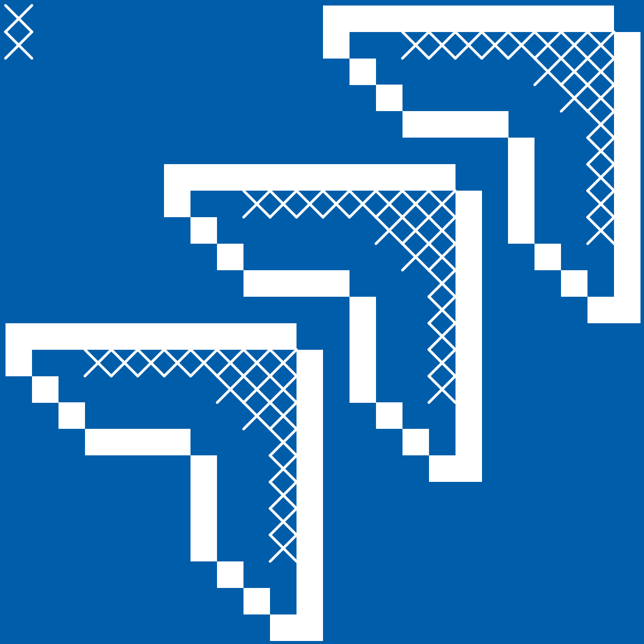
XRAY

Quick Start Guide for **Test Management**



ATLASSIAN

Content

Introduction

Only 6 years after we launched Xray, more than 10 million users at 10.000 customers in 135 countries count our applications as mission-critical.

We believe that Jira Software is a great tool for managing your software development lifecycle.

In this eBook, we'll show you how you can use Xray to infuse quality into every stage of development, so releasing stellar software comes naturally.

With all development requirements naturally linked to testing, you'll never overlook another test. Now, quality is built right into everything you develop.

> *"All companies fit into one of two buckets: either becoming a software company or being disrupted by one. Every industry is being fundamentally altered by software. We'll continue to make it easier for developers to build great software while taking that understanding of how highly effective teams work to help all teams better organize, collaborate, and communicate."*
>
> Mike Cannon-Brooks | Co-Founder, Atlassian

Test Management & Software Development Lifecycle

02

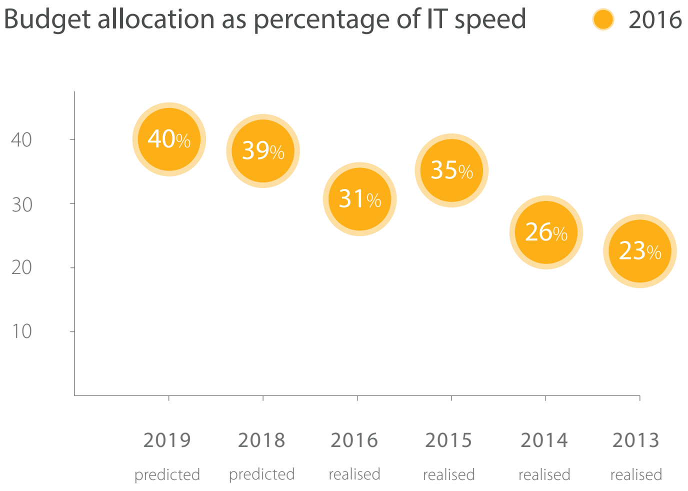
Test Management & Software Development Lifecycle

Everybody agrees that quality assurance is an important part of the software development process. Some critics consider it to be wider than a simple phase of the project development lifecycle. Testing is the straightest path to ensure the quality of the requirements delivery.

QA is a big deal for companies. According to the 2016 World Quality Report, there is a trending increase in the allocated IT budget for QA, with projections reaching 40% in 2019.

