

Requirements Coverage Analysis

Xray provides useful and detailed information about the Requirements Coverage in every project via the **Requirement Coverage Charts** tab panel and also the **Requirement Status** custom field.

- Requirement Coverage Status
 - Requirement conjunction
 - Ignore Requirements with statuses
 - Ignore Requirements with labels
 - Ignore Tests with statuses
- Requirement Test Coverage Section
 - Analysis by version
 - Analysis by Test Plan
- Requirement Coverage Charts
 - Analysis by Version
 - Analysis by Test Plan
 - Analysis by Environment
 - Grouping Results
 - Requirement Filters
- Requirement Status Custom Field
 - No Version
 - Earliest Unreleased Version
 - Unreleased Versions
 - Assigned Versions
- Requirement Coverage Strategy
 - Use versioned Test Executions for Requirement Coverage
 - Use versioned Test Sets for Requirement Coverage
- Troubleshooting

Requirement Coverage Status

The Requirement issue coverage status is calculated based on all Tests that are associated with the Requirement and all the Sub-Requirements Status.

- **OK** - All the Tests associated with the Requirement **PASSED**
- **NOK** - At least one Test associated with the Requirement **FAILED**
- **NOT RUN** - At least one Test associated with the Requirement is **TODO** or **ABORTED** and there are no Tests with status **FAILED**
- **UNKNOWN** - At least one Test associated with the Requirement is **UNKNOWN** and there are no Tests with status **FAILED**
- **UNCOVERED** - The Requirement has no Tests associated with it

The **Status** of a Test for a particular Requirement is calculated based in the Requirement Coverage Strategy.

Requirement conjunction

REQ SUB-REQ	OK	NOK	NOT RUN	UNKNOWN	UNCOVERED
OK	OK	NOK	NOT RUN	UNKNOWN	OK
NOK	NOK	NOK	NOK	NOK	NOK
NOT RUN	NOT RUN	NOK	NOT RUN	UNKNOWN	NOT RUN
UNKNOWN	UNKNOWN	NOK	UNKNOWN	UNKNOWN	UNKNOWN
UNCOVERED	OK	NOK	NOT RUN	UNKNOWN	UNCOVERED



Rationale

Even if you have sub-requirements, when you have tests that are directly linked to the parent requirement, Xray assumes that you are validating the requirement directly. Thus, it's irrelevant if the sub-requirements are uncovered by tests.

Ignore Requirements with statuses

Requirement issues with certain statuses can be excluded from the Requirement Coverage charts. These statuses can be configured in the Xray configuration.

Ignore Requirements with labels

Requirement issues with certain labels can be excluded from the Requirement Coverage charts. These labels can be configured in the Xray configuration.

Ignore Tests with statuses

Test issues with certain statuses can be excluded from the Requirement Coverage calculation. Test issues with ignored statuses will not be considered in the calculation of the Requirement Status, even if they are associated with Requirement issues. These statuses can be configured in the Xray configuration.

Requirement Test Coverage Section

The Requirement Status and the Test Coverage are presented in the Requirement issue view.

Analysis by version

You can choose a specific **analysis version** to calculate the Requirement Status. This can be useful when testing multiple versions of your requirements at the same time or if you need to see the requirement coverage for previous versions. The **analysis version** can be **No Version**, in which case the requirement status will be calculated based on the latest execution for each Test, regardless of the execution version.

Test Coverage +

[Create new Test](#) [Create new Sub Test Execution](#)

Version NOK

↑	CLOSED	Done	WEB-142	Test As a user i can login with my Facebook account	PASS
↑	OPEN	Unresolved	WEB-103	Test As a user i can login with my Facebook account	FAIL
↑	CLOSED	Done	WEB-136	Test As a user i can login with my Facebook account	PASS
↑	OPEN	Unresolved	WEB-135	Test As a user i can login with my Facebook account	EXECUTING
↑	OPEN	Unresolved	WEB-139	Test As a user i can login with my Facebook account	PASS
↑	OPEN	Unresolved	WEB-105	Test As a user i can login with my Facebook account	FAIL
↑	OPEN	Unresolved	WEB-138	Test As a user i can login with my Facebook account	EXECUTING
↑	OPEN	Unresolved	WEB-104	Test As a user i can login with my Facebook account	FAIL
↑	CLOSED	Won't Do	WEB-137	Test As a user i can login with my Facebook account	TODO

Analysis by Test Plan

You can also choose a specific **Test Plan** to calculate the Requirement Status. This will calculate the requirement status based on the executions for that Test Plan. This can be useful when you have multiple test plans and need to see the requirement coverage for a specific Test Plan.

