



**Your Company Name**

# **Test Plan**

Date

www.SDLCforms.com



## Revision History

Date	Version	Author	Change

www.SDLCforms.com

### COPYRIGHT NOTICE

Confidential – ©2015 Documentation Consultants

All rights reserved. These materials are for internal use only. No part of these materials may be reproduced, published in any form or by any means, electronic or mechanical, including photocopy or any information storage or retrieval system, nor may the materials be disclosed to third parties without the written authorization of (Your Company Name).



## Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>4</b>
1.1	Features to be Tested.....	4
1.2	Features Not to be Tested .....	4
1.3	Testing Approach.....	4
1.4	Identifying and Justifying Tests .....	5
1.4.1	Unit Test .....	5
1.4.2	Integration Test.....	5
1.4.3	User Acceptance Test (UAT) .....	6
1.4.4	Load Test.....	6
1.4.5	Operational Readiness Test.....	6
1.4.6	Beta Testing .....	6
1.5	Measuring the Extent of Testing .....	7
1.5.1	Entrance Criteria.....	7
1.5.2	Exit Criteria .....	7
<b>2</b>	<b>Dependencies, Assumptions and Constraints .....</b>	<b>8</b>
2.1.1	Dependencies.....	8
2.1.2	Assumptions .....	8
2.1.3	Constraints .....	9
<b>3</b>	<b>Notification / Escalation Procedures .....</b>	<b>9</b>
<b>4</b>	<b>Measures and Metrics .....</b>	<b>10</b>
4.1	Test Preparation .....	10
4.2	Test Execution and Progress.....	10
4.3	Bug Analysis.....	10
<b>5</b>	<b>Testing Tasks.....</b>	<b>10</b>
<b>6</b>	<b>Suspension Criteria &amp; Resumption of Testing .....</b>	<b>11</b>
<b>7</b>	<b>Status and Issues Meetings.....</b>	<b>11</b>
<b>8</b>	<b>Definitions .....</b>	<b>11</b>
<b>9</b>	<b>Approvals .....</b>	<b>12</b>



The purpose of the test plan is to:

- Provide a central artifact to govern the strategic approach of the test effort; it defines the general approach to be employed when testing the software and when evaluating the results of that testing. Planning artifacts will refer to the test strategy regarding the governing of detailed testing work.
- Provide visible confirmation to test-effort stakeholders that adequate consideration has been given to the governing the test effort and, where appropriate, to have those stakeholders approve the strategy.

*Note: Text displayed in blue italics is included to provide guidance to the author and should be deleted before publishing the document. In any table, select and delete any blue line text; then click Home→Styles and select “Table Text” to restore the cells to the default value.*

## 1 Introduction

The purpose of this Test Strategy is to define the overall approach that will be taken by the Test Team when delivering testing services to all of the projects within the business.

The document helps to clarify the testing activities, roles and responsibilities, processes and practice to be used across successive projects.

*Customize this section and its subsections to fit your project’s needs.*

### 1.1 Features to be Tested

The Test Breakdowns will include details of the Test Scenarios from which the Test Cases will be derived.

### 1.2 Features Not to be Tested

Where it is not possible for the team to test features of a Test Item that would have been expected or that would fall under the scope of testing shown in section 10, Testing Tasks, it will be recorded in the Test Plan.

### 1.3 Testing Approach

The test approach defines the scope and general direction of the test effort. It is a high-level description of the important issues needing to be covered in the test plan and test scripts.

All testing tasks will be conducted in line with the Software Test Life Cycle (STLC) and in support of the Software Development Life Cycle (SDLC). The documents used within the SDLC will be completed both by the Test Team and the project participants that are responsible for providing information and deliverables to the Test Team.



For each testing phase, a detailed test plan shall be developed that identifies the testing requirements specific to that phase.

Specific items to be identified in each test plan shall include:

- Test Items
- Test Execution Procedures
- Test Deliverables
- Test Data Management
- Test Schedule
- Test Environment

## 1.4 Identifying and Justifying Tests

*Describe how tests will be identified and considered for inclusion in the test effort covered by this strategy. Include the sections below that will be in scope for testing your specific project. Unit Test, Integration Test, and User Acceptance are required*

### 1.4.1 Unit Test

Unit testing is the initial testing of new and/or changed code in the system. The purpose of unit testing is to allow the developer to confirm the functionality provided by a single unit or component of code. Additionally, wherein one component cannot function without interacting with another component, the test shall include limited interactions.

Unit testing shall consist of the following:

- **Static testing** – Conducting “walkthroughs” and reviews of the design and coded components.
- **Basic path testing** – Executing path testing based on normal flow.
- **Condition/multi-condition testing** – Executing path testing based on decision points.
- **Data flow testing** – Examining the assignment and use of variables in a program.
- **Loop testing** – Checking the validity of loop constructs.
- **Error testing** – Executing unexpected error conditions.

### 1.4.2 Integration Test

Integration testing confirms that each piece of the application interacts as designed and that all functionality is working. Integration testing includes interactions between all layers of an application, including interfaces to other applications, as a complete end-to-end test of the functionality.