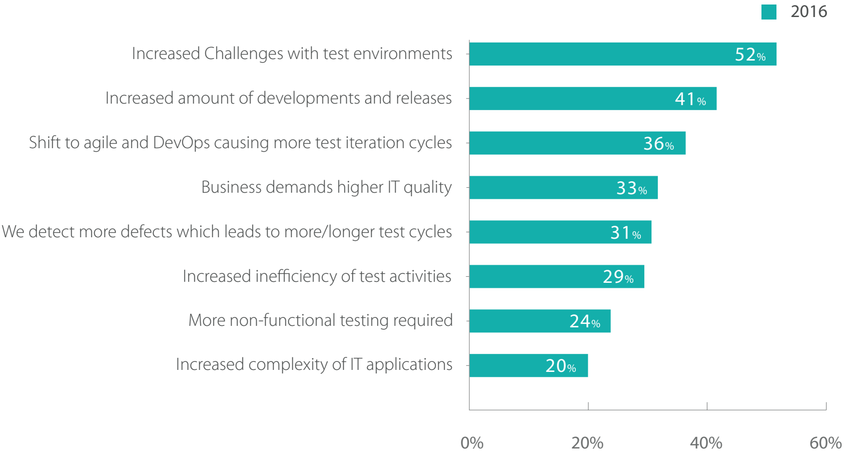
> *QA is a big deal for companies with increased budget allocation.*

Budget allocation as percentage of IT speed



Among other reasons, the increasing number of releases, the shift to agile methodologies, the inefficiencies of test activities (e.g., manual testing, test environment setup and lack of resources) explain the increased spending on QA.

Aspects impacting the increase of QA and Testing budgets



Testing into the new era

Testing should start early in the project.
Quality assurance teams should look for discrepancies between the customer requirements, the project scope proposal, and the actual implementation.

Following development cycles, QA teams come into play by creating different test cases and tracking their respective executions.

They will then report their findings to the development team, including defects, requirements, blocker issues and others. The entire process is repeated and the project moves forward. In fact, with Agile, DevOps and automated testing, tests should be implemented right away, avoiding most of the usual ping-pong between QA and Development teams.

With Continuous Integration and Continuous Delivery on the rise, there is a huge need for more and faster testing, while covering a wider range of areas than the typical functional requirements, such as UI, security and performance.

Is your team ready for the challenge?

Why Jira for Test Management?

03

Why is Jira so special?

Jira is a project management software that allows teams to capture and organize issues, assign work, and follow team activity, spending less time managing and more time developing software.

You don’t need to be a tech ninja to work with Jira. In fact, it is accessible to any person on your team. As a flexible issue tracker, it helps to manage all your different issues within projects, versions, dependencies, relations, defects and much more.

Jira is the right tool to organize work, plan your releases, and manage your projects.

> *Jira keeps everyone on the same page.*
> *From PMO and engineering to creative and business teams.*

What does Jira provide?

Polished User Experience

With Jira you can plan, track, and update work through issues across a fast and intuitive web interface.

Great Reporting

You can choose from several pre-built project reports. With a powerful Jira Query Language (JQL) you can personalize and create your own reports and gadgets, always keeping a good track of what is going on in your project.

Jira loves Agile

Jira Software lets you easily plan your backlog, sprints and track your team's progress.

Powerful Workflow Engine

We love Jira's great workflow engine. It enables teams to follow your unique process. You can either create a workflow that matches your needs and how your team works or just use an out-of-the-box workflow.

Use it for multiple tasks

Beyond bugs, features, tasks, and other project activities, Jira can also be used in other tasks that are not related to software development.

Why Jira for Test Management?

> *Why should we pick Jira when the subject is Test Management and Quality Assurance?*

Jira Software is known worldwide by developers, project managers, and even QA experts for its project management and issue-tracking capabilities. But let´s just jump into the world of quality assurance and test management.

For QA teams, Jira offers some solid features in terms of bug tracking and requirements management. And because the team concept is really important for Jira, it provides a great team collaboration environment so everyone benefits throughout the development cycle.

The question then arises: **Is Jira out-of-the-box the perfect match for QA and Test teams?**

Jira by itself already offers a set of features that can be quite useful for test teams:

- Ability to create test-related issues as new issue types
- Linking between issues (e.g., bugs and requirements)
- Association between issues and versions
- Automatic and manual issue assignment
- Specific workflow definition
- Integration with source code repository
- Search for and within issues
- Issue comments and attachments
- Reporting
- Highly customizable, screens, fields, and notifications
- Ability to import data from other systems

> *But even if you explore Jira's built-in features, are they enough for a QA team?*

Well, not really, because you are missing features specific to Test Management.

