TESTING RECOMMENDATIONS FOR INFORMATION SYSTEMS

CONTENTS PAGE

1. GENERAL INFORMATION

1.01 ***Purpose:*** This section contains a minimum set of recommendations for all phases of testing during system development and maintenance. These recommendations provide a basis for Bell System development organizations to define and implement procedures which will assure adequate testing of information systems. Adequate testing provides the means to determine the quality of the system, its design and its products. Details of test-related activities that should be considered during system development and maintenance can be found in Appendix 1, Testing Performance Aid.

1.02 ***Reason for Issue:*** This section is being issued to assure adequate testing of information systems and thus avoid leaving to chance the quality of those systems. This section was developed by a multicompany project team and replaces Comptroller's M-Letter M-447, Testing. Whenever this section is reissued, the reason for reissue will be stated in this paragraph.

1.03 ***Applicability:*** This section is issued as a guideline and can be applied to all Bell System entities responsible for the development of information systems. The recommendations apply to projects of any size. No particular organizational structure is implied. The recommendations apply to the development of new systems as well as to the enhancement of existing systems.

1.04 ***Organization of This Practice:*** This practice consists of the following four parts:

- Testing Recommendations: Outlines key areas of a testing program and provides recommendations for each area
- Testing Performance Aid (Appendix 1): Outlines the test-related work for system development and maintenance
- Structured Walk-Throughs (Appendix 2): Describes an approach used for the static testing of development products
- Test Plan Guidelines (Appendix 3): Provides an approach for developing test plans.

1.05 *Scope:* The quality of a system and its operating integrity can be assured by adequate testing throughout its entire life cycle—development, operation, and maintenance. This section addresses the development and maintenance stages of the system's life cycle. The recommendations outlined in subsequent paragraphs of this section are directed primarily at management—managers responsible for system development or maintenance; managers responsible for test facilities and administration; general management concerned with developing a more orderly structure for the testing of information systems. (See Section 007-208-310, Project Management, for a description of project management roles, responsibilities, functions, and products.)

1.06 This section provides recommendations for the following test-related areas:

- Static Testing
- Dynamic Testing
- Testing Strategy
- Test Plans
- Test Cases
- Test Data Base
- Test Results
- Regression Testing.

1.07 It is not the intent of this section to provide detailed procedures regarding specific testing techniques or testing methodologies. This is best left to the organizations responsible for the testing of systems.

2. DEFINITIONS

2.01 All terms and acronyms used in this section are defined in the Glossary of System Development Terms and Acronyms (Section 007-200-201).

2.02 ***Testing*** is a process with the capability to expose deficiencies in the quality of a product. In terms of results, testing is a measurement of the total system (Computer Subsystem [CSS] and Personnel Subsystem [PSS]) to perform within its intended operating environment according to the system's requirements.

2.03 There are two basic methods of testing—***static*** and ***dynamic.*** Both methods are required as part of system development and maintenance.

3. TESTING RECOMMENDATIONS

A. Static Testing

3.01 Static testing is the verification (usually manual) of system development products for accuracy, completeness, and consistency.

3.02 Static testing includes reviews, walkthroughs, and inspections. It is especially important during the early phases of system development when the only products that can be tested are developmental products such as the system requirements and the design specifications.

3.03 Static testing has the following four purposes:

- To identify and correct errors with development products before proceeding with further developmental activities
- To improve the quality of design with respect to measurability, correctness, completeness, etc
- To improve communication between members of the development team for the exchange of information concerning design
- To provide feedback to the project manager at recognizable milestones.

Recommendations

3.04 As a minimum, static testing must be performed for the following developmental products:

- System Requirements (eg, Input, Output, Control, Conversion)
- Design Specifications (eg, Hardware, Software, Data Base, PSS/CSS Interface).

3.05 Deliverable documentation for systems must also undergo similar testing reviews. (See Section 007-230-210, Systems Deliverable Documentation.)

3.06 Static testing of additional products will be determined by the project manager. Specific procedures and schedules for static testing will be determined by the development organization with the concurrence of the project manager.

3.07 Additional information regarding static testing procedures is provided in Appendix 2 of this section.

B. Dynamic Testing

3.08 Dynamic testing involves the actual execution of test cases to ensure that the system's software and manual procedures function properly.

