

QA Test Plan

Template

Table of Contents

Table of Contents	1
Overview	2
Test Items	2
Scope	2
Testing Process	3
Testing Flow	3
Testing Process Overview	4
Testing Schedule	4
Roles and Responsibilities	4
Test Strategy	5
Testing Types	5
Test Environments	5
Testing Tools	6
Measures and metrics	6
Testing Analysis and Design	6
Test Execution	6
Bugs Found	7



Document Owner	Wizeline
Revision	1
Last updated	
Owner	
Collaborators	

Overview

<Here you will define the main purpose of the document, define the audience and write a brief introduction describing the project objectives, then the document contents and their objectives.

<The project> will do something, and will be resolving these problems, etc ...

This document is aimed to be used...

This document contains....

>

Test Items

<Here you will enlist the items that will be tested during the testing process.

- *Front end module*
- *Backend module*
- *...*
- *...>*

Scope

<Here you can define the testing scope in terms of test types, you can also define the operating systems, browsers, devices or screen resolution that will be tested.

In Scope:

- *Feature Testing*



- *Smoke Testing*
- *...*
-

Partially in Scope

- *Cross Browser Compatibility*
- *Integration Testing*

Out of scope:

- *Accessibility Testing*
- *Performance Testing*
- *Security Testing*
- *....*

Operating Systems:

- *Mac OS 10.13.6*
- *...*

Browsers:

- *Chrome <version number>*
- *Firefox <version number>*
- *... >*

Testing Process

Testing Flow

<Here you can describe the different testing activities that will be executed during the sprint. You can use following activities as an example:

- *Test case design Phase*
- *Test case execution Phase*
- *Test automation Phase*
- *Bug testing Phase*



You can explain the prerequisites and the results for each Phase, for example:

Test Case design Phase: Phase description

Entry Criteria:

- *Review the sprint backlog*
- *Gather the requirements*
- *Define the features to be tested*

Exit Criteria:

- *Breakdown the features to be tested*
- *Create the test cases for each feature*
- *Estimate the time for testing*
- *Prioritize the features to be tested*

>

Testing Process Overview

<Here you can illustrate the whole testing flow with a JIRA workflow diagram, a numbered list or whatever helps to understand the testing process from the beginning to the end of one sprint. >

Definition of done

<It is good to have a definition of done since the beginning of the project, you will be enlisting here all the criteria that has to be met in order to consider a User Story finished. For example:

- *Acceptance Tests identified*
- *Code Committed to QA environment*
- *Unit Tests passed*
- *Code Peer Reviewed*
- *Acceptance Test passed*
- *Features documented*
- *Scenarios are automated >*



Testing Schedule

<Here you can include a Sprint week diagram that shows how the testing is scheduled for each sprint. You can specify which days the manual testing will be executed, when the Regressions are going to be executed and if you are automating, when the automation is going to be developed. >

Roles and Responsibilities

<Here you will mention every role involved in the Test Strategy:

Role	Responsibilities
------	------------------

>

Test Strategy

Testing Types

<Here you will define each one of the Testing Types that were mentioned in the **Scope** section. You can use a bullet list or a table for each Testing Type. For example:

Feature Testing

Definition	List what is your understanding of Feature Testing for your project.
-------------------	---



Participants	<i>List the names of individuals/departments who would be responsible.</i>
Methodology	<i>Describe how the testing will be conducted. How will the testing activity take place?</i>

If different Testing types are going to be executed depending on the environment, you can specify it here. >

Test Environments

<Here you will enlist the environments where the testing will be executed. You can use a table:

Environment	URL	Description
-------------	-----	-------------

>

Testing Tools

<Here you will enlist all the tools you will be using to perform the testing. You can mention from the tracking tools such as JIRA, Confluence, etc. to the tools you will be using to perform API testing, UI Automation frameworks, CI tools, performance testing tools. You can use a table:

Tool	Description
------	-------------



>

Measures and metrics

<Here you will describe the deliverables QA team will deliver and when they will be delivered>

Testing Analysis and Design

<Some suggested metrics:

- *Test scenarios vs of test cases*
- *Test cases planned vs test tests ready for execution*
- *Estimated planning time vs actual time >*

Test Execution

<Some suggested metrics:

- *Test cases planned vs test case executed*
- *Number of passed, failed, and blocked test cases >*

Bugs Found

<Some suggested metrics:

- *Number of bugs opened vs closed*
- *Number of bugs closed vs reopened*
- *Bug distribution >*

-