To address this, Jira allows you to add functionality, acting as base for Test Management apps.

In the Atlassian Marketplace, you can find several apps that provide Test Management features, evolving Jira to an even more powerful tool and making it more suitable for QA teams.

Integration with the Atlassian Brotherhood

Jira alone is awesome: it is the best issue tracker with great design, great usability, and great features. But it is not standalone at all; it belongs to the Atlassian family with many other tools.

If you feel that you are having a blast by working just with Jira, just hold your breath for the following products of the Atlassian family.

Confluence

Confluence centralizes all the knowledge and documentation for the project team. It allows teams to create, organize, and discuss their work.

Jira Service Desk

Jira Service Desk provides a modern and flexible service desk experience that streamlines customer requests, everything under the power of Jira.

Bamboo

Bamboo is the Continuous Integration Platform of the Atlassian family. Easily tie automated builds, tests and releases in a single workflow.

Bitbucket

Bitbucket is the Git platform for professional teams. It is a code collaboration tool on steroids with flexible deployment models.

+1000 Apps for Jira

Although Jira is awesome, it does not do everything. Apps are software products that extend Jira's set of functionalities.

The Atlassian Marketplace is a platform for Atlassian customers to discover, try, and buy apps for Atlassian applications. It has thousands of apps for Jira, some of them specifically made to leverage your Test Management experience.

Browse the Atlassian Marketplace and look for apps of the "Testing & QA" category.

Choosing your Test Management Tool?

04

Top Features of a Great Test Management Tool

1 Complete Test Management

A Test Management Tool should handle properly the standard Software Testing Life Cycle (STLC) phases, such as test planning, designing and reporting.

Whether you are using Waterfall, Agile or other model for your Software Development Life Cycle, your QA tool should handle it. And since Agile is on the rise, make sure your tool incorporates seamlessly the Agile manifesto.

A complete Test Management Tool should be able to manage both manual or automated tests. Preferably, if Behavior Driven Development (BDD) is adopted, automated tests specifications should be made directly in the tool.

A full, complete and effective Test Management is only assured when you are able to manage tests, requirements, and defects consistently, independently of the process you are following, the type of tests that you are using, and the system under test.

2 Flexibility

A good Test Management tool has configurations for certain features, but an excellent tool will allow you to fully adapt it to your needs.

How? Besides customization of screens and dialogs, organizations may require custom fields, custom status, and custom workflows.
An excellent tool will not force you to use a specific test process; rather, it will allow you to use it in different ways while keeping the focus on Quality Assurance.

Make sure you select a tool that is not just configurable, but also flexible.

3 Focus on Productivity

Testing can be a repetitive, boring task, especially if the team is doing validation using manual tests. Moreover, some tools in the past that should have made testing and managing tasks simpler, did exactly the opposite. If the team starts seeing a tool as an adversary, your overall productivity will be affected.

A Test Management tool should have in mind usability-related aspects, making it obvious and pleasant to work with. A motivated and focused team will dedicate their time to adding value to the project.

4 Reporting & Metrics

Are you ready to make a release? What if you're pressured with time? Maybe you have to drop some feature(s) but which one(s)? You need to take substantiated decisions, right?

The Test Management tool should assist you by means of built-in reports. Make sure you are able to make custom reports or extract custom metrics. Real-time requirement coverage information is crucial.
Make sure you have reports to assure that features developed on previous versions still work great and thus avoid regressions. Look for reports measuring the progress of your testing, including the status of your tests and the defects created during test executions.

Choose a tool that allows you to make decisions based on reports and relevant metrics, not on assumptions.

5 Integration

Ensure that all artifacts are easily managed and accessible from the outside by means of open and fully documented APIs. Thus, you are able to integrate your Test Management tool and extend its feature set without being dependent on vendors.

A good integration will allow you to connect with continuous integration servers, automated testing frameworks and cloud-based testing platforms.

