

A11102 483513

NAT'L INST OF STANDARDS & TECH R.I.C.



A11102483513

Wallace, Dolores R/An overview of comput  
QC100 .U57 NO.500-136 V1986 C.2 NBS-PUB-

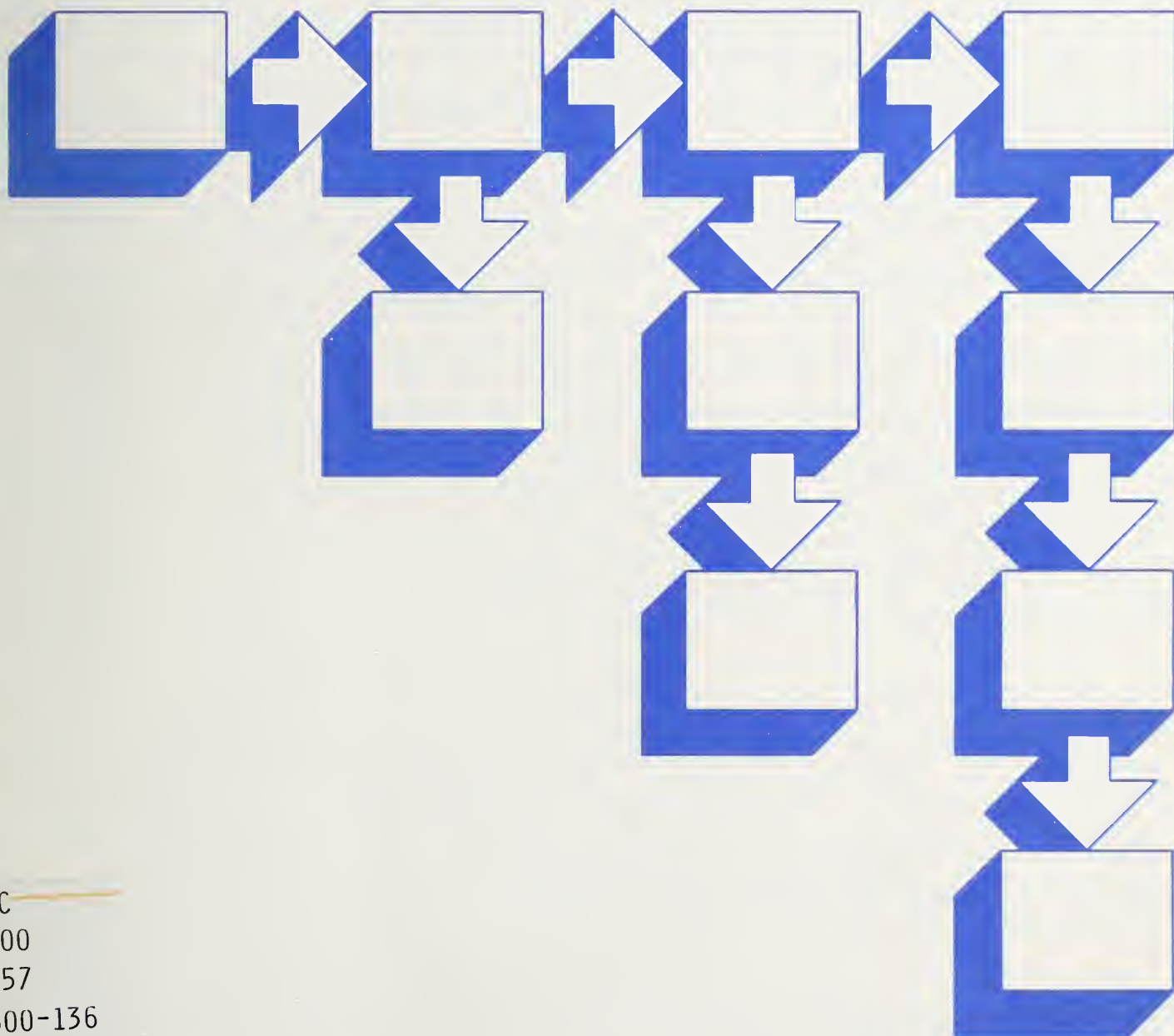
NBS

PUBLICATIONS

# Computer Science and Technology

NBS Special Publication 500-136

## An Overview of Computer Software Acceptance Testing



QC

100

.U57

500-136

1986

C. 2



The National Bureau of Standards<sup>1</sup> was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, the Institute for Computer Sciences and Technology, and the Institute for Materials Science and Engineering.

