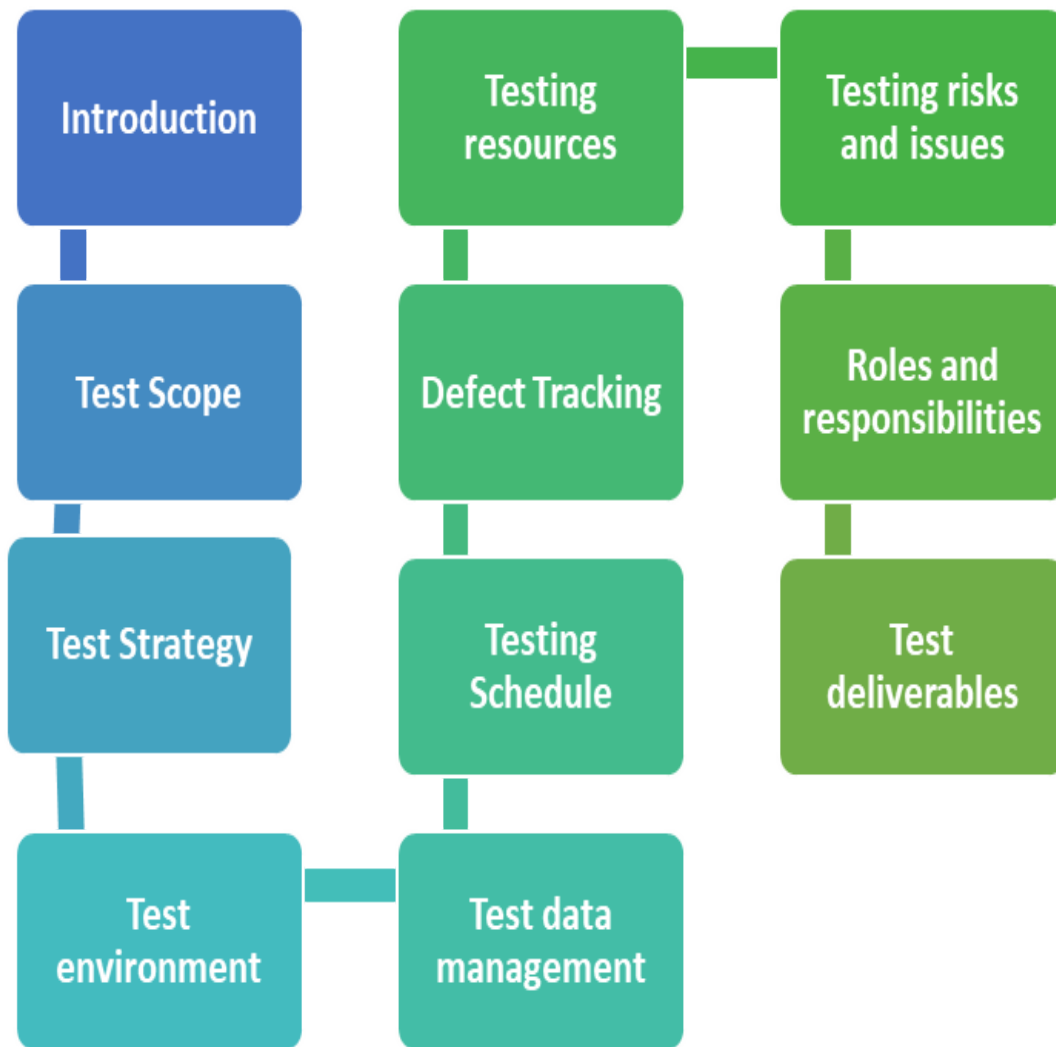


# Test Plan

|                                     |   |
|-------------------------------------|---|
| Test Plan Workflow                  | 2 |
| 1. Introduction                     | 3 |
| 2. Test Scope                       | 3 |
| 3. Test Strategy                    | 4 |
| 3.1. Feature Testing                | 4 |
| 3.2. System and Integration Testing | 4 |
| 3.3. Performance and Stress Testing | 4 |
| 3.4. User Acceptance Testing        | 4 |
| 3.5. Automated Regression Testing   | 5 |
| 3.6. Tools and technologies         | 5 |
| 4. Test environment                 | 5 |
| 5. Test Data management             | 6 |
| 6. Testing Schedule                 | 6 |
| 7. Defect Tracking                  | 7 |
| 8. Testing Resources                | 7 |
| 9. Testing risks and issues         | 7 |
| 10. Roles and responsibilities      | 8 |
| 11. Test deliverables               | 9 |

# Test Plan Workflow

A test plan is a document that outlines the approach, objectives, scope, and activities for testing a software application or system. It provides a comprehensive overview of the testing effort and serves as a guide for the testing team to ensure that all aspects of the software are thoroughly and systematically tested.



