Project Organization

Because Xray uses issue types for mapping test concepts, you can organize your projects to best suit your organizational needs. You can keep your current projects separate from Xray, or combine them together.

Xray is flexible enough to deal with scenarios where, for security or compliance reasons, you have to manage Jira entities on different projects.

Project Organization Use Cases

Below are some common use cases showing how to use Xray in combination with other projects in Jira. Xray issue types support associations from different projects. You can add Tests from one project to Tests Executions on another project, or even mix Tests from different Projects in the same Test Execution or Test Set issue.

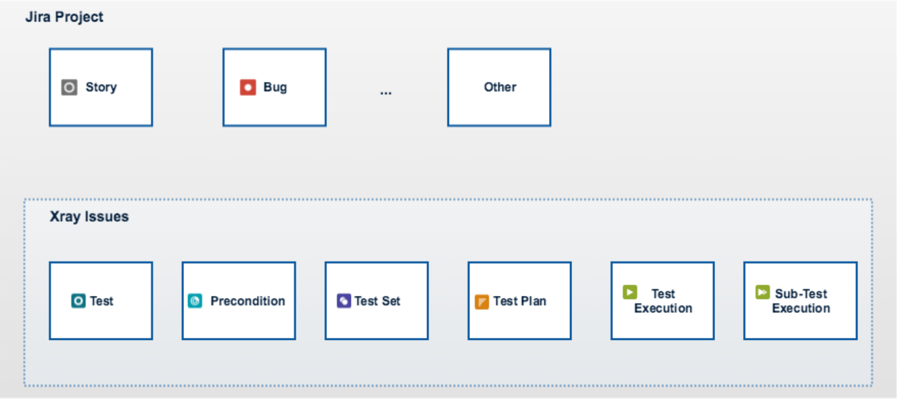
All-in-one

You can use a single project to manage your Requirements and Defects, Test-related issues as well as all your Test Executions.

Recommendation

This is the preferred organization approach as it provides some key benefits over other alternatives:

- **Self-contained**
- Easy to understand and manage
- Promotes team collaboration



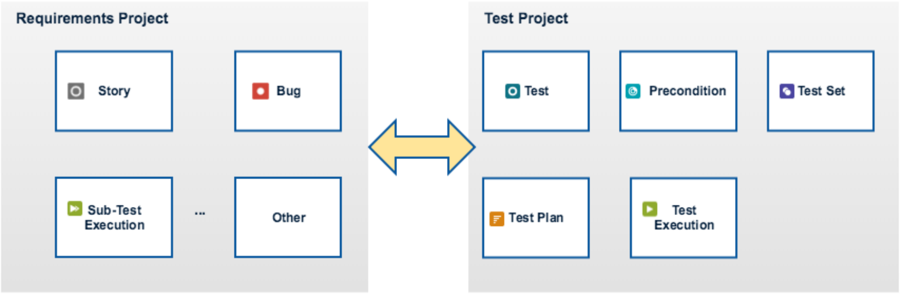
How to set it up

1. Create a new Scrum project or use an existing one.

2. Add Test, Precondition, Test Set, Test Plan, Test Execution and Sub-Test Execution issue types to the project. This can easily be done using a shortcut available in the **Summary** section within the project settings (see Project Settings: Summary) for editing the Issue Type Scheme used by the project. You can also be do it from within Jira administration > Settings > Issues > Issue Type Schemes.
3. In the project settings, within the **Test Coverage** section (see Project Settings: Test Coverage), define the Covered Issue Types (i.e., the testable entities such as the typical requirements, user stories, epics). On the left side, you can see all available Issue Types. Drag the ones you want (e.g., Story, Epic) to the **Covered Issue Types** and save the settings.

Separate my Requirements and Defects from Tests

If you already have a Jira project for managing Requirements and Defects and don't want to have any tests nor executions in this project, you can create a separate companion project just for testing purposes.



