Testing documentation:

Test Strategy->test plan->Test scenario->test condition->test case

Test Strategy:

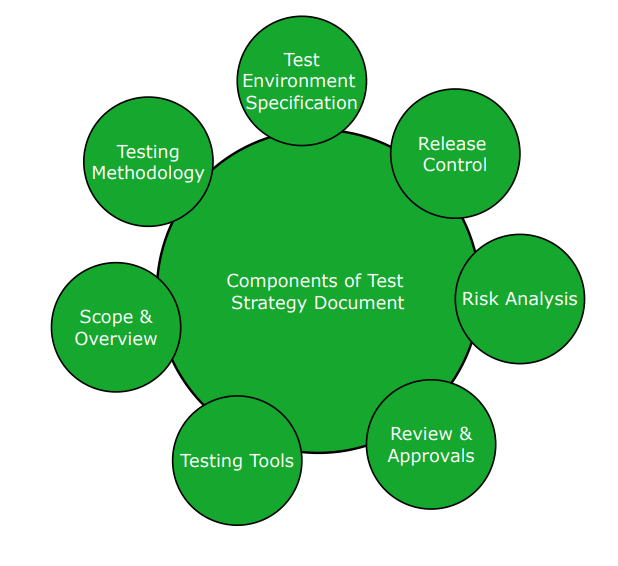
Test Strategy is a set of guidelines that explain the test design and determine how testing needs to be done.

The Test strategy document is a high-level document that outlines the testing technique used in the [Software Development Life Cycle](https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/) and confirms the test kinds or levels that will be performed on the product. One can’t change the test strategy once it’s been written, and it’s been accepted by the Project Manager and development team.

A test strategy is a plan for defining an approach to the Software Testing Life Cycle (STLC). In addition, the test strategy provides the following details, which are required while writing the test document:

* What technique must be used in addition to this?
* Which of the modules will be examined?
* What criteria apply for entry and exit?
* What kind of testing is necessary? Likewise

Test strategy document:



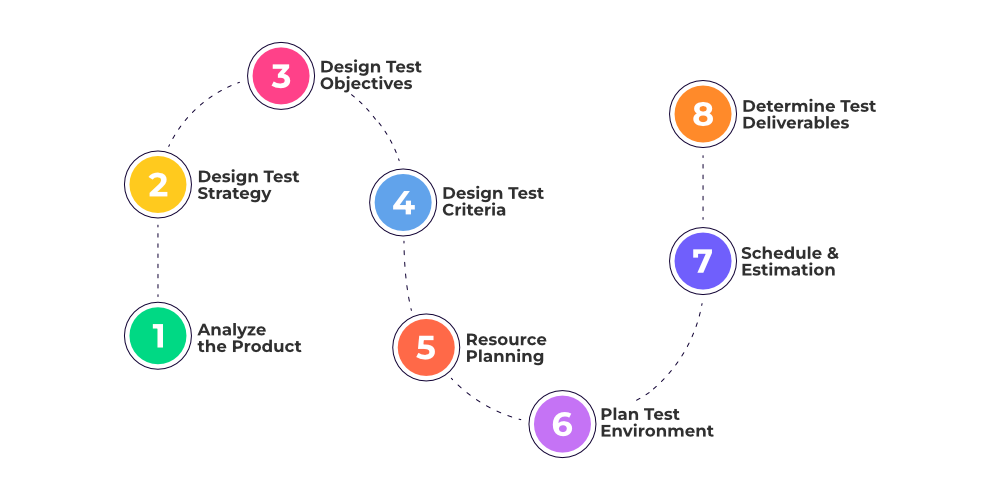
Test Plan:

* A Test Plan can be defined as a document that defines the scope, objective, and approach to test the software application.
* The Test Plan is a document that lists all the activities in a QA project, schedules them, defines the scope of the project, roles & responsibilities, risks, entry & exit criteria, test objective, and anything else that you can think of.
* A test plan is a detailed document which describes software testing areas and activities. It outlines the test strategy, objectives, test schedule, required resources (human resources, software, and hardware), test estimation and test deliverables.
* The test plan is a base of every software's testing. It is the most crucial activity which ensures availability of all the lists of planned activities in an appropriate sequence.
* The test plan is a template for conducting software testing activities as a defined process that is fully monitored and controlled by the testing manager.

