**TEST PLAN**

In software development, a test plan is a detailed document of test strategies It provides information on the strategy, objectives, schedule, resources, estimation, and deliverables used in the testing process. The Test Plan ensures that software testing performs all necessary activities appropriately, making it a key element.

It also serves as a guide for the testing manager to monitor and control the progress of the testing activities. The test lead, manager, and engineer prepare the Test Plan.

**six steps to create an efficient test plan:**

* Define the release scope:
* Schedule timelines:
* Define test objectives:
* Determine test deliverables:
* Design the test strategy:
* Plan test environment and test data

Test plan controlled and written by test managers or leads. It can be shared with stack holders, Dev teams and business analyst.

1. Make sure all stakeholders agree on the **scope** of the plan before beginning development. This will ensure that everyone involved is aware of what needs to be accomplished and their individual roles and responsibilities.

2. Include a **timeline** with deliverables and deadlines to ensure that all tasks are completed on time. This will help keep the project on track and avoid any delays.

3. A test **objective** is a reason or purpose for designing and executing a test.

4. Test **deliverables** are the products of testing that help track testing progress. Deliverables should meet your projects and client’s needs, be identified early enough to be included in the test plan and be scheduled accordingly.

5. Test **strategy** helps determine test cost, test effort, and which features will be in-scope (planned to be tested) versus out-of-scope (not planned to be tested).

6. **Planning** a test environment guarantees precise and robust testing. The test environment includes hardware, software, and network configurations for software testing.

**How To Prepare an Effective Test Plan?**

Follow these necessary steps to prepare an effective test plan:

First up, you have to define your objectives like a boss. Be clear about what exactly you’re trying to achieve with this testing – is it finding bugs, assessing performance, or verifying functionality?

Next, break down the application into manageable components and determine which areas need testing.

Then, design realistic test scenarios that cover all possible user interactions and edge cases. Remember to prioritize these scenarios based on their criticality to save time during execution.

Select the appropriate testing techniques and methodologies that suit your project’s requirements. This will ensure you’re testing efficiently and effectively.

Lastly, create a master schedule including all activities and timelines for each testing phase, from preparation to execution and reporting.

**TEST PLAN MAINLY FOCUS ON: [THE TEMPLATE]**

* What to test?
* What not to test?
* How to test?
* When to test?
* Who will do what test?

It is a dynamic document, and we should always keep it up to date.

|  |
| --- |
| **Test Plan** **Title** Prepared by: John Doe |
| 1. **Introduction** •Executive summary (This should be kept brief) |
| 2. **Testing Resources** •Tester’s name and role |
| 3. **Scope of Testing** •In scope: Modules that are to be tested •Out of scope: Modules that are not to be tested |
| 4. **Testing approaches** •Testing approach and methodology •Types of testing to be performed (e.g., functional, performance, security, usability) |
| 5. **Test Schedule** •Timeline for each testing phase |
| 6. **Risks & Issues** •Risks associated with the testing process •Mitigation strategies for identified risks |

**One-page test plan template**

[**https://www.testrail.com/blog/create-a-test-plan/**](https://www.testrail.com/blog/create-a-test-plan/)

**https://testsigma.com/guides/test-plan-template/**

https://iteratorstesting.com/blog/how-to-write-test-plan