### *The National Measurement Laboratory*

Provides the national system of physical and chemical measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; provides advisory and research services to other Government agencies; conducts physical and chemical research; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

- Basic Standards<sup>2</sup>
- Radiation Research
- Chemical Physics
- Analytical Chemistry

### *The National Engineering Laboratory*

Provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

- Applied Mathematics
- Electronics and Electrical Engineering<sup>2</sup>
- Manufacturing Engineering
- Building Technology
- Fire Research
- Chemical Engineering<sup>2</sup>

### *The Institute for Computer Sciences and Technology*

Conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

- Programming Science and Technology
- Computer Systems Engineering

### *The Institute for Materials Science and Engineering*

Conducts research and provides measurements, data, standards, reference materials, quantitative understanding and other technical information fundamental to the processing, structure, properties and performance of materials; addresses the scientific basis for new advanced materials technologies; plans research around cross-country scientific themes such as nondestructive evaluation and phase diagram development; oversees Bureau-wide technical programs in nuclear reactor radiation research and nondestructive evaluation; and broadly disseminates generic technical information resulting from its programs. The Institute consists of the following Divisions:

- Ceramics
- Fracture and Deformation<sup>3</sup>
- Polymers
- Metallurgy
- Reactor Radiation

<sup>1</sup>Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Gaithersburg, MD 20899.

<sup>2</sup>Some divisions within the center are located at Boulder, CO 80303.

<sup>3</sup>Located at Boulder, CO, with some elements at Gaithersburg, MD.

# Computer Science and Technology

---

NBS Special Publication 500-136

## An Overview of Computer Software Acceptance Testing

Dolores R. Wallace

Center for Programming Science and Technology  
Institute for Computer Sciences and Technology  
National Bureau of Standards  
Gaithersburg, MD 20899

Issued February 1986



**U.S. DEPARTMENT OF COMMERCE**  
Malcolm Baidrige, Secretary

**National Bureau of Standards**  
Ernest Ambler, Director

## **Reports on Computer Science and Technology**

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

Library of Congress Catalog Card Number: 86-600502  
National Bureau of Standards Special Publication 500-136  
Natl. Bur. Stand. (U.S.), Spec. Publ. 500-136, 22 pages (Feb. 1986)  
CODEN: XNBSAV

U.S. GOVERNMENT PRINTING OFFICE  
WASHINGTON: 1986



## **ABSTRACT**

This document provides guidance in planning and managing acceptance testing of computer software. It emphasizes the need for quantitative acceptance criteria and itemized test cases and procedures. It provides a checklist of activities to be performed for planning and managing acceptance testing.

## **KEYWORDS**

software acceptance criteria, software acceptance testing, test cases, test management, test planning, test procedures.

## TABLE OF CONTENTS

1.	INTRODUCTION	1
2.	ACCEPTANCE TESTING DEFINED	1
3.	ACCEPTANCE TEST MANAGEMENT	3
4.	ACCEPTANCE TEST PLAN	4
5.	ACCEPTANCE CRITERIA	6
6.	ACCEPTANCE TEST CASES AND PROCEDURES	7
7.	ANALYSIS OF ACCEPTANCE TEST RESULTS	8
8.	SIGN-OFF	9
9.	SUMMARY	9
10.	REFERENCES	10
	APPENDIX A. ACCEPTANCE TEST CHECKLIST	11
	APPENDIX B. BIBLIOGRAPHY FOR ACCEPTANCE TESTING	15

## 1. INTRODUCTION

Under the Brooks Act, the National Bureau of Standards Institute for Computer Sciences and Technology (ICST) promotes the cost effective selection, acquisition, and utilization of automatic data processing resources within Federal agencies. ICST efforts include research in computer science and technology, direct technical assistance, and the development of Federal standards for data processing equipment, practices, and software.

ICST has published several documents on software verification, validation and testing as part of this responsibility. This report is addressed to those who purchase, develop, test, or use software and to those who manage these software activities. It discusses management responsibilities and provides technical direction for the acceptance testing of computer software. The basic definition of acceptance testing given in the "Guideline for Lifecycle Validation, Verification, and Testing for Computer Software," FIPS PUB 101 [FIPS101] is expanded to include additional analysis activities.

This report is directed principally to the acceptance of custom-built software (e.g., a new software system or a new release of an existing software system developed for a specific customer.) Management responsibilities for acceptance testing are summarized, with a more detailed checklist of activities provided in Appendix A. A list of topics to be included in an acceptance test plan and suggestions for developing quantitative, requirements based acceptance criteria are presented. Different approaches to acceptance testing are cited and acceptance testing activities throughout the software lifecycle are discussed.

## 2. ACCEPTANCE TESTING DEFINED

Acceptance testing is defined as "formal testing conducted to determine whether a software system satisfies its acceptance criteria and to enable the customer to determine whether to accept the system" [FIPS101]. Formal testing includes the planning and execution of several kinds of tests, (e.g., functional, volume, performance tests) to demonstrate that the implemented software satisfies the customer requirements for the software system. In addition to functionality, customer software requirements may include attributes such as hardware/software optimization, user interfaces, and effective program documentation [HOLL73]. Acceptance testing should demonstrate that all of the customer requirements are satisfied. The complete set of system requirements and acceptance criteria form the basis from which to determine the overall approach to acceptance testing and the specific testing and examination methods. The tasks of acceptance testing are shown in Figure 1.