Test Coverage

Create new Test

Create new Sub Test Execution

Test Plan ▾

WEB-148 - As a use...

NOT RUN

↑	CLOSED	Done	WEB-142	Test As a user i can login with my Facebook account	PASS
↑	OPEN	Unresolved	WEB-103	Test As a user i can login with my Facebook account	N/A
↑	CLOSED	Done	WEB-136	Test As a user i can login with my Facebook account	N/A
↑	OPEN	Unresolved	WEB-135	Test As a user i can login with my Facebook account	N/A
↑	OPEN	Unresolved	WEB-139	Test As a user i can login with my Facebook account	ABORTED
↑	OPEN	Unresolved	WEB-105	Test As a user i can login with my Facebook account	N/A
↑	OPEN	Unresolved	WEB-138	Test As a user i can login with my Facebook account	PASS
↑	OPEN	Unresolved	WEB-104	Test As a user i can login with my Facebook account	N/A
↑	CLOSED	Won't Do	WEB-137	Test As a user i can login with my Facebook account	N/A

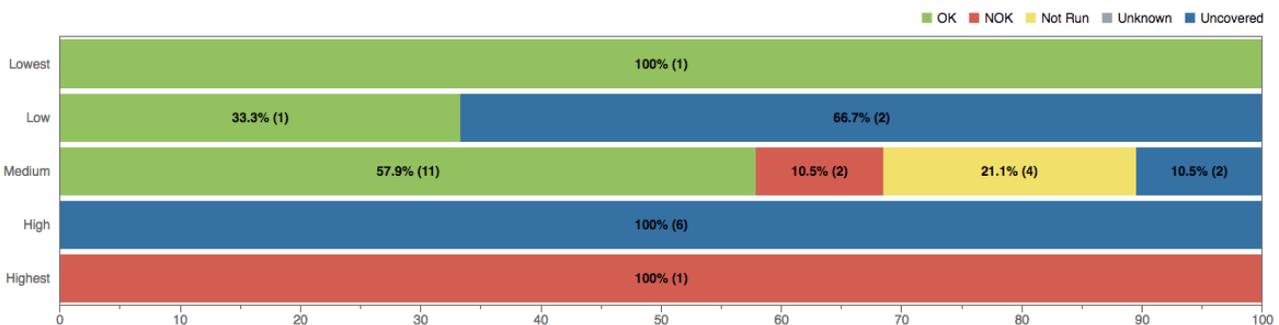
Requirement Coverage Charts

This is divided in two reports:

1. The [Overall Coverage Chart](#) provides an overall view of the current Requirement coverage of a project for a particular Version.
2. The [Daily Historical Requirement Coverage](#) contains a comprehensive stacked bar chart to see the evolution of the requirement coverage over time for a particular Version.

Overall Coverage Chart [Switch report ▾](#)

Version ▾ None (latest execution) ▾ All Environments ▾ Priority ▾ Saved Filter: R1.2.2 Stories ▾



The requirement coverage reports allow you to analyse the test status of requirement issues in the current project, provide an overview of how many requirements are OK, NOK, Not Run and Uncovered, and an overview of the requirement status by version or test plan.

Analysis by Version

You can choose a specific **analysis version** to generate the report. This will calculate the requirement status based on executions for that version. This can be useful when testing multiple versions of your requirements at the same time or if you need to see the requirement coverage for previous versions. The **analysis version** can be **None**, in which case the requirement status will be calculated based in the latest execution for each Test, regardless of the execution version.

Analysis by Test Plan

You can also choose a specific **Test Plan** to generate the report. This will calculate the requirement status based on the executions for that Test Plan. This can be useful when you have multiple test plans and need to see the requirement coverage for a specific Test Plan.

Analysis by Environment

Xray also allows you to specify a **Test Environment** to calculate the status of requirement issues. You can specify an Environment when using both Version or Test Plan analysis. If you want to aggregate the status for all environments, just choose the **All Environments** option. If you choose a specific environment, then Xray will only consider the Test Execution issues that are within this environment when calculating the Requirement Status.

Grouping Results

You can also choose a **grouping field** to group requirements within these charts. You can, for instance, generate a report based on requirement **Priority**, **Resolution**, or any other field of type "Select List" that you have configured for your requirement issues.