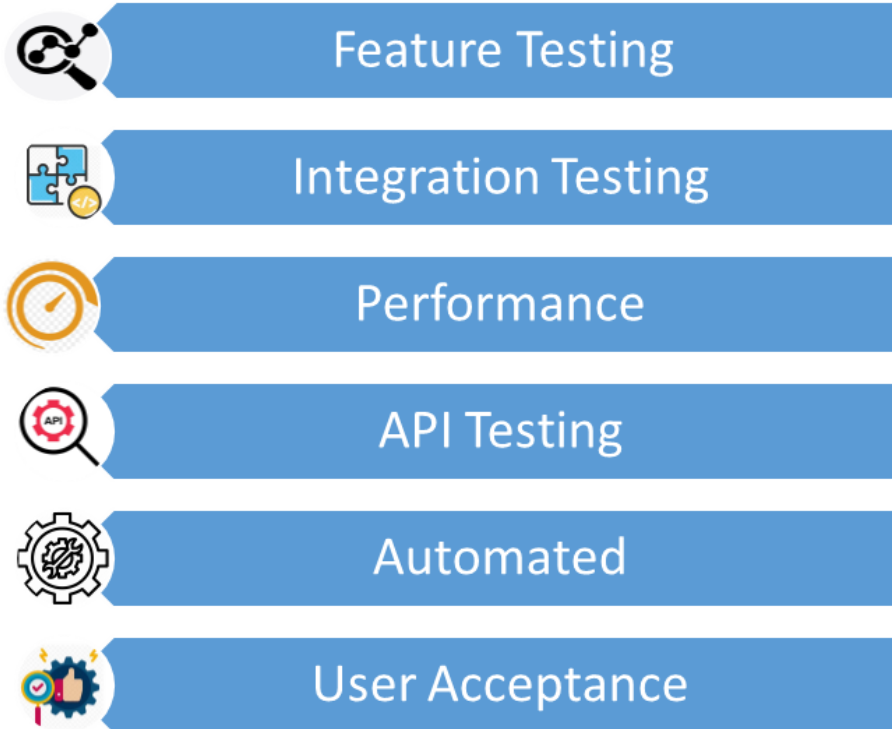
## 1. Introduction

This section describes the purpose of the test plan and provides an overview of the software system being tested. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, the degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning.

## 2. Test Scope

This section defines the scope of the testing, including what parts of the software will be tested, what types of testing will be performed, and what will not be tested. For example - Functional, UI, Responsive, and API testing will be performed and nonfunctional testing like security and performance will not be part of the scope. (Discuss with the client and understand their requirements and accordingly choose the type of testing required)

### In-Scope Testing:



### Out-of-scope testing



## 3. Test Strategy

Overall, the test strategy should provide a clear and comprehensive plan for how testing will be carried out, and how any risks or issues will be managed throughout the testing process. This section describes the overall testing approach, including the testing methods, tools, and techniques that will be used.

### 3.1. Feature Testing

**Definition:** List your understanding of Feature testing. (Feature testing is a type of software testing that focuses on testing the features or functionalities of a software application. This type of testing is used to ensure that the application meets the requirements and specifications of the user or customer.)

**Participants:** Who will be conducting Feature testing on your project? List the individuals who will be responsible for this activity.

**Methodology:** Describe how feature testing will be conducted. Who will write the test scripts for feature Testing?

### 3.2. System and Integration Testing

**Definition:** List your understanding of System Testing and Integration Testing (Individual software modules are combined and tested as a group) for your project.

**Participants:** Who will be conducting System and Integration Testing on your project? List the individuals who will be responsible for this activity.

**Methodology:** Describe how System & Integration testing will be conducted. Who will write the test scripts for Integration testing, what would be the sequence of events of System & Integration Testing, and how will the testing activity take place?

### 3.3. Performance and Stress Testing

**Definition:** List your understanding of Stress Testing for your project.

**Participants:** Who will be conducting Stress Testing on your project? List the individuals who will be responsible for this activity.