3.09 Dynamic testing is based on two complementary approaches—the ***level of detail*** and the ***type of environment***. In the first approach, the level of detail is varied, beginning with the testing of individual system components (eg, programs or work modules) and ending with the testing of the total system. In the latter approach, the testing environment is varied, beginning with a test environment using controlled data and ending with an operating environment using real data. This structure provides a logical building block procedure for testing. Table A shows the relationship between levels of testing detail and the types of testing environment associated with each level.

Recommendations

3.10 When testing is performed, the following three levels of detail must be examined:

- ***Unit Testing*** examines each discrete component (module, program, work module, or procedure) of the system.
- ***Integration Testing*** examines how each component of the system interacts with each other as they are assembled in a stepwise manner. It concentrates on chained programs or work modules and human/machine interfaces.
- ***System Testing*** examines the operation of all components of the system as a whole, according to its system requirements and performance criteria.

3.11 As each level of testing detail is performed, the following types of environment should be considered:

- ***Verification Testing*** examines the logical correctness of each component (module, program, work module, developmental component, or procedure) either individually or together, using controlled data in a test environment.
- ***Validation Testing*** examines the logical correctness of the system using controlled data in the operating environment or one that approximates the operating environment as closely as possible.
- ***Certification Testing*** examines the performance, quality, and reliability of the system to ensure that the system meets its objectives and its performance requirements. This type of testing is conducted in the actual operating environment using real data. Certification emphasizes the performance aspects (eg, volume, response time, fallback, service agreements, recovery, etc) of the system rather than the logical correctness of the system.

3.12 ***Acceptance testing*** is conducted as an extension of system testing and completes certification testing. It is performed by or on behalf of those who will use and operate the system to ensure that the user's needs are met.

3.13 For a Centrally Developed System, an Operating Telephone Company (OTC) may require that acceptance testing be successfully completed as a condition for accepting a system or a system release. The project manager will coordinate acceptance testing requirements with OTC Information Systems Management.

3.14 The testing recommendations outlined above apply to both the initial system release and subsequent scheduled releases. An additional recommendation for the regression testing of subsequent releases is discussed in paragraph 3.38.

TABLE A

LEVEL OF TESTING VERSUS TYPE OF ENVIRONMENT

LEVEL OF TESTING DETAIL	PURPOSE	TESTING CRITERIA	TYPE OF TESTING ENVIRONMENT
UNIT	To prove the logical correctness of each component within the Computer Subsystem and Personnel Subsystem	SPECIFICATIONS	VERIFICATION • Test Environment • Controlled Data
INTEGRATION	To prove the interfaces between components perform logically when integrated into the system	SPECIFICATIONS	VERIFICATION • Test Environment • Controlled Data
			VALIDATION • Operating Environment or Close Approximation • Controlled Data
SYSTEM	To prove the complete system performs according to its system requirements and performance criteria	REQUIREMENTS	VERIFICATION • Test Environment • Controlled Data
			VALIDATION • Operating Environment or Close Approximation • Controlled Data
			CERTIFICATION • Operating Environment • Real Data

C. Testing Strategy

3.15 To develop a comprehensive testing program, all testing activities are planned, budgeted, and integrated into the system's overall development plan. This process begins during the Feasibility Phase as input to the cost/benefit analysis of the proposed system. Plans for test-related activities are further refined in each succeeding phase. For additional details regarding the testing activities performed in each phase, see Appendix 1, Testing Performance Aid.

3.16 Procedures are also developed and maintained to ensure control of the testing process. The procedures should provide project management with the information necessary to measure progress and detect problems during the testing process.

Recommendations

3.17 Detailed phase plans and end of phase reports must contain testing information, including testing activities, schedules, and resources. The project manager is responsible for determining the adequacy of the information.

3.18 Procedures to track and report testing progress on an established interval must be developed. The specific tracking procedures, status reports, and formal sign-off procedures will be determined by the organization(s) responsible for testing with the concurrence of the project manager.

D. Test Plans

3.19 Development of formal test plans is a key element in the testing effort.

Recommendations

3.20 Documented test plans are required for every new system. For revisions or enhancements to existing systems, test plans must be revised, or new ones written, as warranted by the scope of the modification.

3.21 As a minimum, the contents of a test plan for each level of testing must include:

- Objectives, including dependencies and measurable acceptance criteria consistent with the system's requirements
- Individual components to be tested
- Test environment
- Test cases
- Schedules and resources.

