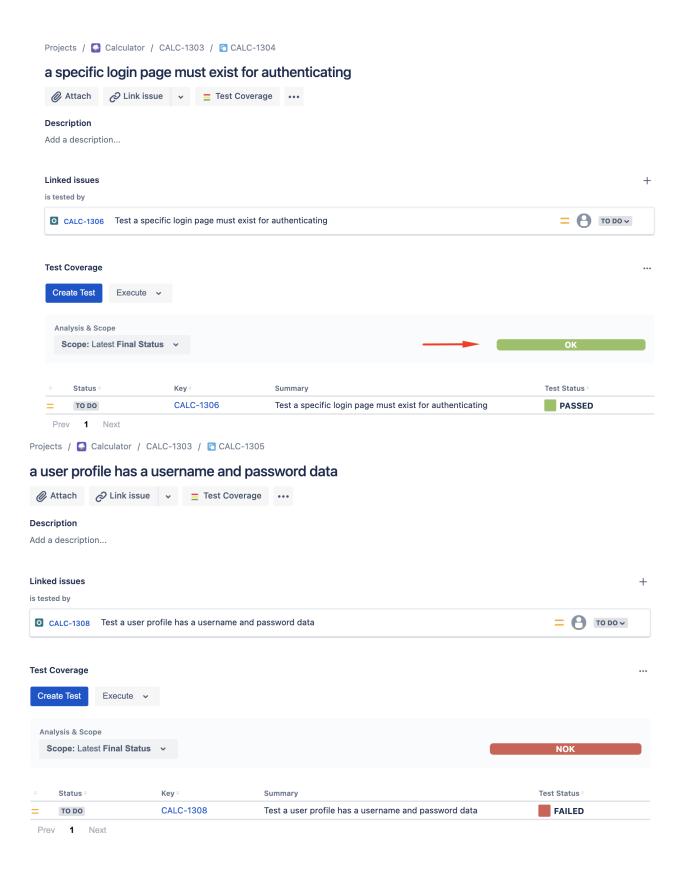
From the Acceptance Criterion issue create Tests and execute them as usual, reporting the final status (e.g., PASSED/FAILED)



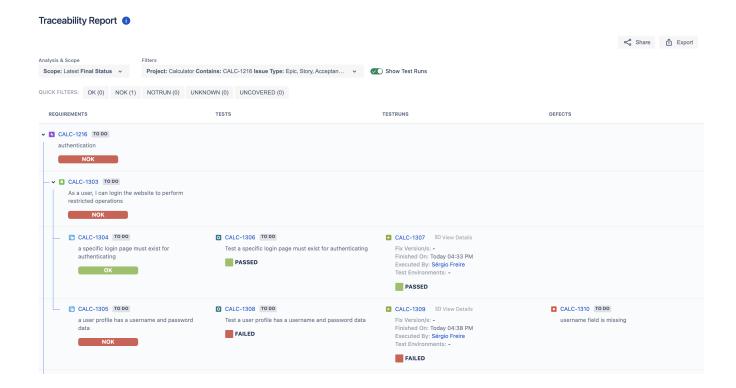
You can then check coverage directly on the Story, which is based on all related testing results (i.e., results from Tests covering all the related Acceptance Criterion issues).



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.



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:



#### Supported features

	Supported?
create one or more AC per Story/requirement	х
uniquely identify each AC	х
ability to create linkable references to AC	x
(.e.g ability to mention AC from somewhere else)	
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	х
story/requirement<=>AC<=>tests<=>test runs<=>defects	

#### FAQ

- Do we have to implement AC as issues?
  - o 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?

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