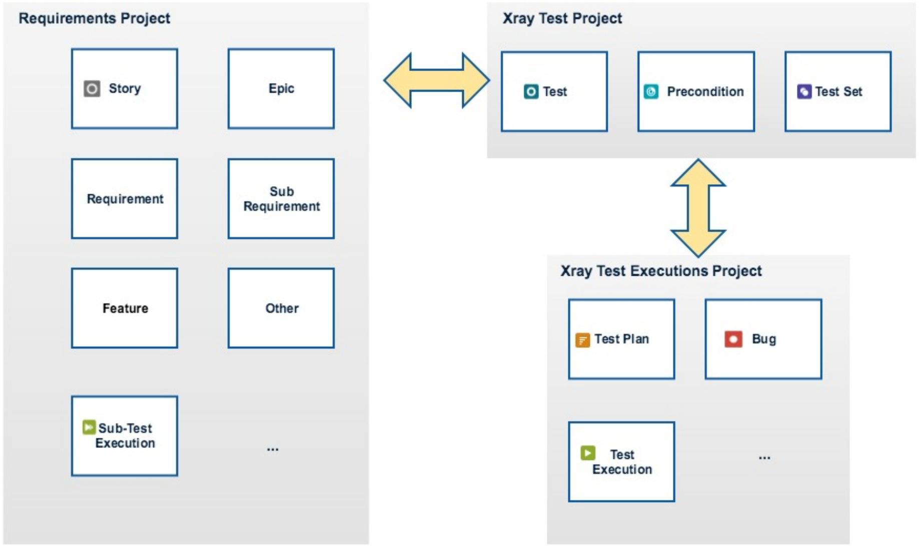
How to set it up

1. Create a Scrum project or use an existing one for managing the "Requirements Project".
2. Create another project (it can also be a Scrum project) for managing the "Test Project".
3. In the "Test Project", add Test, Precondition, Test Set, Test Plan, Test Execution and Sub-Test Execution issue types to the project. Remove all other issue types. This can easily be done using a shortcut available in the **Summary** section within the project settings (see Project Settings: Summary) for editing the Issue Type Scheme used by the project. You can also be do it from within Jira administration > Settings > Issues > Issue Type Schemes.
4. In the "Requirement Project" project settings, within the **Test Coverage** section (see Project Settings: Test Coverage), define the Covered Issue Types (i.e. the testable entities such as the typical requirements, user stories, epics). On the left side, you can see all available Issue Types. Drag the ones you want (e.g., Story, Epic) to the **Covered Issue Types** and save the settings.

Dedicated repository for Tests

Another common use case for Xray is to have a dedicated project for Test (and Precondition and Test Set) issues. Test Executions are the responsibility of other project(s). This separation allows you to manage permissions more effectively; maybe only a few testers have permission to write tests while others only have permission to execute.

Of course, this use case can also be combined with the previous one by having separate projects for managing Requirements and Defects.



Test Execution Versioning

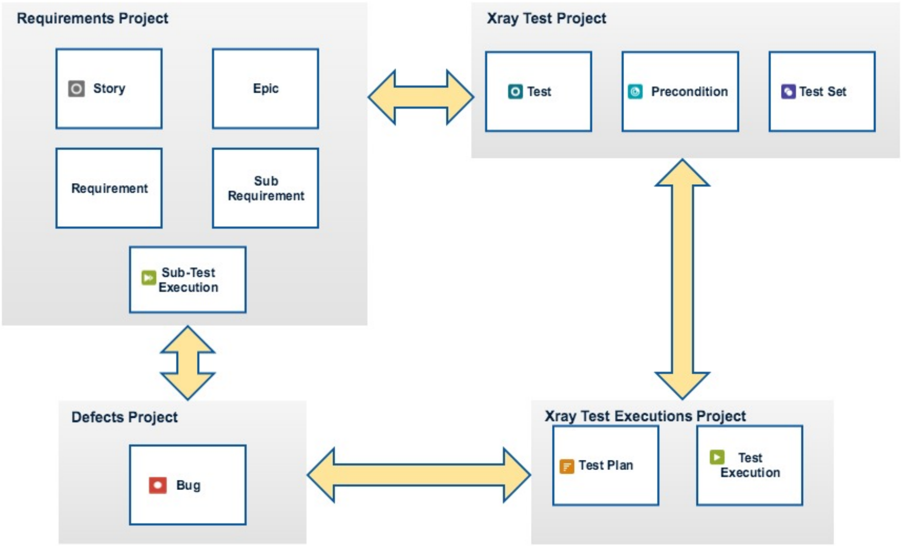
If you plan to have separate projects for managing Requirements/Defects and Tests Executions and you want to analyze Requirements by version, then your Requirement's project version names must match the names of the Test Execution's project.