6 Enterprise-Level Features

Make sure your Test Management tool has the right characteristics so you can trust your QA and your release.

All test-related artifacts should be handled carefully with characteristics expected at the enterprise level.

Consistency is a key enterprise feature for a Test Management tool. Why? Because you want to make sure that when you run a test, all the information of the test specification alongside the respective result persists and is not changed in the future, in case you modify the test specification afterwards.

Similarly, auditable changes on all test-related artifacts are crucial so you know when changes are done.

Ensure accountability by assigning someone responsible for artifacts (e.g., tests, test executions) so they're handled.

Finally, traceability, so you can easily navigate in-context between requirements, tests, executions and defects.

7 Collaboration

All team members contribute to the success of your project and the best way to achieve it is to foster collaboration between them.

A test, similar to a defect, will mostly require feedback from the requirement approval team. Likewise, it may need input from a tester or from someone doing support.

A QA tool should not be isolated from the remaining development ecosystem; it should integrate with all the software development tools and involve all the participants.

8 Good Integration with Jira

It's not really just a matter of good integration with Jira — it's all about making you think of Test Management in the first place.

You may find some tools that kind of act as a "proxy", requiring an external application for you to manage. Beware of all related synchronization issues.

Some tools use their own internal test artifacts and screens, making users feel uncomfortable and limiting the extraction of information or combining it with existing Jira issues.
Make sure the app you choose will naturally improve test management and will benefit from all the good things Jira has to offer.

9 Support

Whether you're thinking short or long-term, think of the support quality. Look for vendors dedicated to their products, who will guarantee long-term satisfaction.

Read reviews mentioning problems, training, support services, and general credibility.

Introducing

XRAY

04

Test Management with Xray for Jira

Xray is one of the most well-known Jira apps, always placing in the Top 10 bestselling apps in the Atlassian Marketplace.
The reasons for this are quite clear. With less than three years in the market, it is already one of the most famous tools among test teams, winning many clients from top-notch standalone apps.

> *Xray issues perfectly map your testing activities, while inheriting the best of Jira.*

Enterprise Testing Features with Jira

Xray uses Jira issues to represent most of the testing-related entities.

Test   Pre-Condition   Test Set   Test Execution   Test Plan

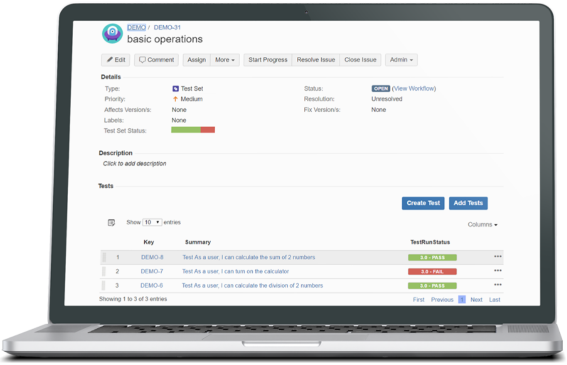
Thus, you are able to leverage all Jira benefits into your tests, such as issue assignment, comments, screens and workflows customization, issue linking, custom fields, and many more. Besides accountability, it's possible to audit changes using the issue's history.

Xray is one of the few tools that ensures data consistency, by storing test definitions for past executions, guaranteeing the integrity of previous test runs, no matter what you do afterwards.
Finally, Xray provides extensive configuration options, to fully adapt it to your specific needs.

Testing-oriented

Xray allows you to maximize all your testing activities, from test design to reporting.
The test design phase includes the specification of both manual and automated tests using Test issues. Sometimes, you'll need to ensure that some conditions are met before executing a Test. Xray allows you to associate one or more Pre-Conditions and reuse them between tests.

Testing that makes sense, simple yet complete and powerful.



