

IEEE Standard for Software Test Documentation

IEEE Standards Board

Approved December 3, 1982

American National Standards Institute

Approved August 19, 1983

Sponsor

**Software Engineering Technical Committee
of the
IEEE Computer Society**

© Copyright 1983 by **The Institute of Electrical and Electronics Engineers, Inc 345 East 47th Street,
New York, NY 10017, USA**

*No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise,
without the prior written permission of the publisher.*

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE which have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least once every five years for revision or reaffirmation. When a document is more than five years old, and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
345 East 47th Street
New York, NY 10017
USA

Foreword

(This Foreword is not a part of ANSI/IEEE Std 829-1983, IEEE Standard for Software Test Documentation.)

Purpose

The purpose of this standard is to describe a set of basic software test documents. A standardized test document can facilitate communication by providing a common *frame of reference* (for example, a customer and a supplier have the same definition for a test plan). The content definition of a standardized test document can serve as a completeness checklist for the associated testing process. A standardized set can also provide a baseline for the evaluation of current test documentation practices. In many organizations, the use of these documents significantly increases the manageability of testing. Increased manageability results from the greatly increased visibility of each phase of the testing process.

This standard specifies the form and content of individual test documents. It does not specify the required set of test documents. It is assumed that the required set of test documents will be specified when the standard is applied. Appendix B contains an example of such a set specification.

Overview

The documents outlined in this standard cover test planning, test specification, and test reporting.

The test plan prescribes the scope, approach, resources, and schedule of the testing activities. It identifies the items to be tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with the plan.

Test specification is covered by three document types:

- 1) A test-design specification refines the test approach and identifies the features to be covered by the design and its associated tests. It also identifies the test cases and test procedures, if any, required to accomplish the testing and specifies the feature pass/fail criteria.
- 2) A test-case specification documents the actual values used for input along with the anticipated outputs. A test case also identifies constraints on the test procedures resulting from use of that specific test case. Test cases are separated from test designs to allow for use in more than one design and to allow for reuse in other situations.
- 3) A test procedure specification identifies all steps required to operate the system and exercise the specified test cases in order to implement the associated test design. Test procedures are separated from test-design specifications as they are intended to be followed step by step and should not have extraneous detail.

Test reporting is covered by four document types:

- 1) A test item transmittal report identifies the test items being transmitted for testing in the event that separate development and test groups are involved or in the event that a formal beginning of test execution is desired.
- 2) A test log is used by the test team to record what occurred during test execution.
- 3) A test incident report describes any event that occurs during the test execution which requires further investigation.
- 4) A test summary report summarizes the testing activities associated with one or more test-design specifications.

Figure 1 shows the relationships of these documents to one another as they are developed and to the testing process they document.

Terminology

The words *shall*, *must*, and the imperative form identify the mandatory material within this standard. The words *should* and *may* identify optional material.

Appendixes

The examples found in Appendix A are meant to clarify the intent of the document descriptions found in the standard. Some suggestions about implementing and using the standard are in Appendix B. Appendix C contains references to related test documentation standards. Appendix D contains references to testing-related documents of general interest which are not focused on test documentation.