Acceptance testing may be included as part of the validation, verification and testing (VV&T) program for computer software, or it may be managed independent of the VV&T program. It is important that the distinction between the objectives of VV&T and acceptance testing is clearly understood. The objectives of VV&T are to discover errors and to show correctness while the objective of acceptance testing is to demonstrate that the software system is acceptable to the customer. Of course, acceptance testing may also uncover additional errors. For both VV&T and acceptance testing, planning efforts begin early in the life cycle. VV&T has high visibility throughout the entire software development process, but for acceptance testing the principal testing activities usually occur after system testing. System testing (typically concerned with integration, file handling, correct parameter passing, validation of the system to the design, etc.) is usually performed in the developer's environment. Acceptance testing is

performed in the production environment to show that the system operates as the customer expects it to [FUJI78].

Acceptance testing is important to both the developer and the customer, even when the organizations are the same. For assurance that the software functions properly, the developer may perform complete acceptance testing, prior to releasing the software product to the customer for the customer's acceptance testing. The IEEE Standard Glossary of Software Engineering Terminology refers to this type of acceptance testing as "qualification testing" [IEEE729]. Qualification testing provides developers the opportunity to locate and modify difficult procedures and causes of program failures before customers try the program. Customers whose first acceptance test of a program is highly unsatisfactory may lose confidence in that program, regardless of later successful acceptance tests on a modified version of the program [GLAS79].

In many instances, the developers send customers a program copy with a baseline set of acceptance tests already executed at the developer's site. Customers then execute the baseline set to verify that the program has been correctly installed. Customers should expand this baseline set to verify that features not included in the developers' test set will function appropriately. Acceptance testing is performed either by, or in the presence of, the software customer, at the customer site.

---

## **TASKS for ACCEPTANCE TESTING**

- o Define the acceptance criteria
- o Determine the test approach
- o Determine organization, resource, other needs
- o Select analysis techniques
- o Write the acceptance test plan
- o Review the acceptance test plan
- o Design test cases
- o Write test procedures
- o Review for test readiness
- o Execute tests
- o Analyze tests
- o Specify accept /reject recommendations

---

**Figure 1. Acceptance test activities**



### 3. ACCEPTANCE TEST MANAGEMENT

Effective acceptance testing requires effective management skills as well as technical expertise. The acceptance test manager should be an experienced staff member and should be included in the early project decisions that will affect the acceptance testing. The acceptance test manager defines the objectives of the task and a plan for meeting them. The manager implements the plan and determines if the implementation has met the objectives [CELE81].

Planning is one of the most important functions of the acceptance test manager. Defining the objectives and a plan to meet them requires careful thought and covers all aspects of the acceptance testing process. The test manager should have the acceptance test plan ready for inclusion in one of the early major project reviews (usually the design review). Among early decisions that affect the planning are those establishing the organization (e.g., who will design the tests, who will execute them) and those determining the methodology for the dynamic testing and other evaluations.

There are various ways of organizing acceptance testing:

1. developer plans the tests; customers witness test performance;
2. developer and customers plan and perform tests together;
3. customers plan and perform tests;
4. customers hire third party to plan tests; customers execute the tests;
5. customers hire third party to plan and execute the tests.

The test objectives and the size and the purpose of the software system may affect the decision of how to organize the acceptance testing. Prior to installation, developers may perform the entire acceptance test, ( "qualification testing") [IEEE729]. For execution in the customer's environment however, the developer may design tests which demonstrate only those features known to work. The customer, on the other hand, should design stressful and unusual cases to assess how bug-free, trouble-free and acceptable the software is [SHOO83]. Third party acceptance testers are sometimes hired by the customer to provide an unbiased opinion. If the customer is a customer-service group for a large computing facility, the end users may actively test the system. Acceptance testing may also provide assurance that the program operating in production status does not adversely affect users of other programs [HOLL73].

In many cases, the customer and developer may work together in setting up the testing procedures. For a small project, a customer may readily perform all tasks for the acceptance testing; for a larger project, the customer may need to interact with the developer and may have many personnel assigned to different acceptance test tasks. The type of software system may affect acceptance test decisions, e.g., the choice of organization to perform acceptance testing. If the system is one whose failure could cause severe damage to life or financial systems, the test manager may be more likely to want an independent evaluation than if system failure would cause only minimum inconvenience.

The test plan should define a general methodology (e.g., testing during a pilot operation, in parallel with the existing system, benchmarking, simulation, etc). When the acceptance testing is to be performed in order to aid a selection process, (e.g., from several similar computer programs or from similar language processors), test cases should be designed that are readily repeatable and test all customer requirements. In this case the acceptance test process is more like a benchmark test [GILS77]. Sometimes the software system may be installed in several locations. Then, the users require a

baseline set of tests to check for verification that the software has been installed successfully and that the software performs as required at all installations.

