

# Test Versioning

When talking about versioning, different things may come to the surface that may somehow be related.

Is it requirements versioning? Test versioning? Are you talking about tracking changes to your artifacts or having explicit versions each time you change an artifact?

Let's see first how Jira handles changes and versions in general.

## How Jira handles versioning

As preliminary information, Jira issues are assigned to a given version by specifying the "Fix Version" field.

Besides this, Jira ensures that changes are tracked under the History section available in the corresponding issue screen, so you know who changed what and when.

If you have a requirement (e.g., a story) for version 1.0 and you want to change it somehow in a future version (e.g., v2.0), normally, you will have to create another requirement issue to specify the new behavior.

However, if you are changing the specification during the development lifecycle of that requirement, you can simply update its description and use Jira workflows to guarantee that is being handled properly.

If you make changes in some of the attributes of an issue, it gets tracked in the History changes; however, the version is not incremented because it is used for another purpose: to tell for which version you're delivering it.

## How does Xray handle "versioning" of Test cases

Versioning of Tests work in the same way as versioning of other issue types in Jira. Since Xray's Tests are Jira issues, we follow the same approach so your users work in the same manner they already do.

Considering that Test issues are like reusable test case templates, should you assign Test issues to a given version? If so, what would be the purpose of it?

The test specification has a lifecycle. Although your tests may live across many versions of your project, they're made in the context of some version, most probably for the version of the requirement they aim to validate. Thus, assigning them to a version serves the purpose of tracking the specification+implementation of the test case, the first time they have been used. This way, you can also track it in the Release as one artifact that needs to be "done" (i.e., specified/reviewed/approved).

But you may want to know the "version" (let's call it revision to avoid misunderstanding) of the Test that you ran in some iteration. Xray does not track revisions of Tests as numbers, in order to avoid unnecessary complexity. It tracks changes and keeps a version of the Test specification within each Test Run.

Xray addresses versioning the Jira way, which we think is more elegant, while at the same time, ensures data consistency:

- Whenever you schedule a Test for execution in a Test Execution, Xray creates an instance of the Test that we call Test Run. That Test Run contains the "version"/revision of the Test specification at that moment.
- You can go to an already recorded Test Run and reset the specification to the last/actual specification on the Test issue. Or, you can merge it, which means that only changed steps will be updated and the other step results are not changed. More info can be found in the [Execute Tests](#) page.
- Normally, you work with the last "version"/revision of the Test (as you do for a user story, etc.). However, if you really want to manage different versions of Tests at same time (assuming that they're actually different), you can create different Tests, one per version.

By persisting the Test specification on the Test Runs, Xray ensures data consistency. The "version" of the Test that was executed in a Test Run, is persisted in the Test Run itself. That means that if you change a Test specification today, that won't affect your already recorded runs for that Test (unless you want to).



### Learn more

Please have a look at [Test Runs](#) for more information on the fields that are persisted at the Test Run level.