Test Plan Template

By - **Pankaj Gupta**

What is a Test Plan?

A test plan is a detailed document that outlines the test strategy,Testing objectives, resources (manpower, software, hardware) required for testing, test schedule, test estimation and test deliverables.

The test plan serves as a blueprint to conduct software testing activities as a defined process which is minutely monitored and controlled by the test manager.

Importance of Test Plan

- Test Plan helps us determine the effort needed to validate the quality of the application under test.
- Help people outside the test team such as developers, business managers, customers understand the details of testing.
- Important aspects like test estimation, test scope, Test Strategy are documented in Test Plan, so it can be reviewed by Management Team and re-used for other projects.

Guidelines for writing a Test Plan

Follow the steps below to create a test plan as per IEEE 829

1. Analyze the product
2. Design the Test Strategy
3. Define the Test Objectives
4. Define Test Criteria
5. Resource Planning
6. Plan Test Environment
7. Schedule & Estimation
8. Determine Test Deliverables

TABLE OF CONTENTS

1. Test Plan Identifier 3
2. Introduction 3
3. Test Environments 3
    3.1. Browser Matrix 3
    3.2. App Matrix 4
4. Scope 4
    4.1. Features to be tested 5
    4.2. Features not to be tested 5
5. Approach 5
6. Pass/ Fail Criteria 5
7. Suspension Criteria 5
8. Test Deliverables 6
9. Testing Tasks 6
10. Environmental Needs 6
11. Roles & Responsibilities 6
12. Schedule 6
13. Risks and Contingencies 7
14. Assumptions & Dependencies 7
15. Tools 7
16. Communications 7
17. Approval 8

1. Test Plan Identifier:

Test Plan Identifier is a unique number to identify the test plan.

Example: RnR_0001

2. Introduction:

Introduction or summary includes the purpose and scope of the project

Example: The objective of this document is to test the functionality of the ‘Rewards and Recognition Program’

3. Test Environments:

A list of test items which will be tested.

Example: Testing should be done on both front end and back end of the application on the Windows/Linux environments.

3.1 Browser Matrix

Here we list down the different browsers and whether the testing is being performed on them or not. Also, we can list the versions of the browsers (example- IE 11)

Browser	Testing performed
Mozilla Firefox	Yes/ No
Google Chrome	Yes/ No
Safari	Yes/ No
Internet Explorer 11	Yes/ No
Internet Explorer 10	Yes/ No
Microsoft Edge	Yes/ No

3.2 App Matrix

Here we list down the apps and native browsers on various platforms and whether the testing is being performed on them or not. Also, we can list the OS version (example- iOS 11), versions of the apps (example- v1.1.2) and the browsers on the apps and their versions (example- Safari 12.2.1)

Browser	Testing performed
iOS	Yes/ No
Android	Yes/ No
Safari on iPhone	Yes/ No
Google Chrome on Android	Yes/ No

4. Scope:

Scope defines the features, functional or non-functional requirements of the software that will be tested.

Defining the scope of your testing project is very important for all stakeholders. A precise scope helps you-

- Give everyone a confidence & accurate information of the testing you are doing
- All project members will have a clear understanding about what is tested and what is not

4.1 Features To Be Tested:

In this section, we list out all the features that will be tested within the project.

Example: The features which are to be tested are Upload beneficiaries, Allocating budget, Reward Activity, Rewards History, etc.

4.2 Features Not To Be Tested:

In this section, we list out the features which are not included in the project.

Example: Only RnR will be tested in corporate dashboard and not optima.

5. Approach:

The overall strategy of how testing will be performed. It contains details such as Methodology, Test types, Test techniques etc.,

Example: Agile testing process is followed for testing RnR

6. Pass/Fail Criteria:

In this section, we specify the criteria that will be used to determine pass or fail percentage of test items.

Example: All the major functionality of the application should work as intended and the pass percentage of test cases should be more than 95% and there should not be any critical bugs.

7. Suspension Criteria:

In this section, we specify when to stop the testing.

Example: If any of the major functionalities are not functional or system experiences login issues then testing should suspend.

8. Test Deliverables:

List of documents need to be delivered at each phase of testing life cycle. The list of all test artifacts.

Examples: Test Cases, Bug Report

9. Testing Tasks:

In this section, we specify the list of testing tasks we need to complete in the current project.

Example: Test environment should be ready prior to test execution phase. Test summary report needs to be prepared.

10. Environmental Needs:

List of hardware, software and any other tools that are needed for a test environment.

11. Roles & Responsibilities:

We specify the list of roles and responsibilities of each test tasks.

Example: Test plan should be prepared by Test Lead. Preparation and execution of tests should be carried out by testers.

12. Schedule:

Complete details on when to start, finish and how much time each task should take place.

Example: Perform test execution – 120 man-hours, Test Reporting – 30 man-hours

13. Risks and Contingencies:

In this section, we specify the probability of risks and contingencies to overcome those risks.

Example: *Risk* – In case of a wrong budget estimation, the cost may overrun.
Contingency Plan – Establish the scope before beginning the testing tasks and pay attention in the project planning and also track the budget estimates constantly.

14. Assumptions:

In this section we list the assumptions that have been made during the preparation of this plan. Example: User is uploading a file with ".csv" extension

15. Tools:

In this section, we make a list of tools required for testing related activities in the project like-

- Bug Tracking Tool (example: Jira, Bugzilla etc.)
- Automation Tools (example: Selenium, Appium etc.)

16. Communications:

In this section, we list down the contact details, like email id, of the people involved in the project. Example-

Name	Designation	Email ID
Pankaj Gupta	QA Lead	pankaj@******.com

17. Approvals:

Who should sign off and approve the testing project

Example: Project manager should agree on completion of the project and determine the steps to proceed further.

—---------- End —----------

BONUS

Bonus with 90% OFF Offer Applied Coupon for more learning

Get Complete Software Testing Knowledge by enrolling in below courses.

Software Manual Testing
https://www.udemy.com/course/software-manual-testing-course/?referralCode=14CF21D04D3C99F3C14E

Software Testing Live Project
https://www.udemy.com/course/software-manual-testing-live-project/?referralCode=1F95EA84C42056B4B1D2

Complete SQL with Practicals
https://www.udemy.com/course/complete-sql-course-qadata-analyticsbusiness-intelligence/?referralCode=D8A52876E0D754934D05

Refer Next Page → Become Automation QA/SDET

Complete Automation QA / SDET Packages - Become Automation Engineer

Java - Selenium
https://www.udemy.com/course/selenium-webdriver-with-java-testng-maven-git-jenkins/?referralCode=50B844906A246649DFDB

Python - Selenium
https://www.udemy.com/course/selenium-using-python/?referralCode=3130F658A75C0F309314

Java - RestAssured : API Testing
https://www.udemy.com/course/api-webservices-automation-testing/?referralCode=B73B2600281FB8674B49

Python Request : API Testing
https://www.udemy.com/course/api-testing-with-python/?referralCode=C6FDD6F97830D9ABA1EA

GIT / Bitbucket for Daily SDET Job
https://www.udemy.com/course/learn-git-from-basic/?referralCode=EA46C26F3FDC939B1835

Learn Complete Java
https://www.udemy.com/course/java-for-beginner/?referralCode=706A7AAAB9849C8DC1F7

Learn Complete Locator with multiple Practicals to be expert in UI Automation Selenium & Appium
https://www.udemy.com/course/learn-element-locators-css-selector-and-xpath-from-scratch/?referralCode=524C2EE783215169E991