Best Practices: Enterprise Test Management

Testing products before deployment is a critical step for enterprises. Those administering the tests must determine when a product is ready to be deployed to production. But before this decision is made, the entire testing process should be carefully planned, managed, and reviewed. With effective test management in place, enterprises can help ensure that products receive comprehensive, well-documented testing and are truly ready for production.

Success test management requires development of a comprehensive testing strategy. Decisions must be reached about who will manage the test project, what measurement tools will be needed, and where and by whom the testing will be performed. Enterprises can apply a few general rules when developing and managing a testing strategy:

- Time durations for each test phase vary according to the complexity and stability of the product during the course of the test execution.
- Test cases or test scenarios are developed to focus on the testing objective for each test phase. Depending on the stability that the product demonstrates during the test phases, these tests may be run multiple times.
- Changes in the product design or additions to the scope of the product late in the cycle can affect an enterprise’s test strategy as well as its resources and budget.

This article is the second in a series of articles examining best practices in enterprise testing. The previous article, “Best Practices: Enterprise Testing Fundamentals,” which appeared in the February 2006 issue of Dell Power Solutions, focused on the fundamentals of unit, product, and system testing in a phased approach. This article focuses on test management.

Developing a test strategy

A test strategy provides a framework for the testing effort. A comprehensive strategy goes beyond the technical requirements of the product or system under test. Figure 1 discusses some factors enterprises should consider when building a test strategy.

Enterprises may also want to consider industry best practices to help them develop their test strategy and processes:

- Industry experts recommend that the frequency of defects discovered during each phase of the test cycle fall within the following ranges: 60 to 65 percent during unit testing; 30 to 35 percent during product testing; and 5 to 10 percent during system testing.

Managing resources

Before testing begins, enterprise test teams should ensure that they and their testing partners (suppliers, contract houses, and so forth) have adequate capacity and resources for the job. Test personnel are needed in a variety of roles:

- A test project engineer or manager leads the test planning and execution.
- Test engineers develop and validate the necessary test cases (including automation).
- Test technicians set up the systems under test, run the test cases, and report any defects.

All necessary prototypes and test tools must be procured, shipped to the testing location(s), configured, and made ready to run the required tests. Test teams must make sure that they have a plan to upgrade hardware as needed during the project. Defect tracking and measurement systems must be ready. All members of the test team must understand the reporting requirements.

Managing the test project

Test project management involves planning the test project, establishing criteria that must be met before starting and concluding each test phase, supervising the test process, documenting the test procedures and results, and analyzing the test results.

Project planning

Managing a project begins well in advance of the start date. Stakeholders should have enough time to review the test strategy and test cases, and the test team should begin physical preparation of the test tools, configurations, and infrastructure.

Entrance and exit criteria

Test entrance and exit criteria should be defined for each phase of the test cycle. The test team must document criteria for entrance to and exit from the unit, product, and system test phases. Entrance criteria focus on component availability and other readiness measurements to start the test phase, whereas exit criteria are used to assess the maturity or progress of the product and readiness to move out of that phase. Figure 2 shows examples of entrance and exit criteria.

Test execution

Managing the execution phase consists primarily of keeping the test effort on track. This means that the test team must make sure that the tests are validating the planned functional coverage or use, defect fixes are being verified, and regression testing is being conducted within the allocated schedule and resources. Test teams often must achieve these goals even when the product design, scope, or implementation details change during the test cycle. Successfully managing test execution requires flexibility, sustained attention to test progress metrics (see the “Measuring success” section in this article), and an ability to make necessary last-minute adjustments.

