Working in regulated environments and ensuring compliance using Xray

Overview

Organizations working in highly regulated markets (e.g., medical/health, pharmaceutical, security, automotive, aerospace, defense), usually have to comply with specific standards or some regulatory requirements that apply to them and the products they are responsible for.

As an example, ensuring safety is essential for organizations developing medical devices.

A strict control over the development process (testing included), release process, and its maintenance is required. Risk management is core to all activities.

In order to have a proper auditable process in place, that is compliant with high-demanding regulatory requirements, we need to:

- Ensure persistence of historical data & change tracking
- Ensue that historical data cannot be tampered
- Ensure that certain items can only be modified if people are allowed to do so
- Ensure it's possible to identify who is responsible for what
- Identify the relationship between entities
- Ensure testing has been performed and that acceptance criteria has been covered
- Proof of the exact testing/checks that were performed
- Ensure diagnosis is facilitated and can be easily performed
- Ensure effective risk management is in place

Ways to ensure regulatory and compliance needs

Process

Overall, the process must be adjustable, controllable, documentable, clear, and tasks must be repeatable and auditable. Diagnosis should also be facilitated, to be able to understand events. On parallel, we must be able to properly manage risks.

Among others, we need to:

- implement a tailored, auditable, and controlled process (e.g., by configuring the project orgazation, by defining the semantics of entities, by enforcing rules & workflows for testing entities)
- support planning, including at testing level (i.e., using Test Plans)
- ensure tasks, including the ones derived from testing, are complete (i.e., by using tailored Xray issues)
- verify and analyze software requirements from multiple perspectives
- support unit, integration and system testing (i.e., by supporting diverse test automation frameworks alongside manual and exploratory testing)
- enable regression testing (i.e., by having tailored Test Sets or Test Plans for that)
- ensure software verification is complete (i.e., using coverage information)
- document released versions, including testing effort (i.e., by generating documents containing all the relevant data, including testing evidence)

- facilitate software problem resolution processes (i.e., by using reports such as the Traceability one)

Implement your own-defined project organization, with fine control over project entities

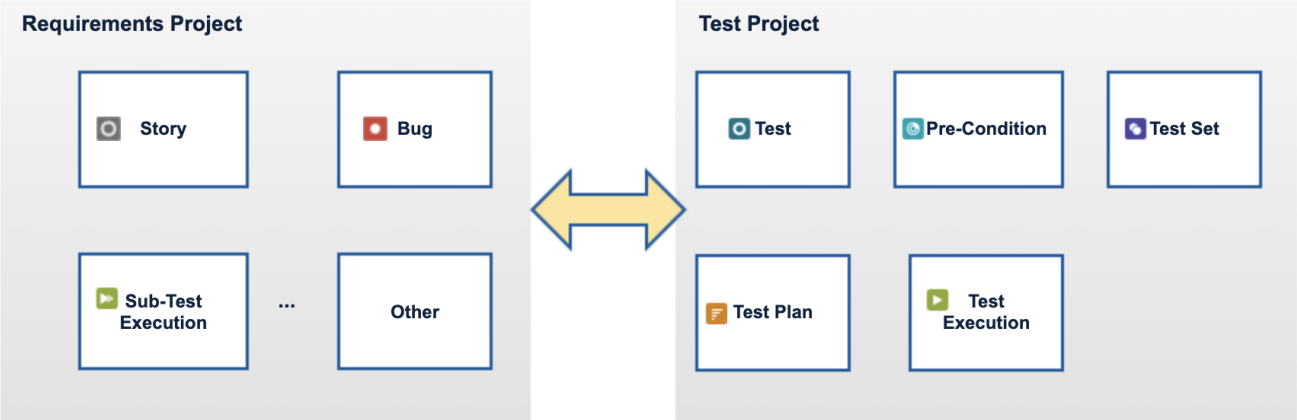
Xray supports different ways for organizing your project related items. This enables teams to adapt usage of the tool with the way they need or to the way they are organized functionally. But more than that: it enables teams to enforce or apply different rules to different entities.

The common scenario is to have an all-in-one approach where standard issue types (e.g. Story, Bug, Task) live alongside testing issue types (e.g. Test, Test Execution, Test Plan).

However, some teams prefer to have testing entities on a separate Jira project to have finer control over the process. Other teams prefer to have defects on a distinct project, or even on a remote Jira instance for security/business reasons.

All these possible project organization scenarios are possible and allow teams to apply more restricted rules on the related entities.

In the following scenario, two Jira projects are used: one mainly for the standard development/coding tasks and another one for testing purpose. This one of multiple configuration possibilities you can implement.



Even though you can be compliant with an all-in-one approach, Xray provides you the flexibility to even better suit your needs.

Learn more:

- Project Organization Use Cases

Implement fine Access control and Permissions

Organizations working in regulated environments need to be able to control the access and the permissions on project artefacts, including the ones derived from testing activities.

Access and permissions for Jira issue based entities (i.e. Test, Pre-Condition, Test Set, Test Plan, Test Execution, Sub-Test Execution) are handled in the same way as for any other issue type in Jira.

It's also possible to restrict execution of Test Runs to its assignee, so that only the assignee can update its contents.

Learn more:

- Advantages of using Jira issues
- Configuration to restrict execution of tests to the Test Run's assignee

Use custom defects with their own semantics and decide where to store them

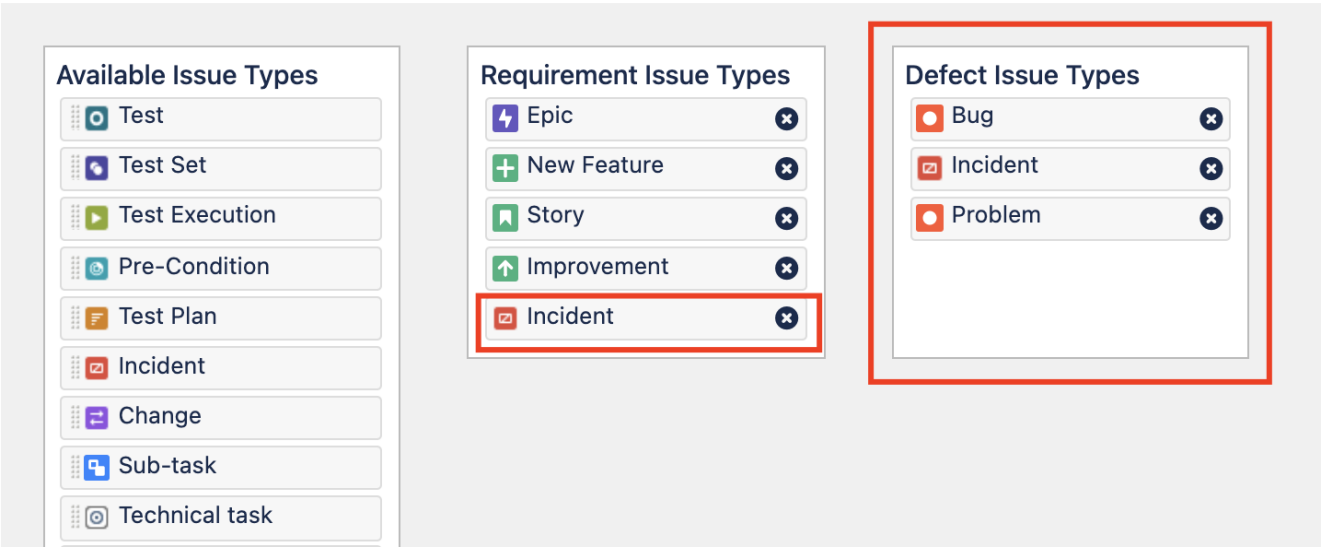
Semantically, and to make it clear and be able to apply different rules, it can make sense to have different issue types than the traditional "Bug."

In Xray, it's possible to define the issue types to be handled as "defects." Therefore, whenever reporting a bug, the team can use the Bug issue type or any other custom issue type where they need.

Defects can also be covered explicitly with tests (if configured as "requirements"/coverable issue types), for explicit verification of the underlying bug fixes.

Issue Type Mapping

Requirements and Defects are two concepts commonly related to Tests. A Requirement represents a singular documented fault in a computer program or system that produces an incorrect or unexpected result, or causes it to behave in unintend and a Defect.



Defects can be reported on the same project alongside other project issues, or can be reported on a separate Jira project. Defects can also be reported on a separate but connected Jira instance, if you wish to do so.

Learn more:

- Issue Type Mapping
- Requirements and Defects

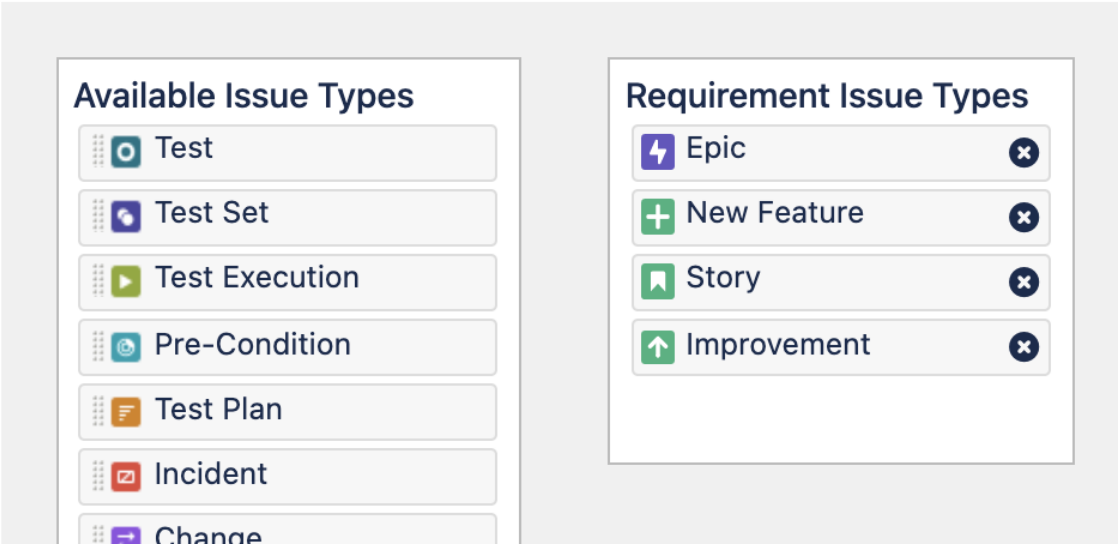
Define what is a requirement and track them at multiple levels

Projects usually have multiple levels of requirements (e.g. Epic>Story). Some broad/complex needs may be addressed in smaller requirements. Value can therefore be decomposed in these different layers and it's important to track it and these layers.

In Xray, a "requirement" is any issue that can be covered with tests, no matter their type. The team can define which issue types can be handled as such, even their own custom issue types (e.g., "Feature").

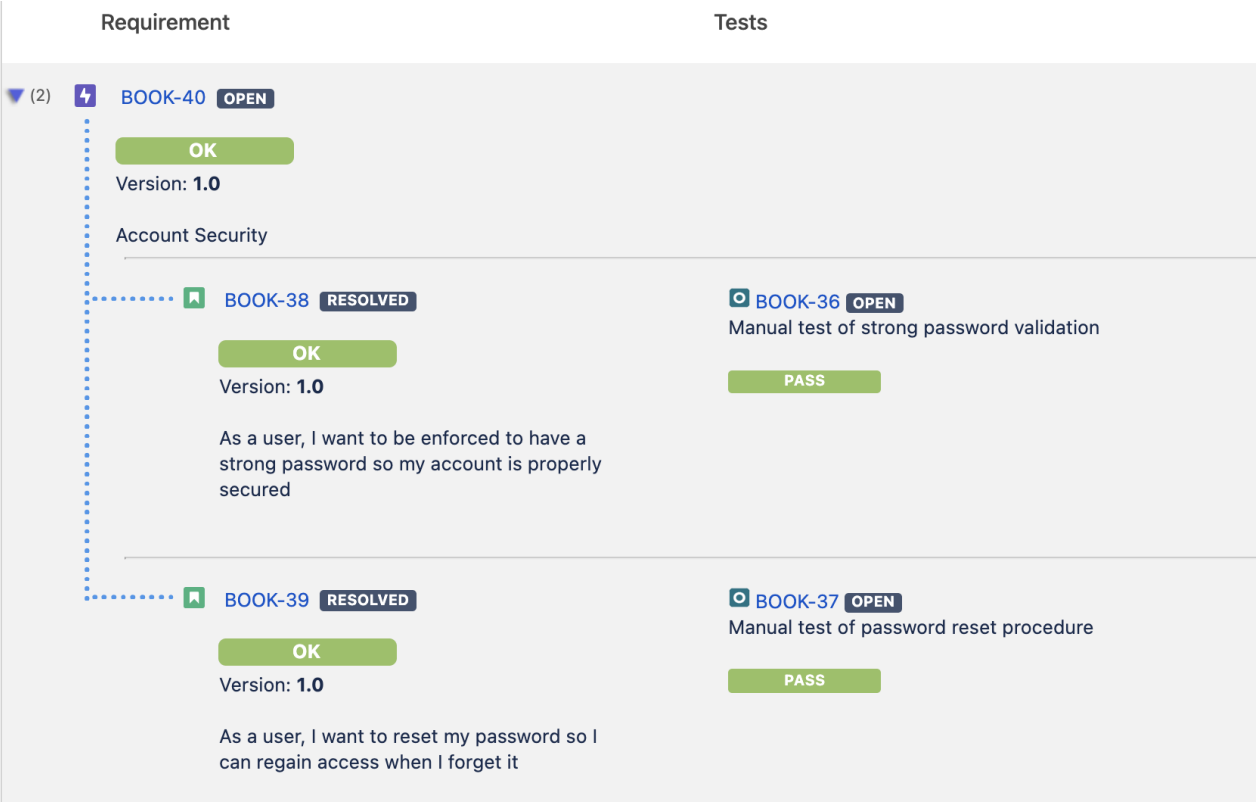
Issue Type Mapping

Requirements and Defects are two concepts commonly related to Tests. A Requ
fault in a computer program or system that produces an incorrect or unexpecte
and a Defect.



