

Document Generator Template: Requirements Report

- Purpose
- Output Example(s)
- How to use
 - Source data
 - Output format
 - Report assumptions
 - Usage examples
 - Export Stories of your project
 - Export Epics associated with a given fix version
 - Export Stories obtained on a given Test Environment
 - Export Story from its detailed view
- Understanding the report
 - Layout
 - "TestingEffortOverview" sheet
 - "DetailedExecutionBreakdown" sheet
- Customizing the report
 - Exercise 1: add a custom issue type that belongs to Requirements in your configuration
 - Exercise 2: track custom test types
- Performance

Purpose

This report lists some details of the selected requirements (Stories or Epics) in Xray, enabling them to be extracted in an Excel format. With this ability to extract the report you can use it for analysis of trends and current testing status, or process this information to generate some metrics, for example, or even share it with someone else who still needs access to Jira.

Possible usage scenarios:

- get an overview of Requirements by Component and/or Priority
- track the time spent on specifying and executing tests associated with a given Requirement
- see all the Test Run statuses count for a given Requirement

Store - [DC](#); [Cloud](#)

Output Example(s)

The following table shows an example of the columns/rows you should expect.

Requirement Key	Summary	Component	Priority	Assignee	Workflow Status	Total Workflow Lifetime	Total Number of Defects	Number of Unique Test Run Assignees
CYPLUG-8	Requirement Debug		Trivial		Open		0	0
CALC-373	S1		Trivial		Awaiting approval		0	0
CALC-328	dummy story		Trivial		To Do		1	1
CALC-319	dummy story		Trivial		To Do		0	0
CAR-22	The infrastructure should	interfaces	Major	Xpand IT Admin	In Progress		2	2
CAR-17	Object Detection	detection	Major		Open		0	0
CAR-4	Sensors should withstand	sensors	Medium		Open	04:20:09	0	1
CAR-3	Sensors	sensors	Critical		Open		1	0
CAR-2	Sensors should support a	sensors	Major		In Progress		1	1
CAR-1	Platform and infrastructure	interfaces,sensors	Critical		Open		0	0
CALC-316	dummy story		Trivial		To Do		0	1
COM-5	Checkout experience	Checkout	Critical		To Do		0	0
COM-4	Details page	Listing	Trivial		Testing		1	1
COM-3	Listing page	Listing	Medium		To Do		0	1
COM-2	Landing page	Main	Trivial		In Progress		0	0
COM-1	Login experience	Authentication	Major		In Progress		0	2
XT-420	As a user, I can login the		Trivial		To Do		0	0
CALC-272	story1		Trivial		Done	04:46:08	0	2

Key	Summary	Component	Priority	Requirement	Test Run Comment	List of Unique Test Run Assignees	Passed	Passed (%)	Failed	Failed (%)	Executing	Executing (%)	To do	To do (%)	Aborted	Aborted (%)
CYPLUG-8	Requirement Debug		Trivial	(UNCOVERED)			0	NaN	0	NaN	0	NaN	0	NaN	0	NaN
CALC-373	S1		Trivial	(NCK)		()	0	NaN	0	NaN	0	NaN	0	NaN	0	NaN
CALC-328	dummy story		Trivial	(NOTRUN)			0	0	1	100	0	0	0	0	0	0
CALC-319	dummy story		Trivial	(NCK)			0	0	0	0	0	0	4	100	0	0
CAR-22	The infrastructure should	interfaces	Major	(NCK)		[Xpand IT Admin]()	1	25	1	25	0	0	2	50	0	0
CAR-17	Object Detection	detection	Major	(UNCOVERED)			0	NaN	0	NaN	0	NaN	0	NaN	0	NaN
CAR-4	Sensors should withstand	sensors	Medium	(OK)	CAR-10 - same line as	()	2	100	0	0	0	0	0	0	0	0
CAR-3	Sensors	sensors	Critical	(NCK)			2	28.57142857	1	14.28571429	1	14.28571429	0	0	1	14.28571429
CAR-2	Sensors should support a	sensors	Major	(NCK)		()	0	0	1	20	0	0	0	0	0	0
CAR-1	Platform and infrastructure	interfaces	Critical	(NCK)			1	25	1	25	0	0	2	50	0	0
CALC-316	dummy story		Trivial	(NOTRUN)		()	0	0	0	0	0	0	1	100	0	0
COM-5	Checkout experience	Checkout	Critical	(NOTRUN)			0	0	0	0	0	0	1	100	0	0
COM-4	Details page	Listing	Trivial	(NCK)		[Xpand IT Admin]	0	0	1	100	0	0	0	0	0	0
COM-3	Listing page	Listing	Medium	(OK)		[Xpand IT Admin]	1	100	0	0	0	0	0	0	0	0
COM-2	Landing page	Main	Trivial	(UNCOVERED)			0	NaN	0	NaN	0	NaN	0	NaN	0	NaN
COM-1	Login experience	Authentic	Major	(NCK)		[Xpand IT Admin]()	0	0	6	66.66666667	0	0	3	33.33333333	0	0
XT-420	As a user, I can login the		Trivial	(UNCOVERED)			0	NaN	0	NaN	0	NaN	0	NaN	0	NaN
CALC-272	story1		Trivial	(NCK)		[Xpand IT Admin]()	5	83.33333333	1	16.66666667	0	0	0	0	0	0