**Methodology:** Describe how Performance & Stress Testing will be conducted. Who will write the test scripts for testing, what would be the sequence of events for Performance & Stress Testing, and how will the testing activity take place?

### 3.4. User Acceptance Testing

**Definition:** The purpose of the acceptance test is to confirm that the system is ready for operational use. During the Acceptance Test, end-users (customers) of the system compare the system to its initial requirements.

**Participants:** Who will be responsible for User Acceptance Testing? List the names of the individuals and their responsibilities.

**Methodology:** Describe how User Acceptance testing will be conducted. Who will write the test scripts for testing, what will be the sequence of events for User Acceptance Testing, and how will the testing activity take place?

### 3.5. Automated Regression Testing

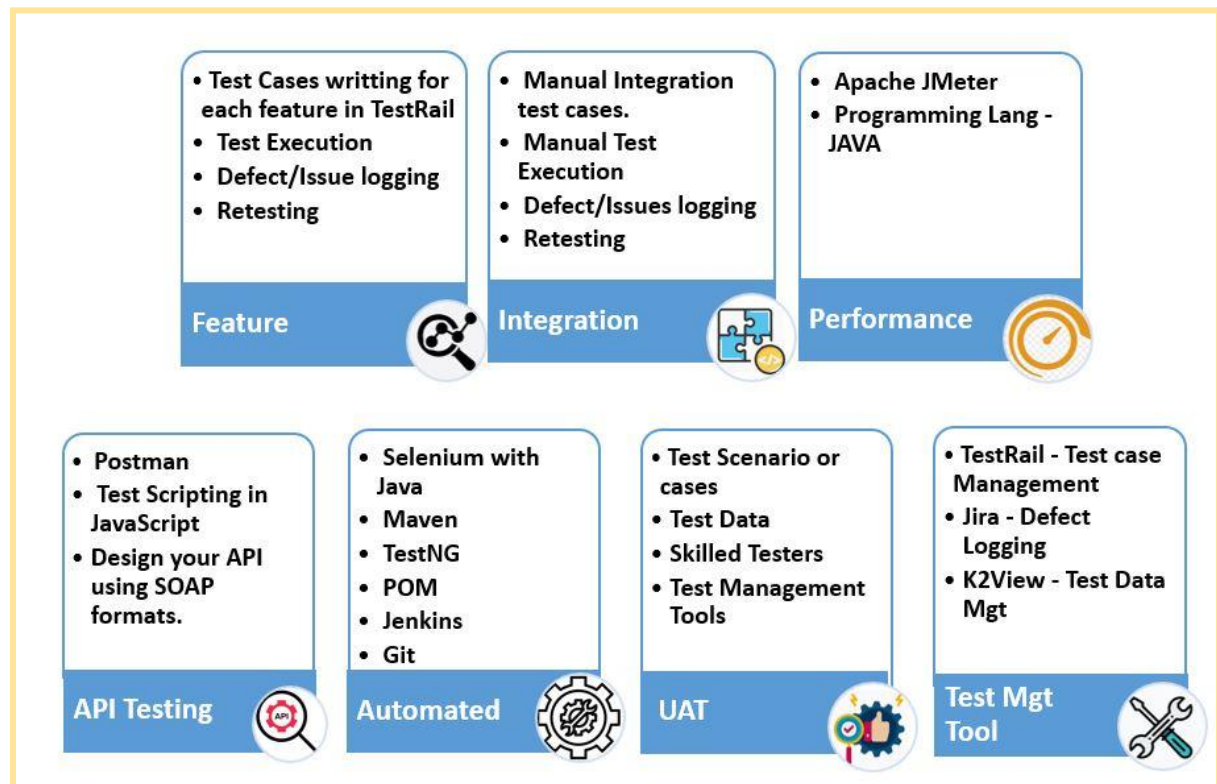
**Definition:** Regression testing is the selective retesting of a system or a component to verify that the modifications have not caused unintended effects and that system or component still works as specified in the requirements.

**Participants:** Who will be responsible for writing automation scripts? List the names of the individuals and their responsibilities.

**Methodology:** Describe how automated regression testing will be conducted. Who will write the test scripts for testing, what will be the sequence of events for automated regression Testing, and how will the testing activity take place?

### 3.6. Tools and technologies

To perform various testing activities we need a set of tools and technologies Below are the different tools and technologies we will be using to achieve different testing objectives.



## 4. Test environment

The test environment should be described in terms of the hardware, software, and network configurations required for testing. This includes any dependencies or third-party systems that will be needed.