DEMO / DEMO-31

basic operations

Details

Type: Test Set
Priority: Medium
Affects Version/s: None
Labels: None
Test Set Status: 🟩🟥

Status: OPEN (View Workflow)
Resolution: Unresolved
Fix Version/s: None

Description

Click to add description

Tests

Show 10 entries

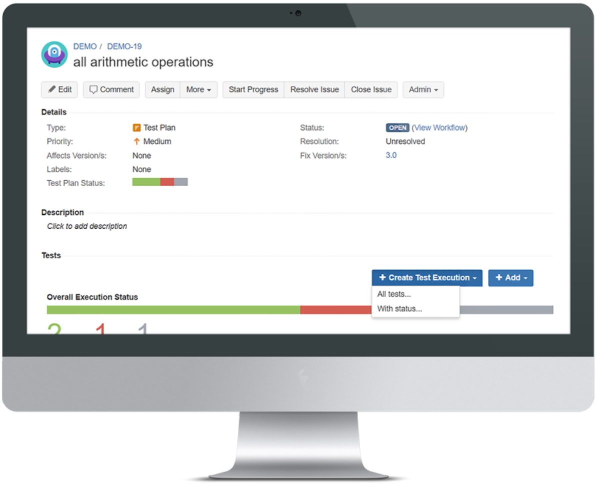
	Key	Summary	TestRunStatus	
1	DEMO-8	Test As a user, I can calculate the sum of 2 numbers	3.0 - PASS	•••
2	DEMO-7	Test As a user, I can turn on the calculator	3.0 - FAIL	•••
3	DEMO-6	Test As a user, I can calculate the division of 2 numbers	3.0 - PASS	•••

Showing 1 to 3 of 3 entries

Tests may be organized the Jira way: using labels, components, priority or other fields. Since this does not scale, Xray introduces the Test Set issue that basically groups a set of Tests by your own criteria. A Test may belong to multiple Test Sets.

Tests may be run right away as unplanned test executions. However, effective test planning is done by creating Test Plan issues, where you can import Tests or Test Sets.

The Test Plan always reflects the current status of your relevant tests, no matter how many executions you do.



all arithmetic operations

Details

Type: Test Plan
Priority: Medium
Affects Version/s: None
Labels: None
Test Plan Status:
Status: OPEN (View Workflow)
Resolution: Unresolved
Fix Version/s: 3.0

Description

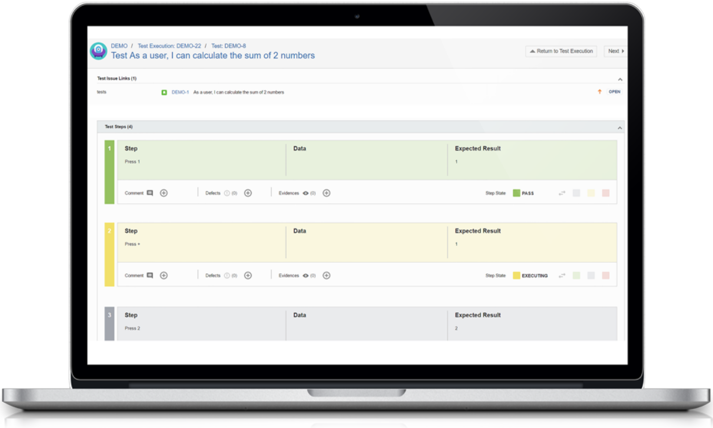
Click to add description

Tests

Overall Execution Status

You might have guessed already: a test execution task is, in fact, a Test Execution issue.

A Test Execution has a list of Tests and respective results after being executed by a user on a specific system version and revision. During the Test Execution, you may leave comments, create defects, attach files, at a step or Test level. All fully traceable, of course.



DEMO / Test Execution: DEMO-22 / Test: DEMO-8

Test As a user, I can calculate the sum of 2 numbers

Test Issue Links (1)

tests DEMO-1 As a user, I can calculate the sum of 2 numbers OPEN

Test Steps (4)

	Step	Data	Expected Result
1	Press 1		1
2	Press +		1
3	Press 2		2

1: Comment Defects (0) Evidences (0) Step State PASS

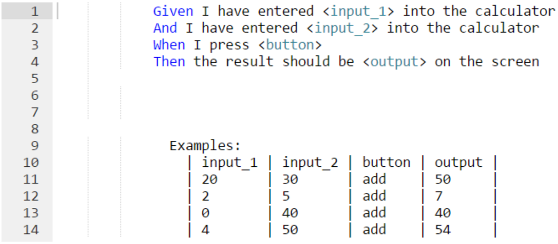
2: Comment Defects (0) Evidences (0) Step State EXECUTING

Automated and Manual Tests

It does not matter how tests are implemented, Xray handles them in the same way. This allows users to have an integrated and consolidated view of test management and the overall requirement coverage.

