

JIRA Software

Quick Start Guide for Test Management



Content

Introduction

Test Management & Software Development Lifecycle

Testing into the new era

Why JIRA for Test Management?

Why is JIRA so special?

What does JIRA provide?

Why JIRA for Test Management?

Integration with Atlassian Brotherhood

+1000 Add-ons for JIRA

Choosing your Test Management Tool

Top features of a great Test Management Tool

Introducing Xray for Jira

Test Management with Xray for JIRA

Conclusion

Introduction

As an Atlassian Expert since 2004, Xpand IT delivers top IT services and products to meet Application Lifecycle Management and Software Development Lifecycle requirements.

We believe that JIRA Software is a great tool, not only for its issue and project tracking capabilities, but also because you can tune it with amazing add-ons to make it an even better and complete platform.

In this ebook, we will focus on test management and quality assurance in JIRA Software. We will share awesome JIRA add-ons used by Fortune 500 companies like Porsche, Audi, Vodafone, Lufthansa, Nike, Audi, Samsung and many others.



"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

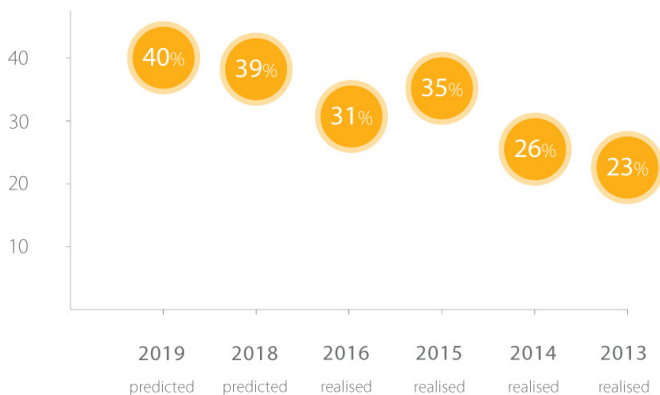
Test Management & Software Development Lifecycle

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

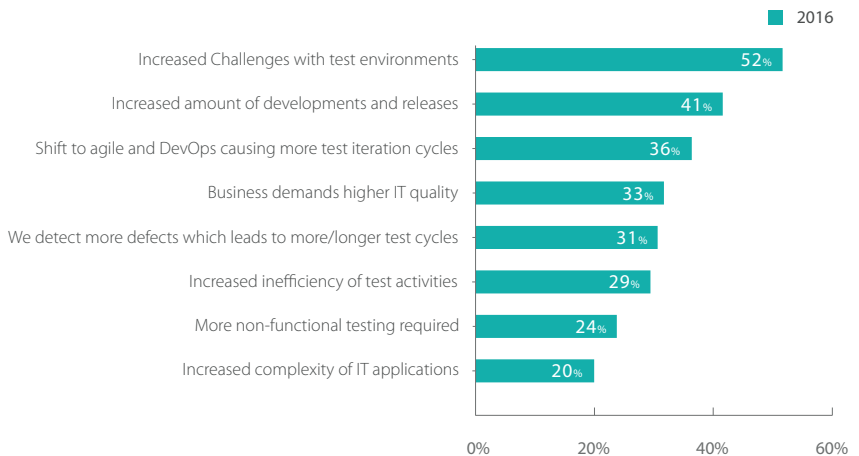
QA is a big deal for companies with increased budget allocation.

Budget allocation as percentage of IT spend ● 2016



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), among other reasons, explain the increased spending on QA.

Aspects impacting the increase of QA and Testing budgets



Testing into the new era

Testing should start early on 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.

In the end, they will report their results to the development team, including defects, requirements, blocker issues and plenty of other issues. 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 & 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?**

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

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

JIRA loves Agile

JIRA Software offers the possibility to easily plan your backlog, sprints and to track your team progress.

Powerful Workflow Engine

We love JIRA's great workflow engine. It offers teams the possibility to follow your unique process for shipping software. You can either create a workflow that matches your needs and the way your team works or just use an out-of-the-box workflow.

