



## PRACTICES GUIDE

### TEST PLAN

Issue Date: <mm/dd/yyyy>  
Revision Date: <mm/dd/yyyy>

## Document Purpose

The purpose of this document is to provide guidance on the way to utilize the Test Plan. This Practice Guide provides an overview, lays out the necessary requirements, describes the best practices, activities, attributes, and related templates, tools, information, and key terminology of industry-leading project management practices.

## Background

The Department of Health and Human Services (HHS) Enterprise Performance Life Cycle (EPLC) is a framework to enhance Information Technology (IT) governance through rigorous application of sound investment and project management principles and industry's best practices. The EPLC provides the context for the governance process and describes interdependencies between its project management, investment management, and capital planning components. The EPLC framework establishes an environment in which HHS IT investments and projects consistently achieve successful outcomes that align with Department and Operating Division goals and objectives.

The EPLC Design Phase initiates the Test Plan, where a final draft test plan is prepared. The Test Plan describes the test cases and test environment specifications, and includes a Requirements Traceability Matrix that maps requirements to the specific tests to be conducted in the Test Phase. This final draft Test Plan will be used in the Development Phase to test components as they are built and integrated. This ensures that all test cases will be adequately evaluated and executed, and system tested to ensure requirements are met.

## Practice Overview

Test planning is the practice of preparing for the testing phase of product development to ensure that what is delivered to the client indeed satisfies the requirements as agreed upon in the requirements and design specification documents. The test plan helps prepares those with a stake in the tests by setting stakeholder expectations and documenting the agreed upon testing approaches.

The test plan is a document that outlines for project stakeholders:

- the product functions to be tested;
- what specific tests will be performed;
- the approach to be take for those tests;
- what to test and what not to test;
- how the tests will be performed;
- who will be responsible for performing each test;
- what results are expected;
- what is considered a successful test and a failed test; and
- the exit criteria for any series of tests as well as for the testing phase as a whole.

A well defined test plan must also describe, in explicit detail, the method, goals, approaches, etc to be used for each test.

The test plan is often created in the early stage of the deployment phase by the testing team, or some group of testing specialists associated with the project. The initial draft of the plan is then reviewed by the project manager and other stakeholders to ensure its compliance with project and business requirements.

According to Harold Kerzner's book Project Management a Systems Approach to Planning, Scheduling, and Controlling (Publisher: John Wiley & Sons Date Published: 1979), approximately 20% - 30% of the overall project's work effort should be allocated to testing. As the risk, size, and/or complexity of the project increases this value may need to increase accordingly. Regardless of how much testing is allocated for

the project, it is important to note that acceptable test results do not necessarily require perfection. Acceptable testing is more about validating what was agreed to be done rather than being perfect or even exceeding expectations. If any special requirements such as specific hardware, training, etc are necessary to successfully test the products, the test plan should detail this information as well.

There are a number of different testing techniques and approaches. However, regardless of which approach is used test planning is made up of three basic phases that include:

### 1. **Preparing for Tests**

Preparation for testing is a vital part of the test planning process. This stage outlines the tests to be performed, and if necessary, creates the environment in which to perform those tests. Some of the documents necessary to effectively prepare for testing include:

- **Test Plan** - describes what testing will be done, to what quality standard, with what resource, within what timeframe, and outlines any risks/issues and how they will be addressed. A well defined test plan should also outline items such as:
  - Items to test such as test product features, interfaces, reporting tools
  - Items not to test such as test product features, interfaces, reporting tools
  - Risks, issues, mitigation strategies, and contingencies plans
  - Testing approach defining methods and tools to be used for testing
  - Item pass/fail criteria specifying what constitutes a successful test
  - Entry and exit criteria specifying what constitutes a completed test
  - Test deliverables such as a test plan, test cases, test tools
  - Environmental needs outlining any requirements for where testing will take place
  - Staffing and training needs
  - Acceptance criteria
- **Test Design Specification** - describes what needs to be tested. This document evolves from requirements and design specification documents. This document also outlines items such as what features to test, the approach to be taken, what the expected test results are, and what constitutes a successful and failed test.
- **Test Case Specification** - produced after the test design is completed these documents are unique to the item being tested. These documents outline the actual tests, procedures, reporting requirements, the individuals involved in the testing process, etc that will be used to validate each of the project's different functional requirements as specified in the requirements and design definition documents.