Xray provides multiple levels of requirements. It also provides different ways to define this hierarchical relationship (e.g., using issue links, or sub-tasks)

Epic and Story issues, usual in Agile environments, can be covered directly by testing. Coverage made on the Story issues can automatically be tracked on the related Epic.



Learn more:

- Issue Type Mapping
- Requirements and Defects

Track the relevant requirements across multiple versions and/or environments

Requirements exist beyond the scope of a specific version. Besides, a specific requirement may be used in different contexts (e.g., browsers, mobile devices).

Therefore we need to be able to track the status of requirements throughout time/versions and also on the different scenarios where they will be used. This (coverage) status is based on the integrated testing results, include the ones from test automation.

We may want to perform this analysis at high-level (i.e., project), or for a/some specific requirement(s).

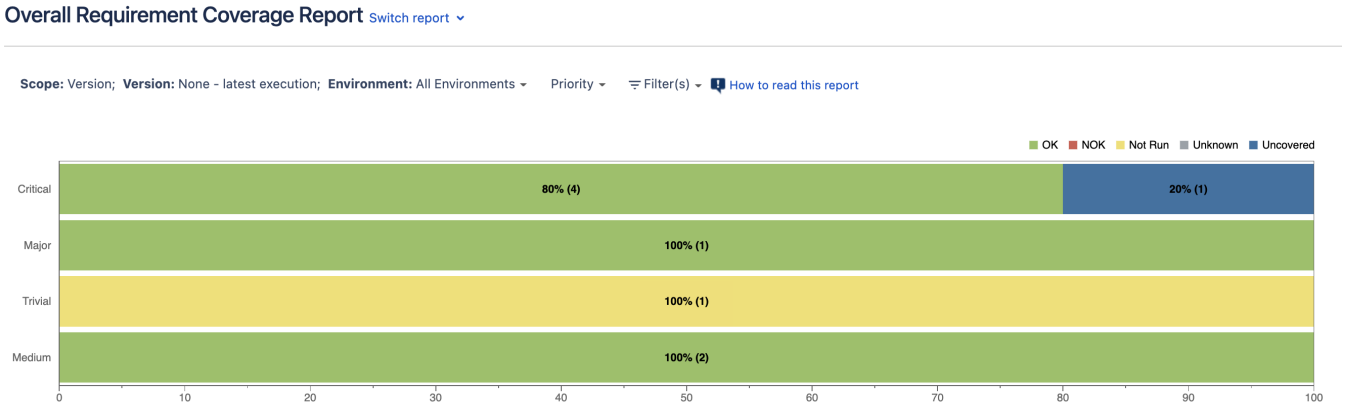
Xray allows you to answer questions such as:

- Are the requirements covered by any tests whatsoever?
- What were the requirements tested for these version? And what's their status based on the related testing? What's their completeness?
- How are the requirements implemented in an earlier version, on this new/other version?

The coverage status for requirements in Xray is multidimensional; in other words, we can analyze requirements from multiple perspectives: the latest testing results for version, or on a given environment, or just considering the results performed in the scope of a Test Plan.

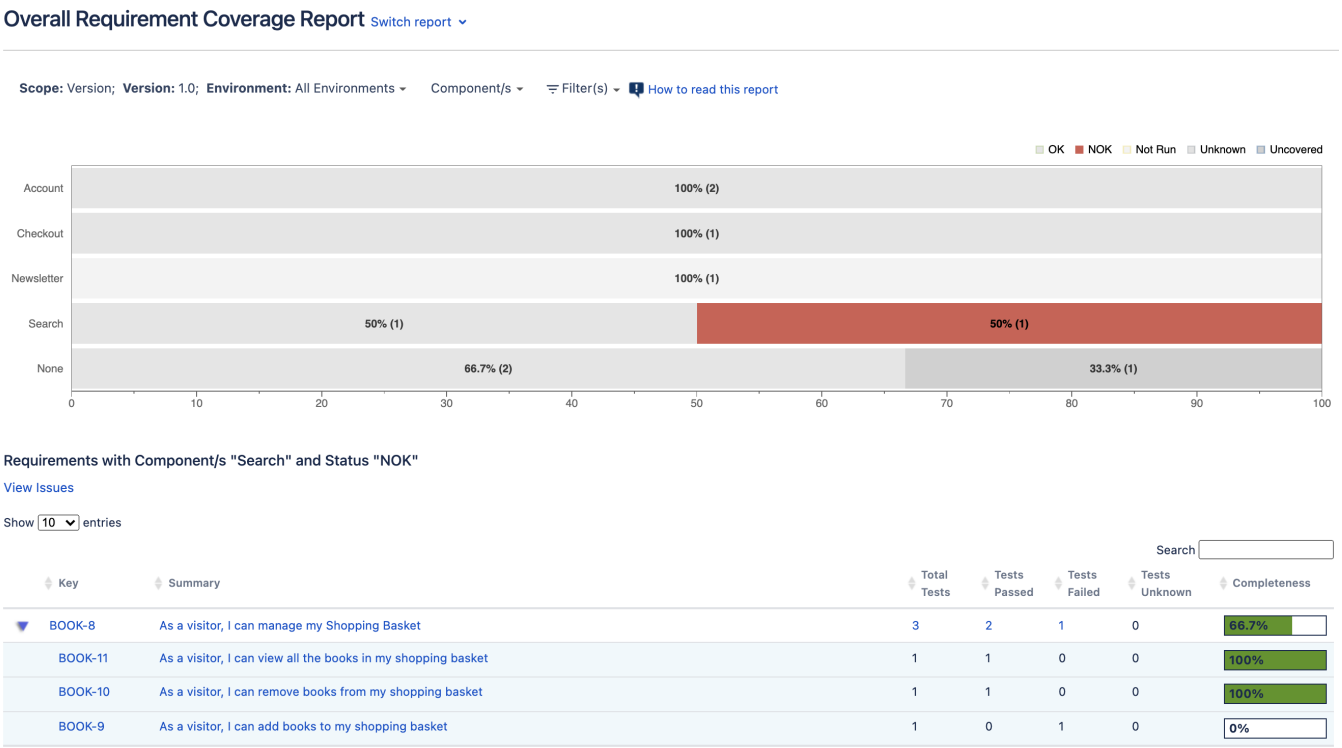
Let's see some examples.

Accordingly with the latest reported results, all medium and major requirements are OK. However, we can see that one requirement is not covered. We can then drill-down if we wish, by clicking on the chart bar.



On version 1.0, 50% of the requirements related to the "Search" component are having problems. We can see that the epic BOOK-8 is 66,7% complete, because one of the related stories (BOOK-9) has 1 test currently failing on version 1.0.

Some related stories (BOOK-10, BOOK-11) are OK though.



Learn more:

- Coverage Analysis
- Overall Requirement Coverage Report

Implement Workflows on testing entities to have explicit control over the process

Teams can implement workflows on all Jira issue based entities (i.e. Test, Pre-Condition, Test Set, Test Plan, Test Execution, Sub-Test Execution).

This will allow them to implement rigorous processes, including on testing entities, leveraged on the flexibility of Jira workflows.

Additionally, Xray provides the ability to further restrict usage of some these entities.

Possible use cases:

- have workflow status to review test specifications
- implement an approval mechanism, having one or more approvers
- make items "read-only" when transitioned to a certain workflow status by setting a Jira property ("jira.issue.editable")
- restrict usage of Tests in a certain workflow status (see how)
- disallow executions for Test Executions in a specific status (see how)
- reopen/review Tests upon changes on related requirements (example)
- enforce that requirements cannot transition to a given status, unless they are covered or even that those tests are successful (example)

Learn more in:

- Using Jira workflows for testing purpose

Sign using e-signatures to track explicit reviewal/approval and ensure data integrity

Electronic/digital signatures are one of the common mechanisms employed to ensure that document, for example, has been reviewed by one or more people and that no changes happened on the signed document meanwhile.

Xray issue based entities (i.e. Test, Pre-Condition, Test Set, Test Plan, Test Execution, Sub-Test Execution) can all be digitally signed, using one of many available Jira apps for that purpose. This is something unique, as it provides full control over the core testing activities; not only Tests can be digitally signed, also Test Sets, or even Test Plans.

Learn more:

- available Jira apps for e-signatures

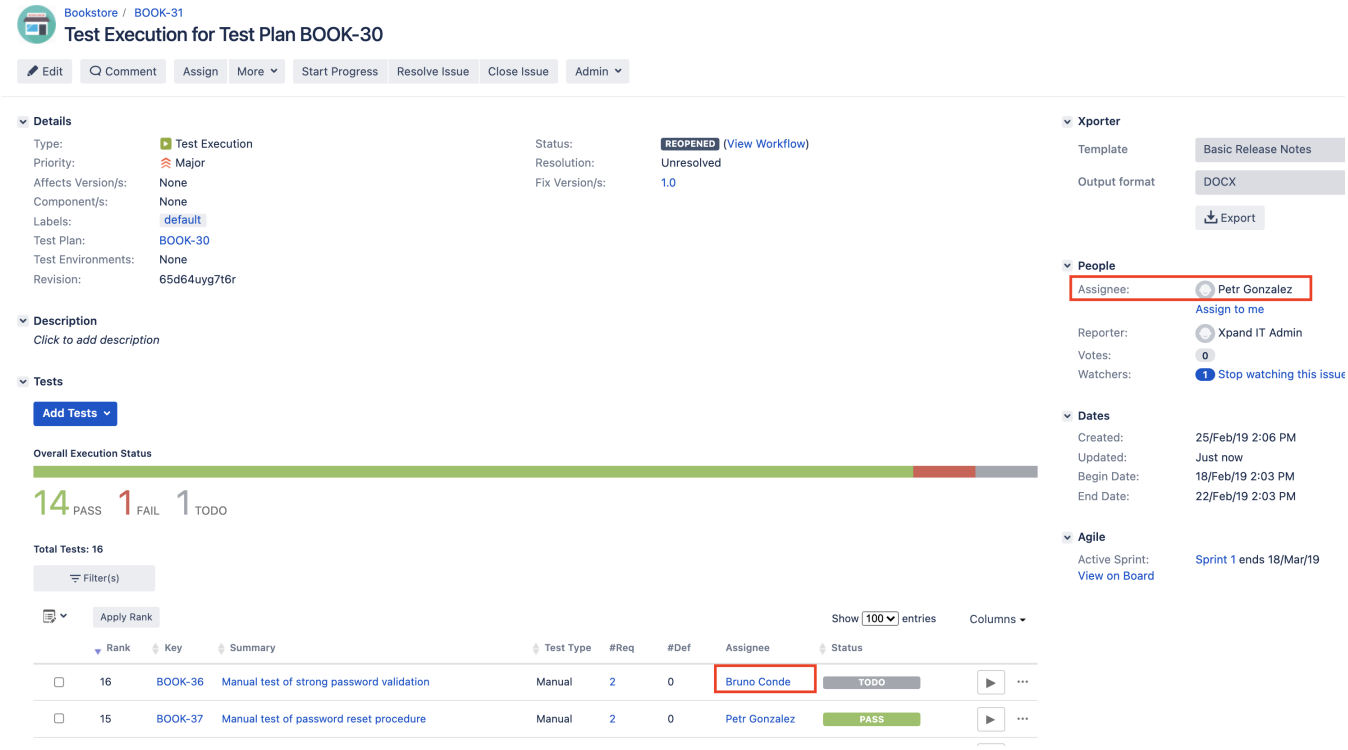
Facilitate reviewing, also during test execution

Review during or after test execution can be implemented.