Integration testing shall consist of the following:

- Verifying links between internal application components.
- Focusing on complete end-to-end processing of programs, threads, and transactions.
- Boundary value analysis (testing modules by supplying input values within, at, and beyond the specified boundaries).
- Cause-effect testing (supplying input values to cause all possible output values to occur).



- Comparison testing (comparing output of system under test with another reference system).
- Security functionality.
- Ensuring traceability to requirements, use cases, user interface (UI) design, and test objectives.
- Testing each business function end-to-end through the application, including positive and negative tests.
- Testing each non-functional requirement.

### **1.4.3 User Acceptance Test (UAT)**

The purpose of user acceptance testing (UAT) is to simulate the business environment and emphasize security, documentation, and regression tests. UAT may be performed by a third party (*i.e.*, TCO) in cases where the general user community is large and may provide different goals and objectives for acceptance testing requirements.

UAT shall be conducted to gain acceptance of all functionality from the user community. UAT shall verify that the system meets user requirements as specified.

### **1.4.4 Load Test**

The purpose of load testing is to identify potential performance problems before they occur in production.

Load testing shall be used to test the performance of the application with near-production (or greater) levels of users accessing the application at the same time as specified by the Supplementary Specifications.

### **1.4.5 Operational Readiness Test**

The purpose of operational readiness testing is to identify any potential issues with the production environment setup before users access the system.

Operational readiness testing shall verify that the application move from the acceptance environment to the production environment was successful.

### **1.4.6 Beta Testing**

In beta testing, a small number of experienced users try the product in a production mode and report defects and deficiencies. The purpose of beta testing is to identify suggested improvements into a general release for the larger user community.

Defects identified during beta testing shall be grouped into two categories: those with significant impact that may not justify immediate implementation and those that can be easily integrated into the project.

Beta testing shall consider the following issues:

- Proper identification of the beta testing group.
- Specific areas for which feedback is requested.
- Specific areas for which feedback is not requested.



## 1.5 Measuring the Extent of Testing

### 1.5.1 Entrance Criteria

Entrance criteria are the required conditions and standards for work product quality that must be present or met prior to the start of a test phase.

Entrance criteria shall include following:

- Review of completed test script(s) for the prior test phase.
- No open critical/major defects remaining from the prior test phase.
- Correct versioning of components moved into the appropriate test environment.
- Testing environment is configured and ready.

### 1.5.2 Exit Criteria

Exit criteria are the required conditions and standards for work product quality that block the promotion of incomplete or defective work products to the next test phase of the component.

Exit criteria shall include the following:

- Successful execution of the test scripts(s) for the current test phase.
- No open critical, major, or average severity defects unless the issue is determined to be low impact and low risk.
- Component stability in the appropriate test environment.



## 2 Dependencies, Assumptions and Constraints

### 2.1.1 Dependencies

*In the following table, list any dependencies identified during the development of this test strategy that may affect its successful execution if those dependencies are not honored. Typically these dependencies relate to activities on the critical path that are prerequisites or post-requisites to one or more preceding (or subsequent) activities.*

*You should account for the responsibilities in which you are relying on other teams or staff members external to the test effort, the timing and dependencies of other planned tasks, and the reliance on certain work products being produced*

Dependency	Potential Impact of Dependency	Owners

### 2.1.2 Assumptions

*In the following table, list any assumptions made during the development of this test strategy that may affect its successful execution if those assumptions are proven incorrect. Assumptions might relate to work you assume other teams are doing, expectations that certain aspects of the product or environment are stable, etc.*

