Company Name

Project Name

Software Test Plan

Version: (n)   **Date: (mm/dd/yyyy)**

1. Revision History

Revision #	Revision Date	Description of Change	Author

2. Distribution

Recipient Name	Recipient Organization	Distribution Method

TABLE OF CONTENTS

.

1. INTRODUCTION

(NOTE 1: THE SOFTWARE TEST PLAN GUIDELINES WERE DERIVED AND DEVELOPED FROM IEEE STANDARD FOR SOFTWARE TEST DOCUMENTATION (829-1998)).

(Note 2: The ordering of Software Test Plan (STP) elements is not meant to imply that the sections or subsections must be developed or presented in that order. The order of presentation is intended for ease of use, not as a guide to preparing the various elements of the Software Test Plan. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content.)

The <u>*Introduction*</u> *section of the Software Test Plan (STP) provides an overview of the project and the product test strategy, a list of testing deliverables, the plan for development and evolution of the STP, reference material, and agency definitions and acronyms used in the STP.*

.

The Software Test Plan (STP) is designed to prescribe the scope, approach, resources, and schedule of all testing activities. The plan must identify the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

1.1 Objectives

(Describe, at a high level, the scope, approach, resources, and schedule of the testing activities. Provide a concise summary of the test plan objectives, the products to be delivered, major work activities, major work products, major milestones, required resources, and master high-level schedules, budget, and effort requirements.)

1.2 Testing Strategy

Testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. (*This may appear as a specific document (such as a Test Specification), or it may be part of the organization's standard test approach. For each level of testing, there should be a test plan and an appropriate set of deliverables. The test strategy should be clearly defined and the Software Test Plan acts as the high-level test plan. Specific testing activities will have their own test plan. Refer to section 5 of this document for a detailed list of specific test plans.)*

Specific test plan components include:

- *Purpose for this level of test,*

- *Items to be tested,*

- *Features to be tested,*

- *Features not to be tested,*

- *Management and technical approach,*

- *Pass / Fail criteria,*

- *Individual roles and responsibilities,*

- *Milestones,*

- *Schedules, and*

- *Risk assumptions and constraints.*

1.3 Scope

(Specify the plans for producing both scheduled and unscheduled updates to the Software Test Plan (change management). Methods for distribution of updates shall be specified along with version control and configuration management requirements must be defined.)

Testing will be performed at several points in the life cycle as the product is constructed. Testing is a very 'dependent' activity. As a result, test planning is a continuing activity performed throughout the system development life cycle. Test plans must be developed for each level of product testing.

1.4 Reference Material

(Provide a complete list of all documents and other sources referenced in the Software Test Plan. Reference to the following documents (when they exist) is required for the high-level test plan:

- *Project authorization,*

- *Project plan,*

- *Quality assurance plan,*

- *Configuration management plan,*

- *Organization policies and procedures, and*

- *Relevant standards.)*

1.5 Definitions and Acronyms

(Specify definitions of all terms and agency acronyms required to properly interpret the Software Test Plan. Reference may be made to the Glossary of

Terms on the IRMC web page.)

2. TEST ITEMS

(Specify the test items included in the plan. Supply references to the following item documentation:

- *Requirements specification,*

- *Design specification,*

- *Users guide,*

- *Operations guide,*

- *Installation guide,*

- *Features (availability, response time),*

- *Defect removal procedures, and*

- *Verification and validation plans.)*

2.1 Program Modules

(Outline testing to be performed by the developer for each module being built.)

2.2 Job Control Procedures

(Describe testing to be performed on job control language (JCL), production scheduling and control, calls, and job sequencing.)

2.3 User Procedures

(Describe the testing to be performed on all user documentation to ensure that it is correct, complete, and comprehensive.)

2.4 Operator Procedures

(Describe the testing procedures to ensure that the application can be run and supported in a production environment (include Help Desk procedures)).

3. FEATURES TO BE TESTED

(Identify all software features and combinations of software features to be tested. Identify the test design specifications associated with each feature and each combination of features.)

4. FEATURES NOT TO BE TESTED

(Identify all features and specific combinations of features that will not be tested along with the reasons.)

5. APPROACH

(Describe the overall approaches to testing. The approach should be described in sufficient detail to permit identification of the major testing tasks and estimation of the time required to do each task. Identify the types of testing to be performed along with the methods and criteria to be used in performing test activities. Describe the specific methods and procedures for each type of testing. Define the detailed criteria for evaluating the test results.)

(For each level of testing there should be a test plan and the appropriate set of deliverables. Identify the inputs required for each type of test. Specify the source of the input. Also, identify the outputs from each type of testing and specify the purpose and format for each test output. Specify the minimum degree of comprehensiveness desired. Identify the techniques that will be used to judge the comprehensiveness of the testing effort. Specify any additional completion criteria (e.g., error frequency). The techniques to be used to trace requirements should also be specified.)