For example:- In one of the projects there were two environments staging and production environment. In the staging environment QAs used to perform all sorts of testing like Smoke, Sanity, Regression, and Retesting. After regression is performed on the staging environment and all or maximum test cases are passed, the code will be pushed to prod for release. In the production environment, a round of sanity and smoke testing will be performed as per the need.

## 5. Test Data management

The strategy should outline how the test data will be collected, stored, and managed, including any requirements for data privacy or security. Need to work on Test data management. In a test plan, test data management involves the following steps:

- **Identification of test data requirements:** This involves identifying the types of data required for testing, such as customer information, transaction data, or user credentials.
- **Creation of test data:** This involves creating test data that meets the requirements of the test cases. For example, if the test case requires customer information, test data with valid customer information needs to be created.
- **Maintenance of test data:** Test data needs to be maintained to ensure that it remains relevant and up-to-date. This involves updating the test data when new features are added or when changes are made to the software application.
- **Security of test data:** Test data needs to be secured to prevent unauthorized access or leakage of sensitive information.

## 6. Testing Schedule

A test schedule is a document that outlines the timing and sequence of testing activities for a software development project. It typically includes the start and end dates of testing, the types of tests to be performed, the resources required for testing, and the deliverables associated with each testing phase. To create the project schedule, the Test Manager needs several types of input as below:

- **Employee and project deadline:** The working days, the project deadline, and resource availability are the factors that affected the schedule
- **Project estimation:** Based on the estimation, the Test Manager knows how long it takes to complete the project. So he can make the appropriate project schedule
- **Project Risk:** Understanding the risk helps Test Manager add enough extra time to the project schedule to deal with the risks

**Example of a test schedule:**

| Testing Activity    | Start Date | End Date   | Resources Required  |
|---------------------|------------|------------|---|
| Unit Testing        | 02/28/2023 | 03/07/2023 | 2 Test Engineers, 1 Automated Testing Tool                                |
| Integration Testing | 03/08/2023 | 03/21/2023 | 3 Test Engineers, 1 Automated Testing Tools                               |
| API Testing         | 03/10/2023 | 03/21/2023 | 2 Test Engineers, 1 Automated Testing Tool                                |
| Performance Testing | 03/22/2023 | 04/04/2023 | 2 Test Engineers, 1 Automated Testing Tools, 1 Dedicated Test Environment |

|                    |            |            |  |
|--------------------|------------|------------|--|
| Acceptance Testing | 04/05/2023 | 04/18/2023 | 2 Test Engineers, 1 Automated Testing Tool, 1 Dedicated Test Environment |
| Regression Testing | 04/19/2023 | 04/26/2023 | 2 Test Engineers, 1 Automated Testing Tool                               |

By following above steps, you can create an effective test schedule that ensures comprehensive testing coverage and delivers high-quality software products.

## 7. Defect Tracking

When testing a software product, it's important to keep track of any problems or defects we find and ensure they are fixed. We call this process defect tracking. Here are some important steps to follow:

- Decide what kinds of problems are important, and how serious they are.
- Use a system to keep track of all the defects you find, so you don't forget any.
- Write down everything you know about each defect, like what happened and when it happened.
- Make sure the right people know about the defects, and assign someone to fix them.
- Keep track of how long it takes to fix each defect and make sure they are fixed before moving on to something else.
- Make sure to check that each defect is really fixed before saying it is done.

By doing this, we can make sure that the software we are working on is of good quality and works well. For bug reporting, we use “**Jira**” and below is the guideline/format to log bugs in it: [Bug Reporting Template](#)

## 8. Testing Resources

Resources are the personnel, tools, and equipment needed to execute the testing activities outlined in a test plan. These resources are typically identified during the planning phase of software testing and may include the following:

- **Test team members:** Testers and test managers who are responsible for planning, designing, and executing the tests.
- **Testing tools:** Software tools that aid in the testing process, such as automated testing tools, performance testing tools, and defect tracking tools.
- **Testing environments:** Hardware and software environments that mimic the production environment where the software will be deployed.
- **Test data:** Data that is required to execute the tests, such as customer data, test scripts, and test cases.
- **Test infrastructure:** The hardware and software infrastructure required to support the testing activities, such as servers, databases, and network connections.
- **Documentation:** The documentation required to support the testing activities, such as test plans, test cases, and test reports.