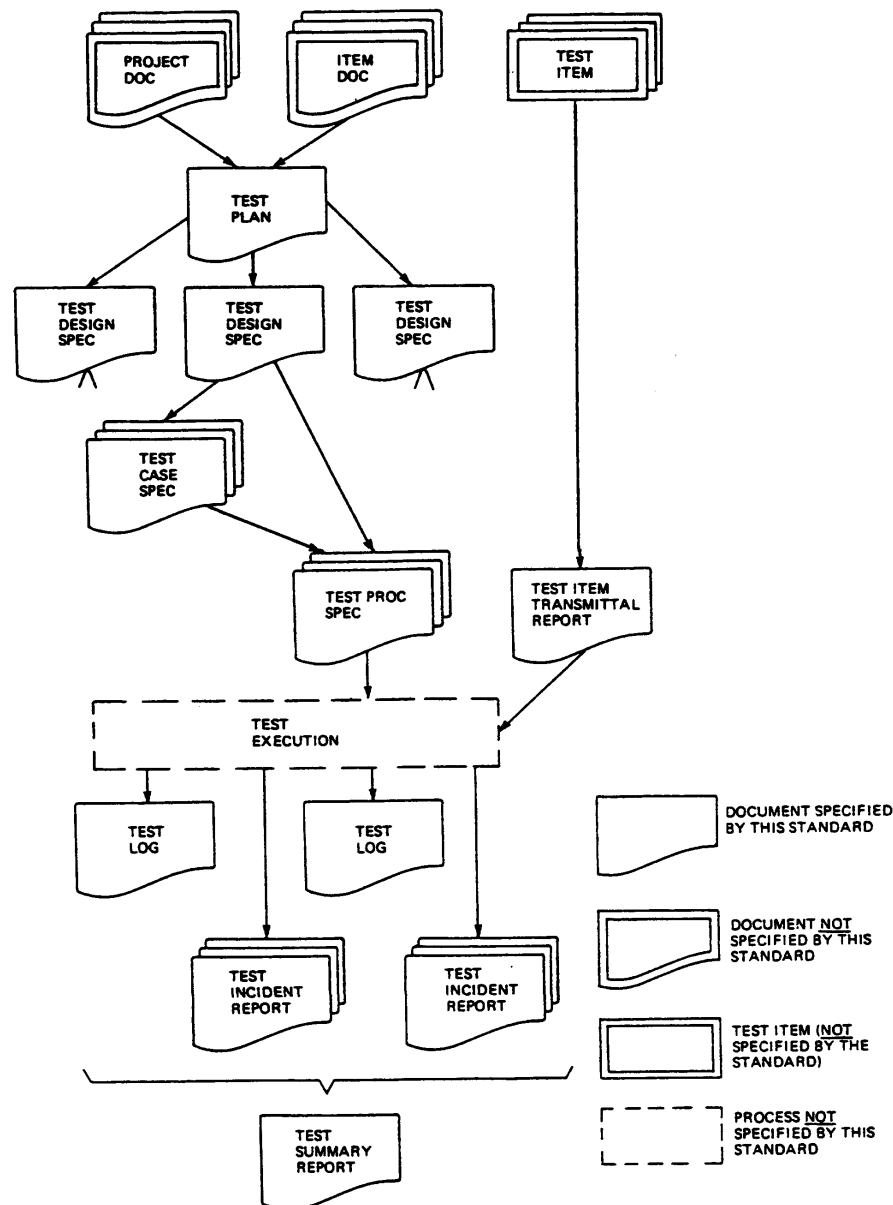


Figure 1 — Relationship of Test Documents to Testing Process

Audience

The standard should be of interest to software users and software procurement personnel; to development, test, and maintenance personnel; to operations and acquisition support managers; to software quality assurance personnel and auditors; and to participants in the legal system.

History

The development of this standard began from discussions within the Software Engineering Standards Subcommittee in 1977. Some initial work was done by a group including Joan Bateman, Leonard Birns, Herb Hecht, and Bob Poston and resulted in an early draft outline. The project authorization request for this effort was approved by the IEEE Standards Board in May, 1980. Following authorization, a series of ten meetings which were held across the country from September, 1980 to May, 1982 produced the first draft submitted for balloting.

Suggestions for improvement of the standard will be welcome. They should be sent to:
Secretary, IEEE Standards Board
Institute of Electrical and Electronics Engineers
345 East 47th Street
New York, NY 10017

Introduction

At the time this standard was approved on December 3, 1982, the IEEE Standards Board had the following membership:

Irvin N. Howell, *Chair*
Edward Chelotti, *Vice Chair*
Sava I. Sherr, *Secretary*

G. Y. R. Allen
J. J. Archambault
James H. Beall
John T. Boettger
J. V. Bonucchi
Edward J. Cohen
Len S. Corey

D. C. Fleckenstein
Jay Forster
Kurt Greene
Joseph L. Koepfinger
Irving Kolodny
John E. May
Donald T. Michael*

A. R. Parsons
J. P. Riganati
Frank L. Rose
Robert W. Seelbach
Jay A. Stewart
Clifford O. Swanson
Robert E. Weiler

This standard was completed by a working group with the following members:

David Gelperin, *Chair*

Rick Adrion
Jules Aronson
Joan Bateman
Charlie Bates
A. Birnbaum
Leonard Birns
John Bowen
Martha Branstad
Fletcher Buckley
Douglas Burt
Ian Burton
John Cain
John Center
George Chapman
Santosh Chokhani
Bruce Clay
D. L. Cooper
Doc Craddock
James Cridge
Patricia Daggett
Bill Dupras
Mary Eads
Sharon Freid
Tom Frost

Tom Gilb
Loretta Guarino
John Hawthorne
Herb Hecht
Mike Hennell
Dan Hocking
Mark Holthouse
Ray Houghton
Paul Howley
John Kalasky
Jason Kazarian
Ed Kit
Denise Krauss
Joseph Krupinski
Tom Kurihara
Costa J. Labovites
Paula Levinton
Leo Lippman
Alex Long
Mable Love
Ben Manny
Albrecht Newmann
Bill Newsom
C. Donald Ostler
Varsha Pai

Art Pollari
Robert Poston
Sam Redwine
Charles Schult
David Schultz
Dennis Sharp
Jerry Smith
Wayne Smith
Joan Spalding
Bob Stewart
S. L. Stewart
Sarah Swales
Barbara Taute
J. R. Taylor
Rudolf Van Megen
David Vatsaas
John Walter
Andrew Weigel
Jonathon Wexler
Paul White
Bruce Wilkins
Harry Wilkinson
F. T. Woodall
Fred Yonda

The standard was approved the Software Engineering Standards Subcommittee of the IEEE Computer Society. At the time it approved this standard, the subcommittee had the following membership:

Fletcher J. Buckley, *Chair*

Russell J. Abbott
A. Frank Ackerman
Leo Beltracchi
Richard L. Bernstein
Nestore G. Biasi
Dennis W. Bragg
Douglas W. Burt
Homer C. Carney

John Center
Won L. Chung
Anonio M. Cicu
Lori A. Clarke
Gilmore G. Cooke
A. J. Cote, Jr
Patricia W. Daggett
George Darling

B. Dasarathy
Noah S. Davids
James A. Dobbins
Mary L. Eads
John D. Earls
Leo G. Egan, Jr
John W. Fendrich
Dennis W. Fife

Joel J. Forman
M. Galinier
Forrest Karl Gardner
David Gelperin
Gary Gladden
Shirley A. Gloss-Soler
John J. Greene
Jack W. Grigsby
Robert M. Gross
David A. Gustafson
Russell T. Gustin
Thomas L. Hannan
Herbert Hecht
Leslie R. Heselton, III
Sam Horvitz
Paul Howley
Shuenn-Chang Hwang
Jim H. Ingram
John P. Kalasky
Laurel V. Kaleda
Thomas M. Kurihara
Domenic V. La-Rosa
Robert A.C. Lane
Gregory N. Larsen
Geoffrey R. Lewis
A. Lip Lim

G. S. Lindsay
Myron Lipow
William M. Lively
Marvin Lubofsky
Don Lundquist
Alan Cheuk-Wai Ma
Andy K. Mahindru
Philip C. Marriott
Mike McCollough
Belden Menkus
Edward F. Miller, Jr
G. Scott Morris
Gene T. Morun
Walter G. Murch
Jack Nebb
Geraldine Rajcula Neidhart
Michael A. Neighbors
Leon Osterweil
Donald J. Ostrom
William E. Perry
Donald J. Pfeiffer
Robert M. Poston

Patricia B. Powell
Farzad Raji
Jean Claude Rault

Samuel T. Redwine
T. L. Regulinski
William E. Riddle
Clarence W. Rutter, III
Norman F. Schneidewind
Antonia D. Schuman
Leonard W. Seagren
Wayne Smith
Harry M. Sneed
Lee Sprague
Edward A. Straker
Kou Chung Tai
Barbara J. Taute
George D. Tice, Jr
Terrence L. Tillmanns
William S. Turner, III
Edwin A. Ulbrich, Jr
Udo Voges
John P. Walter
Andrew H. Weigel
N. P. Wilburn
Martin Wong
David C. Wood
Ted Workman
Alfred W. Yonda
Peter F. Zoll

