IEEE Standard for Software Reviews and Audits

Circuits and Devices

Communications Technology

Computer
Sponsored by the
Software Engineering
Technical Committee of the
IEEE Computer Society

Electromagnetics and Radiation

Energy and Power

Industrial Applications

Signals and Applications

Standards Coordinating Committees

IEEE Std 1028-1988

Published by the Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017, USA.

June 30, 1989

SH12625

IEEE Standard for Software Reviews and Audits

Corrected Edition
June 30, 1989

Corrections have been made on pages 5, 8, 10, 11, 12, 25, 26, 27, 29, and 35. A black bar has been added opposite each correction to aid in identifying the changes made to this printing.

Sponsor

Software Engineering Technical Committee of the IEEE Computer Society

Approved June 9, 1988

IEEE Standards Board

The Institute of Electrical and Electronics Engineers, Inc
345 East 47th Street, New York, NY 10017 USA

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 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

IEEE Standards documents are adopted by the Institute of Electrical and Electronics Engineers without regard to whether their adoption may involve patents on articles, materials, or processes. Such adoption does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standards documents.

Foreword

(This Foreword is not a part of IEEE Std 1028-1988, IEEE Standard for Software Reviews and Audits.)

This Foreword provides the user with the rationale and background of the review and audit procedures outlined in this standard, and their relation to other IEEE standards.

Purpose. This standard defines the review and audit processes applicable to critical and noncritical software, and the specific procedures required for the execution of the reviews and audits described.Also, this standard supports ANSI/IEEE Std 730-1984 by defining review and audit procedures.

This standard does not establish measures or mechanisms for judging individual contributions to product development. Conformance to this standard may be claimed where reviews and audits are performed using procedures defined by this standard.

Additional documents that should be referenced for specific application of this standard are as follows:

(1) ANSI/IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology.
(2) ANSI/IEEE Std 730-1984, IEEE Standard for Software Quality Assurance Plans.
(3) ANSI/IEEE Std 828-1983, IEEE Standard for Software Configuration Management Plans.
(4) ANSI/IEEE Std 983-1986, IEEE Guide for Software Quality Assurance.
(5) ANSI/IEEE Std 1012-1986 IEEE Standard for Software Verification and Validation Plans.

These standards address specific applications to this standard. Although only IEEE standards are referenced in this standard, this standard can be applied in concert with local standards requiring reviews and audits.

Moreover, to support the use of this standard by ensuring uniform classification and codes, use of IEEE Std 1044-1988, IEEE Standard Classification for Software Errors, Faults, and Failures, is suggested.

General Application Intent. This standard applies to all phases of typical software life cycles and provides a standard against which review and audit plans can be prepared and assessed. Maximum benefit can be derived from this standard by planning for its application early in the project life cycle.

This standard can be used with different life cycles provided its intent is followed. ANSI/IEEE Std 730-1984 identifies specific (and minimum) review and audit applications for critical software, while this standard defines the specific processes available to meet these application needs. Planning for reviews and audits, as part of an integrated verification and validation approach, can be aided by following ANSI/IEEE Std 1012-1986.

This standard for software reviews and audits was written in consideration of both the software and its system operating environment. It can be used where software is the total system entity or where it is part of a larger system. Care should be taken to integrate software review and audit activities into any total system development planning; they should exist in concert with hardware system reviews and audits to the benefits of the entire system. To broaden the application possibilities of this standard, simply replace *software element* with *system element* when reading.

This standard was written with the assumption that organizations are void of corruption. This may become an issue where, for example, during the course of an audit, theft or embezzlement is discovered. In such a case, although the audit may proceed, the applicability of this standard is removed.

Appendixes. Although not considered part of this standard, appendixes are provided to assist the reader in understanding:

(1) Specific process applications required by ANSI/IEEE Std 730-1984.
(2) How processes defined in this standard might be applied to meet those application needs.

Audience. This standard was written for those who are responsible for defining, planning, implementing, or supporting software reviews and audits.

Conformance. Conformance to this standard can be claimed when review and audit processes, wherever performed, are performed as defined in this standard.

The following persons were on the balloting committee that approved this document for submission to the IEEE Standards Board.

A. Frank Ackerman
Richard L. Aurbach
Motoei Azuma
H. Jack Barnard
James B. Behm
Leo Beltracchi
Michael A. Blackledge
Gilles R. Bracon
Kathleen L. Briggs
A. Winsor Brown
William Bryan
F. Buckley
Lorie J. Call
Ronald L. Cariola
Harry Carl
John Center
T. S. Chow
J. K. Chung
Won L. Chung
Antonio Cicu
Corey Clinger
Peter Coach
Francois Coallier
Carolyn Collins
Chris Cooke
Rudolph P. Corbett
Stewart Crawford
Taz Daughtrey
Peter A. Denny
Fred M. Discenzo
David A. Dobratz
Michael P. Dotson
David C. Doty
Einar Dragstedt
William P. Dupras
Michael Dutton
Mary L. Eads
Robert G. Ebenau
L. G. Egan
W. D. Ehrenberger
Walter Ellis
Caroline L. Evans
David W. Favor
John Fendrich
Glenn S. Fields
Violet Foldes
Thom Foote-Lennox
Joel Forman
Julian Forster
Richrad C. Fries
Ismael Fuentes
F. Karl Gardner
Leonard B. Gardner
David Gelperin
Anne K. Geraci
Jean A. Gilmore
Shirley A. Gloss-Soler
J. G. Glynn
Andrej Grebenc
Benjamin W. Green
Victor M. Guarnera
Lawrence M. Gunther
David A. Gustafson
Clark Hay
David A. Hill
William A. Hoberg
Charles P. Hollocker
John W. Horch
Laurel V. Kaleda
Harry Kalmbach
Myron S. Karasik
Timothy C. Kasse
George A. Klammer
George Konstantinow
Joseph A. Krvpinsh
Joan Kundig
Tom Kurihara
Lak-Ming Lam
John B. Lane
Robert A. Lane
Gregorie N. Larsen
F. C. Lim
Bertil Lindbert
Ben Livson
Austin J. Maher
Stanley A. Marash
Paulo Cesar Marcondes
Philip C. Marriott
Nicholas L. Marselos
Roger J. Martin
Paul A. Mauro
J. A. McCall
Russell McDowell
Stanley E. McQueen
Manijeh Moghis
Charles S. Mooney
Gary D. Moorhead
D. D. Morton
G. T. Morun
Hirouobu Nagano
Geraldine Neidhart
Dennis E. Nickle
E. J. Oertel
Wilma Osborne
M. T. Perkins
William E. Perry
John Petraglia
Ronald T. Pfaff
Donald J. Pfeiffer
I. C. Pyle
Thomas Steven Radi
Salim Ramji
Jean-Claude Rault
Meir Razy
Donald J. Reifer
R. San Roman
John C. Rowe
Margaret Rumley
Julio Gonzalez Sanz
Stephen R. Schach
Hans Schaefer
Norman Schneidewind
Wolf A. Schnoege
Robert G. Schueppert
David J. Schultz
Gregory D. Schumacher
Leonard W. Seagren
Tony Sgarlatti
Isaac-Shahab Shadman
Gerald P. Share
Robert W. Shillato
David M. Siefert
William G. Singer
Jacob Slonim
G. Wayne Staley
Nick Stewart
W. G. Sutcliffe
Richard H. Thayer
Bob Thibodeau
Paul U. Thompson
George D. Tice
Terrence L. Tillmanns
Valentin W. Tirman
L. F. Tracey
Glenn R. Trebble
Henry J. Trochesset
C. L. Troyanowski
William Stephen Turner, III
Robert B. Urling
David Usechak
Tom Vollman
Dolores R. Wallace
John P. Walter
John W. Walz
Charles J. Wertz
G. Allan Whittaker
Patrick J. Wilson
David L. Winningham
W. M. Wong
Dennis L. Wood
Marian A. Wysocki
Natalie C. Yopconka
Donald Zeleny
Kenneth M. Zemrowski
Peter F. Zoll