The test manager has responsibility for acceptance criteria and overall test procedure. The acceptance criteria should be testable, measurable, complete and should provide adequate coverage of the customer software requirements. The overall test procedure provides for ensuring that test cases and procedures are written and executed for all test requirements. An identification scheme tracing the acceptance criteria to the test requirements, cases, procedures and results may aid in verifying adequate test coverage and completeness of execution. The overall test procedure must also specify techniques for collecting and analyzing the test results.

Appendix A provides an itemized checklist of acceptance test concerns and events. Some of the checklist steps may be eliminated or others added or rearranged according to the needs of the project.

#### **4. ACCEPTANCE TEST PLAN**

The acceptance test plan is prepared during the definition of the requirements and is completed in time for the design review. When written by someone other than the customer, the plan is also reviewed to obtain agreement on what tests must be successfully completed [MULL77]. Two Federal software engineering guidelines [FIPS38, FIPS101] provide outlines for test plans; either format may be tailored to include information for acceptance testing as shown in Figure 2.

---

### **ACCEPTANCE TEST PLAN CONTENTS**

#### **Project Description**

- Brief summary

#### **Management**

- Organization and responsibilities
- Controls and procedures
- Resource and schedule estimates
- Reviews
- Constraints
- Requirements for special tools, data, training

#### **Test Overviews**

- Test objectives
- Acceptance criteria
- Test requirements
- General approach
- Test and examination techniques

#### **Test Execution**

- Data collection and analysis
- Begin, stop conditions
- Anomaly handling
- Configuration control procedures
- Test library

#### **Sign - Off**

---

**Figure 2. Topics to address in acceptance test plan**

The acceptance test plan should provide a brief project overview and sufficient information for proper design, execution, monitoring and analysis of the acceptance testing. The organization responsible for the acceptance testing, staffing, control procedures, review mechanisms, and sign-off authority should be specified. Resource and schedule estimates, test data requirements for test data, software tools, staff training, and other special needs should be included. Contingency planning (e.g., change in project plan affecting timing of acceptance testing) should include, if possible, an alternative method for completing the acceptance testing.

The acceptance test overview describes the acceptance test objectives. The approach to the acceptance test and the details of the test environment should be clearly defined. The test approach may be determined by the installation technique for a software system. For example, a new software system may be installed and tested while customers continue to use an older system; this is an approach referred to as parallel testing. Alternatively, the new software system may be installed and tested in parts, while phasing out the old system. A software system may be installed with different hardware/software configurations at various sites. Real time software may require simulation testing or other techniques. In each of these situations, the selection of test techniques and organization of test cases may be different.

The acceptance test plan presents the acceptance criteria in quantified terms and provides a mechanism for tracing the acceptance criteria to the customer software requirements. The test requirements are defined from the acceptance criteria. The plan should identify a method to ensure that all test requirements are included in the test cases and procedures.

The plan should contain an overview of all the analysis techniques to be used and the procedures for reporting and analyzing the results of acceptance testing. Some techniques may involve manual examination or other methods that do not depend on program execution. Examples of test and examination techniques include but are not necessarily restricted to:

- o functional tests (e.g., a complete payroll process);
- o negative tests (e.g., testing with bad data to check error recovery);
- o special cases (e.g., testing limits or non-existent paths);
- o document examination;
- o stress testing (e.g., large amounts of data, many users);
- o recovery testing (e.g., hardware system crashes, bad user data);
- o maintainability evaluation;
- o user procedure testing (e.g., for starting or stopping system);
- o user friendliness evaluation;
- o security testing.

The acceptance test plan provides general information for the tests; details for the execution and analysis of each test are provided in the test cases and procedures. The plan may also specify the requirements for a test library. A test library that contains test execution histories as well as the test cases and procedures may ensure that acceptance testing of later releases of the software system exercise the appropriate set of tests for a specific release.



## 5. ACCEPTANCE CRITERIA

Acceptance criteria are necessary to determine if the implementation of the acceptance test plan has achieved the test objectives. Broadly, the test objectives are to demonstrate that the software system satisfies customer requirements. The development of the acceptance criteria should occur when the requirements for the software system are initially defined. The requirements document should specify the functions of the software system, descriptions of the system features, and deliverable items, (e.g., user documentation). The acceptance criteria should specify baselines against which the performance of the functional requirements and other features and deliverables of the software system will be evaluated.

Figure 3 lists selected software deliverables. Figure 4 lists typical software qualities for which the deliverables might be examined. These figures do not include all qualities for all deliverables and do not imply inclusion of all items in the software requirements; they serve only as a checklist. Those planning the acceptance tests may examine the software requirements for inclusion of any of these items and define criteria and tests based upon them. For example, when security requirements exist, management must ensure that criteria for security are included in the acceptance criteria.