Special representatives to the Software Engineering Subcommittee were as follows:

William E. Perry, DPMA
Roy P. Pritchett, Jr, EDP
Auditors Association
John Milandin, ANSI Z1

Contents

CLAUSE	PAGE
1. Scope.....	1
2. Definitions.....	1
3. Test Plan.....	2
3.1 Purpose.....	2
3.2 Outline.....	3
4. Test-Design Specification.....	6
4.1 Purpose.....	6
4.2 Outline.....	6
5. Test-Case Specification	7
5.1 Purpose.....	7
5.2 Outline.....	7
6. Test-Procedure Specification	8
6.1 Purpose.....	8
6.2 Outline.....	9
7. Test-Item Transmittal Report.....	10
7.1 Purpose.....	10
7.2 Outline.....	10
8. Test Log	11
8.1 Purpose	11
8.2 Outline.....	11
9. Test-Incident Report	12
9.1 Purpose.....	12
9.2 Outline.....	13
10. Test-Summary Report.....	14
10.1 Purpose.....	14
10.2 Outline.....	14
APPENDIXES	
Appendix A (informative) Examples.....	16
Appendix B (informative) Implementation and Usage Guidelines	37
Appendix C (informative) Related Document Standards	39
Appendix D (informative) Annotated Bibliography	40

IEEE Standard for Software Test Documentation

1. Scope

This standard describes a set of basic test documents which are associated with the dynamic aspects of software testing (that is, the execution of procedures and code). The standard defines the purpose, outline, and content of each basic document. While the documents described in the standard focus on dynamic testing, several of them may be applicable to other testing activities (for example, the test plan and test incident report may be used for design and code reviews).

The standard may be applied to commercial, scientific, or military software which runs on any digital computer. Applicability is not restricted by the size, complexity, or criticality of the software. However, the standard does *not* specify any class of software to which it must be applied. The standard addresses the documentation of both initial development testing and the testing of subsequent software releases. For a particular software release, it may be applied to all phases of testing from module testing through user acceptance. However, since all of the basic test documents may not be useful in each test phase, the particular documents to be used in a phase are *not* specified. Each organization using the standard will need to specify the classes of software to which it applies and the specific documents required for a particular test phase.

The standard does *not* call for specific testing methodologies, approaches, techniques, facilities, or tools, and does *not* specify the documentation of their use. Additional test documentation may be required (for example, code inspection checklists and reports). The standard also does *not* imply or impose specific methodologies for documentation control, configuration management, or quality assurance. Additional documentation (for example, a quality assurance plan) may be needed depending on the particular methodologies used.

Within each standard document, the content of each section (that is, the text which covers the designated topics) may be tailored to the particular application and the particular testing phase. In addition to tailoring content, additional documents may be added to the basic set, additional sections may be added to any document and additional content to any section. It may be useful to organize some of the sections into subsections. Some or all of the contents of a section may be contained in another document which is then referenced. Each organization using the standard should specify additional content requirements and conventions in order to reflect their own particular methodologies, approaches, facilities, and tools for testing, documentation control, configuration management, and quality assurance.

The standard applies to documentation on electronic media as well as paper. Paper must be used for documents requiring approval signatures, unless the electronic documentation system has a secure approval annotation mechanism and that mechanism is used.

2. Definitions

This section contains key terms as they are used in the standard.

design level: The design decomposition of the software item (for example, system, subsystem, program, or module).

pass/fail criteria: Decision rules used to determine whether a software item or a software feature passes or fails a test.

software feature: A distinguishing characteristic of a software item (for example, performance, portability, or functionality).

software item: Source code, object code, job control code, control data, or a collection of these items.

test:

- 1) A set of one or more test cases, or
- 2) A set of one or more test procedures, or
- 3) A set of one or more test cases and procedures.

test case specification: A document specifying inputs, predicted results, and a set of execution conditions for a test item.

test design specification: A document specifying the details of the test approach for a software feature or combination of software features and identifying the associated tests.

test incident report: A document reporting on any event that occurs during the testing process which requires investigation.

test item: A software item which is an object of testing.

test item transmittal report: A document identifying test items. It contains current status and location information.

test log: A chronological record of relevant details about the execution of tests.

test plan: A document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.

test procedure specification: A document specifying a sequence of actions for the execution of a test.

test summary report: A document summarizing testing activities and results. It also contains an evaluation of the corresponding test items.

testing: The process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software item.