How to set it up

1. Create a Scrum project or use an existing one for managing the "Requirements Project".
2. Create another project (it can also be a Scrum project) for managing the "Test Project"
3. Create another project (it can also be a Scrum project) for managing the "Test Executions Project".
4. In the "Test Project", add Test, Precondition and Test Set issue types to the project. Remove all other issue types. This can easily be done using a shortcut available in the **Summary** section within the project settings (see Project Settings: Summary) for editing the Issue Type Scheme used by the project. You can also be do it from within Jira administration > Settings > Issues > Issue Type Schemes.
5. In the "Test Executions Project", add Test Plan, Bug and Test Execution issue types to the project. Remove all other issue types.
6. In the "Requirement Project", add the Sub-Test Execution issue type.
7. In the "Requirement Project" project settings, within the **Test Coverage** section (see Project Settings: Test Coverage), define the Covered Issue Types (i.e., the testable entities such as the typical requirements, user stories, epics). On the left side, you can see all available Issue Types. Drag the ones you want (e.g., Story, Epic) to the **Covered Issue Types** and save the settings.

Completely separate

In this use case, the Tests, Requirements, Executions & Plans and Defects separated and handled in different projects.



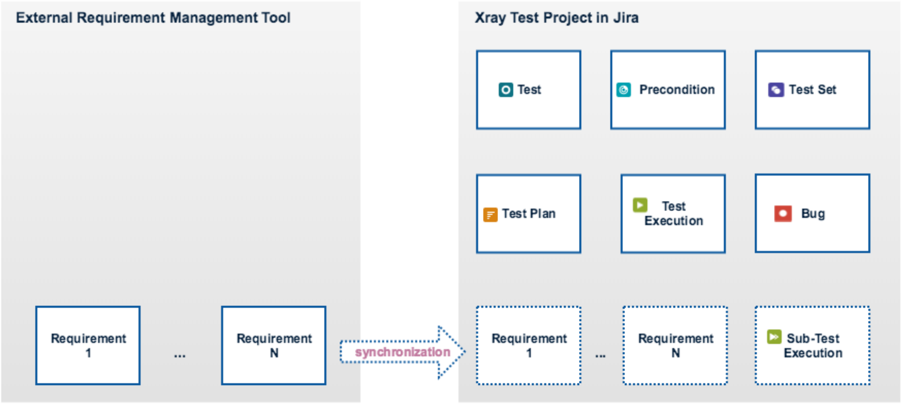
How to set it up

1. Create a Scrum project or use an existing one for managing the "Requirements Project".
2. Create another project (it can also be a Scrum project) for managing the "Test Project".
3. Create another project (it can also be a Scrum project) for managing the "Test Executions Project"
4. Create another project (it can also be a Scrum project), for managing the "Defects Project".
5. In the "Test Project", add Test, Precondition and Test Set issue types to the project. Remove all other issue types. This can easily be done using a shortcut available in the **Summary** section within the project settings (see Project Settings: Summary) for editing the Issue Type Scheme used by the project. You can also be do it from within Jira administration > Settings > Issues > Issue Type Schemes.
6. In the "Test Executions Project", add Test Plan and Test Execution issue types to the project. Remove all other issue types.
7. In the "Requirement Project", add the Sub-Test Execution issue type.
8. In the "Defects Project", add the Bug issue type.
9. In the "Test Executions Project", add Test Plan and Test Execution issue types to the project. Remove all other issue types.
10. In the "Requirement Project" project settings, within the **Test Coverage** section (see Project Settings: Test Coverage), define the Covered Issue Types (i.e., the testable entities such as the typical requirements, user stories, epics). On the left side, you can see all available Issue Types. Drag the ones you want (e.g., Story, Epic) to the **Covered Issue Types** and save the settings.

Isolated repository for Tests

In this more uncommon scenario, you have your requirements outside of Jira in some other tool. You use Jira only as a testing repository for creating Tests, Test Sets, Test Plans and Test Executions.

You certainly can use Xray for this, but you will lose the benefits of requirement coverage and traceability.



ⓘ Users implementing this scenario might gain huge benefits if they are able to synchronize the requirements in the other tool and Jira.

How to set it up

1. Create a Scrum project or use an existing one for managing the "Test Project". This one will be handled like a Requirements Project;
2. In the "Test Project", add Test, Precondition and Test Set issue types to the project. Remove all other issue types, except the issue type to be handled as a requirement and the Bug. This can easily be done using a shortcut available in the **Summary** section within the project settings (see Project Settings: Summary) for editing the Issue Type Scheme used by the project. You can also be do it from within Jira administration > Settings > Issues > Issue Type Schemes.
3. In the "Test Project" project settings, within the **Test Coverage** section (see Project Settings: Test Coverage), define the Covered Issue Types (i.e., the testable entities such as the typical requirements, user stories, epics). On the left side, you can see all available Issue Types. Drag the ones you want (e.g., Story, Epic) to the **Covered Issue Types** and save the settings.