---

### SOFTWARE DELIVERABLES

Requirements specifications	Source listings
Design documents	Software code
Configuration control procedures	Support software
Quality assurance plan	Training
VV&T plan	User manual
VV&T summary report	System manual
Test library	Operation manual

---

**Figure 3. Selected software deliverables**

The acceptance criteria are the baselines against which the software requirements will be tested. They are a verifiable set which can be traced to the software requirements. The criteria may include a functional description of a complete process from which a test case may be developed (e.g., a complete payroll process, an inventory update, or a text-preparation session). The development of a well-defined set of criteria depends on a well-defined set of requirements. Words such as "multi-user," "fast response time," "large sample" are nebulous but the phrases "100 to 120 simultaneous users," "1 second response for clearing the screen, draw 2nd frame," "1000 data points per curve" provide testable requirements. Similar approaches should be used to describe required accuracy for mathematical functions and other functional capabilities. Criteria (e.g., standards, human factors) against which documents will be assessed need to be stated. Finally the criteria should state clearly the acceptance or rejection policy concerning items such as the number and types of unresolved anomalies, the types of patches, and inconsistencies between the code and the documentation and the user procedures and documentation.

Software requirements frequently are changed between their initial definition and the completion of the software system. When this occurs, there can and should be iteration



---

## SOFTWARE QUALITIES

Correctness	Completeness
Functionality	Flexibility
Ease of installation	Integrity
Performance	Portability
Maintainability	Reliability
Standards adherence	Security
Adequacy of documents	Usability
Testability	

---

**Figure 4. Selected software qualities**

between the requirements and the acceptance criteria to ensure that both sets remain well-defined and measurable. At the time of testing or other examination the acceptance criteria may be used to build a matrix relating software requirements to tests showing achievement of the requirements.

### 6. ACCEPTANCE TEST CASES AND PROCEDURES

Design of test cases that completely test a system is often a very difficult task. The total complement of acceptance tests may test individual features, combinations of features, or total system operation and may also specify examinations that do not require program execution.

Ideally, acceptance testing should test all possible features and functions and combinations of them. However, resource constraints may limit testing to only some of the possible combinations of program features for a system. In such situations, test designers need a basis or methodology for selecting features to test, (e.g., features that will be used most frequently, those whose errors would bring down the entire system, or functions that are particularly difficult to transform to code). Test data design approaches such as that developed by Redwine [REDW83], although intended for system testing, may sometimes be extrapolated to acceptance testing. The test cases should also be carefully designed to cover features external to the program functions that are related to performance and stress (e.g., restart capability, large data files, user errors with on-line systems). Readable, understandable and complete requirements specifications and quantitative acceptance criteria aid in test case selection and test design.

Each test case consists of the overall scope, a description of the features to be tested, a stated objective, a design, and a procedure description. The procedure description contains input data, expected output data, and details for the execution of each test. Procedures may contain more information than shown in Figure 5. The goals are to provide all necessary information to the tester at the time of execution and to enable a test to be easily repeated under the same conditions.

Acceptance testing should be performed in a systematic manner. The total acceptance test procedure should specify the order in which the various tests and examinations will be executed. Clearly-defined procedures for the individual test cases enable

---

## WRITTEN TEST PROCEDURES IDENTIFY:

- o Test objective
- o Personnel responsible for test
- o Necessary equipment, location of test
- o Input materials, e.g., support software
- o Input data
- o Expected output
- o Constraints
- o Required test preparation, setup
- o Operator actions, console setups
- o Prerequisite testing
- o Acceptance criteria
- o Retest criteria for failed test
- o Step-by-step execution instructions

---

**Figure 5. Details contained in test procedures**

methodical execution of the tests, repetition of the tests, and easier identification of problems within the acceptance test cases. An acceptance test readiness review, just prior to the acceptance testing, will help to ensure that all preliminary arrangements for test execution have been completed and that the appropriate system configuration and test materials are ready.

## 7. ANALYSIS OF ACCEPTANCE TEST RESULTS

Test design includes design of methods to record and to analyze test results. A method for comparing the expected output data with the actual output data should be planned in advance of the test execution; combinations of automated comparisons or manual examinations may be used. Algorithms necessary for complete analysis of the data should also be specified in advance of the test execution. Some system features (e.g., timing, size, security), may require special software or hardware for recording and comparison. For example, if a one second screen response is required, then there must be a means of recording the response time. In other cases, the users may need to specify means of capturing a picture of the entire screen for comparison with the requirements.

Without clearly-defined specifications, analysis of documentation may be difficult. Requirements for user features need to be quantified in the acceptance criteria. Criteria for a simple system might specify that customers be able to successfully use the system after one hour of training. For complex systems, the criteria might specify user training and experimentation for a longer period of time but with an established limit. Acceptance of a document may depend on examination. For example, a user manual may be examined for standards adherence as well as for descriptions of features, types of input and output data, and procedures for using the system, such as timing estimates and file setups. If the developer's requirements for quality assurance, VV&T, and configuration management are to be checked for adequacy and implementation, then an examination method should be specified in the plan.

## 8. SIGN-OFF

The purpose of acceptance testing is to enable the customer to determine whether or not the system should be accepted. The degree of formality of this acceptance should be decided when the acquisition contract is signed, and should be adhered to during acceptance testing. The level of formality may include test witnessing during key acceptance tests; in other cases, the customers may execute the tests.