Xray allows you to specify Cucumber automated tests in your natural language (e.g., English) using Gherkin. Xray even allows you to manage generic automated tests, no matter what automation framework you're using. Really!

BDD with Cucumber is supported natively but tests may be automated in any framework



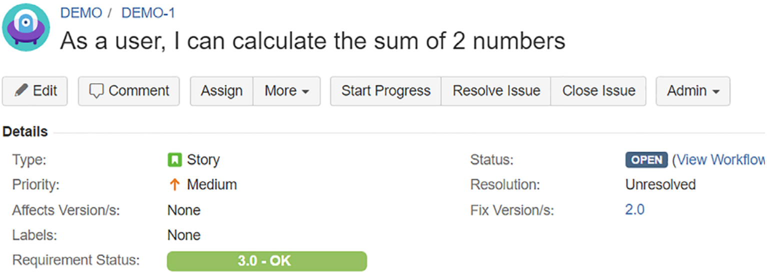
```
Given I have entered <input_1> into the calculator
And I have entered <input_2> into the calculator
When I press <button>
Then the result should be <output> on the screen




  Examples:
    | input_1 | input_2 | button | output |
    | 20      | 30      | add    | 50     |
    | 2       | 5       | add    | 7      |
    | 0       | 40      | add    | 40     |
    | 4       | 50      | add    | 54     |
```

Focus on Requirement Coverage

To ensure the correctness of the features you need to deliver, Xray is extremely oriented to requirement coverage tracking.

Xray automatically calculates the requirement status based on executions of Tests that cover it, making the status available as a queryable custom field that you can expose in screens, filter results or reuse in your dashboards.

Xray is also aware of sub-requirements (e.g., Epic vs. Story), reflecting automatically their coverage status on the parent requirement.



DEMO / DEMO-1

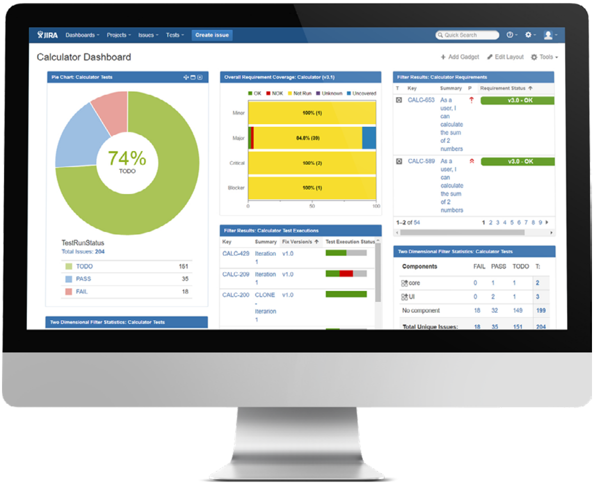
As a user, I can calculate the sum of 2 numbers

Details

Type:	Story	Status:	OPEN (View Workflow
Priority:	Medium	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	2.0
Labels:	None		
Requirement Status:	3.0 - OK		

> *Ensure the quality of your release with real-time requirement coverage status charts reusable in dashboards.*

Real-time requirement coverage reports allow you to evaluate the completeness of your product at any time, and see how it is evolving. Whenever calculating the status, requirements may be grouped by priority, component, etc., making it easier to assess readiness or impacts on relevant parts of your system.



