

Introduction

Testing covers the broadest range of activities required to both *verify* and *validate* deliverables associated with the project efforts and the final system components delivered into production.

- **Verification** addresses the issues surrounding the correctness of the deliverables produced with respect to previous project deliverables and general IT standards and procedures. Verification asks the question "Did we do it right?"
- **Validation** addresses the issues surrounding whether the deliverables produced will actually meet the needs of the user community to which they are targeted. Validation asks "Did we do the right thing?"

Ultimately, validation of the results of an information technology project is the goal of a test plan. However, experience shows that deliverables that can't be verified are difficult-to-impossible to validate.

Continuous Testing

Testing of an e-business system takes place constantly and continuously. The development and testing processes can be said to be intertwined from the day an effort begins to the day it ends. During each individual project effort (e.g. project, sub project, task, activity) three testing activities should be taking place:

1. Deliverables and work products should be verified to assure they conform to project, industry, and user expectations for form and content ("Is it done right?").
2. Verification should specifically include assuring that validation criteria have been defined for the subject described by the deliverables ("How will we determine *after the fact* that we've done the right thing?").
3. Validation against those criteria should be done *to the extent possible* given that many deliverables document future production and so cannot actually be used to demonstrate actual compliance to user needs ("Is it *likely* that we've done the right thing?").

Actual usage validation must wait until system components can actually be used in operation. This final testing, carried out at the latest points in project efforts, represents the highest cost correction path if defects are encountered, making early and continuous testing critical to the success of an e-business project.

Testing is a process, not a phase!

Setting Objectives

All systems have processes they perform and data they maintain. All systems also have certain attributes they must possess if the system is to be considered a success. Critical success factors are usually a combination of what the system must *do* (processes on data) and what the system must *be* (attributes) to be successful.

Testing needs to identify and focus on verifying and validating that these critical success factors are met. For factors considered important to the project, specific objectives should be defined by management as a guide to project team members and any independent test team staff.

Test objectives should be reviewed with all project stakeholders for completeness and agreement before moving on to more detailed test strategy development or detailed test planning.

Test Factors

Each project will prioritize the factors it considers critical depending upon unique characteristics of the system being built, its intended customer base, and the risks to the business that are associated with the system. Projects should select and prioritize from the following factors when setting high-level test objectives:

- **CORRECTNESS** - Extent to which the system must satisfy stated requirements. All requirements must be shown to be clear, complete, unambiguous, measurable, and testable. All design features must be shown to conform to the requirements.
- **AUTHORIZATION** - Extent to which the system processing requires management authorization. All relevant authorization rules must be defined as requirements and the design must include a method for implementing those rules.
- **INTEGRITY** - Extent to which the system capabilities must be accurate. System and database integrity requirements must be defined and mapped to the controls that are designed.
- **AUDIT TRAIL** - Extent to which the system capabilities must be traced. Reconstruction requirements must be defined, and an audit trail designed and implemented, to meet those requirements.
- **CONTINUITY OF PROCESSING** - Extent to which processing must be able to continue. The impact of any possible system failures must be defined during requirements and appropriate contingency plans developed during analysis and design.
- **SERVICE LEVELS** - Extent to which schedules must be met. The desired service levels must be identified as requirements and the method for achieving the desired service levels must be included in the design.
- **ACCESS CONTROLS** - Extent to which access to computer and systems must be restricted. The requirements statement must include a definition of access needs for each system function and the appropriate access procedures must be designed to support those needs.
- **METHODOLOGY** - Extent to which the development plan and process must be followed. The project plan must conform to all methodology and standards requirements in order to serve as a baseline for compliance testing.
- **RELIABILITY** - Extent to which the system cannot fail during operation. Tolerances must be established for all functions during requirements and data and process integrity controls must be designed to assure that the system functions within the established tolerances.
- **EASE OF USE** - Effort required to learn and use the system.
- **MAINTAINABLE** - Effort required to locate and fix errors in operational components of the system.
- **PORTABLE** - Extent to which the system can operate in multiple environments and/or on multiple browsers.
- **COUPLING** - Extent to which the system must integrate with other systems. Integration requirements must be defined and appropriate integration controls must be designed.
- **PERFORMANCE** - Extent to which the system must be able to perform its functions within specified limits. Operational constraints must be defined as requirements and all design features are tested against these constraints.
- **EASE OF OPERATIONS** - Effort required to operate the system. The target implementation environment and staff must provide specific operating requirements.

Note that there are no *unimportant* test factors. The priorities to be identified by the project represent *relative* priority within the overall testing effort. More test planning and execution will be conducted for those factors prioritized higher than others, whether such priority is based on an assessment of risk, cost, importance, visibility, or some other criteria.

Setting specific test objectives focuses attention and resources during the project!