At the time this standard was approved, the Software Reviews and Audits Working Group of the Software Engineering Technical Committee of the IEEE Computer Society had the following members:

Charles P. Hollocker, Chairman **Tim Kasse, Cochairman**

William A. Hoberg, Secretary

A. Frank Ackerman	Violet Foldes	Stan McQueen
Jack Barnard	Gordon Grunewald	Belden Menkus
William Bryan	Clark Hay	William Perry
Ronald L. Cariola	Peter Hung	William N. Sabor
Francois Coallier	David R. Janson	Isaac Shadman
Christopher M. Cooke	Harry Kalmbach	David J. Schultz
Rudy Corbett	Myron Karasik	Richard H. Thayer
Stewart Crawford	Adi Kasad	William S. Turner, III
Vince Dowd	Robert A. Kessler	Marian Wysocki
Robert G. Ebenau	Tom Kurihara	Natalie C. Yopconka
Michael Fagan	Philip C. Marriot	

When the IEEE Standards Board approved this standard on June 9, 1988, it had the following membership:

Donald C. Fleckenstein, *Chairman* **Marco W. Migliaro,** *Vice Chairman*

Andrew G. Salem, *Secretary*

Arthur A. Blaisdell	John W. Horch	L. Bruce McClung
Fletcher J. Buckley	Jack M. Kinn	Donald T. Michael*
James M. Daly	Frank D. Kirschner	Richard E. Mosher
Stephen R. Dillon	Frank C. Kitzantides	L. John Rankine
Eugene P. Fogarty	Joseph L. Koepfinger*	Gary S. Robinson
Jay Forster*	Irving Kolodny	Frank L. Rose
Thomas L. Hannan	Edward Lohse	Helen M. Wood
Kenneth D. Hendrix	John E. May, Jr	Karl H. Zaininger
Theodore W. Hissey, Jr	Lawrence V. McCall	Donald W. Zipse

*Members Emeriti

Contents

SECTION PAGE

FIGURE

TABLE

APPENDIXES

APPENDIX TABLES

IEEE Standard for Software Reviews and Audits

1. Scope and References

1.1 Scope. The purpose of this standard is to provide definitions and uniform requirements for review and audit processes. It does not establish the need to conduct specific reviews or audits; that need is defined by local policy. Where specific reviews and audits are required, standard procedures for their execution must be defined.

This standard provides such definition for review and audit processes that are applicable to products and processes throughout the software life cycle. Each organization shall specify where and when this standard applies and any intended deviations from this standard.

1.2 References. This standard shall be used in conjunction with the following publications:

[1] ANSI/IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology.[1]

[2] ANSI/IEEE Std 829-1983, IEEE Standard for Software Test Documentation.

2. Definitions

audit. An independent evaluation of software products or processes to ascertain compliance to standards, guidelines, specifications, and procedures based on objective criteria that include documents that specify

(1) The form or content of the products to be produced
(2) The process by which the products shall be produced
(3) How compliance to standards or guidelines shall be measured

review. An evaluation of software element(s) or project status to ascertain discrepancies from planned results and to recommend improvement. This evaluation follows a formal process (for example, management review process, technical review process, software inspection process, or walkthrough process).

software element. A deliverable or in-process document produced or acquired during software development or maintenance. Specific examples include but are not limited to

(1) Project planning documents (for example, software development plans, and software verification and validation plans)
(2) Software requirements and design specifications
(3) Test effort documentation (ie, as described in ANSI/IEEE Std 829-1983 [2])[2]
(4) Customer-deliverable documentation
(5) Program source code
(6) Representation of software solutions implemented in firmware
(7) Reports (for example, review, audit, projectect status) and data (for example, defect detection, test)

For other definitions, including verification and validation, see ANSI/IEEE Std 729-1983 [1].

[1]ANSI/IEEE publications are available from IEEE Service Center, 445 Hoes Lane, PO Box 1331, Piscataway, NJ 08855-1331, or from the Sales Department, American National Standards Institute, 1430 Broadway, New York, NY 10018.

[2]The numbers in brackets correspond to those of the references in 1.2.

3. Introduction

Reviews and audits can be used in support of objectives associated with software quality assurance, project management, and configuration management, or similar control functions, as shown in Table 1. Those quality objectives, valid throughout software development and maintenance, include the need for evaluation, verification, validation, and compliance confirmation. For the sake of completeness, Test is identified in Table 1 as the principal process of validation. Testing, however, is outside the scope of this standard. Processes defined in this standard include

(1) Management review
(2) Technical review
(3) Software inspection
(4) Walkthrough
(5) Audit

To support compliance confirmation, auditing activities may include performing reviews and tests to determine product consistency with respect to established baselines (ie, software requirements specifications or design descriptions) or to witness the correctness of implementation.