3.22 Upon completion of the test plan or major revisions to it, the plan must be reviewed by the developing organization and accepted by the project manager before it can be implemented. Refer to Appendix 3 of this section for further guidelines on test plan content.

E. Test Cases

3.23 A test case represents a specific set of data and associated procedures used to test a specific function or a logical grouping of functions. To prepare for dynamic testing, test cases should be developed in conjunction with the building of the test data bases. For large or complex systems, test data generator packages should be considered to assist the test case development effort.

Recommendations

3.24 Test cases must consist of both "friendly" and "unfriendly" data and must adequately represent conditions that will be encountered in a live environment.

3.25 Friendly data are those which meet valid input requirements and test for correct component processing under normal conditions. They are directed primarily at testing the system's logic and procedures. Valid inputs are specified in the Input Requirements, Group Element Definition, etc. Unfriendly data are those that do not satisfy valid input requirements, but test the functioning of a component or system under abnormal and unpredictable conditions. They are directed primarily at testing the system's controls. Including unfriendly data in test cases increases the likelihood of detecting those errors that result from improper input to the system.

3.26 Expected results must be included with each test case (eg, run times, response times, outputs, data base updates, error handling, volumes, recovery capabilities, etc).

F. Test Data Base

3.27 A test data base is a permanent collection of files, tables, and data which is used to measure the ability of a system to perform to a defined confidence level. Since the users possess an in-depth knowledge of the data that the system will process, the development of a test data base should be a cooperative effort of those responsible for testing and the system users.

3.28 The system design specifications provide the basis for the development of the test data base. An arbitrary collection of live data, regardless of size, is not sufficient to populate a test data base. It is impossible to accurately assess the degree to which criteria in the design specifications are represented in such data. Copies of live files can be an excellent volume supplement, but are not a replacement for a test data base.

3.29 System developers are responsible for maintaining the privacy and integrity of live data used during system testing.

Recommendations

3.30 The test data base and its related environment must be created from system design specifications. The test data base must logically parallel the proposed system.

3.31 The test data base must be maintained throughout the system life cycle with adequate resources allocated to keep it current.

G. Test Results

3.32 Test results provide a direct measure of the system's completeness, accuracy, and performance. They are compared with the expected results as specified in the related test cases used for the test. The test results should be documented for future reference (troubleshooting, problem resolution, etc).

Recommendation

3.33 As a minimum, the system test results must be documented to include the following information:

- Actual test results (eg, run times, response times, access and accountability controls, outputs, data base updates, service objectives, error handling, volumes, recovery capabilities, etc)
- Deviations from expected results and identification of the probable causes (eg, procedural documentation, inadequate test cases, design faults, program bugs, training deficiencies, environmental)
- Modifications Requests (MRs) encountered, including those corrected and those still outstanding
- Descriptions of any agreed-to solutions for correcting unacceptable deviations or the rationale for accepting other deviations.

H. Regression Testing

3.34 Acceptance of a system by its user marks a new stage in its life cycle. Since change is a normal condition for a system, it is expected that the system will be enhanced and maintained throughout its life cycle.

3.35 Two questions must be addressed to all system changes:

- To what extent does the change impact the existing system?
- Does the existing system work as planned after incorporating the change?

3.36 In addition to performing the static and dynamic testing outlined in previous paragraphs, a program of regression testing for all changes will satisfactorily answer these questions.

3.37 Regression testing examines a modified module or system using a known set of data, or examines modified data using a known set of modules or the system. It ensures that the modification has not altered any process or data outside the scope of the modification. The regression test package is made up of test cases created during system development.

Recommendations

3.38 A program of regression testing must be established for all scheduled system releases after the system's initial release. This does not preclude the establishment of a regression testing program prior to initial release. Such a program is encouraged especially for those systems whose development cycle exceeds 1 year. The regression test package and test data base must be maintained to ensure that they are kept current.

3.39 Acceptance testing, which checks for the proper operation of a release in the live environment, is required for each scheduled system release.

4. RELATED SECTIONS

4.01 The following sections provide additional test-related information:

(a) Project Management—Section 007-208-310.

(b) Review Procedures for the Conversion and Operations Impact of Centrally Developed Systems—Section 007-210-320.*

(c) System Control Standards—Section 007-209-201.

(d) Developmental Documentation and Developmental Documentation Specifications—Sections 007-227-305 and 007-227-310, respectively.

(e) Systems Deliverable Documentation—Section 007-230-210.

*Check Divisional Index 007 for availability.