



## **BEST PRACTICE**

### **Performing Tests**

Test execution is the operation of a test cycle. Each cycle needs to be planned, prepared for, executed and the results recorded. This section addresses these activities involved in performing tests:

- Test platforms
- Test cycle strategy
- Use of tools in testing
- Test execution
- Executing the Unit Test plan
- Executing the Integration Test Plan
- Executing the System Test Plan
- When is Testing Complete?
- Concerns

#### **Platforms**

Building the Test Environment platforms must be established for conducting tests.

#### **Test Cycle Strategy**

Each execution of testing is referred to as a test cycle. Ideally the cycles are planned and included in the test plan. However, as defects are uncovered, and change is incorporated into the software, additional test cycles may be needed. Software testers should determine the number and purpose of the test cycles to be used during testing.

#### **Use of Tools in Testing**

A well-organized and structured file system and a good text editor are a minimum support set. A more powerful support set includes data reduction and report generation tools.

**Test Documentation** - The preparation of a Test Plan and issuing a Test Analysis Report is recommended. The Test Plan should identify test milestones and provide the testing schedule and requirements. The Test Analysis Report should summarize and document the test results and findings. The analysis summary should present the software capabilities, deficiencies, and recommendations.

#### **Test Execution**

- In a life cycle approach to testing, test performance can occur throughout the project life cycle, from testing requirements through conducting user



- acceptance testing. This discussion will focus on the performance of the dynamic testing that is planned for an application.
- The more detailed the test plan, the easier this task becomes for the individuals responsible for performing the test.
  - The Test Manager is responsible for conducting the Test Readiness Review prior to the start of testing. The purpose of this review is to ensure that all of the entrance criteria for the test phase have been met, and that all test preparations are complete.
  - The test plan should contain the procedures, environment, and tools necessary to implement an orderly, controlled process for test execution, defect tracking, coordination of rework, and configuration & change control. This is where all of the work involved in planning and set-up pays off.
  - For each phase of testing, the planned tests are performed and the actual results are compared to the documented expected results. When an individual performs a test script, they should be aware of the conditions under test, the general test objectives, as well as specific objectives listed for the script. All tests performed should be logged on a Test Execution Log, or in a tool such as Mercury's Test Director, by the individual performing the test.
  - The Test Log is a simple worksheet or spreadsheet that records test activities in order to maintain control over the test. It includes:
    - The test ID
    - Test activities
    - Start and Stop times
    - Pass or fail results
    - Comments
    - Be sure to document actual results. Log the incidents into the defect tracking system once a review determines it is actually a defect.
  - When the development team communicates the defect resolution back to the test team, and the fix is migrated to the test environment, the problem is ready for retest and execution of any regression testing associated with the fix.

### **Perform Unit Testing**

Unit testing is normally performed by the programmer that developed the program. Unit testing is performed many ways, most of which work, but the result of unit testing should be that the unit is defect free. In other words, the program specifications perform as specified. Integration testing should not occur until the units included in integration testing are defect free.

### **Perform Integration Test**

Integration testing should begin once unit testing for the components to be integrated is complete, the objectives in this stage of testing are to validate the



application design, and prove that the application components can be successfully integrated to perform one or more application functions.

### **Perform System Test**

System test should begin as soon as a minimal set of components has been integrated and successfully completed integration testing. System test ends when the test team has measured system capabilities and corrected enough of the problems to have confidence that the system will operate successfully in production. The major steps are outlined below:

- Set up system test environment, mirroring the planned production environment as closely as possible.
- Establish the test bed.
- Identify test cases that will be included in the system test.
- Identify test cycles needed to replicate production where batch processing is involved.
- Assign test cases to test cycles; note that in applications where the processing is real-time the test sets may still need to be grouped by planned days if the sequence of test execution is critical.
- Assign test scripts to testers for execution.
- Review test results and determine whether problems identified are actually defects.
- Record defect in tracking system, making sure the developer responsible for fixing the defect is notified.
- When the defect is fixed and migrated to the test environment, re-test and validate the fix. If the defect is fixed, close the defect log. If the defect is not fixed, return it to the developer for additional work.

### **When is Testing Complete?**

The Test Manager must be able to report, with some degree of confidence that the application will perform as expected in production and whether the quality goals defined at the start of the project have been met.

The Test Manager may use a set of test metrics, including Mean Time Between Failure or the percentage of coverage achieved by the executed tests, to determine whether the application is ready for production. Other factors, such as the number of open defects and their severity levels, must also be taken into consideration. Finally, the risk associated with moving the application into production, as well as the risk of not moving forward, must be taken into consideration.

### **General Concerns**

There are three general concerns testers have in performing tests:

- Software is not in a testable mode for this test level.



- There is inadequate time and resources.
- Significant problems will not be uncovered during testing.

## References

Guide – CSTE Common Body Of Knowledge, V6.1