Use it for multiple tasks

From bugs, features, tasks and other project activities, JIRA can also be used in other tasks that are not related with 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 world-wide by developers, project managers and even QA experts for its project management and issue-tracking capabilities. But let's just jump to the world of quality assurance and test management.

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

But then the question 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 (ex: 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.

There are ways to address this: JIRA offers the possibility to add functionality, acting as building ground for Test Management Add-ons.

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

Integration with Atlassian Brotherhood

JIRA alone is awesome: it is the best issue tracker with great design, great usability and great features. But it is not alone 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.

HipChat

HipChat is the Atlassian group and video chat tool that allows discussions to move throughout different tools and share progress instantaneously.

Bamboo

Bamboo is the Continuous Integration Platform of the Atlassian family. Easily tie automated builds, tests and release 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 ADD-ONS FOR JIRA

Although JIRA is awesome, it does not do everything. Add-ons are software products that extend JIRA's set of functionalities.

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

Browse the Atlassian Marketplace and look for add-ons of the "Testing & QA" category.



Choosing your **Test Management Tool**

Top Features of a Great Test Management Tool

1 Complete Test Management

A Test Management Tool should handle properly the standard STLC (Software Testing Life Cycle) 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 likewise. Preferably, if BDD (Behaviour Driven Development) 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 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 enforce you with a specific test process, 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, make sure it's flexible.

3 Focus on Productivity

Testing can be a repetitive, boring task, namely if the team is doing validation using manual tests.

Moreover, the past has brought us tools that instead of making testers and managers 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 on 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 consubstantiated 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 take 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 expectable at 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 with the respective result gets persisted and is not changed in the future, namely if 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 a responsible to 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, similarly to a defect, will mostly require feedback from the requirement approval. 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 extract of information or combining it with existing JIRA issues.

Make sure the add-on 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 in short or long terms, 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



Test Management with Xray for JIRA

Xray is one of the most well-known JIRA add-ons, having always a place in the Top 10 bestselling add-ons 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

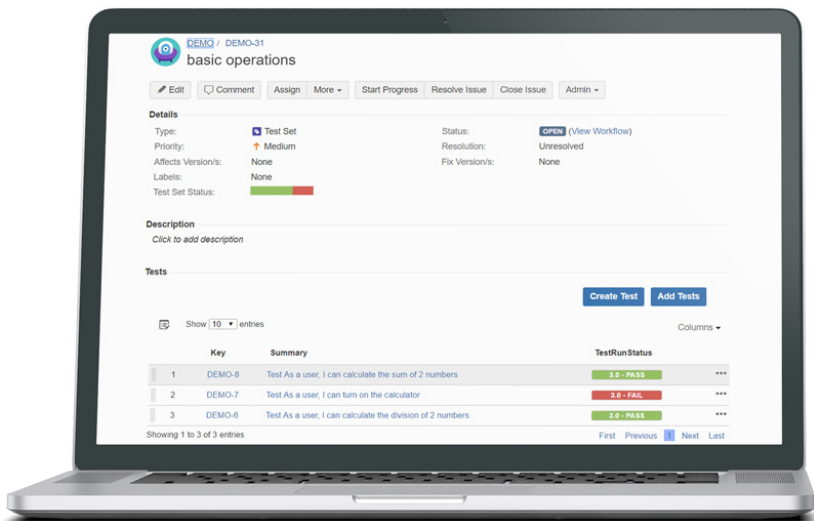
Thus, you are able to leverage all JIRA benefits into your tests, such as issue assignment, comments, screens and workflows customization, issue linking, custom fields and many more. Besides accountability, it's possible to audit changes using issue's history. Xray is one of the few tools that assures 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 associate one or more Pre-Conditions and reuse them between tests.