5.1 Component Testing

(Testing conducted to verify the implementation of the design for one software element (e.g., unit, module) or a collection of software elements. Sometimes called unit testing. The purpose of component testing is to ensure that the program logic is complete and correct and ensuring that the component works as designed.)

5.2 Integration Testing

(Testing conducted in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated. The purpose of integration testing is to ensure that design objectives are met and ensures that the software, as a complete entity, complies with operational requirements.)

5.4 System Testing

(Testing to ensure that the application operates like a complete system.)

5.6 Security Testing

(Testing done to ensure that the application systems control and auditability features of the application are functional.)

5.7 Recovery Testing

(Testing done to ensure that application restart and backup and recovery facilities operate as designed.)

5.8 Performance Testing

(Testing done to ensure that that the application performs to customer expectations (response time, availability, portability, and scalability)).

5.9 Regression Testing

(Testing done to ensure that that applied changes to the application have not adversely affected previously tested functionality.)

5.10 Acceptance Testing

(Testing conducted to determine whether or not a system satisfies the acceptance criteria and to enable the customer to determine whether or not to accept the system. Acceptance testing ensures that customer requirements' objectives are met and that all components are correctly included in a customer package.)

5.11 Beta Testing

(Testing, done by the customer, using a pre-release version of the product to verify and validate that the system meets business functional requirements. The purpose of beta testing is to detect application faults, failures, and defects.)

6. PASS / FAIL CRITERIA

(Specify the criteria to be used to determine whether each item has passed or failed

testing.)

6.1 Suspension Criteria

(Specify the criteria used to suspend all or a portion of the testing activity on test items associated with the plan.)

6.2 Resumption Criteria

(Specify the conditions that need to be met to resume testing activities after suspension. Specify the test items that must be repeated when testing is resumed.)

6.3 Approval Criteria

(Specify the conditions that need to be met to approve test results. Define the formal testing approval process.)

7. TESTING PROCESS

(Identify the methods and criteria used in performing test activities. Define the specific methods and procedures for each type of test. Define the detailed criteria for evaluating test results.)

7.1 Test Deliverables

(Identify the deliverable documents from the test process. Test input and output data should be identified as deliverables. Testing report logs, test incident reports, test summary reports, and metrics' reports must be considered testing deliverables.)

7.2 Testing Tasks

(Identify the set of tasks necessary to prepare for and perform testing activities. Identify all intertask dependencies and any specific skills required.)

7.3 Responsibilities

(Identify the groups responsible for managing, designing, preparing, executing, witnessing, checking, and resolving test activities. These groups may include the developers, testers, operations staff, technical support staff, data administration staff, and the user staff.)

7.4 Resources

(Identify the resources allocated for the performance of testing tasks. Identify the organizational elements or individuals responsible for performing testing activities. Assign specific responsibilities. Specify resources by category. If

automated tools are to be used in testing, specify the source of the tools, availability, and the usage requirements.)

7.5 Schedule

(Identify the high level schedule for each testing task. Establish specific milestones for initiating and completing each type of test activity, for the development of a comprehensive plan, for the receipt of each test input, and for the delivery of test output. Estimate the time required to do each test activity.)

(When planning and scheduling testing activities, it must be recognized that the testing process is iterative based on the testing task dependencies.)

8. ENVIRONMENTAL REQUIREMENTS

(Specify both the necessary and desired properties of the test environment including the physical characteristics, communications, mode of usage, and testing supplies. Also provide the levels of security required to perform test activities. Identify special test tools needed and other testing needs (space, machine time, and stationary supplies. Identify the source of all needs that is not currently available to the test group.)

8.1 Hardware

(Identify the computer hardware and network requirements needed to complete test activities.)

8.2 Software

(Identify the software requirements needed to complete testing activities.)

8.3 Security

(Identify the testing environment security and asset protection requirements.)

8.4 Tools

(Identify the special software tools, techniques, and methodologies employed in the testing efforts. The purpose and use of each tool shall be described. Plans for the acquisition, training, support, and qualification for each tool or technique.)

8.5 Publications

(Identify the documents and publications that are required to support testing activities.)

8.6 Risks and Assumptions

(Identify significant constraints on testing such as test item availability, test resource availability, and time constraints. Identify the risks and assumptions associated with testing tasks including schedule, resources, approach and documentation. Specify a contingency plan for each risk factor.)

9. CHANGE MANAGEMENT PROCEDURES

(Identify the software test plan change management process. Define the change initiation, change review, and change authorization process.)

10. PLAN APPROVALS

.

(Identify the plan approvers. List the name, signature and date of plan approval.)