How to use

This report can be generated from different places/contexts, including:

- Issue view screen
- Issue search page (main search page or as a bulk operation)



Learn more

General information about all the existing places available to export from and how to perform it is available on the Exporting ([DC](#), [Cloud](#)) page.

Source data

This report is applicable to:

- 1 or more Story or Epic issues

Output format

The standard output format is .XLSX so you can open it in Microsoft Excel, Google Sheets, and other tools compatible with this format. From those tools, you can generate a .CSV file.

Report assumptions

The template has a set of assumptions that you have to make sure your Jira/Xray environment complies with:

1. Issue types having the name: "Story", "Epic".
2. Direct issue links existing between requirements and tests (otherwise you get blank cells for most metrics related to the testing effort).
3. Time calculations require certain statuses or logging (see the "Understanding the report" section below).

If any of these assumptions is not met, you need to update the template or the environment accordingly.

Usage examples

Export Stories of your project

1. from the Issue Navigator/Search, search by the issueType (i.e., "Story") from your project (e.g., "BOOK") and then use bulk export or ExportDocument Generator

example of JQL expression to use

```
project = "BOOK" and issuetype="Story" order BY created DESC
```

Export Epics associated with a given fix version

1. from the Issue Navigator/Search, search by the release (i.e., "fixVersion") of your project (e.g., "EWB") and then use bulk export or ExportDocument Generator

example of JQL expression to use

```
project = "BOOK" and issuetype= "Epic" and fixVersion=1.2 order BY created DESC
```

Export Stories obtained on a given Test Environment

1. from the Issue Navigator/Search, search by Stories assigned to that Test Environment (e.g., "chrome") and then use bulk export or ExportDocument Generator

example of JQL expression to use

```
project = "BOOK" and issuetype= "Story" and testEnvironments = chrome order BY created DESC
```

Export Story from its detailed view

1. open the Story issue and export it using this template

Understanding the report

The report shows information about the Requirements in a list form.

Layout

The report is composed of two sheets - "TestingEffortOverview" and "DetailedExecutionBreakdown" - with the information on the "Requirements" (by default, Stories and Epics). Each sheet will present a line per each Requirement.

The DC version has 2 extra sheets with graphs based on the "DetailedExecutionBreakdown" tab (variables needed for the graphs are not yet available in Cloud).

"TestingEffortOverview" sheet

Column	Notes
Requirement Key	Issue key of the requirement
Summary	Summary of the requirement
Component	Component(s) of the requirement
Priority	Priority of the requirement
Assignee	Assignee of the requirement
Workflow Status	Workflow status of the requirement
Total Workflow Lifetime	The difference between "Created" and "Resolved" dates. If resolution is empty, the calculation will fail.
Total Number of Defects	Total number of linked defects (directly or through linked Tests)
Number of Unique Test Run Assignees	Number of unique assignees across all the associated test runs. "Unassigned" will count as 1.
Total Number of Test Runs	Total number of associated test runs
Total Number of Failed Test Runs	Total number of associated test runs with the "Fail" status
Total Number of Test Runs with Comments	Total number of associated test runs that have comments (regardless of the status)
Total Time Spent in Execution	Total elapsed time calculated from "Started On" and "Finished On" fields from test runs
Total Number of Linked Tests	Sum of the 4 columns below
Total Number of Linked Manual Tests	Total number of linked tests with "manual" type (regardless of the link type)