3. Test Plan

3.1 Purpose

To prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

3.2 Outline

A test plan shall have the following structure:

- 1) Test-plan identifier
- 2) Introduction
- 3) Test items
- 4) Features to be tested
- 5) Features not to be tested
- 6) Approach
- 7) Item pass/fail criteria
- 8) Suspension criteria and resumption requirements
- 9) Test deliverables
- 10) Testing tasks
- 11) Environmental needs
- 12) Responsibilities
- 13) Staffing and training needs
- 14) Schedule
- 15) Risks and contingencies
- 16) Approvals

The sections shall be ordered in the specified sequence. Additional sections may be included immediately prior to *Approvals*. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test plan or available to users of the plan.

Details on the content of each section are contained in the following sections.

3.2.1 Test-Plan Identifier

Specify the unique identifier assigned to this test plan.

3.2.2 Introduction

Summarize the software items and software features to be tested. The need for each item and its history may be included.

References to the following documents, when they exist, are required in the highest level test plan:

- Project authorization
- Project plan
- Quality assurance plan
- Configuration management plan
- Relevant policies
- Relevant standards

In multilevel test plans, each lower-level plan must reference the next higher-level plan.

3.2.3 Test Items

Identify the test items including their version/revision level. Also specify characteristics of their transmittal media which impact hardware requirements or indicate the need for logical or physical transformations before testing can begin (for example, programs must be transferred from tape to disk).

Supply references to the following item documentation, if it exists:

- Requirements specification
- Design specification
- Users guide
- Operations guide
- Installation guide
- Reference any incident reports relating to the test items.
- Items which are to be specifically excluded from testing may be identified.

3.2.4 Features to be Tested

Identify all software features and combinations of software features to be tested. Identify the test-design specification associated with each feature and each combination of features.

3.2.5 Features Not to be Tested

Identify all features and significant combinations of features which will not be tested and the reasons.

3.2.6 Approach

Describe the overall approach to testing. For each major group of features or feature combinations, specify the approach which will ensure that these feature groups are adequately tested. Specify the major activities, techniques, and tools which are used to test the designated groups of features.

The approach should be described in sufficient detail to permit identification of the major testing tasks and estimation of the time required to do each one.

Specify the minimum degree of comprehensiveness desired. Identify the techniques which will be used to judge the comprehensiveness of the testing effort (for example, determining which statements have been executed at least once). Specify any additional completion criteria (for example, error frequency). The techniques to be used to trace requirements should be specified.

Identify significant constraints on testing such as test-item availability, testing-resource availability, and deadlines.

3.2.7 Item Pass/Fail Criteria

Specify the criteria to be used to determine whether each test item has passed or failed testing.

3.2.8 Suspension Criteria and Resumption Requirements

Specify the criteria used to suspend all or a portion of the testing activity on the test items associated with this plan. Specify the testing activities which must be repeated, when testing is resumed.

3.2.9 Test Deliverables

Identify the deliverable documents. The following documents should be included:

- Test plan
- Test design specifications
- Test case specifications
- Test procedure specifications
- Test item transmittal reports

- Test logs
- Test incident reports
- Test summary reports
- Test input data and test output data should be identified as deliverables.
- Test tools (for example, module drivers and stubs) may also be included.

3.2.10 Testing Tasks

Identify the set of tasks necessary to prepare for and perform testing. Identify all intertask dependencies and any special skills required.