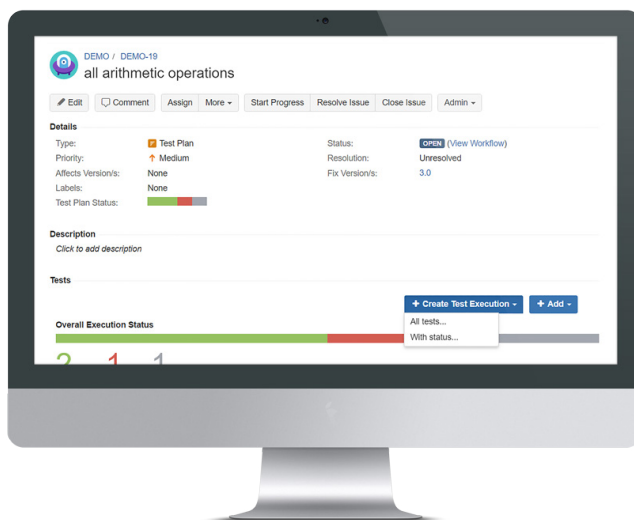
Testing that makes sense, simple yet complete and powerful.



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.

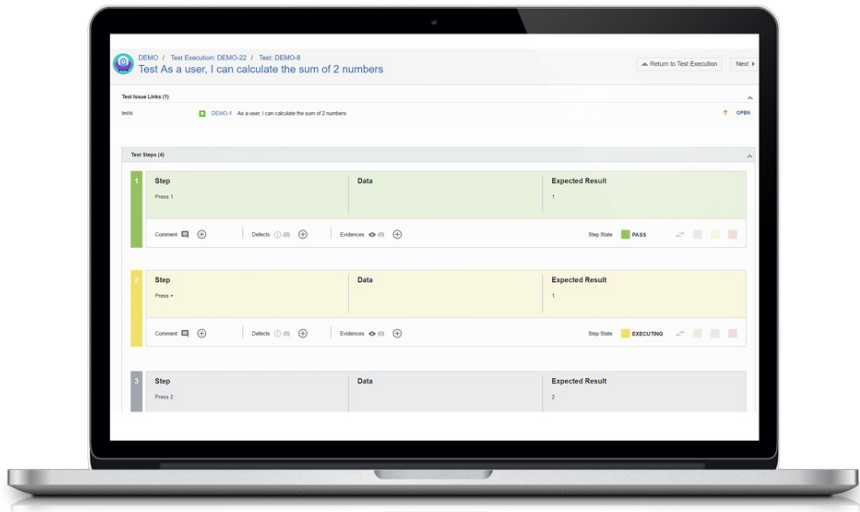
Tests may be run right away as unplanned test executions. However, effective test planning is done by creating Test Plan issues, where you can import Tests or Test Sets to.

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



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.