Requirement Filters

In the Requirement Coverage charts, you can filter requirements to show in the charts. Use the basic filter fields:

- Fix Version (you can also choose whether to include the previous versions -- equivalent in JQL to fixVersion <= X. The default value for this option can be changed in the Xray configuration.)
- Workflow Status
- Resolution
- Assignee
- Key or Summary
- Component

or use a previously "Saved Filter" with Requirement issues.

Fix Version: v1.0, Status: Open

FILTERS | **SAVED FILTER**

Fix Version	Assignee	Component	Issue Key or Summary
v1.0	All		
Status	Resolution		
Open	All		

Include Previous Versions

Apply Clear Cancel

Each time a user accesses the "Requirement Coverage" project page, the chart will be generated with the user's last chosen options. This includes the report and all the filters for the requirement issues.

There is also an options menu where you can choose the visualization type for the charts:

- **hierarchical** - only the parent requirement issues will be presented in the charts. Sub-requirements can still be visualized in the overall details table.
- **flatten** - the requirement issues will not consider the parent/child relationship. All parent and child requirements will be considered for the chart.

Requirement Status Custom Field

The Requirement Status custom field is created when Xray is installed and associated with the Jira Default Issue Screen. This custom field is only shown for a particular issue if its issue type is defined as a Requirement in the Xray configuration.

The Requirement Status displays a color bar reflecting the current Status of the Requirement coverage. When you hover over this bar, a detail bar appears showing all the tests for the Requirement and their current status. The number of Tests in each status is also displayed and can be clicked to show the Test issues in the issue search page.

Because the status of a Requirement can depend on the Version, you can choose from the following strategies. These strategies can be changed in the Xray configuration.

No Version

When this option is enabled, the status of each Test associated with the Requirement is calculated based in the latest Test Run, regardless of the version it was executed.

Earliest Unreleased Version

When this option is enabled, only the earliest unreleased Fix Version is displayed in the Custom Field.

Unreleased Versions

When this option is enabled, all unreleased Fix Versions are displayed in the Custom Field.

Assigned Versions

When this option is enabled, only explicitly assigned Fix Versions are displayed in the Custom Field.

 **Calculator** / CALC-13

Multiplication and Division Operations

Edit Comment Assign More ▾ Start Progress Resolve Issue Workflow ▾

Details

Type:	<input checked="" type="checkbox"/> Requirement	Status:	→ Open (View Workflow)
Priority:	↑ Major	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	v1.0
Labels:	None		
Requirement Status:	v1.0 - NOTRUN		

Labels:	None
Requirement Status:	<div style="display: flex; align-items: center;"><div style="width: 50%; height: 15px; background-color: #4CAF50; text-align: center; color: white; font-weight: bold;">1</div><div style="width: 50%; height: 15px; background-color: #FFEB3B; text-align: center; color: white; font-weight: bold;">1</div></div>

Requirement Coverage Strategy

The requirement coverage strategy defines how the Requirement Status must be calculated. The strategy must be configured in the Xray configuration.

Use versioned Test Executions for Requirement Coverage

When this option is enabled in Xray configuration, the **Status** of a Requirement for a specific Fix Version **V** is calculated based on the latest Test Run with Fix Version **V** of each associated Test ([check how the latest Test Run is chosen](#)). With this strategy, a Requirement can have different statuses, one for each Fix Version in the project.

Use versioned Test Sets for Requirement Coverage

When this option is enabled in the Xray configuration, the requirement coverage will be calculated based on the Test Sets.

The main goal of this option is to use Test Sets as containers for tests of different versions related to the same requirement. This allows for regression Test Sets in subsequent versions.

Requirement Tests must be associated with Test Sets of a specific Fix Version to be considered for Requirement Coverage of that particular version. A given Test **T** is considered for requirement coverage of Requirement **R** in the version **V** if:

- 1) **T** is associated with **R** with link type relation "tests/is tested by"
- 2) **T** is associated with any Test Set with version **V**