Calculator Dashboard

TestRunStatus
Total Issues: 204

TODO	151
PASS	35
FAIL	18

Filter Results: Calculator Test Executions

Key	Summary	Fix Version/s	Test Execution Status
CALC-429	Iteration 1	v1.0	
CALC-209	Iteration 1	v1.0	
CALC-200	CLONE - Iterarion 1	v1.0	

Filter Results: Calculator Requirements

T	Key	Summary	P	Requirement Status
	CALC-653	As a user, I can calculate the sum of 2 numbers		v3.0 - OK
	CALC-589	As a user, I can calculate the sum of 2 numbers		v3.0 - OK

1–2 of 54 1 2 3 4 5 6 7 8 9

Two Dimensional Filter Statistics: Calculator Tests

Components	FAIL	PASS	TODO	T:
core	0	1	1	2
UI	0	2	1	3
No component	18	32	149	199
Total Unique Issues:	18	35	151	204

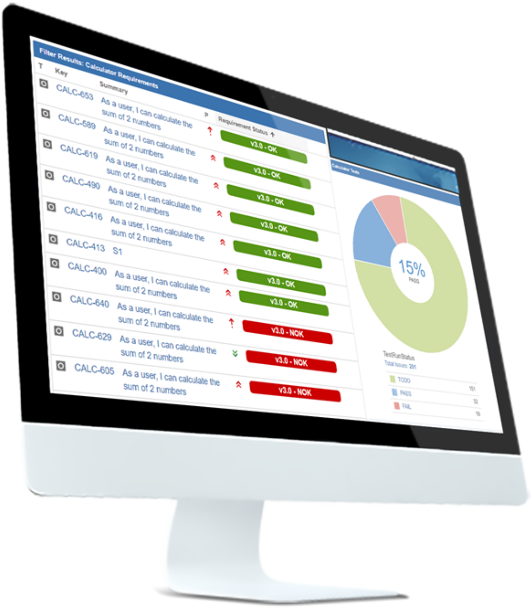
Integration

Xray includes a REST API out-of-the-box on top of Jira’s own API, so you will be able to manage all test-related entities from the outside world.
This same API facilitates the integration with CI tools, such as Bamboo, Jenkins, and others.

To make your life easier, you're able to import tests or test steps from Excel/CSV files.
Test results may also be imported, from your local or CI environment. Xray supports importing test results from Cucumber, Xamarin Test Cloud, Behave or any other framework using JSON.

Reports and Gadgets

Enhanced reporting using Jira gadgets (e.g., pie charts, tables), JQL, and through specific custom fields, besides the built-in gadgets for requirement coverage overview and evolution.



Filter Results: Calculator Requirements

T	Key	Summary	P	Requirement Status ↑
▣	CALC-653	As a user, I can calculate the sum of 2 numbers	↑	v3.0 - OK
▣	CALC-589	As a user, I can calculate the sum of 2 numbers	⇈	v3.0 - OK
▣	CALC-619	As a user, I can calculate the sum of 2 numbers	⇈	v3.0 - OK
▣	CALC-490	As a user, I can calculate the sum of 2 numbers	⇈	v3.0 - OK
▣	CALC-416	As a user, I can calculate the sum of 2 numbers	⇈	v3.0 - OK
▣	CALC-413	S1	⇈	v3.0 - OK
▣	CALC-400	As a user, I can calculate the sum of 2 numbers	⇈	v3.0 - OK
▣	CALC-640	As a user, I can calculate the sum of 2 numbers	↑	v3.0 - NOK
▣	CALC-629	As a user, I can calculate the sum of 2 numbers	⇊	v3.0 - NOK
▣	CALC-605	As a user, I can calculate the sum of 2 numbers	⇈	v3.0 - NOK

Calculator Tests

TestRunStatus
Total Issues: 201

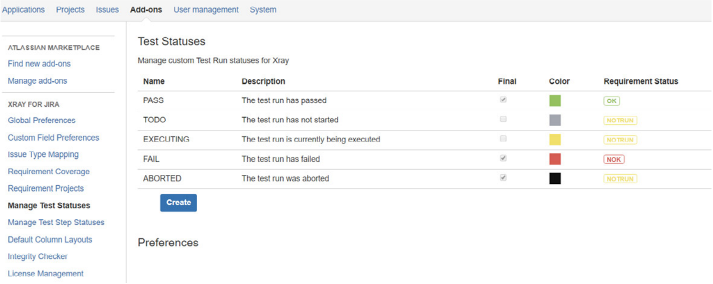
🟩 TODO	151
🟦 PASS	32
🟥 FAIL	18

Export to fully customized docs

Build your own custom reports in Word, Excel or PDF. Use the Xporter for Jira app when you need to make a nice formal test acceptance document, include the requirement traceability matrix, or generate analysis and graphics on an Excel sheet.