### 2. **Performing Tests**

The schedule of what test cases will be executed when is outlined within the project's test plan which will be executed by the project's testing team. Depending on what is being tested at which point within the project's life cycle, informal testing may be performed by developers, quality assurance, users, etc. However, final testing will be performed by individuals identified within the test plan. One essential prerequisite to testing is to at least have a completed software module to test, preferably a fully functional version of the software to be tested. Once ready, methods of testing may include:

- **Compatibility Testing** - tests the system, or a component of it, against existing systems, hardware, software, etc to ensure compatibility.
- **Conformance Testing** - verifies the systems conformance to industry/organizational standards, Federal/organizational mandates and regulations, etc.
- **Functional Testing** - verifies that the system indeed conforms to documented functional specifications and ensures that the product delivered indeed satisfies the requirements as agreed upon by the client.
- **Load Testing** - executes performance tests and stress tests to ensure that the system can handle all expected, and exceptional, levels of demands upon the system.
- **Performance Testing** - executed to better understand the scalability of the system, to benchmark performance, identify performance bottlenecks, etc.
- **Regression Testing** - retests previously tested system components to ensure that any reported defects have been corrected and that no new quality issues have been introduced.
- **Stress Testing** - tests that evaluate the system, or component of it, at or beyond the limits of its specified requirements to determine the load under which it fails and how.
- **System Testing** - tests the entire system from end-to-end.

- **Unit Testing** - tests individual components or modules of the system for the purpose of finding defects.
- **User Acceptance Testing** - tests executed by the user to ensure that the product delivered indeed satisfies the requirements as agreed upon by the client.
- **Vulnerability Assessment Testing** - tests that identify, quantify, and prioritize system vulnerabilities

**Test Log/Reporting** - As each test is performed it is important to document the details and results of those tests. A test log is commonly used to accomplish this. The log records information regarding what test cases have been run and the results of those tests. This log includes items such as a description of the test, the assignment of each test to one or more individuals, a target date by which the test needs to be completed, and other related information. If a test has failed, a test incident report should be filed to record the issue for future resolution.

### 3. Validating/Evaluating Testing

This simply involves evaluating the test results to verify that the product tests performed in accordance with defined requirements as agreed upon by the client. The confirmed results are then recorded using a test summary document. A test summary document records all pertinent information about the testing of the product. It would document items such as:

- An assessment of actual testing vs. planned testing
- A critical assessment about the quality of the system
- The number of incidents raised and any remaining outstanding issues

## Best Practices

The following best practices are recommended for **Test Planning**:

- **Plan for Testing** - Plan for approximately 20% - 30% of the project's work effort for testing. As the risk, size, and/or complexity of the project increases this value may need to increase accordingly.
- **User Groups** - Identify user groups for each scheduled test as early in the project's life as possible. Solicit commitment from each group to perform the required tests.
- **Segregate Testing** - Validating that user requirements have been fulfilled is vital to project success. However, there is often a difference between valid testing performed by the project team and testing performed by the actual end-users. Whenever possible user acceptance testing (UAT) should be conducted by a group made up of individuals different from anyone on the actual project team.
- **Facility** - Identify any required facilities needed to perform each scheduled test as early in the project's life as possible. Solicit commitment from the facility owner to allow its use for testing.
- **Active Stakeholder Participation** - Access is needed for users that have the authority and ability to provide and obtain information regarding the tests being performed.
- **Review and Approve** - Review all tests and testing documents to ensure that they do indeed test the product requirements as agreed upon in the requirement and design specification documents. Agree upon pass/fail criteria for tests and if needed a go/no-go stage gate for production roll-out.
- **Traceability** - Tests should be directly traceable to satisfy defined requirements.
- **Assumptions/Constraints** - Record and address all assumptions, constraints, issues, etc.

## Practice Activities

For software development projects the following practice activities are appropriate:

- **Prepare** - Prepare for testing by creating a test plan and documenting design specifications and test cases.
- **Perform** - Perform the planned tests and log the results within a test log. Report those results to the appropriate stakeholders.
- **Validate/Evaluate** - Validate test results conform to the expected outcomes. Evaluate the test results to better understand the outcomes and if needed, make changes to the product.