**Test plan**

a test plan defines your testing team’s test strategy, goals, and scope, which ultimately work together to ensure that all your software components are tested sufficiently before a release.

six steps to create an efficient test plan:

**1.Define the release scope**

it’s important to define the scope of testing for your release. This means defining the features or functions that need to be included in the release,

considering any constraints and dependencies that can affect the release, and determining what type of release it is.

**2.Schedule timelines**

Specify release deadlines to help you decide your testing time and routine.

**3.Define test objectives**

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

These objectives ultimately help guide and define the scope of testing activities.

* Functional testing objectives
* Performance testing objectives
* Security testing objectives
* Usability testing objectives

**4.Determine test deliverables**

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.Design the test strategy**

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.Plan test environment and test data**

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

**Follow these procedures to set up the test environment:**

* Determine your hardware and program requirements: Select test environment devices and software, including operating systems, browsers, databases, and testing tools.
* Install the required software: Once prerequisites are established, install the necessary tools on the test environment. This may require setting up a separate server to host the application and installing a database management system or other tools.
* Configure the network: Make sure that firewall protocols, IP addresses, and DNS settings, among other network configurations, are identical between the test and production environments.
* Create the test data: Prepare the test material for the application’s testing. Test data can be created manually with data from the production environment, retrieved from an existing production environment and database, or created via automated Data Generation Tools.
* Access the builds: Ensure that the builds that the testers will be testing are accessible. One example is setting up a file-sharing or version control system to allow testers access to the most current builds.
* Verify the test environment: After setting it up, check that your test environment fulfils the requirements.