**Example:** The Test Plan gives information about who is going to test at what time. **For Example,** Module 1 is going to be tested by “X tester”. If tester Y replaces X for some reason, the test plan has to be updated.

| **S No.** | **Test Plan** | **Test Strategy** |
| --- | --- | --- |
| 1. | It’s developed from a set of software requirements (SRS). | It comes from a Business Requirement paper (BRS). |
| 2. | The test lead or manager is in charge of preparing it. | The project manager or the business analyst creates it. |
| 3. | The test plan’s components include the test plan’s id, features to be tested, test techniques, testing tasks, features pass or fail criteria, test deliverables, responsibilities, and timetable, among others. | The components of a test strategy include objectives and scope, documentation formats, test processes, team reporting structure, client communication strategy, and so on. |
| 4. | After the requirements have been approved, the test plan is written. | The test strategy comes first, followed by the test plan. |
| 5. | The test plan should be simple and straightforward. | The test approach serves as a general guide for the project at hand. |

How to create a test plan:



Types of test plan:

3 types:

### **Master Test Plan**

Master Test Plan is a type of test plan that has multiple levels of testing. It includes a complete test strategy.

### **Phase Test Plan**

A phase test plan is a type of test plan that addresses any one phase of the testing strategy. For example, a list of tools, a list of test cases, etc.

### **Specific Test Plans**

Specific test plan designed for major types of testing like security testing, load testing, performance testing, etc. In other words, a specific test plan designed for non-functional testing.

Test plan document:



**1. Objective:** It describes the aim of the test plan, whatever the good process and procedure they are going to follow to give quality software to customers. The overall objective of the test is to find as many defects as possible and to make software bug-free.

The test objective must be broken into components and sub-components. In every component following activities should be performed.

* List all the functionality and performance to be tested.
* Make goals and targets based on the application feature.

**2. Scope:**It consists of information that needs to be tested concerning an application. The scope can be divided into two parts:

* **In-Scope:** The modules that are to be tested rigorously.
* **Out Scope:** The modules that are not to be tested rigorously.

**Example:**In an application A, B, C, and D features have to be developed, but the B feature has already been designed by other companies. So the development team will purchase B from that company and perform only integrated testing with A, B, and C.

**3. Testing Methodology:**The methods that are going to be used for testing depend on application to application. The testing methodology is decided based on the feature and application requirements.

Since the testing terms are not standard, one should define what kind of testing will be used in the testing methodology. So that everyone can understand it.

**4. Approach:** The approach of testing different software is different. It deals with the flow of applications for future reference. It has two aspects:

* **High-Level Scenarios:** For testing critical features high-level scenarios are written. For Example, login to a website, and book from a website.
* **The Flow Graph:**It is used when one wants to make benefits such as converging and merging easy.

**5. Assumption:**In this phase, certain assumptions will be made.

**Example:**

* The testing team will get proper support from the development team.
* The tester will get proper knowledge transfer from the development team.
* Proper resource allocation will be given by the company to the testing department.

**6. Risk:**All the risks that can happen if the assumption is broken. For Example, in the case of wrong budget estimation, the cost may overrun. Some reason that may lead to risk is:

* Test Manager has poor management skills.
* Hard to complete the project on time.
* Lack of cooperation.

**7. Mitigation Plan:**If any risk is involved then the company must have a backup plan, the purpose is to avoid errors. Some points to resolve/avoid risk:

* Test priority is to be set for each test activity.
* Managers should have leadership skills.
* Training course for the testers.

**8. Roles and Responsibilities:** All the responsibilities and role of every member of a particular testing team has to be recorded.

**Example:**

* **Test Manager:**Manages the project, takes appropriate resources, and gives project direction.
* **Tester:** Identify the testing technique, verify the test approach, and save project costs.

**9. Schedule:**Under this, it will record the start and end date of every testing-related activity. For Example, writing the test case date and ending the test case date.

**10. Defect Tracking:**It is an important process in software engineering as lots of issue arises when you develop a critical system for business. If there is any defect found while testing that defect must be given to the developer team. There are the following methods for the process of defect tracking:

* **Information Capture:**In this, we take basic information to begin the process.
* **Prioritize:**The task is prioritized based on severity and importance.
* **Communication:** Communication between the identifier of the bug and the fixer of the bug.
* **Environment:** Test the application based on hardware and software.

**Example:**The bug can be identified using bug-tracking tools such as Jira, Mantis, and Trac.

**11. Test Environments:**It is the environment that the testing team will use i.e. the list of hardware and software, while testing the application, the things that are said to be tested will be written under this section. The installation of software is also checked under this.

**Example:**

* Software configuration on different operating systems, such as Windows, Linux, Mac, etc.
* Hardware Configuration depends on RAM, ROM, etc.

**12. Entry and Exit Criteria:**The set of conditions that should be met to start any new type of testing or to end any kind of testing.