In Xray, similarly to other issues, Test Execution issues have an assignee. We can use that as the main responsible for the Test Execution, perhaps for defining the list of tests to be run and then for reviewing its results.

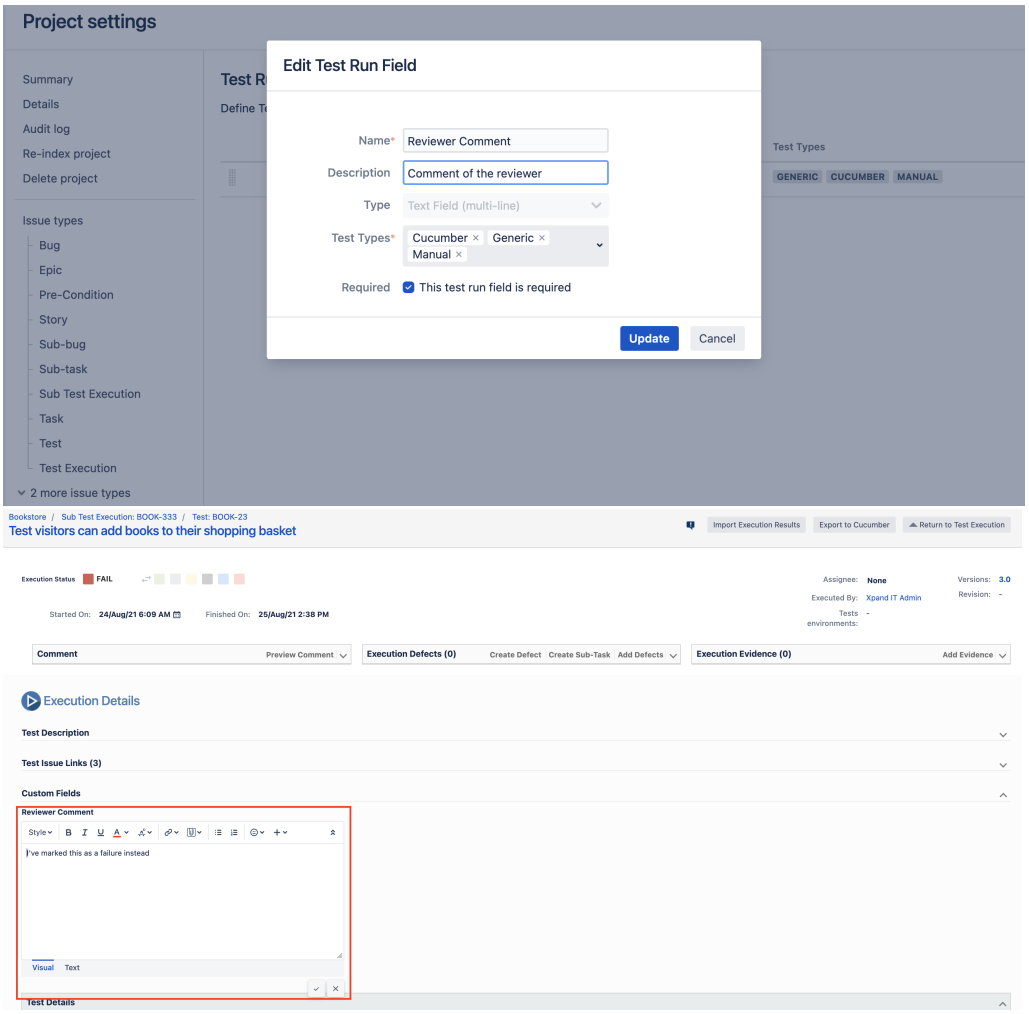
The Tests to be run on Test Executions (i.e., the Test Runs) can have their own specific assignee. Therefore, we can have a Test Execution responsible and the individual testers performing the identified testing.

This Test Execution as a whole is assigned to Petr Gonzalez but one of the Test Runs is assigned to another user (Bruno Conde). This can be used as means to have one person responsible for the Test Execution and reviewing the related reported results, while at the same time allowing for someone else to perform the actual testing.



This enables the implementation of a reviewal process, where Test Execution related results can be reviewed after they're reported; the Test Execution can be transitioned to a status (e.g., "PENDING_REVIEW") and then transitioned to another one (i.e., "CLOSED") when it's reviewed.

We can also create some specific Test Run custom field (e.g., "Reviewer Comment") and make it required for all runs, or just for the ones related to manual scripted test cases.



During execution of a test and its revision, we may find it useful to report a step as having an acceptable failure (e.g., "IRRELEVANT_FAIL"). We can then decide to mark the overall Test Run with a status (e.g., "FAIL_DISCARD") that won't mark the related requirement as being NOK, such as UNKNOWN or even OK if we want.

Test Statuses

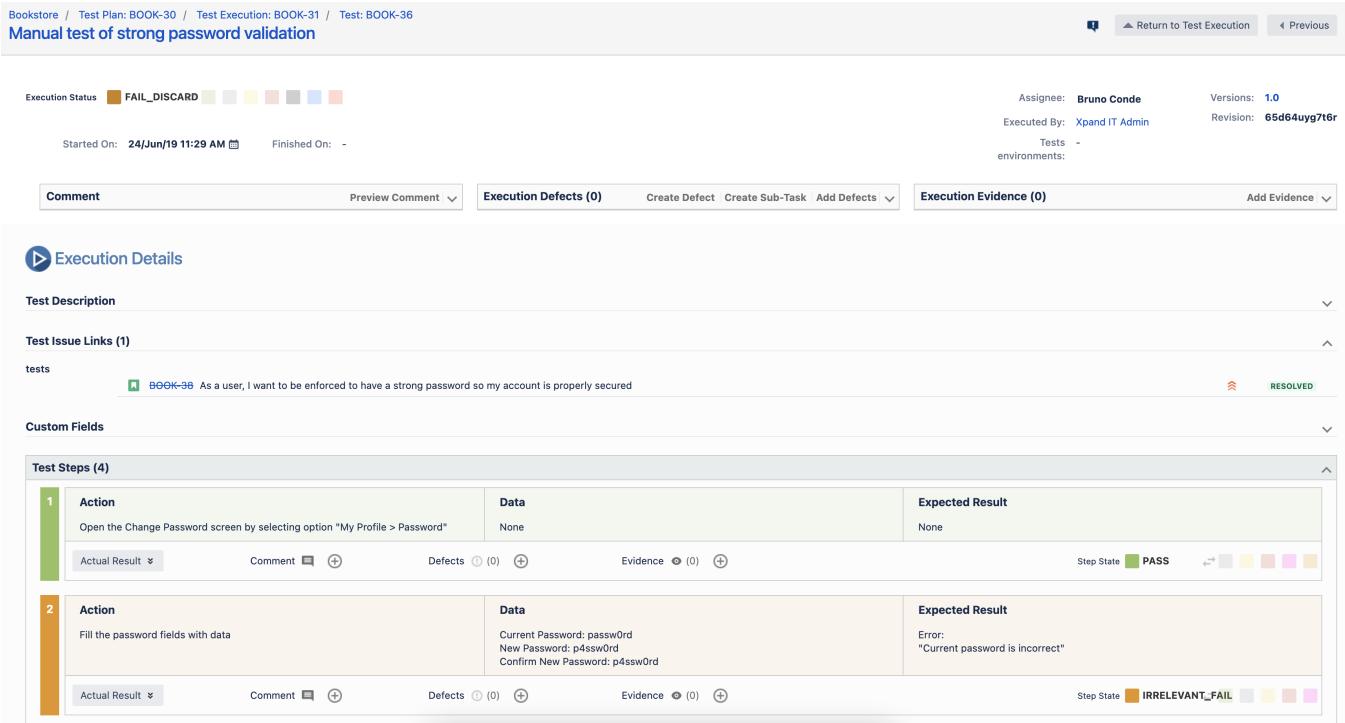
Manage custom Test Run statuses for Xray

Name	Description	Final	Color	Requirement Status
PASS	The test run has passed	☑		OK
TODO	The test run has not started	☐		NOTRUN
EXECUTING	The test run is currently being executed	☐		NOTRUN
FAIL	The test run has failed	☑		NOK
ABORTED	The test run was aborted	☑		NOTRUN
BLOCKED		☐		NOTRUN
PENDING		☑		UNKNOWN
FAIL_DISCARD	There was a minor issue but nothing relevant	☐		UNKNOWN

Test Step Statuses

Manage custom Test Step statuses for Xray

Name	Description	Color	Test Status
PASS	The test step has passed		PASS
TODO	The test step has not started		TODO
EXECUTING	The test step is currently being executed		EXECUTING
FAIL	The test step has failed		FAIL
SKIP	The test step has been skipped.		PASS
IRRELEVANT_FAIL	There was a minor issue with this step but nothing relevant		FAIL_DISCARD



Use detailed, trackable and yet flexible Test Planning

Even though plans can change (especially in an Agile context), being able to define a plan, no matter its level of detail, is something usually required.

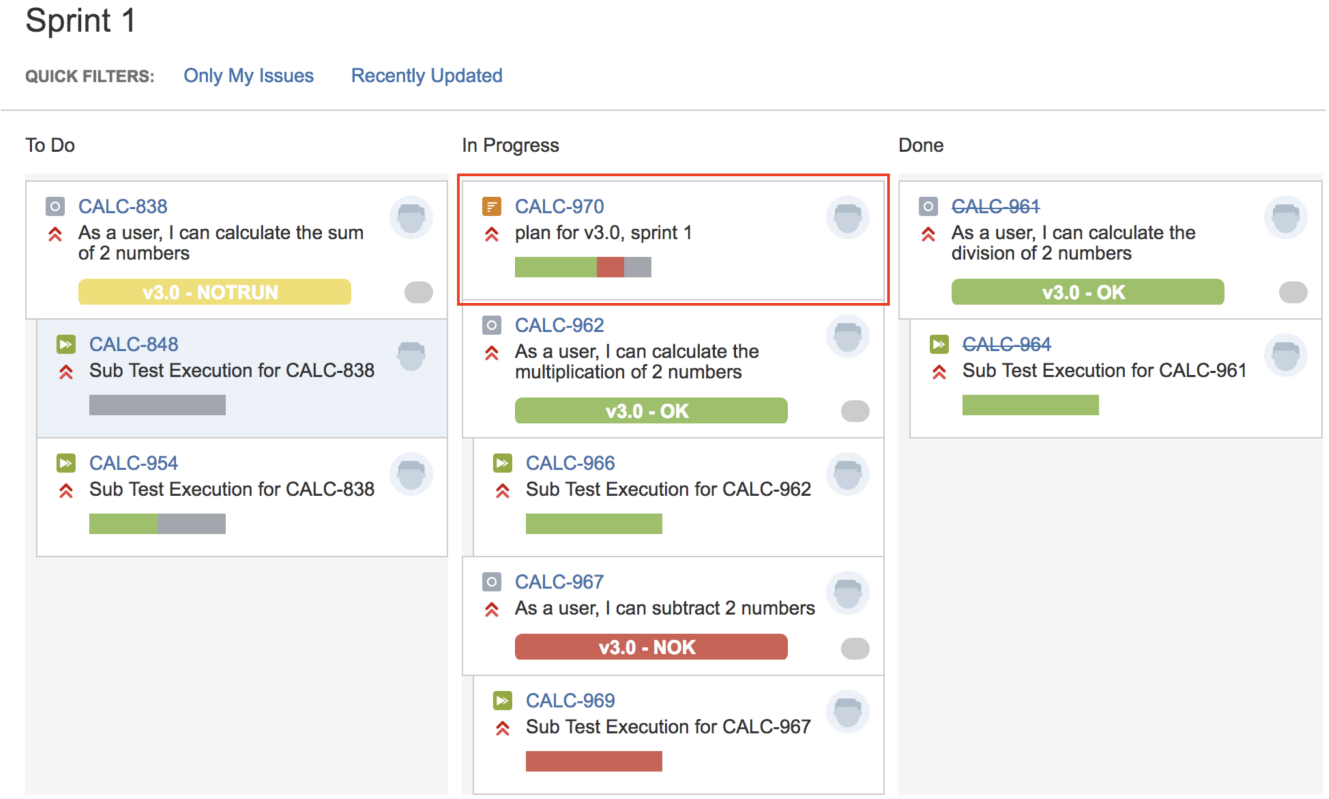
Some testing activities, including verification, can be planned upfront.

