The Value Of Testing In A Migration Project

Testing For Success

The definition and creation of a viable test plan is critical to the success of a migration project. Used during various stages of the project, test plans serve to validate the baseline system, assist the developers in performing unit and system tests, and provide the framework for final acceptance testing.

Specific test plan details vary from project to project, however *Sector7* employs the same overall approach with the same basic requirements for all project test plans. This document describes test plan creation and usage at *Sector7*.

Overview

Generally speaking, the more time spent up front putting together a detailed test plan, the greater the chances for success in meeting project deliverables, schedules and expectations. Like many other similar exercises however, the law of diminishing returns comes into play at some point in time.

The danger lies in developing a plan with procedures so complex or comprehensive that it becomes impractical or too time-consuming to execute efficiently. The challenge then, is to define a test plan sufficiently detailed to ensure application integrity and validity; yet simple enough to allow efficient and repeatable execution.

It is, of course, important to consider that the time required to run the tests is dependent upon the speed of the platform on which the testing is being performed.

Subject Matter Experts

Subject Matter Experts, individuals thoroughly familiar with the application and all of its operational details, must ultimately construct test plans. The individual tests should be documented in sufficient detail such that testers not necessarily familiar with the application can efficiently execute them.

In some cases, comprehensive automated test suites exist and can be employed effectively for validation of migration projects. In cases where they do not exist, we recommend the use of manual procedures, scripted out in step-by-step detail. Undertaking the development and implementation of an automated test procedure can be an involved and significant task.

We realize that creating a test plan can be a resource intensive and time-consuming task. To help alleviate these problems, *Sector7* has experienced test managers available to work directly with clients in developing test plans tailored to their specific environment and project requirements. The rest of this appendix will describe the manner in which *Sector7* projects utilize test plans as well as the steps we recommend for test plan creation:

Test Plan Usage

Baseline Validation

The "baseline" system forms the agreed-upon starting point for a migration project. The overall objectives of baseline validation include verification of the fact that source code received from the client can be built and validated on the source platform at *Sector7*. This assures we are migrating the correct version of the code, and that we have all necessary source modules.

Successful execution of the test plan validates the system and forms the reference point for functionality comparisons. Successful test plans include mechanisms for capturing and saving output in the form of screen-prints or hard-copy reports.

Unit Testing

During the development and debugging stages, programmers need to be able to run parts of the application to recreate and track down problems. Although programmers may not execute entire scripts each time they wish to test, the tests should be sufficiently well documented such that basic program functionality can be inferred. Naturally, this assumes the client provides SMEs (Subject Matter Experts) to train *Sector7* personnel on application details and interpreting test results.

System Testing

After all modules are migrated and successfully unit tested the project enters the system test phase. During this phase the target environment is created, data is migrated or converted and the code moved to an isolated test system.

Installation and configuration procedures are devised and tested, and trial deployments performed to identify and document setup procedures, necessary hardware and software components along with the required revision levels. After the target environment is stabilized the test plan is used to perform system tests and ensure the migrated application functionality is complete and correct.

When allocating time for system testing, we normally allow triple the time it takes to perform the baseline test (e.g. if the baseline takes 60 hours to run end-to-end, then we allow 180 hours for system testing).

Acceptance Testing

Acceptance testing performed by the client at project completion validates the results of the entire migration project. It forms the basic agreement by both parties that the migrated code on the target platform is functionally equivalent to the code on the original source platform. It is imperative the plan be agreed upon by both parties prior to this task to ensure successful project conclusion.

In terms of project deliverables and risk management, this task represents the most critical utilization of the test plan. Care must be taken therefore, to ensure the plan specifically address application "hot spots" (i.e. heavy usage/traffic) as well as functional areas devoted to supporting "mission critical" business processes. Interfaces to external systems almost always present high exposure, so test plans usually stipulate individual detailed tests for each interface.

The final acceptance test generally takes the same amount of time as the baseline. The assumption is that once all the system testing is done, we then run through the baseline/acceptance from end to end on the basis that it will be clean and ready to pass over to the client.