Moreover, since quality assurance deals with products and processes, Fig 1 provides a view of how various quality assurance processes can be mapped to examine product and project issues.

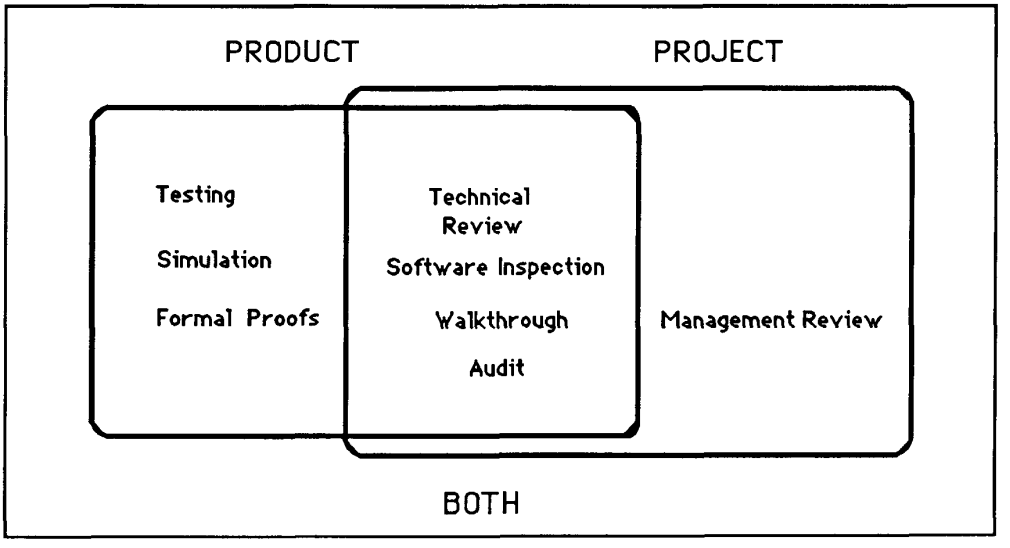
Table 1
Some Principal Processes for Achieving Quality Objectives

Objectives	Principal Processes Include
Evaluation	Management review, Technical review
Verification	Inspection, Walkthrough
Validation	Test
Compliance, Confirmation	Audit

The examination of project issues (both technical and managerial) occurs at various phases during the project life cycle. The results of such examinations are meant to permit improvement of the methods of ensuring software quality and the ability to meet time and cost constraints. The evaluation of software elements occurs during the production of the element(s) and on its completion. This ensures that the element(s) completely and correctly embodies its baseline specifications.

Any standard process has prerequisite conditions; those conditions necessary, though not in themselves sufficient for the process to reach completion. For reviews and audits, these conditions are identified in 3.1 and 3.2 respectively.

Fig 1
Relation of Some Quality Assurance Processes to Products and Projects



Section 3.3 presents the template used to define review and audit processes in the following sections.

3.1 Review Process Prerequisites. The objective of reviewing software elements is to evaluate software or project status to identify discrepancies from planned results and to recommend improvement where appropriate.

The following managerial offices are to achieve that objective:

3.1.1 Management

(1) Responsible for reviews, and the anticipated product rework

(2) Responsible for performance of reviews and reporting of review results against project milestone events

(3) Provides the necessary resources of time, personnel, budget, and facilities required to plan, define, execute, and manage the reviews

3.1.2 Development Staff

(1) The staff has a level of development expertise and product knowledge sufficient to comprehend the software under review.

(2) Training and orientation in the use of the review processes are provided to the staff.

3.1.3 Development Process Planning

(1) Standard entry and exit criteria have been defined for development phases.

(2) The product criteria (such as readability and modularity) are sufficiently defined by development standards.

(3) The product is partitioned into manageable, reviewable units.

(4) The software elements to be reviewed and the review process to be employed are identified in the project planning documents (for example, Software Quality Assurance Plan, Software Verification Plan, and Software Project Management Plan.

3.1.4 Review Process Planning

(1) Each type of review to be employed is fully specified.

(2) Administrative procedures are defined to enable the review to be initiated, executed, recorded, and acted on.

(3) The responsibility is assigned for the planning, specification, administration, and maintenance of the review process.

3.2 Audit Process Prerequisites. The objective of software auditing is to provide an objective evaluation of products and processes to confirm compliance to standards, guidelines, specifications, and procedures. The following requirements are prerequisite to achieve that objective:

(1) Objective audit criteria exist (for example, contracts, requirements, plans, specifications, standards) against which software elements and processes can be evaluated.

(2) Audit personnel are selected to promote team objectivity. They are usually independent of any direct responsibility for the products and processes examined and may be from an external organization.

(3) Audit personnel are given sufficient authority by appropriate management to perform the audit.

3.3 Procedural Description Template. The following descriptions establish the minimum template used to plan, prepare, and execute any review or audit. The review and audit process descriptions in this standard follow this common template:

().1 *Objective:* Process goals.

().2 *Abstract:* Process overview.

().3 *Special Responsibilities:* Roles unique to this specific process.

().4 *Input:* Products to which the process is applied, and supporting information.

().5 *Entry Criteria:* Conditions that must be satisfied before the process can begin.

().6 *Procedures:* Standard steps followed in performing the process.

().7 *Exit Criteria:* Conditions that must be satisfied before this process is considered complete.

().8 *Output:* The minimum set of deliverables resulting from process completion.

().9 *Auditability:* Description of evidence needed to determine at a later date that the process was followed.

Additional template sections have been added to these basic nine sections when needed.

Procedure descriptions in section "().6" will be presented as to include, but not be limited to, the following steps:

().6.1 Planning
().6.2 Overview
().6.3 Preparation
().6.4 Examination

Additional procedure subsections have been added where needed.

4. The Management Review Process

4.1 Objective. The objective of a management review is to provide recommendations for the following:

(1) Making activities progress according to plan, based on an evaluation of product development status;
(2) Changing project direction or to identify the need for alternative planning;
(3) Maintaining global control of the project through adequate allocation of resources.