Test Environments

Use Test Environments to execute the same Test in different environments (e.g., devices, browsers) and have a joint view of the overall Test Execution status.



Test Statuses

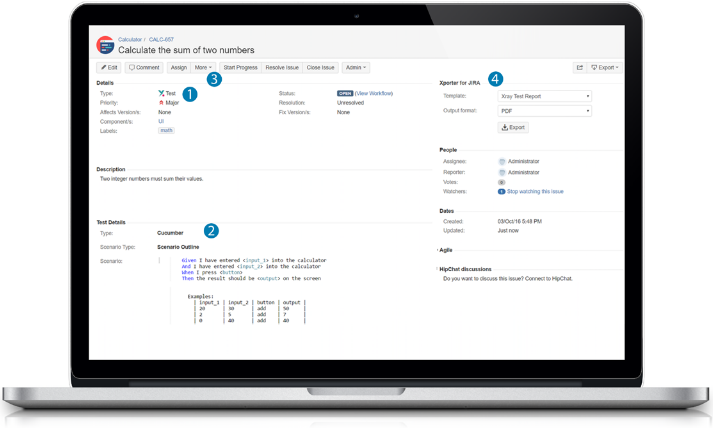
Manage custom Test Run statuses for Xray

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

Preferences

Extensive Configuration

Xray administration preferences allow the customization of some features: how the requirement coverage works, which issue types should be considered “requirements” or “defects”, projects to be used in Xray, custom Test and step statuses, and many more.



Calculate the sum of two numbers

Details

Type: Test
Priority: Major
Affects Version/s: None
Component/s: UI
Labels: math

Status: OPEN (View Workflow)
Resolution: Unresolved
Fix Version/s: None

Description

Two integer numbers must sum their values.

Test Details

Type: Cucumber
Scenario Type: Scenario Outline
Scenario:

```
Given I have entered <input_1> into the calculator
And I have entered <input_2> into the calculator
When I press <button>
Then the result should be <output> on the screen

  Examples:
    | input_1 | input_2 | button | output |
    | 20      | 30      | add    | 50     |
    | 2       | 5       | add    | 7      |
    | 0       | 40      | add    | 40     |
```

Xporter for JIRA

Template: Xray Test Report
Output format: PDF

People

Assignee: Administrator
Reporter: Administrator
Votes: 0
Watchers: 1 Stop watching this issue

Dates

Created: 03/Oct/16 5:48 PM
Updated: Just now

HipChat discussions

Do you want to discuss this issue? Connect to HipChat.

In this screen, you can see that Xray is a seamlessly integrated app for Jira.

The picture shows how a normal Jira issue can look like, including the capacity to link to other issues, clone, attach files and plenty other, such as:

1 Test details including priority, components, label, status and fix version.

2 Automated Test definition written on Cucumber.

3 Inside "More" tab, you can export your resolved tests directly from Cucumber.

4 Xporter is another app for Jira allowing you to export issues and build reports using custom templates.

Conclusion

Historically, software development and quality assurance were one and the same. If you built it, you also tested it. But then software grew up, and as it got more and more complex, dev and QA needed to split up in order to do their job right.

But instead of these two teams remaining close friends, they **grew far apart**. Each in their own world, operating in different environments, using their own workflows, and speaking different languages.

How can we expect our software to be at the highest quality, when quality and development are so disconnected?

With Native Quality Management, you embed the Test Management process into the development workflow, Jira. That way teams work side by side to release consistently quality software - together.

Now it's quality's turn. It's time to usher in a new wave, a whole new approach to delivering high-quality software – by embedding it throughout the entire development lifecycle, instead of treating it like an afterthought.

It's time to give your SDLC the quality upgrade it badly needs – **by transforming it into the Software Quality Lifecycle**.

This is the power of Native Quality Management.

Let's build better software. Together.

getxray.app