The recommendation to accept or reject the software may be written with caveats. For example, a recommendation to reject may cite situations in which the software would cause serious effects if anomalies were not corrected. Another recommendation may cite reasons to accept software even with some uncorrected defects or some components missing. The sign-off procedures, including the name of person or agent of the person who has sign-off authority, should be clearly stated.

## 9. SUMMARY

This report addresses management concerns and technical direction for the acceptance testing of computer software. The acceptance test manager should define the following items prior to acceptance test execution:

- o comprehensive acceptance test plan
- o quantified acceptance criteria
- o test methodology
- o detailed checklist of all acceptance test activities
- o itemized, repeatable test cases and procedures
- o methods for collection and evaluation of test results
- o iteration procedures for changes to test items
- o sign-off criteria.

Acceptance testing of computer software enables customers to determine whether or not a software system satisfies its requirements and is ready to be accepted. Acceptance testing requires careful planning during requirements definition. Developers and customers need to work together to determine a set of quantified acceptance criteria for the software functional requirements and other features of the software products. Acceptance testing may include various types of analyses in order to demonstrate that these criteria have been met. Detailed test and examination procedures help to ensure that a specific analysis technique is correctly applied to the software feature for which it was designed.

Developers may perform the acceptance test at their sites in order to satisfy themselves that the software system qualifies for acceptance. Final acceptance testing, however, is generally performed in the customer environment, either by, or in the presence of, the customer.



## 10. REFERENCES

[CELE81]

Celentano, A., C. Ghezzi, F. Liguori, "A Systematic Approach to System and Acceptance Testing," *Computer Program Testing*, B. Chandrasekaran and S. Radicchi, (ed.), North Holland Publishing, 1981.

[FIPS38]

"Guidelines for Documentation of Computer Programs and Automated Systems," Federal Information Processing Standards Publication 38, National Bureau of Standards, February, 1976.

[FIPS101]

"Guideline for Lifecycle Validation, Verification, and Testing of Computer Software," Federal Information Processing Standards Publication 101, National Bureau of Standards, June, 1983.

[FIP102]

"Guideline for Computer Security Certification and Accreditation," Federal Information Processing Standards Publication 102, National Bureau of Standards, September, 1983.

[FUJI78]

Fujii, Marilyn S., "A Comparison of Software Assurance Methods," *Proceedings of the Software Quality Assurance Workshop*, Nov.15-17,1978, San Diego, CA, Association for Computing Machinery, New York, New York, 1978.

[GILS77]

Gilsinn, David E., J.Trotter Hardy, Charles L. Sheppard, "The NBS Validation Routines for the BASIC Language," *IEEE Proceedings of COMPSAC 77, 1st International Computer Software and Applications Conference*, The Institute for Electrical and Electronics Engineers, New York, NY, 1977.

[GLAS79]

Glass, R.L., *Software Reliability Guidebook*, Prentice Hall, 1979.

[HOLL73]

Holland, J. Grant, "Acceptance Testing for Application Programs", *Program Test Methods*, Hetzel, W.C., (ed), Prentice Hall, Englewood Cliffs, NJ, 1973.

[IEEE729]

"IEEE Standard Glossary of Software Engineering Terminology," ANSI/IEEE Std. 729-1983, The Institute for Electrical and Electronics Engineers, 345 East 47th St., New York, NY 10017, 1983.

[MULL77]

Mullin, Frank J., "Software Test Management for \$2M and Up," *IEEE Proceedings of COMPSAC 77, 1st International Computer Software and Applications Conference*, The Institute for Electrical and Electronics Engineers, New York, NY, 1977.

[REDW83]

Redwine, S.T., "An Engineering Approach to Software Test Data Design," *IEEE Transactions on Software Engineering*, Vol.SE-9, Number 2, March 1983.

[SHOO83]

Shooman, M.L., *Software Engineering*, McGraw Hill, 1983.



## APPENDIX A. ACCEPTANCE TEST CHECKLIST

The following checklist is provided as an aid to the acceptance test planner. The list is intended only as guidance and is not intended to be complete or rigid; the planner should add, delete or change items according to specific project needs.

### Build glossary

- o acronym definitions
- o special definitions used in test plan or other documents

### Cite reference documents

- o standards
- o project documents
- o other applicable documents

### Develop acceptance criteria

- o measurable, quantitative, traceable to software requirements
- o aid in understanding requirements

### Change acceptance criteria

- o after review
- o after changes to requirements

### Identify test objectives

- o general objectives
- o specific to test groupings

### Determine test methodology

- o benchmark testing
- o baseline
- o parallel
- o pilot operation
- o in stages, after installation of each component
- o by features
- o by combination of features
- o by simulation
- o on site, total system
- o by end user usage
- o standard system tests
- o combinations of above
- o other

#### Determine types of tests and evaluations

- o functional
- o boundary
- o error handling
- o stress
- o data management
- o user interfaces
- o performance
- o configuration for different environments
- o installation procedures
- o recovery procedures
- o document examination
- o maintainability evaluation
- o procedures for system execution
- o other