The management review process can be applied to new development or to maintenance activities.

4.2 Abstract. A management review is a formal evaluation of a project level plan or project status relative to that plan by a designated review team.

During the review meeting the entire review team examines plans or progress against applicable plans, standards, and guidelines, or both. Each problem area identified by the review team is recorded. When critical data and information cannot be supplied, then an additional meeting shall be scheduled to complete the management review process.

4.3 Special Responsibilities. The review leader is responsible for the administrative tasks pertaining to the review, for assuring that the review is conducted in an orderly manner, and for issuing the management review report (see 4.8).

The individual(s) responsible for formally reporting the status of the project under review will ensure that the project status and all supporting documentation are available for distribution.

Each member of the review team is responsible for the following:

(1) Being adequately prepared for the meeting;
(2) Ensuring that the review meets its objectives.

4.4 Input. The minimum input to the management review process is as follows:

(1) A statement of objectives for the management review
(2) A list of issues to be addressed
(3) Current project schedule and cost data
(4) Reports (for example, managerial review reports, technical review reports, audit reports) from other reviews or audits, or both, already completed
(5) Reports of resources assigned to the project
(6) Data on the software elements completed or in progress

Additional applicable reference material can be supplied by project management or requested by the review leader.

4.5 Entry Criteria

4.5.1 Authorization. The need for conducting certain management reviews is initially established in the appropriate project planning documents (for example, Software Quality Assurance Plan, Software Development Plan, or Software Verification and Validation Plan). Under these plans, completion of a specific software element (for example, a Planning Document, a Requirements Specification, a Design Document, and Code Specification), or completion of a phase can trigger a management review.

In addition to those management reviews required by a specific plan, other management reviews may be formally announced and held at the request of software quality management, functional management, project management or the customer, according to local procedures.

4.5.2 Initiating Event. A management review is conducted when a selected review leader does the following:

(1) Establishes or confirms a statement of objectives for the meeting;
(2) Judges that the software element(s) and any other documentation or reports pertinent to the review are sufficiently complete and their status available to support the review objectives.

4.6 Procedures

4.6.1 Planning. During the planning step, the review leader, in conjunction with project management, does the following:

(1) Identifies the review team
(2) Schedules the meeting and identifies a meeting place
(3) Distributes input materials to the participants, allowing adequate time for preparation

4.6.2 Overview. A qualified person from the project under examination shall conduct an overview session for the review team when requested by the review leader. This overview can occur as a part of the examination meeting (see 4.6.4) or as a separate meeting.

4.6.3 Preparation. Each person on the review team individually studies the material and prepares for the review meeting. Any required presentations to the review team are prepared.

4.6.4 Examination. During the management review the review team holds one or more meetings to do the following:

(1) Examine project status and determine if it complies with the expected status according to a predefined plan;
(2) Examine project status and determine if it is overly constrained by external and internal factors not originally considered in the project plan;
(3) Record all deviations from the expected status accenting risks;
(4) Generate a list of issues and recommendations to be addressed by higher level management;
(5) Generate a list of issues and recommendations to be addressed by other responsible individuals, or organizations who affect the project;
(6) Recommend what course of action should be taken from this point on;
(7) Recommend authorization for additional reviews or audits;
(8) Identify other issues that must be addressed.

4.6.5 Rework. Any plan adjustments or product rework resulting from the management review is not considered as part of the management review process, except where needed as additional input to complete the examination process.

4.7 Exit Criteria. The management review is considered complete when

(1) All issues identified in the review Statement of Objectives have been addressed.
(2) The management review report has been issued (see 4.8).

4.8 Output. The output from the management review process is a Management Review Report that identifies the following:

(1) The project being reviewed;
(2) The review team;
(3) Inputs to the review;
(4) Review objectives;
(5) Action item ownership and status;
(6) A list of issues and recommendations identified by the review team that must be addressed for the project to meet its milestone;
(7) Recommendations regarding any further reviews and audits, and a list of additional information and data that must be obtained before they can be executed.

Although this standard sets minimum requirements for report content, it is left to local standards to prescribe any report format requirements.

4.9 Auditability. The management review report is an auditable item. It is recommended that this report be traceable to and from the appropriate project planning documents (for example, Software Development Plan, Software Quality Assurance Plan, or other applicable plans, or a combination of these).

5. The Technical Review Process

5.1 Objective. The objective of a technical review is to evaluate a specific software element(s) and provide management with evidence that

(1) The software element(s) conform to its specifications;
(2) The development (or maintenance) of the software element(s) is being done according to plans, standards, and guidelines applicable for the project;
(3) Changes to the software element(s) are properly implemented, and affect only those system areas identified by the change specification.

Moreover, to further tailor this review process for an individual software element, specific objectives are to be identified in a Statement of Objectives made available before the review meeting. The technical review concept can be applied to new development or to maintenance activities.

5.2 Abstract. A technical review is a formal team evaluation of a software element(s). It identifies any discrepancies from specifications and standards or provides recommendations after the examination of alternatives, or both. This examination may require more than one meeting.

5.3 Special Responsibilities. Roles for the technical review include:

5.3.1 Leader. The review leader is responsible for conducting a specific review. This includes administrative tasks pertaining to the review and ensuring that the review is conducted in an orderly manner. The review leader is also responsible for issuing the review report.

5.3.2 Recorder. The recorder is responsible for documenting findings (for example, defects, inconsistencies, omissions, and ambiguities), decisions, and recommendations made by the review team.

5.3.3 Team Member. Each team's members is responsible for their own preparation, and for ensuring that the review meets its objectives. Together, they are responsible for formulating recommendations in such a way that management can act on them promptly.

Management is responsible for acting on the review team recommendations in a timely manner.

5.4 Input. The minimum input to the technical review process is as follows:

(1) A statement of objectives for the technical review
(2) The software element(s) being examined
(3) Specifications for the software element(s) being examined
(4) Any plans, standards, or guidelines against which the software element(s) are to be examined

Additional applicable reference material can be made available by the individual(s) responsible for the software element, when requested by the review leader.

5.5 Entry Criteria

5.5.1 Authorization. The need for conducting technical reviews of specific software elements is defined by project planning documents.