<table>
<thead>
<tr>
<th>Factor to consider</th>
<th>Impact on testing strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the product under test a new architecture, first release, or sustaining effort?</td>
<td>Determines where to focus test coverage and technical strategy—for example, new feature validation, integration testing, or regression testing</td>
</tr>
<tr>
<td>Was the product under test developed in house, provided by a supplier, or jointly developed?</td>
<td>Determines the level of knowledge about the system’s internal structure or coding and whether low-level or “white box” testing is possible; drives decisions about who performs testing as well as when and where to test</td>
</tr>
<tr>
<td>Will future, similar test efforts be needed?</td>
<td>Drives decisions about staffing and test automation</td>
</tr>
<tr>
<td>Will testing be performed by staff, a contract test house, or a supplier?</td>
<td>Affects the test costs, training, and retention of technical knowledge; drives the audit strategy (degree of revalidation to perform on testing conducted outside the enterprise)</td>
</tr>
<tr>
<td>Will testing be performed in a single geographic region or in multiple regions?</td>
<td>Determines the need to evaluate certain factors: ownership and accountability for overall test effort; a clear delineation of responsibilities and tasks; the impact of working in different time zones; communication strategies, methods, and frequencies; and cultural factors such as trust, communication protocols, and holiday scheduling</td>
</tr>
<tr>
<td>How many configurations and prototype units are budgeted and available for testing?</td>
<td>Affects appropriate staffing levels; determines whether configuration variations can be tested in parallel or serially; helps establish a sound technical test strategy</td>
</tr>
<tr>
<td>Which configurations will be used most in deployment or pose the greatest risk to successful deployment?</td>
<td>Determines the test depth and frequency of test repetition; affects the test automation strategy</td>
</tr>
</tbody>
</table>

Figure 1. Factors that affect a test strategy

<table>
<thead>
<tr>
<th>Example test phase entrance criteria</th>
<th>Example test phase exit criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>• All test cases are available.</td>
<td>• All test cases were attempted.</td>
</tr>
<tr>
<td>• All test resources are available (both personnel and hardware).</td>
<td>• Test pass rates were high enough to warrant transition to the next phase.</td>
</tr>
<tr>
<td>• All technical specifications and user documentation are available for this phase of the test effort.</td>
<td>• All defect fixes were verified and regression on adjacent areas was successfully performed.</td>
</tr>
<tr>
<td>• Correct revision levels of hardware and software are installed.</td>
<td>• No unresolved high-severity defects exist that would adversely affect transition to the next phase.</td>
</tr>
</tbody>
</table>

Figure 2. Example test phase entrance and exit criteria
Test documentation

Test documentation at a minimum should include a test plan, test cases, and a test summary report. Suggested contents for each type of document are as follows:

- **Test plan**: Includes what is being tested; what is excluded from testing; the features to be verified; the configurations to be used; and an estimate of the resources and time required for testing.
- **Test cases**: Include the goal of each test; a detailed description of the test procedures; the required environment for test execution; whether testing will be automated or manual; and the criteria by which to judge whether testing passed or failed.
- **Test summary report**: Includes the dates and revisions for test execution; whether entrance and exit criteria were met; the test case pass/fail rates; how many defects were detected, during which test phase they were detected, and their severity level; whether any defects remained in the product when it was shipped or deployed; and a description of those defects.

Post-project assessment

Soon after testing is completed, the test team should review what went well and what went poorly during the project. In particular, they should capture the historical metrics for the project, as described in the “Measuring success” section in this article and in Figure 3. The results of this assessment should be used to improve planning and execution of future testing.

Post-project assessment may include how long testing took compared to the test plan, how many resources were employed compared to the test plan, and a comparison of the test effort with previous similar efforts.

Measuring success

In general, test measurements are sound only insofar as they are balanced. A balanced set of test measurements includes assessments of productivity, financials, quality, and timeliness.

As part of the test management strategy, test teams should establish measurement systems with three distinct objectives: operational measurements to track real-time progress through test execution (assessing the quality and release readiness of the product); historical measurements to assess the project (comparing the actual results of the test project with the test plan); and business measurements to drive organizational improvement over time (comparing performance to that of similar past projects or to industry benchmarks). Figure 3 provides example metrics that can be used.

Learning from test projects

At times, the complexity of managing a test project can rival the complexity of the systems or products being tested. Enterprises must develop a strategy, a test plan, and test cases in advance to meet the unique requirements of their environment. Furthermore, test teams must keep their projects on track through technical difficulties, product changes, and resource constraints. By taking balanced measurements of their progress, test processes, and product quality, test teams can help their enterprises create successful products—and the lessons learned from each test effort can help improve planning and execution for future projects.

Cynthia Lovin is a senior consultant test engineer in Dell’s Product Group Global Test Department, with 16 years of test- and quality-engineering experience in companies ranging from startups to Fortune 500 enterprises. She has Six-Sigma Green Belt certification and a B.A. in Business Administration from The University of Texas at Austin.

Tony Yaptangco is the director of the system test group within the Dell Product Group Global Test Department. He has 25 years of experience in various software and test environments, including 16 years in engineering management positions. Tony has a B.S. in Computer Science from San Diego State University and an M.S. in Engineering Management from National Technological University.