#### Cite deliverables

- o deliverables to be accepted
- o methods of examination

#### Determine resource needs

- o hardware
- o software
- o personnel
- o scheduling
- o support tools
- o training
- o other

#### Determine developer/customer roles

#### Make staffing recommendations

- o for anomaly corrections
- o for test designs, cases, procedures, and execution
- o for test support (define requirements for data sets, build-extract files)
- o scheduling
- o assisting users in the acceptance testing
- o for test library management and production control-copy-rename data sets;
- o to initiate, modify, monitor test system
- o technical support- for system performance, backup-recovery during acceptance tests
- o test witnesses
- o other

#### Determine specific techniques

- o for testing
- o for data collection
- o for statistics on timing, scheduling
- o for test identification
- o for library entries, exits
- o to record results generated by system
- o to record results on performance
- o to analyze results

#### Determine policies

- o when to begin
- o when to stop for corrections
- o when to STOP - testing complete
- o what to do for "unexpected" anomaly
- o how to determine system adequately tested
- o retest policy, both before and after change
- o for test witnessing
- o for sign-off
- o other

#### Write the test plan

- o project summary
- o test objectives
- o acceptance criteria
- o test requirements
- o management information

#### Design test cases and procedures

- o state specific objective
- o specific test
- o test criteria
- o command languages
- o data (input, expected output)
- o resources (time, hardware, software, personnel)
- o constraints

#### Perform readiness tasks

- o review test plans
- o install support software
- o train staff
- o review for test readiness

#### Execute tests

- o arrange for computer facilities
- o get all data/test cases together
- o check that test witness present, if required
- o run test according to pre-set procedures
- o test complete-succeed
- o test complete-fail
- o test incomplete-fail

#### Collect test results

#### Analyze test results

#### Make acceptance decisions

- o for acceptance, follow sign-off procedures
  
- o for rejection,
  - \* check for error in test materials
  
- or
  
- \* return product to developer and  
locate proper iteration level  
for acceptance test.



## APPENDIX B. BIBLIOGRAPHY FOR ACCEPTANCE TESTING

[CELE81]

Celentano, A., C. Ghezzi, F. Liguori, "A Systematic Approach to System and Acceptance Testing," *Computer Program Testing*, B. Chandrasekaran and S. Radicchi, (ed.), North Holland Publishing, 1981.

[CHAN81]

Chandrasekaran, B. and S. Radicchi, (ed) *Computer Program Testing*, North Holland Publishing, 1981.

[CHO80]

Cho, Chin-Kuei, *An Introduction to Software Quality Control*, John Wiley & Sons, 1980.

[DEUT82]

Deutsch, Michael, *Software Validation and Verification*, Prentice-Hall, 1982.

[DUNN84]

Dunn, Robert H., *Software Defect Removal*, McGraw Hill, 1984.

[FAIR80]

Fairley, R.E., "Software Validation and Pre-implementation Issues," *Software Development Tools*, W.E. Riddle and R.E. Fairley, Springer-Verlag, 1980.

[FAIR80]

Fairley, R.E., "Software Testing Tools," *Computer Program Testing*, Chandrasekaran, B and S. Radicchi, (ed.) North Holland Publishing, 1981.

[FIPS101]

"Guideline for Lifecycle Validation, Verification, and Testing of Computer Software," Federal Information Processing Standards Publication 101, National Bureau of Standards, June, 1983.

[FIPS102]

"Guideline for Computer Security Certification and Accreditation," Federal Information Processing Standards Publication 102, National Bureau of Standards, September, 1983.

[FUJI78]

Fujii, Marilyn S., "A Comparison of Software Assurance Methods," *Proceedings of the Software Quality Assurance Workshop*, Nov.15-17,1978, San Diego, CA, ACM, New York, New York, 1978.

[GILS77]

Gilsinn, David E., J.Trotter Hardy, Charles L. Sheppard, "The NBS Validation Routines for the BASIC Language," *IEEE Proceedings of COMPSAC 77, 1st International Computer Software and Applications Conference*, The Institute for Electrical and Electronics Engineers, New York, NY, 1977.

[GLAS79]

Glass, R.L., *Software Reliability Guidebook*, Prentice Hall, 1979.

[GRUE73]

Gruenberger, F., "Program Testing: The Historical Perspective," *Program Test Methods*, Hetzel, W.C. (ed), Prentice Hall, Englewood Cliffs, NJ, 1973.

[HALL78]

Hallin, T.G., Hansen, R.C., "Toward a Better Method of Software Testing," *IEEE Proceedings of COMPSAC 78, 2nd International Computer Software and Applications Conference*, The Institute for Electrical and Electronics Engineers, New York, NY, 1978.

[HETZ73]

Hetzel, W.C., (ed) *Program Test Methods*, Prentice Hall, Engelwood Cliffs, NJ, 1973.