Assumptions	Potential Impact of Assumption	Owners





### 2.1.3 Constraints

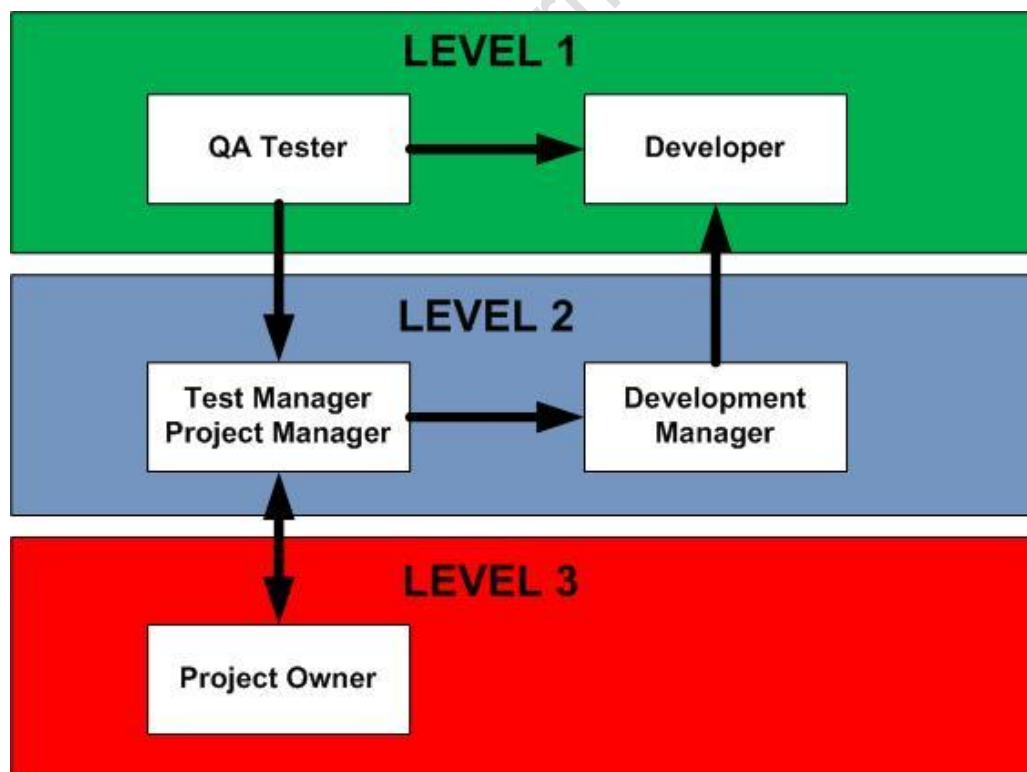
*In the following table, list any constraints placed on the test effort that have had a negative affect on the way in which this test strategy has been approached.*

Constraint On	Impact Constraint has on Test Effort	Owners

## 3 Notification / Escalation Procedures

*Replace the illustration with an illustration that pertains to your process.*

The following diagram shows the notification and escalation paths to be followed for the duration of the project Testing Phase.





---

## 4 Measures and Metrics

During the Project Initiation Phase, the Test Team will publish a set of measures and metrics related to the test activities of their Planning & Analysis and Execution phases. The Test Plan also defines the milestone dates for key deliverables such as the Test Plan and these are metrics captured for ongoing statistical process analysis across successive projects.

### 4.1 Test Preparation

- Number of Test Scenarios v. Number of Test Cases
- Number of Test Cases Planned v. Ready for Execution
- Total time spent on Preparation v. Planned time.

### 4.2 Test Execution and Progress

- Number of Tests Cases Executed v. Test Cases Planned
- Number of Test Cases Passed, Failed and Blocked
- Total Number of Test Cases Passed by Test Item / Test Requirements
- Total Time Spent on Execution vs. Planned Time

### 4.3 Bug Analysis

- Total Number of Bugs Raised and Closed per Test Run
- Total Number of Bugs Closed v. Total Number of Bugs Re-Opened
- Bug Distribution Totals by Severity per Test Run
- Bug Distribution Totals by Test Item by Severity per Test Run.

---

## 5 Testing Tasks

The Testing Tasks that the Test Team will deliver cover the following scope:

- **Fully In Scope:** Functional and Regression Testing
- **Partially in Scope:** Cross Browser Compatibility, Integration in the Large.
- **Out of Scope:** Performance testing, Automated Regression, all forms of Non-Functional, Accessibility Compliance Testing, Security Testing, User Documentation Review.



## 6 Suspension Criteria & Resumption of Testing

Suspension Criteria	Resumption Requirements
Severity 1 issue is logged and requires fixing before further testing can take place (a Blocking Issue)	The issue will need to be fixed before the Test Item is returned to the Test Team for testing.
Significant differences exist between observed behavior of the Test Item and that shown in Test Scenario, Test Case or as expected from the previous version of the technology.	Development, the Test Team and PM must come to a conclusion on resolving the issue and agreeing a definition of the expected behavior.
A Test Item sent for testing fails more than 20% of Developer Unit Tests.	The Test Item must be fixed or Unit Tests refactored if out of date and then demonstrated to pass with <20% failure rate.

## 7 Status and Issues Meetings

During Development/Unit Testing, it is recommended that a weekly status meeting be held at the same scheduled time on the same day each week. This meeting will support a communication process to inform the project team of status, problems or issues, and changes. The Agenda will include a review of open high and medium issues, Unit and Integration Test plans, and any action items from the previous week.

Meeting minutes should be taken that include a list of attendees, discussion points, and action items.

## 8 Definitions

Test Phases	Focus
Unit Test	Testing individual modules and programs, and testing them in sufficient context to insure that work flows correctly through the affected business process.
Integration/System Test	Validates that all processes, including customizations and interfaces, work together to support the business functions
Volume/Stress Test	Validates that critical functions will meet production performance requirements
User Test	Validating production-ready functionality and data integrity
Regression Test	Ensures that the application doesn't negatively impact previously migrated objects/modules. Re-tests the application to ensure that the application performs correctly after a package upgrade or change.



## 9 Approvals

*Update the table to reflect your organization.*

The following personnel are required to approve the Test Plan.

Approval By	Approval	Date
Test Manager		
Test Department Manager		
Product Owner		
Development Manager		
Project Manager		