Unscheduled technical reviews may be conducted at the request of functional management, project management, software quality management, or software engineers, according to local procedures.

5.5.2 Initiating Event. A technical review may not be conducted until

(1) A statement of objectives for the review is established;
(2) The responsible individual(s) for the software element(s) indicate readiness for review;
(3) The technical review leader is satisfied that the software element(s) are sufficiently complete for a technical review to be worthwhile

5.6 Procedures

5.6.1 Planning. A review leader is assigned responsibility for the following planning activities:

(1) Identify, with appropriate management support, the review team
(2) Schedule and announce the meeting place
(3) Distribute input materials to participants, allowing adequate time for their preparation

5.6.2 Overview. A technically qualified person shall conduct an overview session for the review team when requested by the review leader. This overview can occur as a part of the review meeting (see 5.6.4) or as a separate meeting.

5.6.3 Preparation. During the preparation step, each review team member individually examines the software element(s) and related materials in preparation for the review meeting.

5.6.4 Examination. At the review meeting the entire team reviews the software element, evaluating its condition relative to applicable guidelines, specifications and standards, or evaluating alternative problem solutions. Specifically, the review team performs the following tasks:

(1) Examines the software element(s) under review and verifies that it complies with the specifications and standards to which it must adhere. All deviations from the specifications and standards are recorded.
(2) Documents technical issues, related recommendations, and the individual responsible for getting the issues resolved.
(3) Identifies other issues that must be addressed.

After the software element(s) have been reviewed, a report is generated to document the meeting, list deficiencies found in the software element, and describe any recommendations for management.

When deficiencies are sufficiently critical or numerous, the review leader must recommend that an additional review process (that is, management review, technical review, or walkthrough) be applied to the reworked software element(s) after the deficiencies have been resolved. This, at a minimum, covers areas changed to resolve deficiencies.

5.7 Exit Criteria. A Technical Review is complete when
(1) All issues identified in the review Statement of Objectives have been addressed
(2) The technical review report (see 5.8) has been issued

5.8 Output. The output from the technical review process is a technical review report that identifies the following:
(1) The review team members
(2) The software element(s) reviewed
(3) Specific inputs to the review
(4) A list of unresolved software element(s) deficiencies
(5) A list of management issues
(6) Action item ownership and status
(7) Any recommendations made by the review team on how to dispose of unresolved issues and deficiencies

Although this standard sets minimum requirements for report content, it is left to local standards to prescribe any report format requirements.

5.9 Auditability. The technical review report is an auditable item. It is recommended that this report be traceable to/from the appropriate project planning documents (for example, Software Development Plan, Software Quality Assurance Plan).

6. The Software Inspection Process

6.1 Objective. The objective of a software inspection is to detect and identify software element defects. This is a rigorous, formal peer examination that does the following:
(1) Verifies that the software element(s) satisfy its specifications
(2) Verifies that the software element(s) conform to applicable standards
(3) Identifies deviation from standards and specifications
4) Collects software engineering data (for example, defect and effort data)
(5) Does not examine alternatives or stylistic issues

6.2 Abstract. Software inspections are conducted by peers, and typically comprise three to six participants. The process is led by a moderator impartial to the software element(s) under examination. The moderator is not the author. Defect resolution is mandatory, and rework is formally verified.

Defect data shall be systematically collected and stored in an inspection data base. Minimum requirements for data collection are presented in 6.10. When this data is analyzed, steps can be taken to improve both the product and the development process, and to assess the effectiveness of the inspection process.

6.3 Special Responsibilities. All team members, including those with special roles, are inspectors. Special roles required for the inspection process include

6.3.1 Moderator. The moderator is the chief planner and meeting manager for the inspection process, and is responsible for issuing the inspection reports (see 6.8). The moderator may also perform the role of recorder.

6.3.2 Reader. At the meeting(s), the reader leads the inspection team through the software element(s) in a comprehensive and logical fashion, generally paraphrasing sections of

the work and by reading line-by-line where required by local standards.

6.3.3 Recorder. The recorder is charged with documenting defects detected at the meeting(s) and recording inspection data required for process analysis.

6.3.4 Inspector. The role of inspector is to identify and describe defects in the software element. Inspectors must be knowledgeable of the inspection process. They are chosen to represent different viewpoints at the meeting (for example, requirements, design, code, test, independent test, project management, quality management). Only those viewpoints pertinent to the inspection of the element are present.

6.3.5 Author. The author is responsible for the software element(s) meeting its inspection entry criteria, for contributing to the inspection based on special understanding of the software element, and for performing any rework required to make the software element(s) meet its inspection exit criteria. The author is not allowed to perform any other (that is, moderator, reader, or recorder) role.

6.4 Input. The necessary inputs to the inspection consist of the following:

(1) The software element(s) to be inspected; They must satisfy the inspection entry criteria;
(2) The approved software element(s) specification;
(3) The inspection checklist. See Appendix B for sample checklists;
(4) Any standards and guidelines against which the software element is to be inspected;
(5) All necessary inspection reporting forms.

For a reinspection, the previous inspection defect list is also required.

6.5 Entry Criteria

6.5.1 Authorization. Inspections are planned for, and documented in the appropriate project planning documents (for example, the overall project plan, software quality assurance plan, or software verification and validation plan).

6.5.2 Initiating Event. The software inspection process can be triggered by the following:

(1) Software element(s) availability
(2) Project plan compliance
(3) SQAP or SVVP schedule compliance
(4) Scheduled reinspection
(5) At the request of functional management, project management, software quality management, or software engineering

6.5.3 Minimum Entry Criteria Before Pannning Inspection. Before planning the inspection, the following minimum entry criteria must be met:

(1) The software element(s) conform to project standards of content, and format
(2) All prior milestones are satisfied, as identified in the the appropriate planning documents
(3) All required supporting documentation is available
(4) For a reinspection, all items noted on the defect list must be satisfied

6.6 Procedures

6.6.1 Planning. During the planning step, the author assembles the inspection package materials for the moderator.

The moderator is responsible for assuring that the materials meet the inspection entry criteria. The moderator is also responsible for assuring the selection of the inspection team and the assignment of their inspection meeting roles, for scheduling the inspection meetings, and for the distribution of the inspection materials.