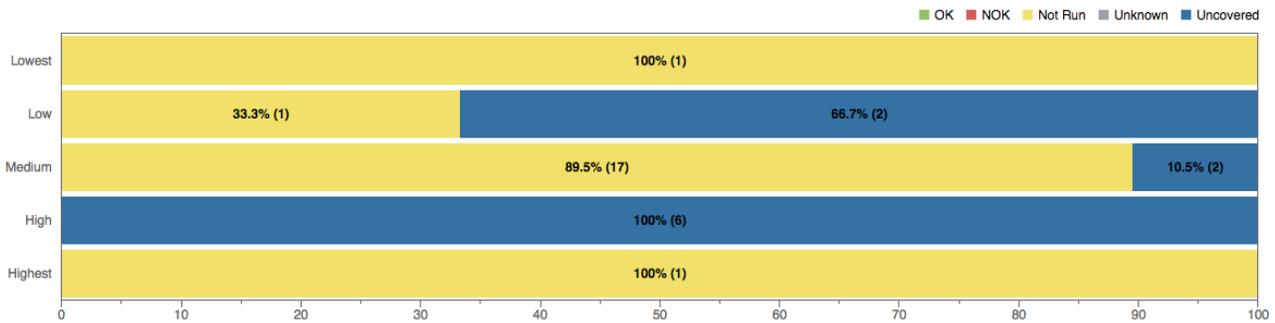
Once Test **T** is considered for coverage in version **V**, its status is calculated based on the latest Test Run with the specified Fix Version **V** ([check how the latest Test Run is chosen](#)).

Troubleshooting

"I have an Xray Project to Test my Requirements project. I've created a Test and I have already executed the Test, but I always get requirements status Covered but Not Run."

Overall Coverage Chart [Switch report](#)

Version [1.0](#) [IOS](#) [Priority](#) [Saved Filter: R1.2.2 Stories](#)



You might be using environments on Test Executions that are affecting the aggregated status for the Tests. If you have a couple of Test Executions, one with a specific environment (e.g., Android) and another without any environment, then Xray will also consider the empty environment when calculating the aggregated status (this is, of course, if all Test Executions are within the same Version or Test Plan).



You have not configured your versions accordingly. Remember that requirements coverage is calculated with [Test Execution](#) and/or Test Set for a Requirement Fix version.



You must have the same version name between projects. The Test Execution **fix version** must have the same value as the Requirement Issue **fix version**.

Create Amazing Addon Mobile App

Key: CAAMA | Lead: admin | Category: None | URL: No URL

Overview Administration

Summary

Issue Types

- Bug
- Improvement
- New Feature
- Pre-Condition
- Sub-task
- Task
- Test
- Test Execution
- Test Set

Workflows

Screens

Fields

Versions

Components

Roles

Permissions

Issue Security

Notifications

HipChat Integration

Issue Collectors

Test CAAMA

Key: TC | Lead: admin | Category: None | URL: No URL

Overview Administration

Summary

Issue Types

- Bug
- Improvement
- New Feature
- Pre-Condition
- Sub-task
- Task
- Test
- Test Execution
- Test Set

Workflows

Screens

Fields

Versions

Components

Roles

Permissions

Issue Security

Notifications

HipChat Integration

Issue Collectors

Atlassian JIRA Project Management Software (v4.4.5464203-sha1:78a0362) - About JIRA | Report a problem

The JIRA site is for non-production use only.

Atlassian

Test CAAMA / TC-4

Create SQLite database - T.E

[Edit](#) | [Comment](#) | [Assign](#) | [More](#) | [Start Progress](#) | [Resolve Issue](#) | [Close Issue](#) | [Admin](#)

Type: **Test Execution** | Status: **Unresolved** [View Workflow](#)
 Priority: **Major** | Resolution: **None**
 Affects Versions: **None** | Fix Versions: **v1.1.0**
 Labels: **None**

Assignee: **admin**
 Reporter: **admin**
 Views: **1**
 Watchers: **1** (Stop watching this issue)

Description
 Click to add description

Test Execution
 Click to add a Begin Date
 Click to add an End Date
 Click to add Revision Description

Dates
 Created: 2 days ago
 Updated: 2 days ago

Tests
[+ Add](#)

Overall Execution Status
 TOTAL TESTS: 1

FILTERS
 Test Set: All | Assignee: All | Status: All | Search: [Clear](#)

Show 10 entries

Key	Summary	Test Type	AtRisk	IdEst	Test Sets	Assignee	Status
TC-2	Create the database exclusively Test	Manual	1	0	admin	Unresolved	

Showing 1 to 1 of 1 entries

[First](#) | [Previous](#) | [Next](#) | [Last](#)

Activity
[All](#) | [Comments](#) | [Work Log](#) | [History](#) | [Activity](#) | [Comments](#)

There are no comments yet on this issue.

[Comment](#)