BEST PRACTICE
Final Test Reports

Test reports should be prepared at the conclusion of each level of testing. This might include:

- Unit Test Report
- Integration Test Report
- System Test Report
- Acceptance Test Report

The test reports are designed to report the results of testing as defined in the Test Plan. Without a well-developed Test Plan, which has been executed in accordance with the plan, it is difficult to develop a meaningful test report.

All final test reports should be designed to accomplish the following three objectives:

- Define the scope of testing – this is normally a brief recap of the Test Plan.
- Present the results of testing.
- Draw conclusions and recommendations from those test results.

The final test report may be a combination of electronic data and printed information. For example, if the Function Test Matrix is maintained electronically, there is no reason to print that, as the detail is available electronically if needed. The printed report will summarize that data, draw the appropriate conclusions, and present recommendations.

The final test report has the following objectives:

- Inform the developers what works and what does not work.
- Provide information to the users of the software system so that they can determine whether the system is ready for production; and if so, to assess the potential consequences and initiate appropriate actions to minimize those consequences.
- After implementation, help the project trace problems in the event the application malfunctions in production. Knowing which functions have been correctly tested and which ones still contain defects can assist in taking corrective action.
- Use the test results to analyze the test process for the purpose of preventing similar defects from occurring in the future. Accumulating the results of many test reports to identify which components of the software development process that are defect-prone. These defect-prone

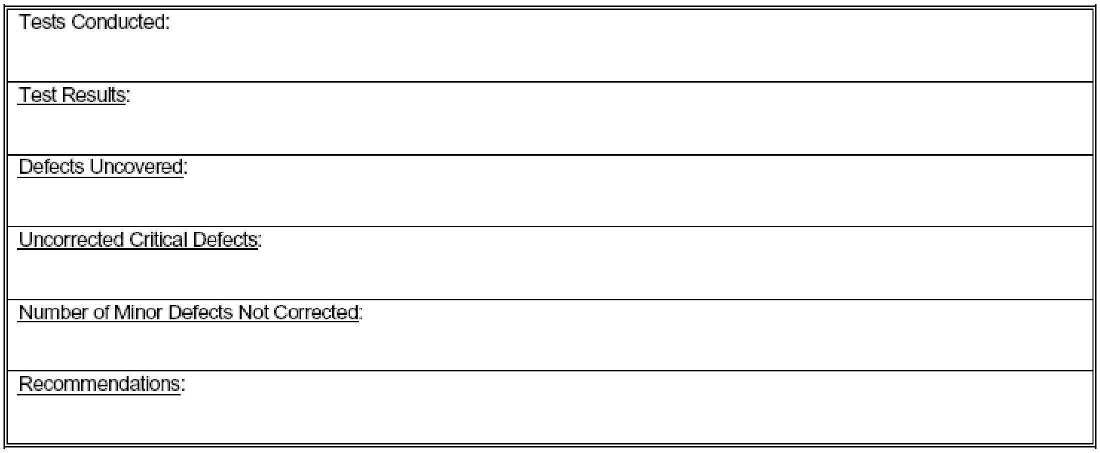
components identify tasks or steps that, if improved, could eliminate or minimize the occurrence of high-frequency defects.

Description of Test Reports

There is no generally accepted standard regarding the type, content and frequency of test reports. However, it is reasonable to assume that some type of report should be issued after the conclusion of each test activity. This would include reports at the conclusion of these tests activities:

- Unit test
- Integration test
- System test

The individual who wrote the unit normally conducts unit testing. The objective is to assure all the functions in the unit perform correctly, and the unit structure performs correctly. The report should focus on what was tested, the test results, defects uncovered and, what defects have not been corrected, plus the unit tester’s recommendations as to what should be done prior to integration testing. Figure 79 illustrates an example of a Unit Test Report.



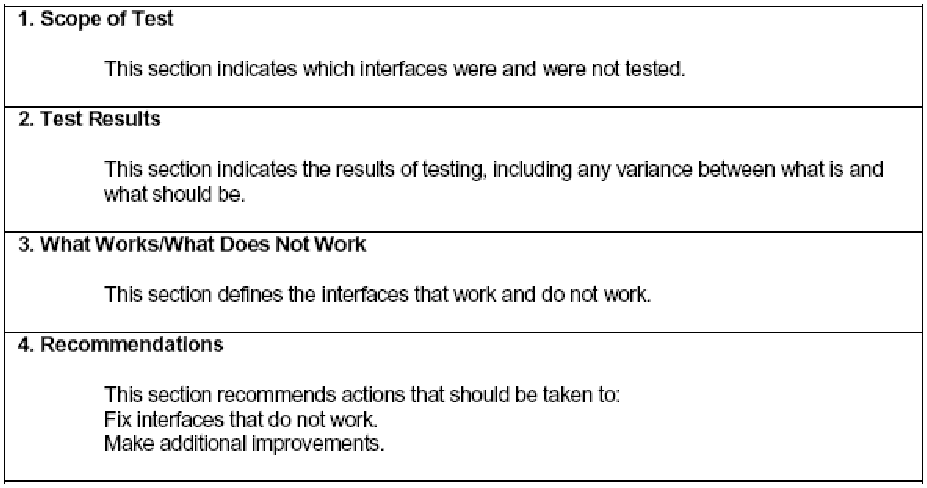
Tests Conducted:
Test Results:
Defects Uncovered:
Uncorrected Critical Defects:
Number of Minor Defects Not Corrected:
Recommendations:

Figure 79. Unit Test Report Example

Integration Test Report

Integration testing tests the interfaces between individual projects or units. A good Test Plan will identify the interfaces and institute test conditions that will validate interfaces. Given this, the integration report follows the same format as the Unit Test Report, except that the conditions tested are the interfaces.