6.6.2 Overview. If scheduled, an overview presentation of the software element(s)to be inspected is conducted by the moderator, and the author makes the presentation. This overview is used to educate the other inspectors concerning the software element(s) and may also be attended by other project personnel who could profit from the presentation.

6.6.3 Preparation. It is the individual responsibility of all inspectors to become thoroughly familiar with the software element, using the inspection checklist, the information provided in the overview, and the software specification.

6.6.4 Examination. The inspection meeting shall follow this agenda:

6.6.4.1 Introduce Meeting. To open the meeting, the moderator introduces the participants, and describes their roles. The moderator states the purpose of the inspection and directs the inspectors to focus their efforts toward defect detection, not solution-hunting.

The moderator reminds the inspectors to direct their remarks to the reader and to comment only on the product, not the author. It may also be useful to resolve any special procedural questions raised by the inspectors.

6.6.4.2 Establish Preparedness. The moderator asks for individual preparation times and records the total on the inspection report. It is the moderator's responsibility to reschedule the meeting if the inspectors are not adequately prepared.

6.6.4.3 Review the Inspection Checklist. The moderator reviews the inspection checklist with the team to ensure that the product has been adequately studied before the inspection meeting.

6.6.4.4 Read Software Elements and Record Defects. The reader presents the materials to the inspection team. The inspection team examines the software objectively, and the moderator focuses this part of the meeting on creating the defect list. The recorder enters each defect, location, description, and classification on the defect list. During this time, the author answers any specific questions and contributes to defect detection based on his/her special understanding of the software element.

6.6.4.5 Review the Defect List. At the end of the inspection meeting, the moderator must have the defect list reviewed with the team to ensure its completeness and accuracy.

6.6.4.6 Make Exit Decision. The purpose of the exit decision is to bring an unambiguous closure to the inspection meeting. The exit decision determines if the materials meet the inspection exit criteria and prescribes any appropriate rework verification. Specifically, the inspection team identifies the software element(s) disposition as one of the following:

(1) *Accept.* The software element is accepted *as is* or with only minor rework (for example, that would require no further verification)
(2) *Verify Rework.* The software element is to be accepted after the moderator verifies rework.
(3) *Reinspect.* Schedule a reinspection to verify rework after revision.

At a minimum, a reinspection examines the product areas changed to resolve defects identified in the last inspection.

6.6.5 Rework. During rework, the author revises the materials, addressing all items on the inspection defect list.

6.6.6 Follow-Up. The software inspection process provides follow-up on two levels:

(1) Verifying rework per inspection disposition
(2) Reporting inspection data

6.7 Exit Criteria. A single exit criterion applied to all inspections is that all of the defects that have been detected are resolved. Each project shall develop its own criteria to meet the needs of its specific products and development environment.

The moderator will ensure that the materials comply with the exit criteria and that the inspection results have been reported before certifying that the inspection is complete.

6.8 Output. For each software element(s), the reports produced by the inspection are:

(1) The defect list, containing the defect location, description, and categorization
(2) The inspection defect summary summarizing the number of defects identified by each defect category
(3) The inspection report, containing
    (a) The number of participants
    (b) The inspection meeting duration
    (c) The size of the materials inspected
    (d) The total preparation time of the inspection team
    (e) The disposition of the software element
    (f) An estimate of the rework effort and rework completion date

Information reports are retained for subsequent analysis as prescribed in the appropriate project planning documents. Although this standard sets minimum requirements for report content, it is left to local standards to prescribe any report format requirements.

6.9 Auditability. Auditability is provided by the following:

(1) Documented inspection procedures
(2) Retained inspection reports
(3) Retained inspection defect data

6.10 Data Collection Requirements. Software inspections provide data for the analysis of the quality of the software elements, the effectiveness of the development procedures, and the

efficiency of the inspection process. To enable these analyses, defects that are identified at an inspection meeting are categorized by defect type, class, and severity.

6.10.1 The defect type identifies software element(s) attribute, and for example may be identified as affecting the software element's conformance to

(1) Standards compliance
(2) Capability
(3) Procedure
(4) Interface
(5) Description

6.10.2 Defect class characterizes evidence of nonconformance, and for example, may be categorized as

(1) Missing
(2) Wrong
(3) Extra

6.10.3 Additional defect classes used when inspecting documents could include

(1) Ambiguous
(2) Inconsistent

6.10.4 Defects are then ranked by severity, for example, as

(1) *Major*. Defects that would result in failure of the software element(s) or an observable departure from specification
(2) *Minor*. Defects that would affect only the nonfunctional aspects of the software element

In addition to counts of defects by type, class, and severity, retained inspection data shall contain the identification of the software element, the date and time of the inspection, the moderator, the preparation and inspection times, the volume of the materials inspected, and the disposition of the inspected software element. The recording of this information is required in setting local guidance for inspection effort and volume. Common practice would suggest up to 1.5 h preparation, up to 2 h meeting time, and whatever volume might be comfortably handled in that period of time.

The management of inspection data requires capabilities for the storage, entry, access, updating, and summarization and reporting of the categorized defects. The frequency and types of the inspection analysis reports, and their distribution, are left to the users of this standard.

7. The Walkthrough Process

7.1 Objective. The objective of a walkthrough is to evaluate a software element. Although long associated with code examinations, this process is also applicable to other software elements (for example, architectural design, detailed design, test plans/procedures, and change control procedures). The major objectives are to find defects, omissions, and contradictions; to improve the software element(s); and to consider alternative implementations.

Other important objectives of the walkthrough process include exchange of techniques and style variations, and education of the participants. A walkthrough may point out efficiency and readability problems in the code, modularity problems in design or untestable design specifications.

The walkthrough may be used in either new development or maintenance activities.

7.2 Abstract. During the walkthrough meeting, the author makes an overview presentation of the software element(s) under review. This is followed by a general discussion from the participants after which the presenter *"walks through"* the software element in detail. As the walkthrough progresses, errors, suggested changes, and improvements are noted and written. When the walkthrough is finished, the notes are consolidated into one report, which is distributed to the author and other appropriate personnel. A copy of the report is placed in the project file for auditability.

