A Template for a Test Plan

1. GENERAL INFORMATION

1.1 Summary. Summarize the functions of the software and the tests to be performed.

1.2 Environment and Pretest Background. Summarize the history of the project. Identify the user organization and computer center where the testing will be performed. Describe any prior testing and note results that may affect this testing.

1.3 Test Objectives. State the objectives to be accomplished by testing.

1.4 Expected Defect Rates. State the estimated number of defects for software of this type.

1.5 References. List applicable references, such as:

a) Project request authorization.
b) Previously published documents on the project.
c) Documentation concerning related projects.

2. PLAN

2.1 Software Description. Provide a chart and briefly describe the inputs, outputs, and functions of the software being tested as a frame of reference for the test descriptions.

2.2 Test Team. State who is on the test team and their test assignment(s).

2.3 Milestones. List the locations, milestone events, and dates for the testing.

2.4 Budgets. List the funds allocated to test by task and checkpoint.

2.5 Testing (systems checkpoint). Identify the participating organizations and the system checkpoint where the software will be tested.

2.5.1 Schedule (and budget). Show the detailed schedule of dates and events for the testing at this location. Such events may include familiarization, training, data, as well as the volume and frequency of the input. Resources allocated for test should be shown.

2.5.2 Requirements. State the resource requirement, including:

a) Equipment. Show the expected period of use, types, and quantities of the equipment needed.
b) Software. List other software that will be needed to support the testing that is not part of the software to be tested.
c) Personnel. List the numbers and skill types of personnel that are expected to be available during the test from both the user and development groups. Include any special requirements such as multishift operation or key personnel.

2.5.3 Testing Materials. List the materials needed for the test, such as:

a) System documentation
b) Software to be tested and its medium
c) Test inputs
d) Test documentation
e) Test tools

2.5.4 Test Training. Describe or reference the plan for providing training in the use of the software being tested. Specify the types of training, personnel to be trained, and the training staff.

2.5.5 Test to be Conducted. Reference specific tests to be conducted at this checkpoint.

2.6 Testing (system checkpoint). Describe the plan for the second and subsequent system checkpoint where the software will be tested.

3. SPECIFICATIONS AND EVALUATION

3.1 Specifications

3.1.1 Business Functions. List the business functional requirement established by earlier documentation.

3.1.2 Structural Functions. List the detailed structural functions to be exercised during the overall test.

3.1.3 Test/Function Relationships. List the tests to be performed on the software and relate them to the functions in paragraph 3.1.2.

3.1.4 Test Progression. Describe the manner in which progression is made from one test to another so that the entire test cycle is completed.

3.2 Methods and Constraints.

3.2.1 Methodology. Describe the general method or strategy of the testing.

3.2.2 Test Tools. Specify the type of test tools to be used.

3.2.3 Extent. Indicate the extent of the testing, such as total or partial. Include any rationale for partial testing.

3.2.4 Data Recording. Discuss the method to be used for recording the test results and other information about the testing.

3.2.5 Constraints. Indicate anticipated limitations on the test due to test conditions, such as interfaces, equipment, personnel, databases.

3. SPECIFICATIONS AND EVALUATION *(continued)*

3.3 Evaluation.

3.3.1 Criteria. Describe the rules to be used to evaluate test results, such as range of data values used, combinations of input types used, maximum number of allowable interrupts or halts.

3.3.2 Data Reduction. Describe the techniques to be used for manipulating the test data into a form suitable for evaluation, such as manual or automated methods, to allow comparison of the results that should be produced to those that are produced.

4. TEST DESCRIPTIONS

4.1 Test (Identify). Describe the test to be performed (format will vary for on-line test script).

4.1.1 Control. Describe the test control, such as manual, semiautomatic or automatic insertion of inputs, sequencing of operations, and recording of results.

4.1.2 Inputs. Describe the input data and input commands used during the test.

4.1.3 Outputs. Describe the output data expected as a result of the test and any intermediate messages that may be produced.

4.1.4 Procedures. Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination.

4.2 Test (Identify). Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

Template for Test Problem Documentation

1/1

Test Problem Documentation

Name of Software Tested

Problem Description

Actual Results

Expected Results

Effect of Deviation

Cause of Problem

Location of Problem

Recommended Action

Test Report (Template)

1/1

1. General Information

1.1 *Summary.* Summarize both the general functions of the software tested and the test analysis performed.

1.2 *Environment.* Identify the software sponsor, developer, user organization, and the computer center where the software is to be installed. Assess the manner in which the test environment may be different from the operation environment, and the effects of this difference on the tests.

1.3 *References.* List applicable references, such as:
a. Project request (authorization).
b. Previously published documents on the project.
c. Documentation concerning related projects.

2. Test Results and Findings

Identify and present the results and findings of each test separately in paragraphs 2.1 through 2.*n*.

2.1 *Test (identify)*

2.1.1 *Validation tests.* Compare the data input and output results, including the output of internally generated data, of this test with the data input and output requirements. State the findings.

2.1.2 *Verification tests.* Compare what is shown on the document to what should be shown.

2.*n* *Test (identify).* Present the results and findings of the second and succeeding tests in a manner similar to that of paragraph 2.1.

3. Software Function Findings

Identify and describe the findings on each function separately in paragraphs 3.1 through 3.*n*.

3.1 *Function (identify)*

3.1.1 *Performance.* Describe briefly the function. Describe the software capabilities designed to satisfy this function. State the findings as to the demonstrated capabilities from one or more tests.

3.1.2 *Limits.* Describe the range of data values tested. Identify the deficiencies, limitations, and constraints detected in the software during the testing with respect to this function.

3.*n* *Function (identify).* Present the findings on the second and succeeding functions in a manner similar to that of paragraph 3.1.

4. Analysis Summary

4.1 *Capabilities.* Describe the capabilities of the software as demonstrated by the tests. Where tests were to demonstrate fulfillment of one or more specific performance requirements, compare the results with these requirements. Compare the effects any differences in the test environment versus the operational environment may have had on this test demonstration of capabilities.

4.2 *Deficiencies.* Describe the deficiencies of the software as demonstrated by the tests. Describe the impact of each deficiency on the performance of the software. Describe the cumulative or overall impact on performance of all detected deficiencies.

4.3 *Risks.* Describe the business risks if the software is placed in production.

4.4 *Recommendations and estimates.* For each deficiency, provide any estimates of time and effort required for its correction, and any recommendations as to:
a. The urgency of each correction.
b. Parties responsible for corrections.
c. How the corrections should be made.

4.5 *Option.* State the readiness for implementation of the software.