[HOLL73]

Holland, J. Grant, "Acceptance Testing for Application Programs", *Program Test Methods*, Hetzel, W.C., (ed), Prentice Hall, Englewood Cliffs, NJ, 1973.

[IEE729]

"IEEE Standard Glossary of Software Engineering Terminology," ANSI/IEEE Std. 729-1983, The Institute for Electrical and Electronics Engineers, Inc., 345 East 47th St., New York, NY 10017, 1983.

[LOES78]

Loesh, R.E., P. Molko, B. Larman, "Project Galileo Software Management Plan," Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, August, 1978, Report 652-510.

[METZ83]

Metzger, Philip W., *Managing a Programming Project*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[MULL77]

Mullin, Frank J., "Software Test Management for \$2M and Up," *IEEE Proceedings of COMPSAC 77, 1st International Computer Software and Applications Conference*, The Institute for Electrical and Electronics Engineers, New York, NY, 1977.

[MUSA79]

Musa, John D., "Software Reliability Measures Applied to System Engineering," *AFIPS 1979 National Computer Conference Proceedings*, AFIPS Press, Montvale, NJ, Vol. 48, 1979, pp.941-947.

[MYER79]

Myers, Glenford, *The Art of Software Testing*, Wiley-Interscience Publications, John Wiley & Sons, 1979.

[NG73]

Ng, Edward W., "Mathematical Software Testing Activities," *Program Test Methods*, W.C. Hetzel (ed), Prentice Hall, Englewood Cliffs, NJ, 1973.

[PAIN77]

Painter, James A., "Software Testing in Support of Worldwide Military Command and Control System ADP," *IEEE Proceedings of COMPSAC 77, 1st International Computer Software and Applications*, The Institute for Electrical and Electronics Engineers, New York, NY, 1977.

[REDW83]

Redwine, S.T., "An Engineering Approach to Software Test Data Design," *IEEE Transactions on Software Engineering*, March 1983, Vol. SE-9, No 2.

[SHOO83]

Shooman, M.L., *Software Engineering*, McGraw Hill, 1983.

[WASS80]

Wasserman, A.I., "Software Tools and the User Software Engineering Project," *Software Development Tools*, W.E.Riddle and R.E.Fairley, (ed), Springer-Verlag, 1980.

[YOUN73]

Youngberg, E.P., "A Software Testing Control System", *Program Test Methods*, W.C. Hetzel, Prentice Hall, Englewood Cliffs, NJ, 1973.

U.S. DEPT. OF COMM. <b>BIBLIOGRAPHIC DATA SHEET</b> <i>(See instructions)</i>	1. PUBLICATION OR REPORT NO. NBS/SP-500/136	2. Performing Organ. Report No.	3. Publication Date Feb. 1986
4. TITLE AND SUBTITLE Computer Science and Technology: An Overview of Computer Software Acceptance Testing			
5. AUTHOR(S) Dolores R. Wallace			
6. PERFORMING ORGANIZATION <i>(If joint or other than NBS, see instructions)</i> National Bureau of Standards Department of Commerce Gaithersburg, MD 20899		7. Contract/Grant No.	8. Type of Report & Period Covered Final
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS <i>(Street, City, State, ZIP)</i> Same as item 6			
10. SUPPLEMENTARY NOTES Library of Congress Catalog Card Number 86-500502 <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.			
11. ABSTRACT <i>(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)</i> This document provides guidance in planning and managing acceptance testing of computer software. It emphasizes the need for quantitative acceptance criteria and itemized test cases and procedures. It provides a checklist of activities to be performed for planning and managing acceptance testing.			
12. KEY WORDS <i>(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)</i> Software acceptance criteria, software acceptance testing, test cases, test management, test planning, test procedures.			
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input checked="" type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161			14. NO. OF PRINTED PAGES 22 15. Price



**ANNOUNCEMENT OF NEW PUBLICATIONS ON  
COMPUTER SCIENCE & TECHNOLOGY**

Superintendent of Documents,  
Government Printing Office,  
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Bureau of Standards Special Publication 500-.

Name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_

(Notification key N-503)

U.S. GOVERNMENT PRINTING OFFICE : 1981 O-340-997 : 1650)



# NBS *Technical Publications*

## *Periodical*

---

**Journal of Research**—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. Issued six times a year.

## *Nonperiodicals*

---

**Monographs**—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

**Handbooks**—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

**Special Publications**—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

**Applied Mathematics Series**—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

**National Standard Reference Data Series**—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396).

NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

**Building Science Series**—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

**Technical Notes**—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

**Voluntary Product Standards**—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

**Consumer Information Series**—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

*Order the above NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

*Order the following NBS publications—FIPS and NBSIR's—from the National Technical Information Service, Springfield, VA 22161.*

**Federal Information Processing Standards Publications (FIPS PUB)**—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

**NBS Interagency Reports (NBSIR)**—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

**U.S. Department of Commerce**  
National Bureau of Standards  
Gaithersburg, MD 20899

Official Business  
Penalty for Private Use \$300