7.3 Special Responsibilities. The following roles are established for the walkthrough:

7.3.1 Moderator. The walkthrough moderator is responsible for conducting a specific walkthrough, handling the administrative tasks pertaining to the walkthrough, and ensuring that the walkthrough is conducted in an orderly manner. The moderator is also responsible for preparing the statement of objectives to guide the team through the walkthrough.

7.3.2 Recorder. The recorder is responsible for writing all comments made during the walkthrough that pertain to errors found,

questions of style, omissions, contradictions, suggestions for improvement, or alternative approaches.

7.3.3 Author. The author is the person responsible for the software element(s) being examined, and presents the materials.

Each member of the walkthrough team is responsible for reviewing any input material prior to the walkthrough, and participating during the walkthrough to ensure that it meets its objective.

Roles may be shared among the walkthrough members. The author and walkthrough moderator may be the same person.

7.4 Input. The minimum input to the walkthrough process is as follows:

(1) A statement of objectives for the walkthrough
(2) The software element under examination
(3) Standards that are in effect for the development of the software element
(4) Specifications for the software element(s) under review

7.5 Entry Criteria

7.5.1 Authorization. The need for conducting walkthroughs shall be established in the appropriate planning documents, (for example, a software quality assurance plan (SQAP) or a software development plan). In addition, under either of these plans, completion of a specific software element(s) trigger the walkthrough for that element. Additional walkthroughs should be conducted during development of the software element(s) at the request of functional management, quality management, or the author, according to local procedures.

7.5.2 Initiating Event. A walkthrough is conducted when

(1) The author of the software element(s) to be reviewed indicates it is ready, and is prepared to present the materials.
(2) The moderator is appointed per local standards or practices and has concluded that the software element is sufficiently complete to conduct a walkthrough.

7.6 Procedures

7.6.1 Planning. During the planning phase, the walkthrough moderator does the following:

(1) Identifies the walkthrough team
(2) Schedules the meeting and selects the meeting place
(3) Distributes all necessary input materials to the participants, allowing for adequate preparation time

7.6.2 Overview. An overview presentation is made by the author as part of the walkthrough meeting. Before that meeting, however, individual preparation is still required.

7.6.3 Preparation. During the preparation phase participants shall review the input material that was distributed to them and prepare a list of questions and issues to be discussed during the walkthrough.

7.6.4 Examination. During the walkthrough meeting

(1) The presenter makes an overview presentation of the software element(s) under examination.
(2) The author walks through the specific software element(s) so that members of the walkthrough team may ask questions or raise issues regarding the software element, or document their concerns, or both.
(3) The recorder writes comments and decisions for inclusion in the walkthrough report.

At the completion of the walkthrough, the walkthrough team may recommend a follow-up walkthrough. This follow-up would follow the standard for walkthroughs, and would, at a minimum, cover areas changed by the author.

Within a reasonable time after the walkthrough, the walkthrough leader shall issue a walkthrough report detailing the walkthrough findings. Minimum content requirements are stated in 7.8.

7.7 Exit Criteria. The walkthrough process is complete when

(1) The entire software element(s) have been *"walked through "* in detail
(2) All deficiencies, omissions, efficiency issues, and suggestions for improvement have been noted
(3) The walkthrough report has been issued

7.8 Output. The output of the walkthrough process is a walkthrough report listing those deficiencies, omissions, efficiency issues, and suggestions for improvement that were recorded during the walkthrough.

The walkthrough report contains the following:

(1) Identification of the walkthrough team
(2) Identification of the software element(s) being examined
(3) The statement of objectives that were to be handled during this walkthrough meeting
(4) A list of the noted deficiencies, omissions, contradictions, and suggestions for improvement
(5) Any recommendations made by the walkthrough team on how to dispose of deficiencies and unresolved issues

If follow-up walkthroughs are suggested, this suggestion shall also be mentioned in the report.

Although this standard sets minimum requirements for report content, it is left to local standards to prescribe any report format requirements.

7.9 Auditability. The walkthrough report, meeting minutes, and other materials upon which conclusions are based shall be included in project documentation files. The walkthrough moderator files these documents.

8. The Audit Process

8.1 Objective. The objective of software auditing is to provide an objective compliance confirmation of products and processes to certify adherence to standards, guidelines, specifications, and procedures.

8.2 Abstract. Audits are performed in accordance with documented plans and procedures. The audit plan establishes a procedure to conduct the audit and for follow-up action on the audit findings.

In performing the audit, audit personnel evaluate software elements and the processes for producing them against objective audit criteria, such as contracts, requirements, plans, specifications, or procedures, guidelines, and standards.

The results of the audit are documented and are submitted to the management of the audited organization, to the entity initiating the audit, and to any external organizations identified in the audit plan. The report includes a list of the items in noncompliance or other issues for subsequent review and action. When stipulated by the audit plan, recommendations are reported in addition to the audit results.

8.3 Special Responsibilities. It is the responsibility of the audit team leader to organize and direct the audit and to coordinate the preparation and issuance of the audit report. The audit team leader shall ensure that the audit team is prepared to conduct the audit, and that audit procedures are performed and reports (see 8.8) issued in accordance with audit scope.

The entity initiating the audit is responsible for authorizing the audit. Management of the auditing organization assumes responsibility for the audit, and the allocation of the necessary resources to perform the audit.

Those whose products and processes are being audited provide all relevant materials and resources and correct or resolve deficiencies cited by the audit team.

8.4 Input. The following inputs are required to ensure the success of the audit:

(1) The purpose and scope of the audit.
(2) Objective audit criteria, such as contracts requirements, plans, specifications, procedures, guidelines, and standards.
(3) The software elements and processes to be audited and any pertinent histories.
(4) Background information regarding the organization responsible for the products and processes being audited (for example, organization charts).

8.5 Entry Criteria. The need for an audit is established by one of the following events:

(1) A special project milestone has been reached. The audit is initiated per earlier plans (for example, the software quality assurance plan, software development plan).
(2) External parties (for example, regulatory agencies, or end users) demand an audit at a specific calendar date or project milestone. This may be in fulfillment of a contract requirement, or as a prerequisite to contractual agreement.

(3) A local organizational element(s) (for example, project management, functional management, systems engineering, internal quality assurance/control) has requested the audit, establishing a clear and specific need.
(4) A special project milestone, calendar date, or other criterion has been met and, as part of the auditing organization's charter, it is to respond by initiating an audit.