Test Plan Creation

Defining the Testing Approach

The first step in creating a test plan is to define the testing approach. This need not be a complex or detailed process; it is often sufficient to simply indicate how different application areas will be tested. For example: on-lines will be tested by operators manually entering data from scripts, batch jobs will be submitted and reports saved, etc.

Besides specifying how the tests will be performed, consideration must be given to the logistics necessary to make the tests repeatable (e.g. restoring data to known state before each test run, etc.). In addition, the manner in which results will be captured, saved and compared to the baseline validation system should be carefully defined and documented.

Perhaps most critical to the beginning of testing is the definition of what activities and results constitute completion of the project. These activities are known collectively as *"Acceptance Criteria."* These acceptance criteria, included as part of the test plan, include a list of specific tests where the migrated code results will be compared with the results generated from testing the original code. The completion of testing and the comparison, or resolution of discrepancies encountered, is used to indicate completion of the project. The customer and *Sector7* are required to approve the specific conditions for acceptance testing.

Defining the Application Subset to Test

After deciding upon an overall test methodology, the next step defines a subset of the application to test. Obviously, it is not practical to attempt to validate every possible code path in a large complex application. Our experience shows that testing 70-80% of the modules is generally sufficient to validate the migration, depending on application flexibility and complexity.

There are several reasons for this:
- Analysis and profiling reveals the vast majority of application utilization is generally performed using a subset of the full application functionality.
- Profiling also reveals that even within highly utilized application functions, specific code paths are executed much more frequently than others.
- Applications are usually constructed with subroutines or library calls performing complex or common application functions. When these routines are indirectly tested within one functional test, there is usually no need to validate them from other functional tests as well.
- Migration projects are unique in the sense that they do not introduce new functionality into the application. The same overall approach is usually taken in migrating all modules (i.e. automated tools, procedures, etc.). This generally means that corrections to migration-related defects resolve all instances of the problem.

These reasons combined with the fact that test plans must be complete and detailed, yet simple enough to execute efficiently, support the approach of defining an application subset for testing. The difficulty lies with choosing the appropriate modules for inclusion in the plan. *Sector7* recommends that QA personnel developing the test plan consult application SMEs as well as technical architects to ensure the proposed plan includes sufficient coverage.

Defining Tests for External System Interfaces

Sector7 recommends the application subset approach not be taken for testing external system interfaces. These components normally represent high risk factors and are sufficiently different enough from each other to warrant individual test plans. This argument is reinforced by the fact that SMEs for the external systems are not usually available, and sometimes the tests must be performed by, or requires extensive support from, technicians.

Training Sector7 on Test Procedures and Interpreting Results

Throughout the migration project, engineers and testers perform partial or complete test procedures for unit and system testing. Furthermore, it is often necessary during problem analysis and troubleshooting, to run application tests under the debugger for investigation and resolution.

It is therefore necessary for client testers or SMEs to provide training for *Sector7* migration engineers and testers to enable them to run the tests independently and verify the correctness and completeness of observed results.

Define Test Results Approval Process

It is extremely important to specify the exact method by which test results will be captured, reported and approved. Forms or other control vehicles should be defined and created, sufficiently general enough to cover all types of individual test results. Other factors covered in this task include designating which individuals are responsible for approval, mechanics for communication and follow-up and establishment of turn-around time frame expectations.

Script Each Functional Test

Sector7 recommends that client QA personnel or SMEs actually produce written scripts for each functional test. These detailed documents list each prompt the tester sees when performing a test and indicate the data entry keystrokes necessary to advance to the next step within the test. Screen prints are not necessary, although they do help in quickly identifying format problems. The scripts also list expected results or output generated in response to keystrokes or actions taken by the tester.

The client needs to allocate sufficient time for this activity, as development of test scripts is not a trivial task. Similar to programs, all but the simplest of scripts must be executed multiple times and debugged by the author to ensure they accurately reflect the actual application dialog with the user.

Approval and Agreement by All Parties

The final step in defining a test plan is to obtain approval and agreement from all involved parties that the plan is viable and complete enough to fulfill the needs described above.