**Entry Condition:**

* Necessary resources must be ready.
* The application must be prepared.
* Test data should be ready.

**Exit Condition:**

* There should not be any major bugs.
* Most test cases should be passed.
* When all test cases are executed.

**Example:**If the team member reports that 45% of the test cases failed, then testing will be suspended until the developer team fixes all defects.

**13. Test Automation:**It consists of the features that are to be automated and which features are not to be automated.

* If the feature has lots of bugs then it is categorized as Manual Testing.
* If the feature is frequently tested then it can be automated.

**14. Effort Estimation:**This involves planning the effort that needs to be applied by every team member.

**15. Test Deliverables:** It is the outcome from the testing team that is to be given to the customers at the end of the project.

Before the testing phase:

* Test plan document.
* Test case document.
* Test design specification.

During the testing phase:

* Test scripts.
* Test data.
* Error logs.

After the testing phase:

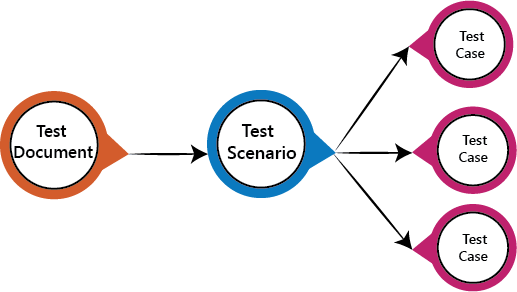
* Test Reports.
* Defect Report.
* Installation Report.

It contains a test plan, defect report, automation report, assumption report, tools, and other components that have been used for developing and maintaining the testing effort.

**16. Template:**This is followed by every kind of report that is going to be prepared by the testing team. All the test engineers will only use these templates in the project to maintain the consistency of the product.

Test Scenario:

A Test Scenario is a statement describing the functionality of the application to be tested.



## Example of Test scenarios

Here we are taking the **Gmail application** and writing test scenarios for different modules which are most commonly used such as **Login, Compose, Inbox, and Trash**

### **Test scenarios on the Login module**

* Enter the valid login details (Username, password), and check that the home page is displayed.
* Enter the invalid Username and password and check for the home page.
* Leave Username and password blank, and check for the error message displayed.
* Enter the valid Login, and click on the cancel, and check for the fields reset.
* Enter invalid Login, more than three times, and check that account blocked.
* Enter valid Login, and check that the **Username** is displayed on the home screen.

### **Test scenarios on Compose module**

* Checks that all users can enter email ides in the **To, Cc, and Bcc**.
* Check that the entire user can enter various email ids in To, Cc, and Bcc.
* Compose a mail, send it, and check for the confirmation message.
* Compose a mail, send it, and check in the sent item of the sender and the inbox.
* Compose a mail, send it, and check for invalid and valid email id (valid format), check the mail in sender inbox.
* Compose main, discard, and then check for conformation message and check-in draft.
* Compose mail click on save as draft and check for the confirmation message
* Compose mail click on close and check for conformation save as drafts.

### **Test scenarios on Inbox module**

* Click on the inbox, and verify all received mail are displayed and highlighted in the inbox.
* Check that a latest received mail has been displayed to the sender email id correctly.
* Select the mail, reply and forward send it; check in the sent item of sender and inbox of the receiver.
* Check for any attached attachments to the mail that are downloaded or not.
* Check that attachment is scanned correctly for any viruses before download.
* Select the mail, reply and forward save as draft, and check for the confirmation message and checks in the Draft section.
* Check all the emails are marked as read are not highlighted.
* Check all mail recipients in **Cc** are visible to all users.
* Checks all email recipients in **Bcc** are not visible to the users.
* Select mail, delete it, and then check in the **Trash** section.

### **Test scenario on Trash module**

* Open trash, check all deleted mail present.
* Restore mail from Trash; check-in the corresponding module.
* Select mail from trash, delete it, and check mail is permanently deleted.

Test condition:

Test conditions are the static rules should be followed to test an application.

Examples test Conditions:  
#1) Enter the country name as “India” and check for the addition of the country.  
#2) Leave blank fields and check if the country gets added.

Test case:

(test case for manual testing and test scripts for automation testing)

* A test case refers to the actions required to verify a specific feature or functionality in software testing.
* The test case details the steps, data, prerequisites, and postconditions necessary to verify a feature

Positive and negative test cases:

Positive test case eg; for login page

* Valid username and password combination successfully logs the user in.
* Successful login using a valid email address as the username

Negative test case eg;

* Entering an incorrect password for a valid username.
* Entering an incorrect username for a valid password