3.2.11 Environmental Needs

Specify both the necessary and desired properties of the test environment. This specification should contain: the physical characteristics of the facilities including the hardware, the communications and system software, the mode of usage (for example, stand-alone), and any other software or supplies needed to support the test. Also specify the level of security which must be provided for the test facilities, system software, and proprietary components such as software, data, and hardware.

Identify special test tools needed. Identify any other testing needs (for example, publications or office space). Identify the source for all needs which are not currently available to the test group.

3.2.12 Responsibilities

Identify the groups responsible for managing, designing, preparing, executing, witnessing, checking, and resolving. In addition, identify the groups responsible for providing the test items identified in 3.2.3 and the environmental needs identified in 3.2.11 .

These groups may include the developers, testers, operations staff, user representatives, technical support staff, data administration staff, and quality support staff.

3.2.13 Staffing and Training Needs

Specify test staffing needs by skill level. Identify training options for providing necessary skills.

3.2.14 Schedule

Include test milestones identified in the Software Project Schedule as well as all item transmittal events.

Define any additional test milestones needed. Estimate the time required to do each testing task. Specify the schedule for each testing task and test milestone. For each testing resource (that is, facilities, tools, and staff), specify its periods of use.

3.2.15 Risks and Contingencies

Identify the high-risk assumptions of the test plan. Specify contingency plans for each (for example, delayed delivery of test items might require increased night shift scheduling to meet the delivery date).

3.2.16 Approvals

Specify the names and titles of all persons who must approve this plan. Provide space for the signatures and dates.

4. Test-Design Specification

4.1 Purpose

To specify refinements of the test approach and to identify the features to be tested by this design and its associated tests.

4.2 Outline

A test-design specification shall have the following structure:

- 1) Test-design-specification identifier
- 2) Features to be tested
- 3) Approach refinements
- 4) Test identification
- 5) Feature pass/fail criteria

The sections shall be ordered in the specified sequence. Additional sections may be included at the end. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test-design specification or available to users of the design specification.

Details on the content of each section are contained in the following sections.

4.2.1 Test-Design-Specification Identifier

Specify the unique identifier assigned to this test-design specification. Supply a reference to the associated test plan, if it exists.

4.2.2 Features to be Tested

Identify the test items and describe the features and combinations of features which are the object of this design specification. Other features may be exercised, but need not be identified.

For each feature or feature combination, a reference to its associated requirements in the item requirement specification or design description should be included.

4.2.3 Approach Refinements

Specify refinements to the approach described in the test plan. Include specific test techniques to be used. The method of analyzing test results should be identified (for example, comparator programs or visual inspection).

Specify the results of any analysis which provides a rationale for test-case selection. For example, one might specify conditions which permit a determination of error tolerance (for example, those conditions which distinguish valid inputs from invalid inputs).

Summarize the common attributes of any test cases. This may include input constraints that must be true for every input in the set of associated test cases, any shared environmental needs, and any shared special procedural requirements, and any shared case dependencies.

4.2.4 Test Identification

List the identifier and a brief description of each test case associated with this design. A particular test case may be identified in more than one test design specification. List the identifier and a brief description of each procedure associated with this test-design specification.

4.2.5 Feature Pass/Fail Criteria

Specify the criteria to be used to determine whether the feature or feature combination has passed or failed.

5. Test-Case Specification

5.1 Purpose

To define a test case identified by a test-design specification.

5.2 Outline

A test-case specification shall have the following structure:

- 1) Test-case-specification identifier
- 2) Test items
- 3) Input specifications
- 4) Output specifications
- 5) Environmental needs
- 6) Special procedural requirements
- 7) Intercase dependencies

The sections shall be ordered in the specified sequence. Additional sections may be included at the end. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test-case specification or available to users of the case specification.

Since a test case may be referenced by several test-design specifications used by different groups over a long time period, enough specific information must be included in the test-case specification to permit reuse.

Details on the content of each section are contained in the following sections.

5.2.1 Test-Case-Specification Identifier

Specify the unique identifier assigned to this test-case specification.

5.2.2 Test Items

Identify and briefly describe the items and features to be exercised by this test case.

For each item, consider supplying references to the following item documentation.

- 1) Requirements specification
- 2) Design specification
- 3) Users guide