Xray provides an entity specifically for this purpose: the Test Plan. The Test Plan can be used more strategically, to define and track and assign testing activities to be performed.

In Xray, teams can have one or more Test Plans to track different testing efforts, for example related with different types of tests, or different risks, or even explicitly for regression testing.

The Test Plan can be as detailed as needed and documents can be added and linked to it. A Test Plan is also assigned to a user; it can be prioritized, labelled, and be explicitly one work item of the Sprint or of the Kanban board.

In the following example, we can see the Test Plan being tracked as an item and handled as part of the Sprint, similarly to other on-going project issues.



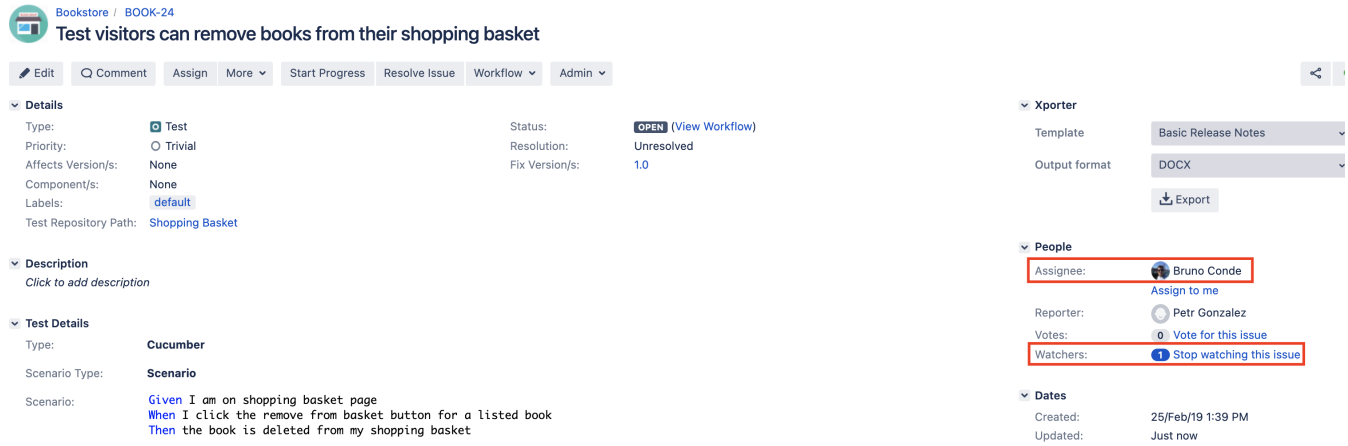
Learn more:

- Planning Tests
- Tips for planning tests
- Some examples of using Test Plans in Scrum context
- Agile Board Enhancements

Track testing tasks with full Assignment and Accountability

Xray issues (i.e. Test, Pre-Condition, Test Set, Test Plan, Test Execution, Sub-Test Execution) can be assigned and watchers can be added to them.

That means, for example, that whenever a Test is modified, you can get notified.



Ensure issues get done by assigning them to users and tracking them using workflows (more info ahead), no matter if you're dealing with the specification of a test case or the execution of a bunch of tests within a given Test Execution.

Individual Test Runs can also be assigned and it's possible to notify the assignee using traditional email notifications, just like any other issue in Jira.

Learn more:

- Email Notifications

Enable Customization to fine-tune the tool to your overall rigorous process

For defining a more rigorous process, teams need the ability to customize usage of the tool.

This includes, among others, the ability to:

- add custom fields on any testing entity, including all Jira issue based entities (i.e. Test, Pre-Condition, Test Set, Test Plan, Test Execution, Sub-Test Execution) and also on Test steps and on Test Runs
- implement additional customizations (e.g., using ScriptRunner, Automation for Jira, Cprime Power Apps)
- integrate with other Jira apps to widen the feature set provided by Xray itself

Learn more:

- Advantages of using Jira issues
- Integration with ScriptRunner
- Integration with Automation for Jira
- Integration with Cprime Power Scripts, Power Custom Fields, and Power Actions

Data

Record all meaningful testing evidence

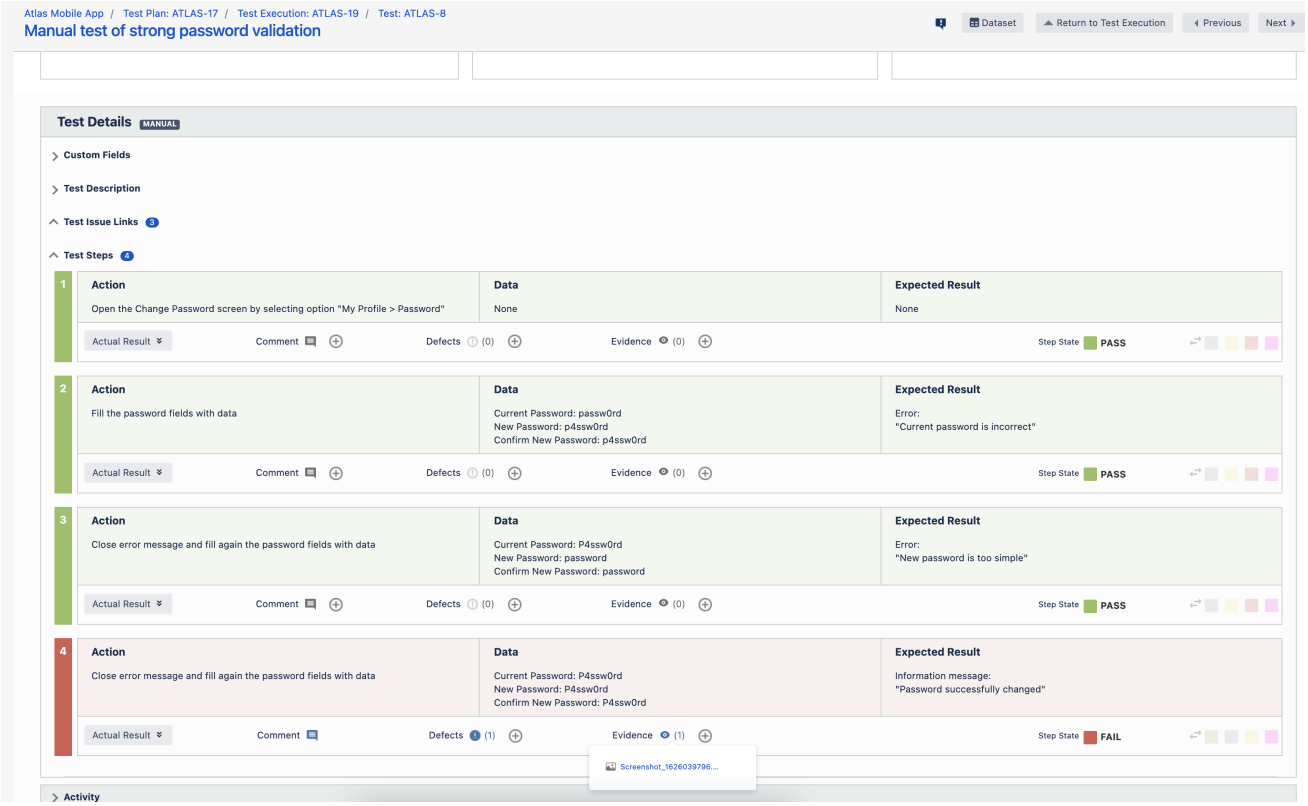
Teams can follow different approaches/styles for testing: scripted (i.e. test cases and "automated" test scripts) and exploratory.

No matter the testing approach, Xray can provide visibility of testing results, including evidence, in one place: Jira.

Xray allows you to identify:

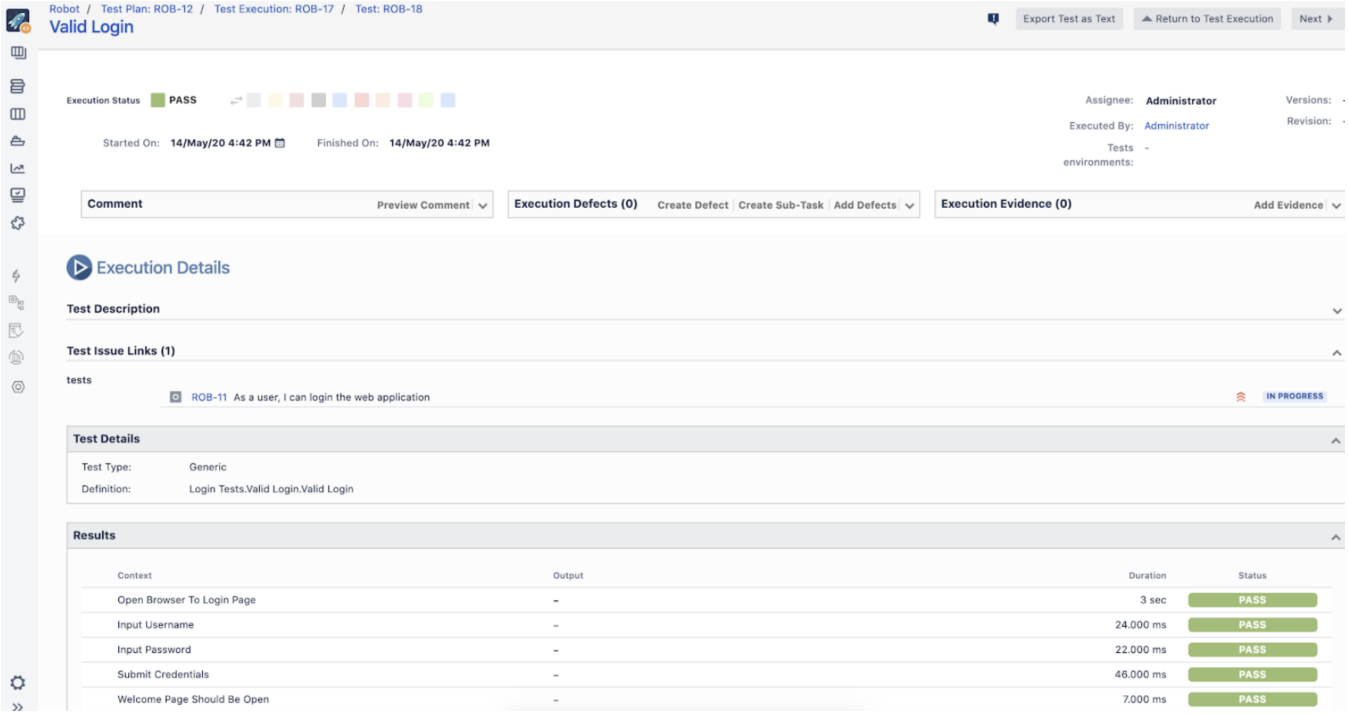
- version and revision of the SUT
- environment (e.g., browser, device)
- test result/status
- detailed results (e.g. steps)
- screenshots, videos or any other file-based evidence
- execution logs
- any reported comments

Whenever manually executing scripted test cases, we can report results at the step level, and attach some screenshot as evidence, for example:

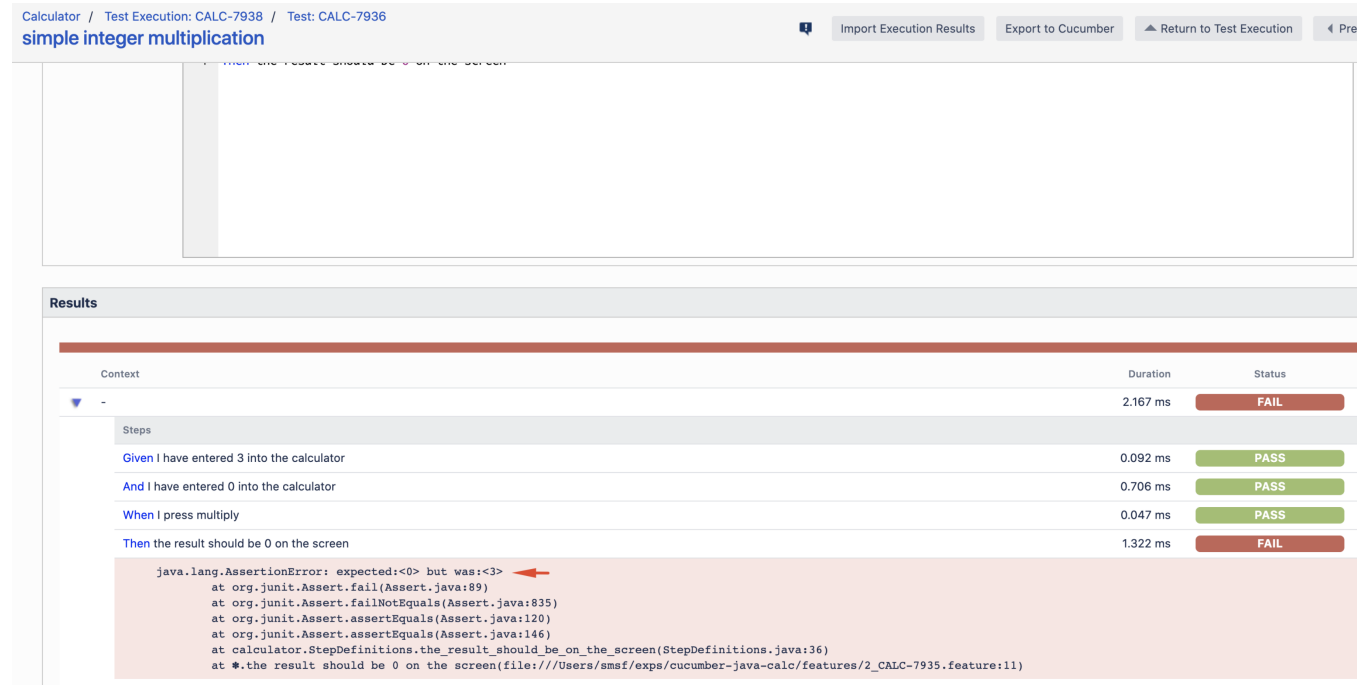


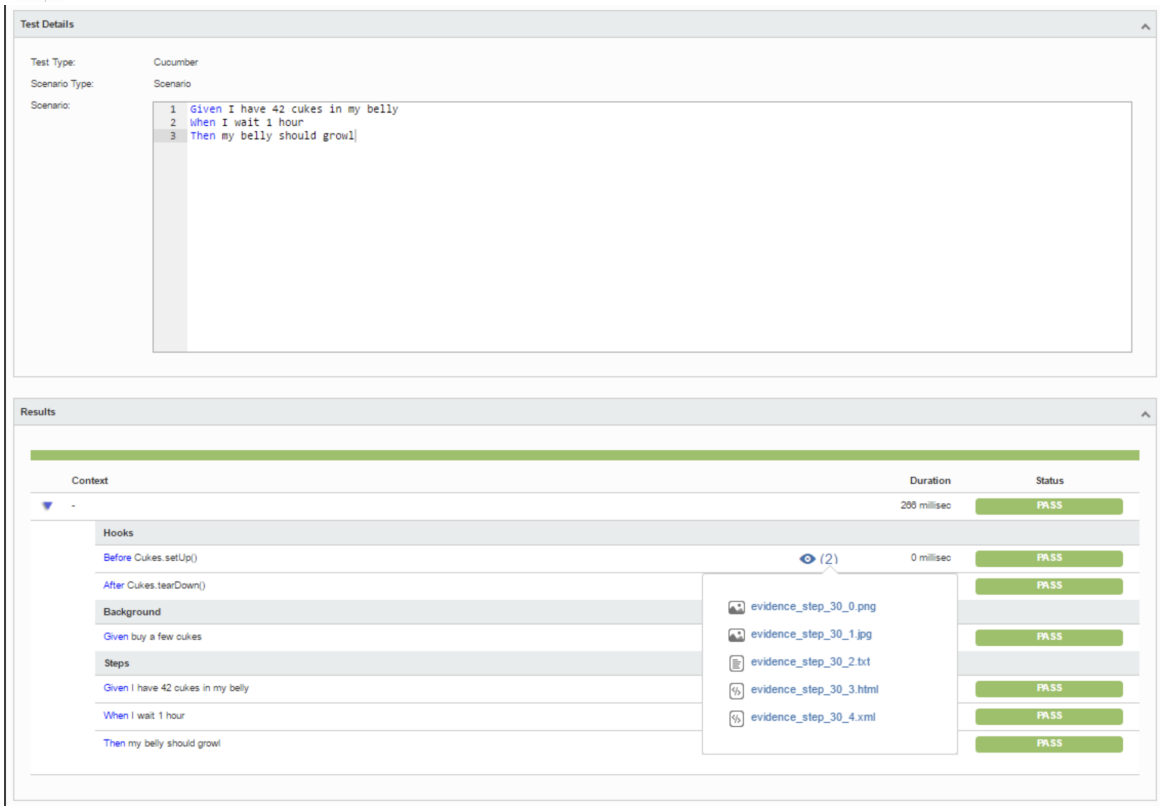
Detailed information from multiple test automation frameworks can also be imported to Xray, and tracked similarly to other tests.

This includes automation frameworks such as Robot Framework, where you can see the detailed keyword-level information.

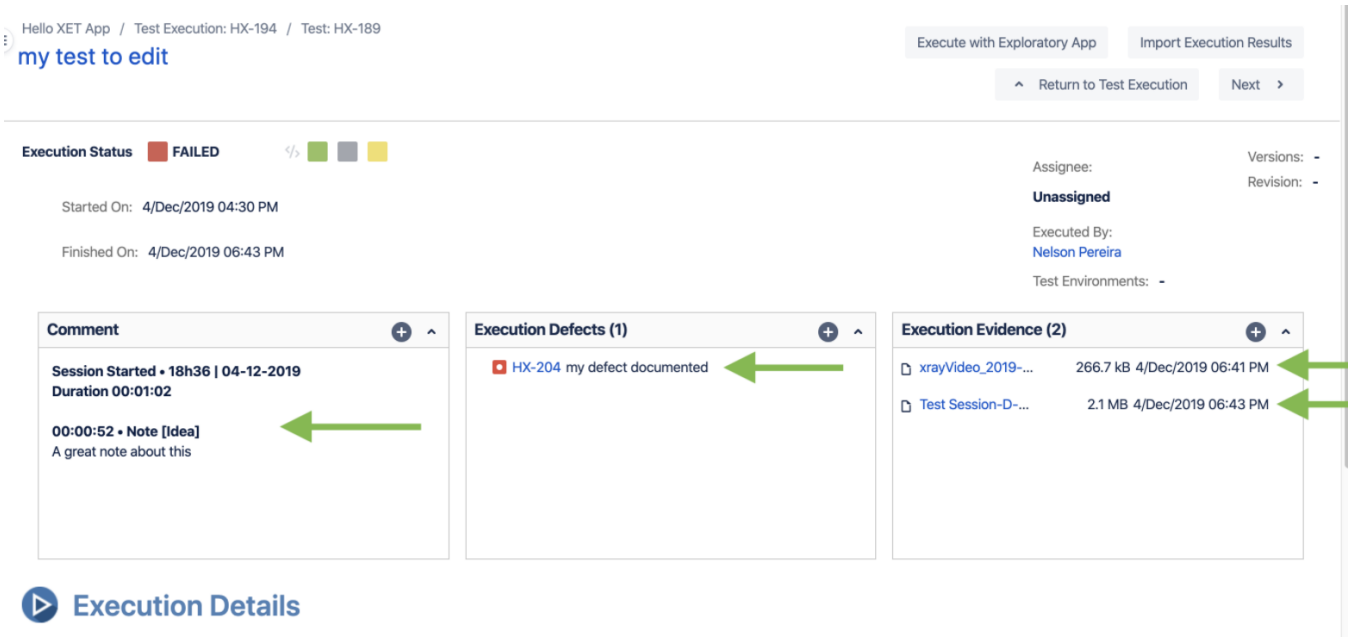


If adopting BDD frameworks, such as Cucumber, we can also see detailed information about the Gherkin statements and deep-dive into the automation logs or evidence (e.g., screenshots) taken by the automation.





Xray also supports exploratory testing using a desktop app (Xray Exploratory App) that can integrate with Xray to bring the best of both worlds: have exploratory testing evidence tracked in Jira, be accessible by the team, and reflect on the related requirements.



Learn more:

- Execute Tests
- Vast collection of test automation tutorials showcasing integrations with Xray
- Performing exploratory testing and integrating it with Xray

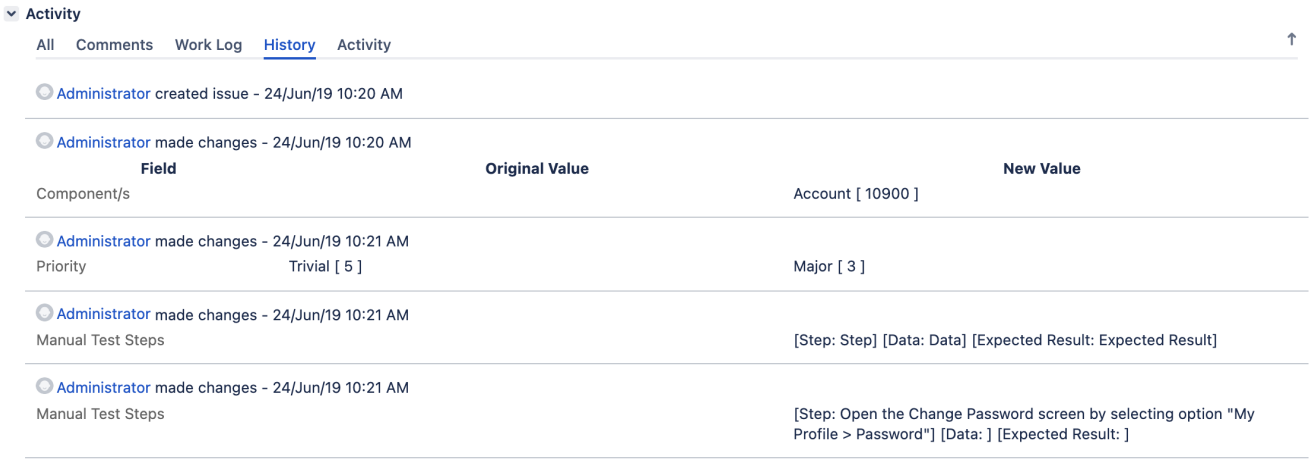
Ensure data persistence, history and change tracking

In order to enable auditing and facilitate diagnosis, data must be stored, and changes, whenever applicable, need to be clearly identified.

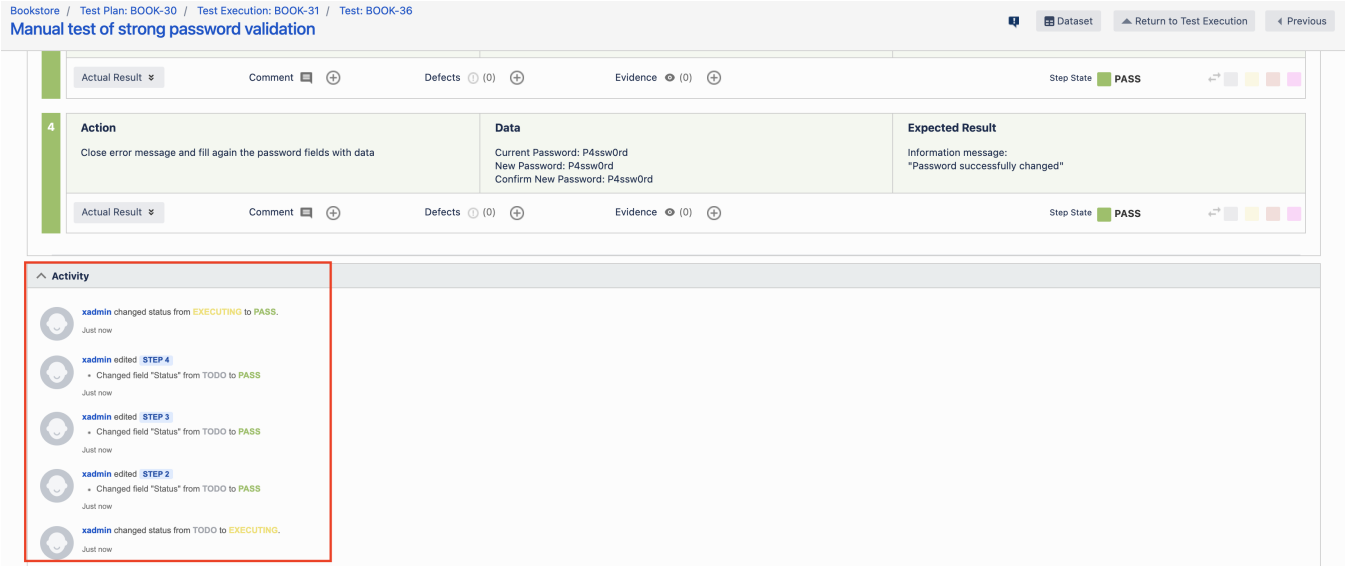
In Xray, all data is persisted and can be easily tracked using a historical timeline.

All changes on Jira issue based entities (i.e. Test, Pre-Condition, Test Set, Test Plan, Test Execution, Sub-Test Execution) are tracked similarly to other issue types in Jira.

One of those examples include Tests themselves.