Total Number of Linked Cucumber Tests	Total number of linked tests with "cucumber" type (regardless of the link type). By default, "cucumber" is defined as a list of 'cucumber,gherkin,behave,behat'. See the customization section below.
Total Number of Linked Exploratory Tests	Total number of linked tests with "exploratory" type (regardless of the link type)
Total Number of Linked Non-Exploratory Generic Tests	Total number of linked tests with "generic" type (regardless of the link type)
Total Time Specifying Tests (in minutes)	Total aggregated time calculated from logged time across test runs. If time is not being logged, the calculation will need to be adjusted based on "Created"/"Resolved" dates (see "Total Workflow Lifetime" column) or similar date fields.

"DetailedExecutionBreakdown" sheet

Column	Notes
Key	Issue key of the requirement
Summary	Summary of the requirement
Component	Component(s) of the requirement (repeated on this tab in the DC version only to support graphs)
Priority	Priority of the requirement (repeated on this tab in the DC version only to support graphs)
Requirement Status (DC) / TestRunStatus (Cloud)	DC: overall requirement coverage status Cloud: list of test run statuses
Test Run Comments	List of test run comments
List of Unique Test Run Assignees	List of unique assignees across test runs. "Unassigned" will be represented as "[]"
Total Number of Defects	Test Plan(s) linked to the Test Execution
Number of Unique Test Run Assignees	Defects linked to the Test Execution (at either the test run or the test step level)
Passed	Number of runs in the passed status.
Passed (%)	Percentage of runs in the passed status.
Failed	Number of runs in the failed status.
Failed (%)	Percentage of runs in the failed status.
Executing	Number of runs in the executing status.
Executing (%)	Percentage of runs in the executing status.
To do	Number of runs in the to do status.
To do (%)	Percentage of runs in the to do status.
Aborted	Number of runs in the aborted status.
Aborted (%)	Percentage of runs in the aborted status.

Customizing the report

The common customization actions are:

- adding/removing columns
- changing the level of detail in the columns

As this report is column-based, if some columns are not relevant to you, you should be able to delete them. Make sure that no temporary variables are created in the cells of those columns that are used in other subsequent columns.

Exercise 1: add a custom issue type that belongs to Requirements in your configuration

At the top of the template, locate this line

```
#{if (%{'${IssueTypeName}'.equals('Epic') || '${IssueTypeName}'.equals('Story')}}}
```

Add, remove, or rename types in brackets after "equals", for example

```
#{if (%{'${IssueTypeName}'.equals('Story') || '${IssueTypeName}'.equals('Request') || '${IssueTypeName}'.equals('FeatureIdea') })}
```

Exercise 2: track custom test types

On the "TestingEffortOverview" tab, columns O-R provide the breakdown by 4 common test types. You can add more columns or modify the code in one of the existing ones, depending on which type you are interested.

For example, if you have a custom "Robot" type, you can add it to the first line of the "Non-exploratory Generic" column, then the conditional line below will automatically account for the new entry, no further changes are needed.

```
${set(genericTestTypes, 'generic, Robot')}  
...  
#{if (%{'${genericTestTypes}', '.indexOf('${Links[j]}.Test Type', '.toLowerCase()) >= 0')}}
```

Keep in mind that if you add columns, you will also need to update the formula in the "Total Number of Linked Tests" column.

You can further finetune the content and formatting via JavaScript, you can find more useful snippets in [this tutorial for Document Generator \(DC, Cloud\)](#)

Performance

Performance can be impacted by the information that is rendered and by how that information is collected/processed.

The number of Test Executions and Test Runs, depending on scenarios, can be considerably high, especially with CI/CD. As this report sums up quite a lot of information, please use it wisely.