As testing is completed on each project, integration test reports could be issued for the individual projects. Figure 80 is an example of such a report, indicating it should discuss the scope of the test, the test results, what works and does not work, and recommendations.



1. Scope of Test

This section indicates which interfaces were and were not tested.

2. Test Results

This section indicates the results of testing, including any variance between what is and what should be.

3. What Works/What Does Not Work

This section defines the interfaces that work and do not work.

4. Recommendations

This section recommends actions that should be taken to:
Fix interfaces that do not work.
Make additional improvements.

Figure 80. Integration Test Report Example

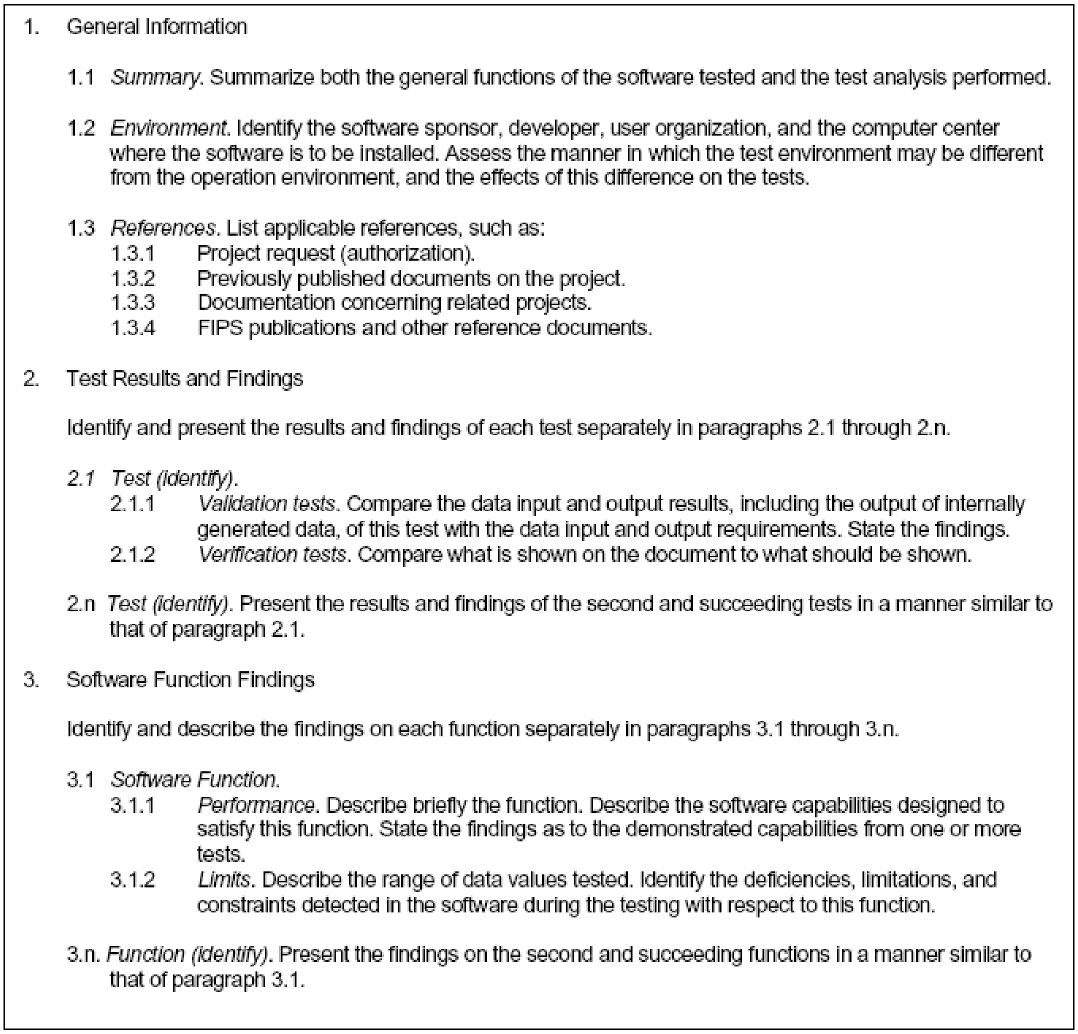
In any report on testing, it is important to show the scope; otherwise, the reader will assume that exhaustive testing has occurred, which is never the case. Testing is a risk-oriented activity in which resources should be expended to minimize the major risks. Exhaustive testing is neither possible, practical, nor economical. Thus, testing is never designed to assure that there are no defects remaining in the software and the scope will explain what the testers accomplished.

The remainder of the report is straightforward, describing the result of the testing – specifically, what works and what does not work in recommendations. Where detailed interim test reports are available, the “what works” and “what does not work” sections may merely reference those reports or attach those reports.

The recommendations section is a critical part of the report, because the reader is usually removed from the project being tested and the technical recommendations provided by the testers can help with the reader’s business decision. For example, testers may indicate that there is a 50/50 probability that the system will terminate abnormally in production due to dating problems. A business decision might then be made to put the software into operation, but develop effective backup recovery procedures in case the termination occurs.

System Test Report

Test Planning presented a system test plan standard that identified the objectives of testing, what was to be tested, how it was to be tested, and when tests should occur. The System Test Report should present the results of executing that Test Plan. Figure 81 illustrates the test reporting standard that is based on the test plan standard.



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:
1.3.1 Project request (authorization).
1.3.2 Previously published documents on the project.
1.3.3 Documentation concerning related projects.
1.3.4 FIPS publications and other reference documents.

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 *Software Function*.
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.

Figure 81. System Test Report Standard Example

References

Guide – CSTE Common Body Of Knowledge, V6.1