Test Runs also have an activity section where its changes are tracked.



Learn more:

- Test Runs

Ensure no data tampering

Historical results (i.e. Test Run details) can't be modified. The same applies to past changes made on Jira issue based entities (i.e. Test, Pre-Condition, Test Set, Test Plan, Test Execution, Sub-Test Execution).

It is also possible to make Xray issue-based entities read-only. Besides, test specification cannot be modified during execution of the related test.

Export data in human-readible formats and automate data snapshots

All core entities, including Test Runs (i.e. results), can be exported to PDF, Word, or Excel documents, using fully customizable documents in terms of layout.

This is a way to obtain a formal, readable, copy of the relevant test data, no matter if it's related to test specification or execution.

Two options exist to achieve this: either using the built-in Document Generator capabilities, or by using the more complete and general purpose Xporter App.

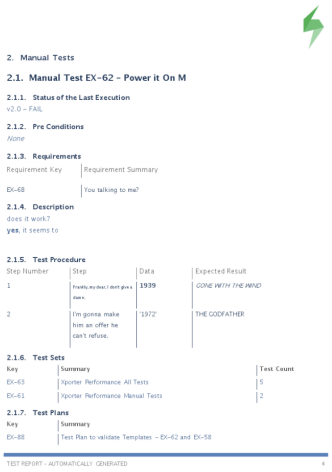
You can start by defining a template, or downloading and customizing an existing one from the template store, with all the data to be embedded, logos and other information (e.g., legal disclaimers)

Test Report

This template creates a test report from the selection of multiple issues, created using Xray for JIRA.

Download Template

Download Sample File



2. Manual Tests

2.1. Manual Test EX-62 - Power it On M

2.1.1. Status of the Last Execution
v2.0 - FAIL

2.1.2. Pre Conditions
None

2.1.3. Requirements

Requirement Key	Requirement Summary
EX-68	You talking to me?

2.1.4. Description
does it work?
yes, it seems to

2.1.5. Test Procedure

Step Number	Step	Data	Expected Result
1	Frankly, my dear, I don't give a damn.	1939	*GONE WITH THE WIND*
2	I'm gonna make him an offer he can't refuse.	'1972'	THE GODFATHER

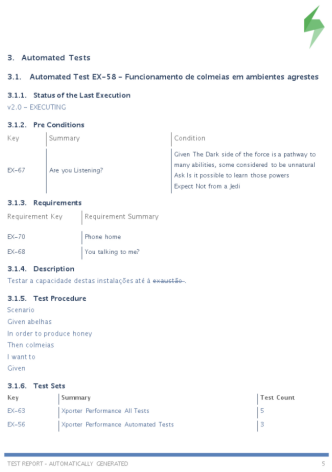
2.1.6. Test Sets

Key	Summary	Test Count
EX-63	Xporter Performance All Tests	5
EX-61	Xporter Performance Manual Tests	2

2.1.7. Test Plans

Key	Summary
EX-88	Test Plan to validate Templates - EX-62 and EX-58

TEST REPORT - AUTOMATICALLY GENERATED 4



3. Automated Tests

3.1. Automated Test EX-58 - Funcionamento de colmeias em ambientes agrestes

3.1.1. Status of the Last Execution
v2.0 - EXECUTING

3.1.2. Pre Conditions

Key	Summary	Condition
EX-67	Are you Listening?	Given The Dark side of the force is a pathway to many abilities, some considered to be unnatural Ask Is it possible to learn those powers Expect Not from a Jedi

3.1.3. Requirements

Requirement Key	Requirement Summary
EX-70	Phone home
EX-68	You talking to me?

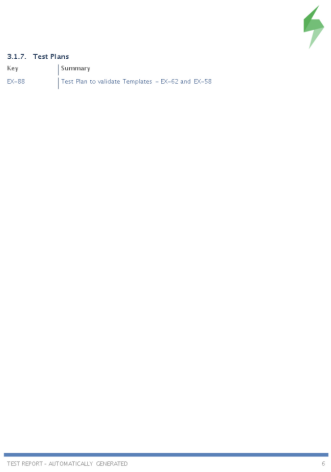
3.1.4. Description
Testar a capacidade destas instalações até à ~~exaustão~~.

3.1.5. Test Procedure
Scenario
Given abelhas
In order to produce honey
Then colmeias
I want to
Given

3.1.6. Test Sets

Key	Summary	Test Count
EX-63	Xporter Performance All Tests	5
EX-56	Xporter Performance Automated Tests	3

TEST REPORT - AUTOMATICALLY GENERATED 5



3.1.7. Test Plans

Key	Summary
EX-88	Test Plan to validate Templates - EX-62 and EX-58

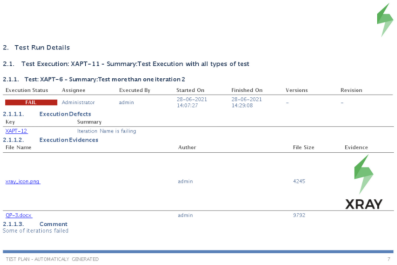
TEST REPORT - AUTOMATICALLY GENERATED 6

Test Plan Advanced with Cover Page (Xray 5)

This template provides a way to see a Test Plan with a little more detail. You can see the Test Run details, Test Steps details, attachments, evidences, and defects of normal and parameterized tests. This template is compatible with versions above 5.x.x

Download Template

Download Sample File



2. Test Run Details

2.1. Test Execution: XAPT-11 - Summary:Test Execution with all types of test

2.1.1. Test: XAPT-6 - Summary:Test more than one iteration 2

Execution Status	Assignee	Executed By	Started On	Finished On	Versions	Revision
FAIL	Administrator	admin	28-06-2021 14:07:27	28-06-2021 14:29:08	-	-

2.1.1.1. Execution Defects

Key	Summary
XAPT-12	Iteration Name is failing

2.1.1.2. Execution Evidences

File Name	Author	File Size	Evidence
xray_icon.png	admin	4245	
QP-5.docx	admin	9792	

2.1.1.3. Comment
Some of iterations failed

TEST PLAN - AUTOMATICALLY GENERATED 7

2.1.1.4. Test Description
Test more than one iteration

2.1.1.5. Test Issue Attachments
XAPT-6 has 0 Attachments.

2.1.1.6. Parameters
There are 6 parameters on this Test Run.

Key	Value
Browser	Google
Browser	Safari
Browser	
Browser	Firefox
Environment	Staging
Environment	Production

2.1.1.7. Iterations
This Test Run has 8 iterations

2.1.1.7.1. Iteration Overall Execution Status

List Of Statuses: PASS: 12.5%, FAIL: 25.0%, ABORTED: 25.0%, EXECUTING: 25.0%, TODO: 12.5%

TODO	12.5%
EXECUTING	25.0%
PASS	12.5%
FAIL	25.0%

TEST PLAN - AUTOMATICALLY GENERATED 8

ABORTED	25.0%

2.1.1.7.2. Test Run Iteration 1 details

Status	PASS
Total of Parameters	2
Parameters	Browser = Google,Environment = Staging

2.9.1.2.1. Iteration Pre-Condition XAPT-7 Definition
Manual Pre condition

2.1.1.7.3. Iteration 1 Parameters details

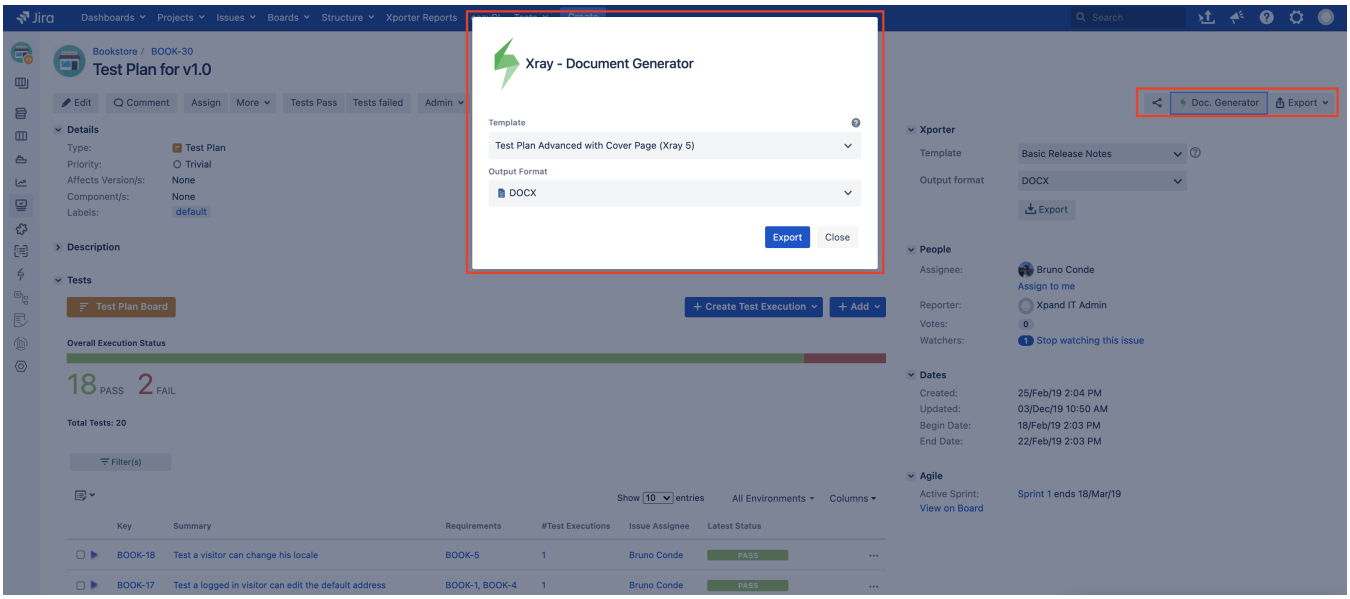
Key	Value
Browser	Google
Environment	Staging

2.1.1.7.4. Iteration 1 Test Steps details
This Manual Test has 3 Steps

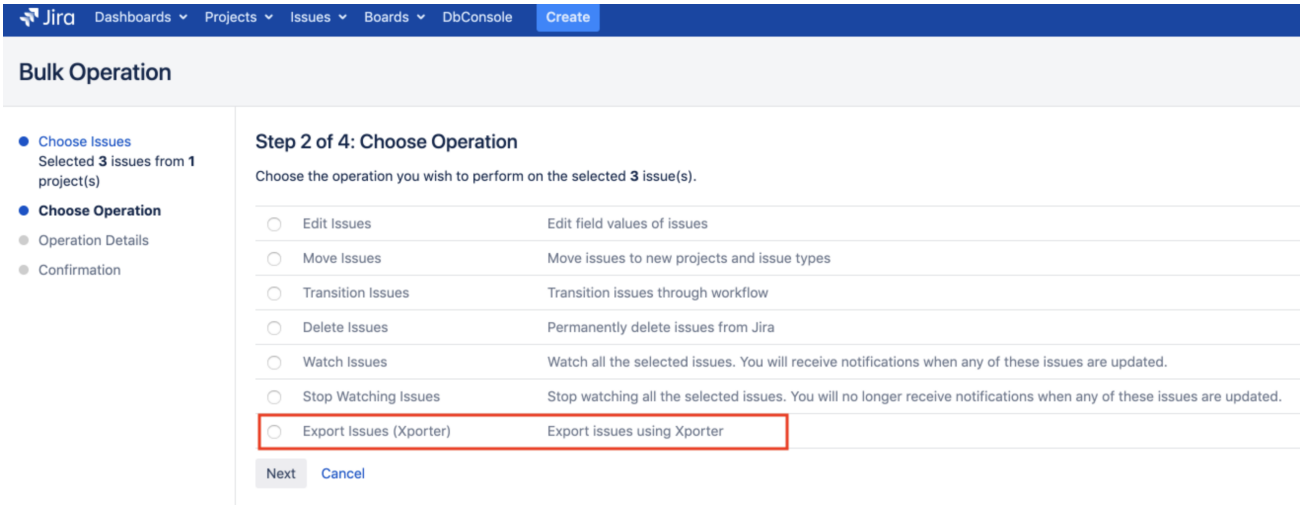
	Action	Data	Expected Result	Attachments	Comment	Defects	Evidences	Status
1	Open menu Document Generator		Menu opened with success					PASS
2	Install template from store	template name = Offline Test Report	Template installed with success					PASS
3	Upload a new template manually		Template added with success	Test Run - Parameterized Tests.docx				PASS

TEST PLAN - AUTOMATICALLY GENERATED 9

With Xporter and DocGen it's possible to export the information of an issue (e.g., a Test Plan), right from the issue screen (among other places).