Automated and Manual Tests

It does not matter how tests are implemented, Xray handles them in the same way, allowing 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

```
1 |
2 |
3 |
4 |
5 |
6 |
7 |
8 |
9 |
10 |
11 |
12 |
13 |
14 |
```

```
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

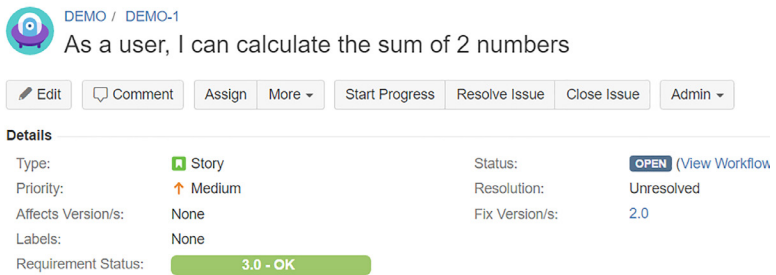


Focus on Requirement Coverage

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.



The screenshot shows a JIRA issue titled "As a user, I can calculate the sum of 2 numbers". The issue is of Type "Story", Priority "Medium", and Status "OPEN". It has no labels and affects version 3.0. The requirement status is "3.0 - OK", indicated by a green bar. The page includes buttons for Edit, Comment, Assign, More, Start Progress, Resolve Issue, Close Issue, and Admin. A "Details" section is visible below the buttons.

DEMO / DEMO-1

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

Edit Comment Assign More Start Progress Resolve Issue Close Issue Admin

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 allows 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.



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.

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.

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.



Export to fully customized docs

Build your own custom reports in Word, Excel or PDF. Use the [Xporter for JIRA Add-on](#) whether you need to make a nice formal test acceptance document, include the requirement traceability matrix, or make some analysis and graphics in 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.

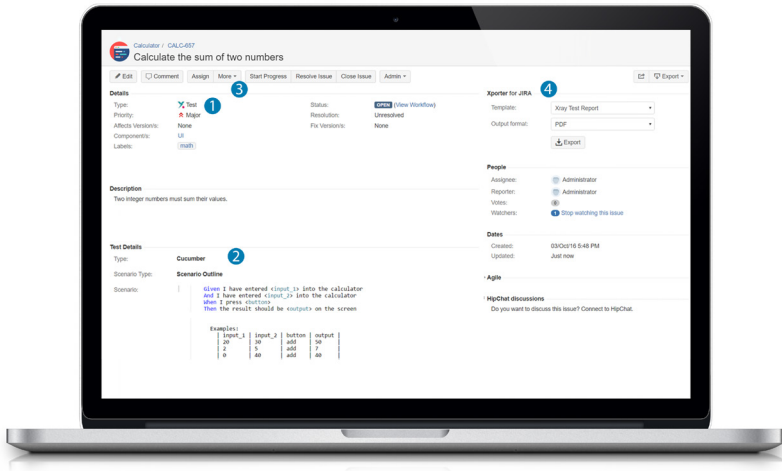
The screenshot shows the 'Test Statuses' configuration page in JIRA Xray. The page has a sidebar with navigation links: Applications, Projects, Issues, Add-ons, User management, and System. The main content area is titled 'Test Statuses' and 'Manage custom Test Run statuses for Xray'. It contains a table with the following data:

Name	Description	Final	Color	Requirement Status
PASS	The test run has passed	<input checked="" type="checkbox"/>	Green	OK
TODO	The test run has not started	<input type="checkbox"/>	Grey	NO RESULT
EXECUTING	The test run is currently being executed	<input type="checkbox"/>	Yellow	NO RESULT
FAIL	The test run has failed	<input checked="" type="checkbox"/>	Red	NOOK
ABORTED	The test run was aborted	<input checked="" type="checkbox"/>	Black	NO RESULT

Below the table is a 'Create' button. At the bottom of the page, there is a 'Preferences' link.

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.



In this screen you can see that Xray is a seamlessly integrated add-on 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 add-on for JIRA allowing you to export issues and build reports using custom templates.

Conclusion

We all know that the process of testing can be, for a software tester, an everyday challenge. Rather than just an activity, software testing must be considered as one of the most important practices inside the Software Development Life Cycle.

It must assure that all the requirements are met, business and technical, functional and non-functional, and above all, that the product works as expected.

Why do we need several tools for different purposes when we can use the same platform for our Software Development process?

In this ebook we tried to go over the main reasons why JIRA is one of the best tools for your QA and Testing teams.

We have generically enumerated the top main requirements for a Test Management Tool and the specific points to have in mind if you look for it in the Atlassian Marketplace.

Finally, we have shown how Xray for JIRA fulfils the needs of teams wishing to do Test Management in JIRA.

Xray for JIRA is a comprehensive and enterprise proven Test Management Tool for automated and manual tests.

JIRA is awesome. Xray for JIRA makes it even better for Test Management.

Thanks for reading our eBook!
Share your thoughts on social networks using the hashtag **#XPebooks**.



London

1 Primrose Street London, EC2A 2EX
United Kingdom

Phone **+44 845 867 0875**

Lisbon

Edifício Mar do Oriente, Alameda dos Oceanos
Lote 1.07.1Y – Fração 2.3
1990-203 Lisboa, Portugal

Phone **+351 21 896 7150**

Viana do Castelo

Rua de Fornelos 77 4900-709 Viana do Castelo
Portugal

Phone **+351 21 896 7150**