## 9. Testing risks and issues

Identifying and managing testing risks and issues is an essential part of a test plan. Risks are potential problems that may occur during the testing process, while issues are actual problems that have occurred and require resolution. Below are some examples of testing risks and issues that should be considered in a test plan:

- **Test data availability:** If the test data required to execute the tests is not available or is inaccurate, it can impact the testing process and the validity of the results.
- **Test environment availability:** If the required testing environments are not available or are not set up properly, it can delay the testing process and impact the quality of the results.
- **Test case coverage:** If the test cases do not cover all the required scenarios, it can result in insufficient testing and potential defects being missed.
- **Test automation challenges:** If the automated testing tools are not set up properly or are not suitable for the project, it can impact the efficiency and effectiveness of the testing process.
- **Resource availability:** If the testing team does not have the necessary resources, such as time, personnel, or tools, it can impact the testing process and the quality of the results.
- **Defect management:** If defects are not managed properly, it can impact the testing process, the release of the software, and the user experience.

## 10. Roles and responsibilities

In a test plan outline the roles of the individuals involved in the testing process and define their responsibilities throughout the testing lifecycle. The following are some examples of roles and responsibilities that may be included in a test plan:



- **Test Manager:** The person responsible for managing the testing process, including planning, executing, and reporting on testing activities. The Test Manager should ensure that the testing objectives are achieved and that the test plan is executed within the required timelines and budget.
- **Test Lead:** The person responsible for leading the testing team and coordinating testing activities. The Test Lead should ensure that testing

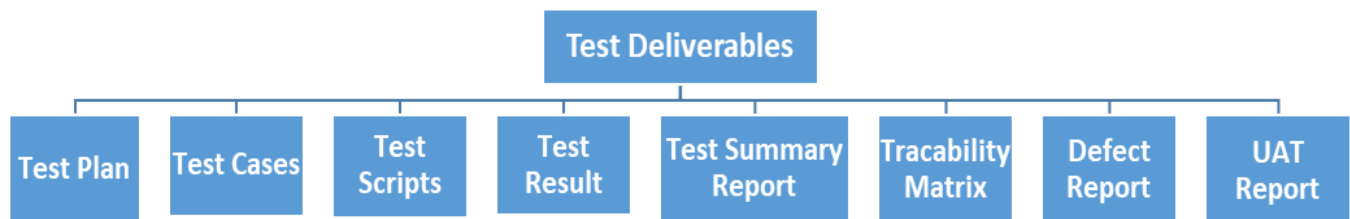


activities are executed as per the test plan and that the testing team is aligned with the testing objectives.

- **Test Analyst:** The person responsible for designing and executing test cases, reporting and tracking defects, and analyzing test results.
- **Automation Engineer:** The person responsible for developing and maintaining automated test scripts and ensuring that the automated testing tools are used effectively.
- **Subject Matter Expert (SME):** The person responsible for providing domain-specific knowledge and expertise, such as business requirements or regulatory compliance, to support testing activities.
- **Developers:** The individuals responsible for developing and delivering the software being tested. They may be required to provide support to the testing team in debugging defects or in addressing testing-related issues.
- **Business Owners / Stakeholders:** The individuals responsible for owning and validating the software being tested. They may be required to provide feedback and sign off on testing activities and results.

## 11. Test deliverables

These are the outputs or documents that are created during the software testing process. These deliverables are used to communicate the results of testing to stakeholders, including developers, project managers, and customers. Here are some examples of test deliverables.



- **Test plan:** A document that outlines the approach, objectives, scope, and schedule for testing a software application.
- **Test cases:** A set of detailed instructions or steps that describe how to test a specific aspect of the software application. Test cases typically include the expected results, preconditions, and any necessary inputs.
- **Test scripts:** Automated scripts used to execute test cases. Test scripts can be created using a variety of tools and programming languages.
- **Test Results:** A summary of the test results, including any defects or issues found during testing. Test results can be communicated through various means, such as test reports or defect logs.
- **Test summary report:** A document that provides an overview of the testing activities and results. This report summarizes the number of test cases executed, the number of defects found, and any recommendations for future testing.
- **Traceability matrix:** A document that links test cases to requirements or specifications. This helps ensure that all requirements are tested and that there are no gaps in testing coverage.
- **Defect report:** A document that describes any defects or issues found during testing, including the severity, priority, and steps to reproduce the defect.
- **User acceptance test (UAT) report:** A document that summarizes the results of UAT, which is testing conducted by end-users to ensure that the software application meets their needs and expectations.