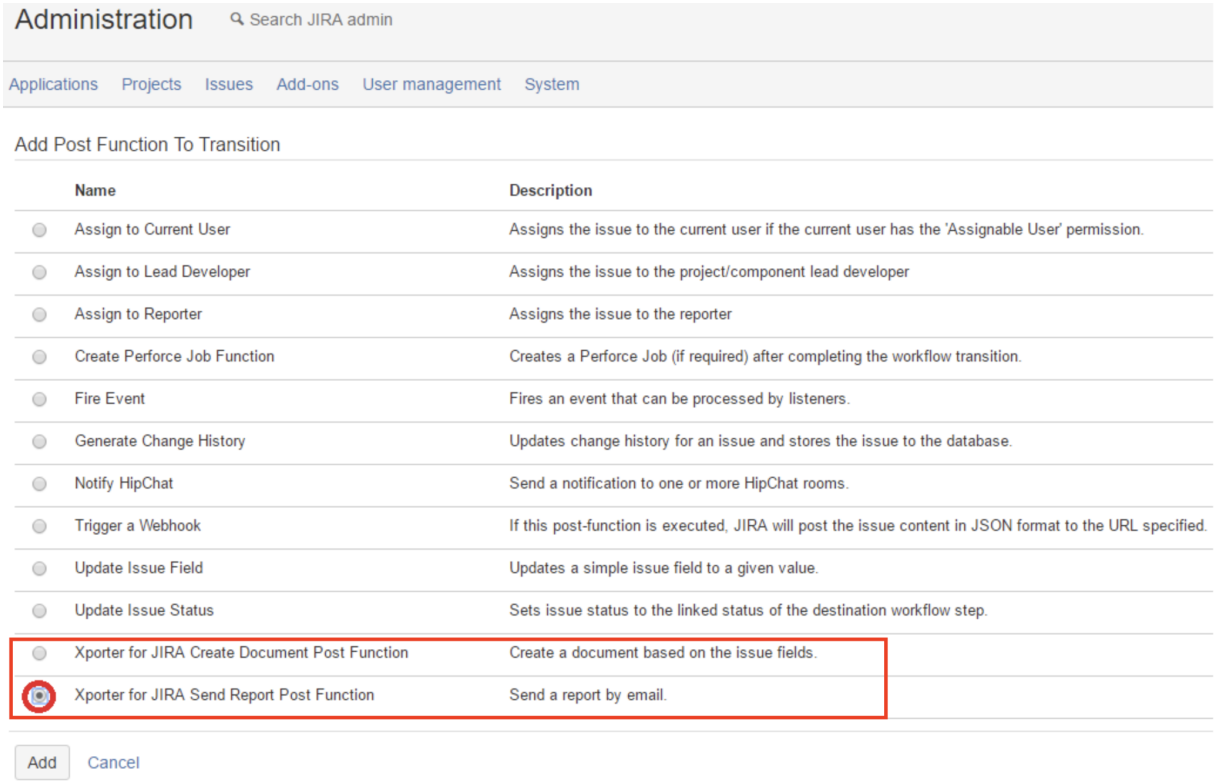
With Xporter it's also possible to export issues in bulk; this can be useful to export a set of Tests or Test Executions, for example.



With any of the two solutions, and just for reference, it's possible to:

- generate a custom PDF document with test cases and their specification, and the link to the related requirements (e.g., the typical "notebook of tests)
- generate a custom Word document with the current detailed results of a Test Plan, and all related iterations (i.e. Test Executions)

With Xporter it's possible to automate the generation of these documents and, for example, generate them upon a transition of a workflow, or attach to an existing Confluence page, or even send them to some email.



Learn more:

- Built-in Document Generator
- Xporter app
- Xporter Template Store

Risks

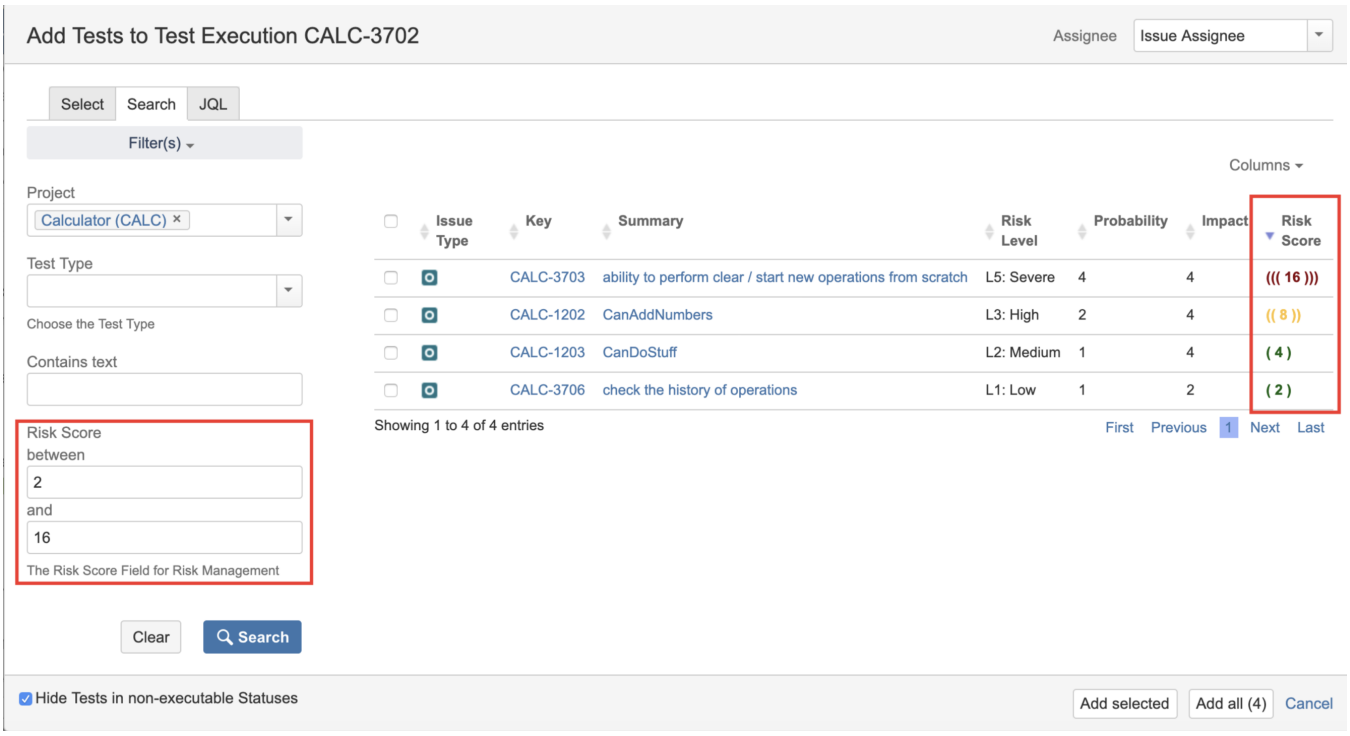
Handle risks with proper Risk Management and Risk-Based Testing

Handling risks is an intrinsic part of good testing and is essential to organizations that work in highly regulated environments.

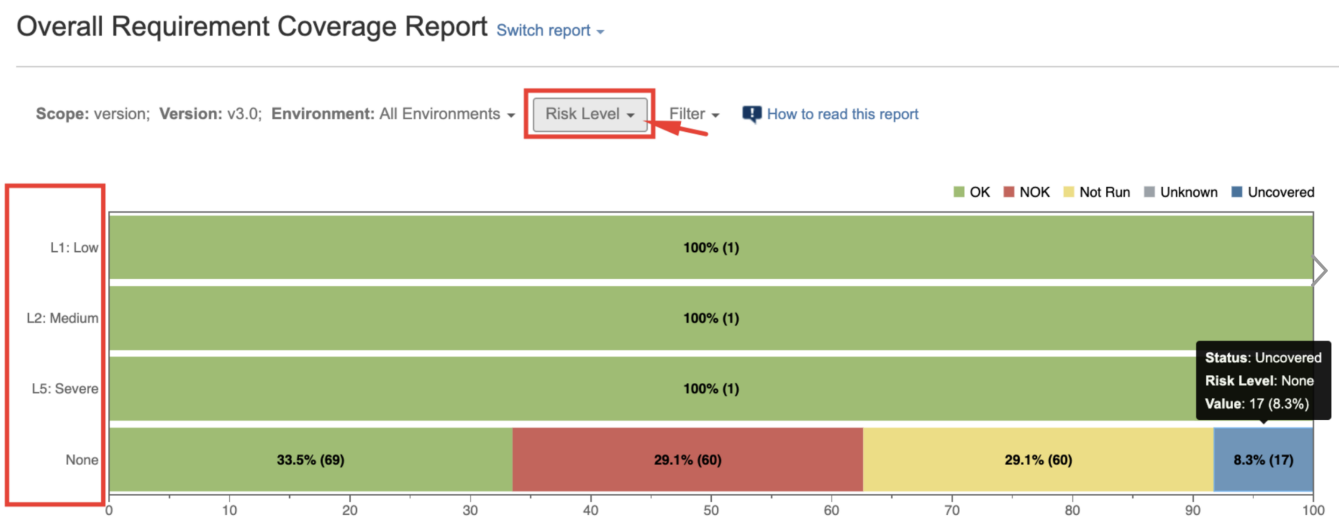
Xray supports Risk-Based Testing (RBT) and allows you to define risks at different levels: project, requirement or at the test-level.

Depending on the exact need, implementation of risk management can be accomplished using Jira built-in capabilities or through an additional app.

No matter the approach, we can use risks to pick the requirements and/or the Tests that we need based on risk criteria.



Whenever risks are defined at requirement level, we can analyze coverage for all requirements on our project and, for example, decide to make a release or not based on the status of the most critical requirements.



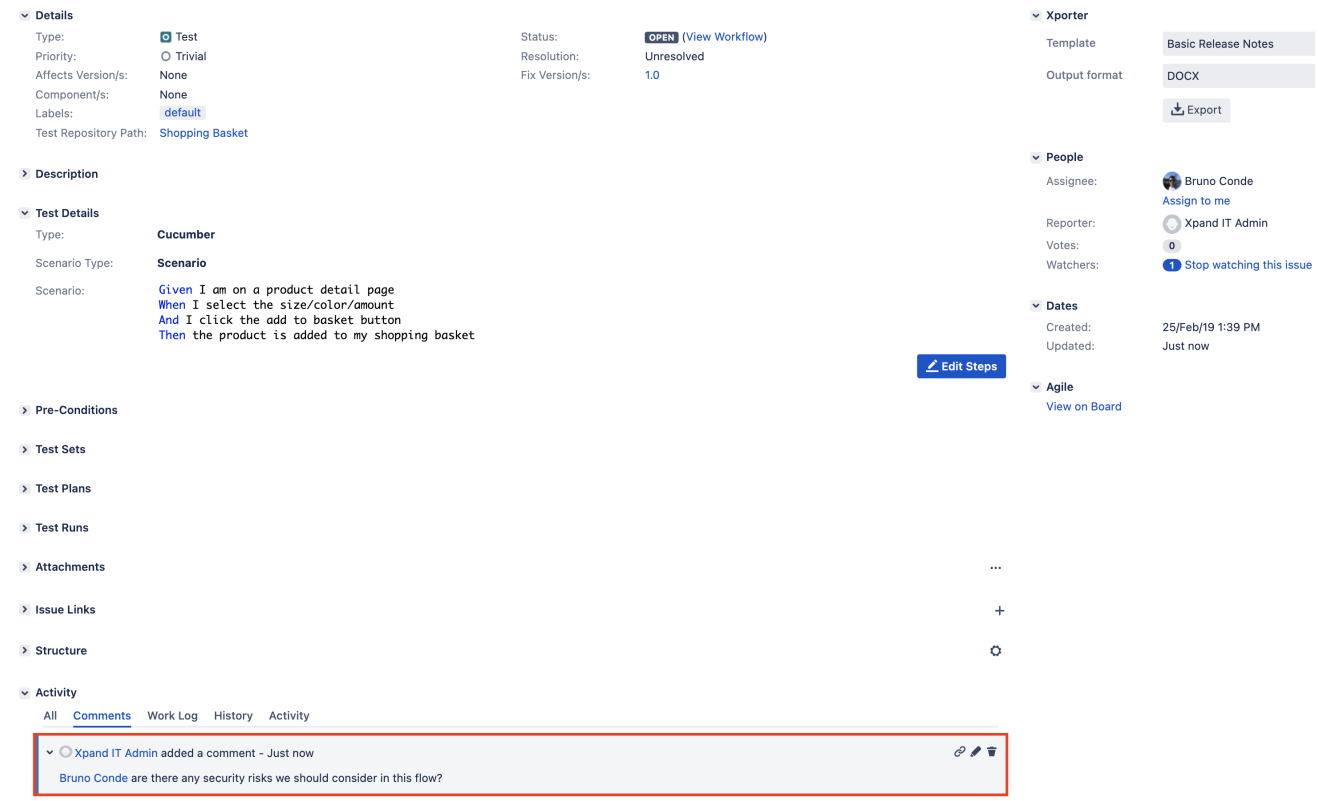
Learn more:

- Performing Risk-Based Testing (RBT) with Xray
- Risk Management
- Risk-Based Testing

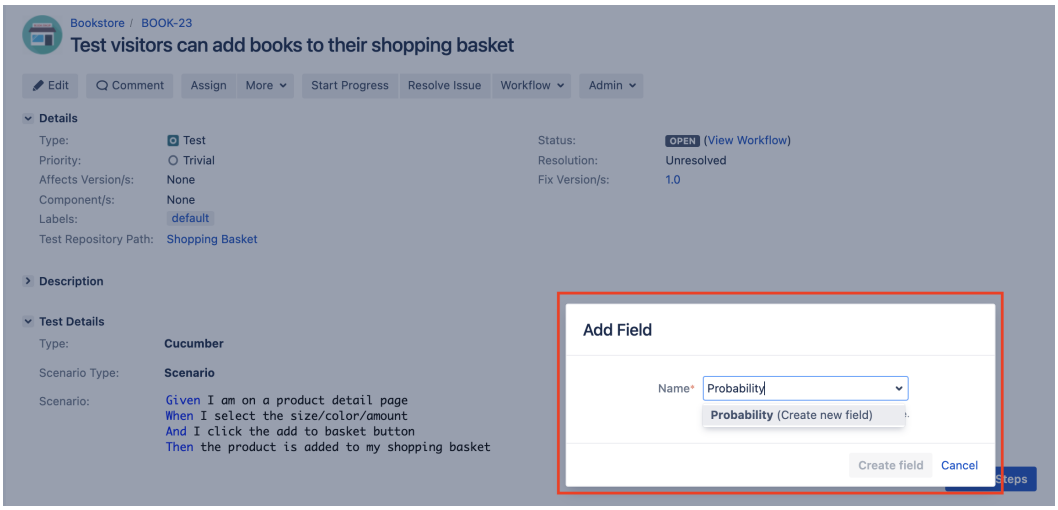
Mitigate risks by fostering clarity and facilitating collaboration, including on all testing activities

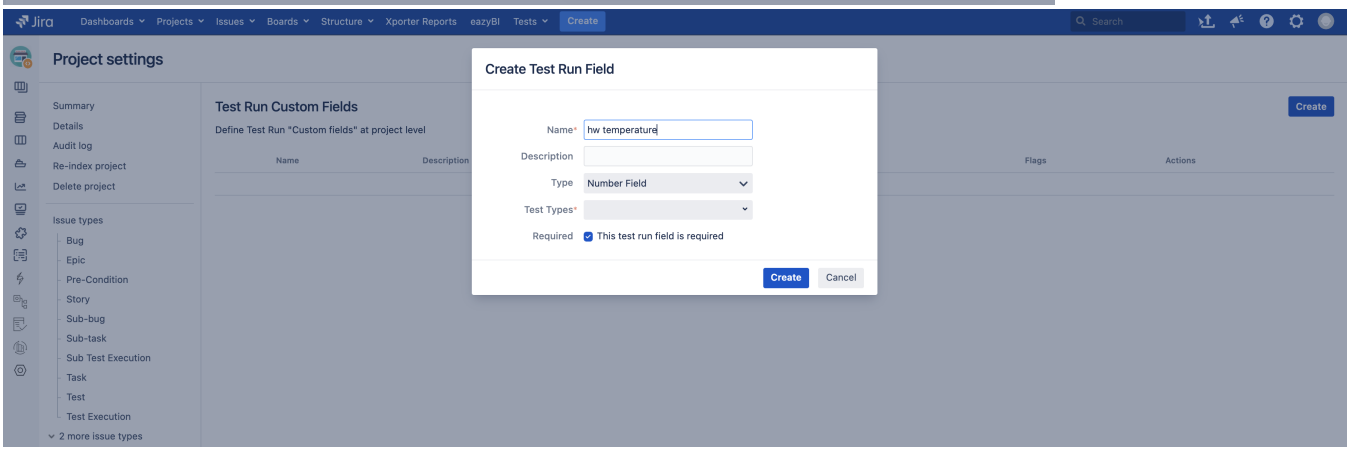
The best way to avoid misunderstandings is to clarify them ASAP and make sure doubts don't get lost. This is, in itself, a way of risk mitigation.

Comments and evidence (e.g., reference documentation) can be added on all Xray issue based entities (i.e. Test, Pre-Condition, Test Set, Test Plan, Test Execution, Sub-Test Execution) and also on Test Runs.



If your team decides you need one or more specific fields to embed some relevant information, then you can create custom fields on all of the previous entities, including on Test Steps and also on Test Runs, if you wish to do so.





Learn more:

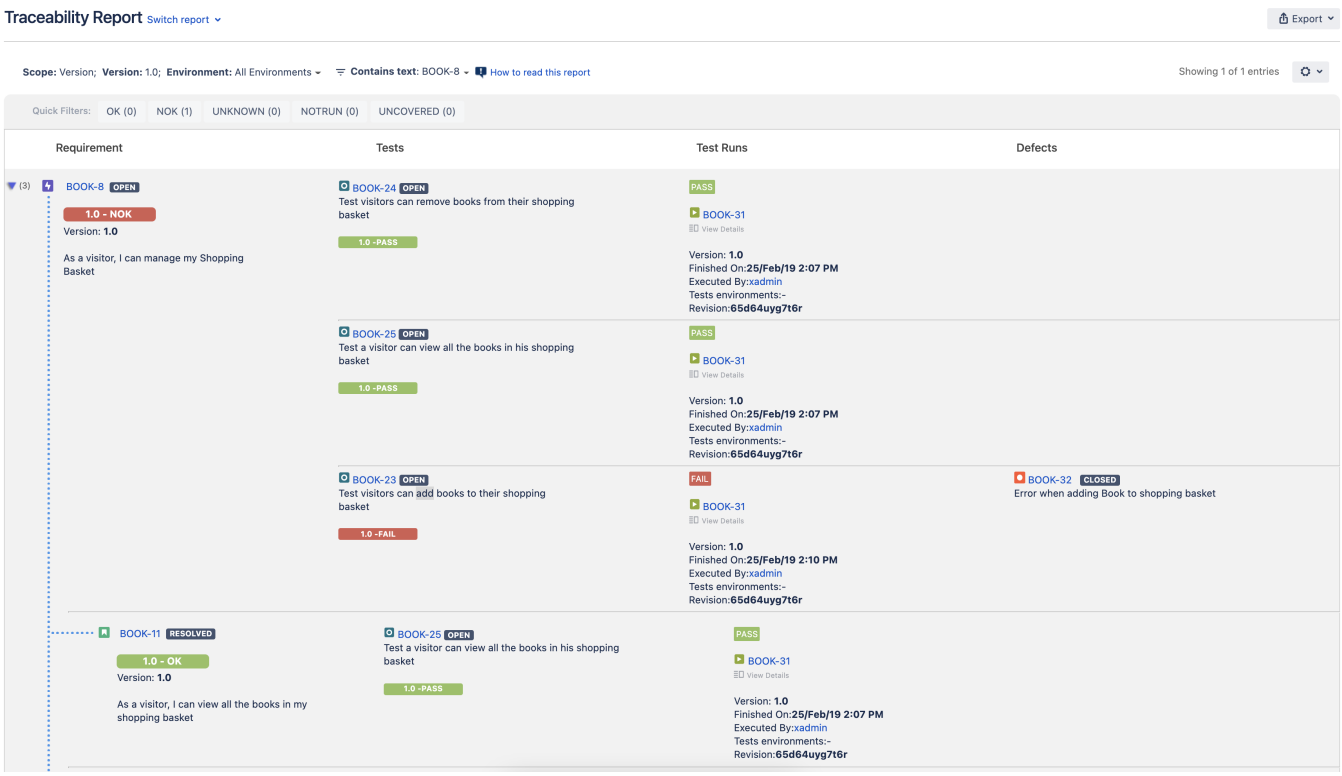
- Advantages of using Jira issue types in Test Management
- Configuring Test Step Custom Fields
- Configuring Test Run Custom Fields

Trace relationship between entities, up to the code

Xray provides full traceability between requirements, tests, their runs, and reported defects. This traceability not only provides the relation between the entities but also their status based on testing.

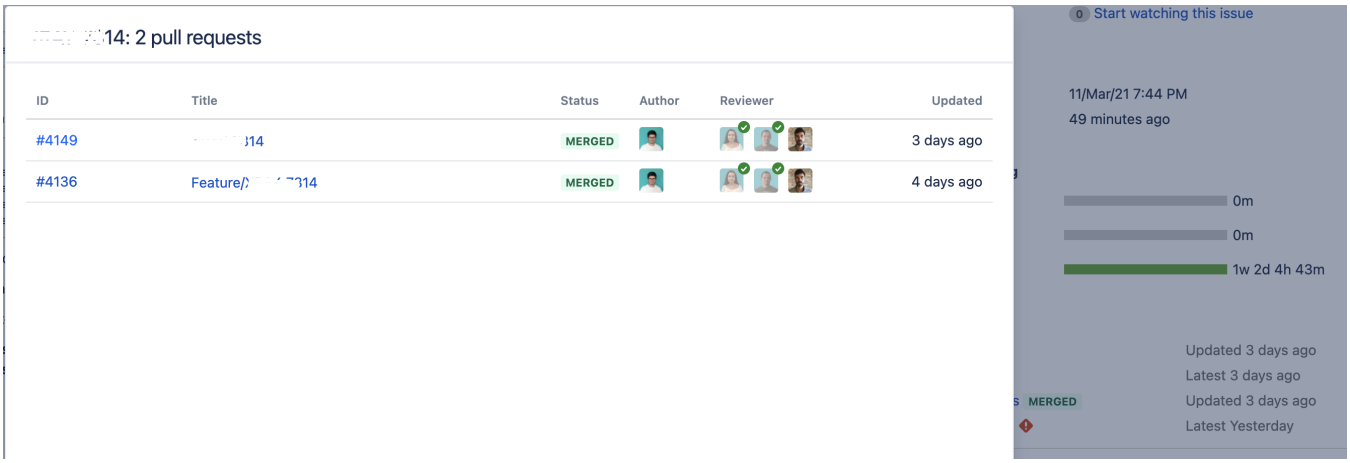
Traceability can be evaluated for a specific version and/or a specific environment, as its contents will depend on the testing results obtained for that specific context.

In the following screenshot, we can see an Epic that is NOK on version 1.0 even though one of the related Story issues is OK. In this example, the latest result for one of the tests (BOOK-23) that cover the Epic was a failure. It's also possible to see that a defect (BOOK-32) has been reported on that Test Run.



This helps auditing and diagnosing, knowing exactly what is impacting on what.

In Jira we can even look at the code of the requirements (e.g., Epic or Story issues) to facilitate analysis. The same can also be applied to automated tests that will have a corresponding Test issue that we can reference on the code commits. This helps analyze if a potential reported bug is on the features or even on the test automation code.



Facilitated diagnosis and Root-Cause Analysis

Root Cause Analysis (RCA) involves answering questions like these:

- **What** is the root **cause** of a problem?
- **When** was a problem introduced?
- **Why** was the problem introduced?
- **Where** was the problem introduced at?
- **How** can it be prevented to happen in the future?

Xray helps answer these questions in multiple ways, using all the capabilities mentioned earlier.

To find when a problem was introduced, for example, we can look at coverage information which can be used to identify the status of the requirement on each version. We can then use the Traceability Report or the Test Runs List Report to see in which revision it was reported, comments and other testing evidence left.

Learn more:

- [Xray] Xray's features for handling Root Cause Analysis

Learning

Facilitated training resources to improve knowledge and ensure a seamless adoption

Understanding a tool, how it works, its recommend use, possible configurations and extensions/integrations is essential to make the most out of it and fine tune it to the team and overall process needs.

Besides the extensive user guide documentation available online, there are also many courses on Xray Academy to understand not only Xray essentials but also more specific/advanced topics, such as test automation.

Learn more:

- Xray Academy
- Xray server/datacenter user guide

Standards and regulatory requirements

Standards and regulation that apply to organizations depend on the sector(s) they are working at and the type of products being developed and maintained.

For Medical Device Regulation (MDR), for example, these are some of the potential standards to consider:

- ISO 13485
- IEC 62304:2006
- ISO 14791:2019
- US FDA CFR Part 11