8.6 Procedures

8.6.1 Planning. The auditing organization shall develop and document an audit plan for each audit. This plan shall, in addition to restating the audit scope, identify the following:

(1) Project processes to be examined (provided as input) and the time frame for audit team observation.
(2) Software required to be examined (provided as input) and their availability. Where sampling is used, a statistically valid sampling methodology shall be used to establish selection criteria and sample size.
(3) Reports shall be identified (that is, results report, and optionally, the recommendations report and their general format defined). Whether recommendations are required or excluded shall be explicitly stated.
(4) Report distribution
(5) Required follow-up activities
(6) Requirements: necessary activities, elements and procedures to meet the scope of the audit
(7) Objective audit criteria: provides the basis for determining compliance (provided as input)
(8) Audit procedures and checklists
(9) Audit personnel: required number, skills, experience, and responsibilities
(10) Organizations involved in the audit (for example, the organization whose products and processes are being audited)
(11) Date, time, place, agenda, and intended audience of *overview* session (optional)

The audit team leader shall ensure that the audit team is prepared and includes members with the necessary experience and expertise.

Notification of the audit shall be provided to the involved organizations at a reasonable amount of time before the audit is performed, except for unannounced audits. The notification shall be written and shall include the scope and the identification of processes and products to be audited, and to identify the auditors.

8.6.2 Overview. An optional overview meeting with the audited organization is recommended to *kick-off* the examination phase of the audit. The overview meeting, led by the audit team leader, provides the following:

(1) Overview of existing agreements (for example, audit scope, plan, related contracts);
(2) Overview of production and processes being audited;
(3) Overview of the audit process, its objectives, and outputs;
(4) Expected contributions of the audited organization to the audit process (that is, the number of people to be interviewed, meeting facilities, etc);
(5) Specific audit schedule.

8.6.3 Preparation

8.6.3.1 The following preparations are required by the audit team:

(1) Understand the organization: it is essential to identify functions and activities performed by the audited organization, and to identify functional responsibility.
(2) Understand the products and processes: it is a prerequisite for the team to learn about the products and processes being audited through readings and briefings.
(3) Understand the objective audit criteria: it is important that the audit team become familiar with the objective audit criteria to be used in the audit.
(4) Prepare for the audit report: it is important to choose the administrative reporting mechanism that will be used throughout the audit to develop the report that follows the layout identified in the audit plan.
(5) Detail the audit plan: choose appropriate methods for each step in the audit program.

8.6.3.2 In addition, the audit team leader shall make the necessary arrangements for

(1) Team orientation and training as needed

(2) Facilities for audit interviews
(3) Materials, documents, and tools required by the audit procedures
(4) The software elements to be audited (for example, documents, computer files, personnel to be interviewed)
(5) Scheduling interviews

8.6.4 Examination. Elements that have been selected for audit shall be evaluated against the objective audit criteria. Evidence shall be examined to the depth necessary to determine if these elements comply with specified criteria.

The audit shall, as appropriate to its scope:
(1) Review procedures and instructions
(2) Examine the work breakdown structures
(3) Examine evidence of implementation and balanced controls
(4) Interview personnel to ascertain the status and functioning of the processes and the status of the products
(5) Examine element documents
(6) Test the element(s)

8.6.5 Reporting. Subsequent to the audit examination, the audit team will issue a draft audit report to the audited organization for review and comments.

Audit team rework of the audit report occurs before formal results reporting. This rework is performed, in concert with the draft report review, resolve any misunderstandings or ambiguities while maintaining objectivity and correctness. It also serves to ensure report usability by adding consistency to the level of report details and by adding any freshly verified information. The recommended practice is to involve representatives of the audited organization in reviewing audit results, as
(1) An ongoing, periodic activity integrated into the overall examination schedule, or
(2) As a closure step to the examination period

Involving the audited organization contributes to report quality through interaction and the possible delivery of any further evidence.

The audit team shall conduct a post audit conference to review with audited organization staff the deficiencies, findings, and (if applicable) recommendations. Comments and issues raised by the audited organization must be resolved.

The final audit report shall then be prepared, approved, and issued by the audit team leader to the organizations specified by the audit plan.

8.7 Exit Criteria. An audit shall be considered complete when
(1) Each element(s) within the scope of the audit has been examined;
(2) Findings have been presented to the audited organization;
(3) Response to draft findings have been received and evaluated;
(4) Final findings have been formally presented to the audited organization and initiating entity;
(5) The audit report has been prepared and submitted to recipients designated in the audit plan;
(6) The recommendation report, if required by plan, has been prepared and submitted to recipients designated in the audit plan;
(7) All of the auditing organization's follow-up actions included in the scope (or contract) of the audit have been performed.

8.8 Output. As a standard framework for audit reports, the draft and final audit reports shall, as a minimum, contain the following:
(1) *Audit identification.* Report title, audited organization, auditing organization, date of the audit.
(2) *Scope*. Scope of the audit, including an enumeration of the standards, specifications, practices, and procedures constituting the objective audit criteria against which the audit was conducted of the software elements and processes audited.
(3) *Conclusions.* A summary and interpretation of the audit findings including the key items of nonconformance.
(4) *Synopsis*. A listing of all the audited software elements, processes, and associated findings.
(5) *Follow-up*. The type and timing of audit follow-up activities.

Additionally, when stipulated by the audit plan, recommendations shall be provided to the audited organization, or the entity that initiated the audit. Recommendations shall be reported separately from results.

8.9 Auditability. The materials documenting the audit process must be maintained by the audit organization for a stipulated period of time subsequent to the audit and include the following:

(1) All work programs, checklists, etc are fully annotated
(2) Team staffing
(3) Interview notes, observation notes
(4) Compliance test evidence
(5) Copies of examined items, annotated
(6) Report draft(s) with response by audited organization
(7) Follow-up memo(s) as necessary

Appendixes

(These Appendixes are not a part of IEEE Std 1028-1988, IEEE Standard for Software Reviews and Audits, but are included here for information only.)

Appendix A

Guide to Process Applications for Critical Software

A1.

ANSI/IEEE Std 730-1984 [A1] has identified a minimum set of review and audit process applications for critical software (for example, where failure could impact safety, or cause large financial or social losses).

(1) Software requirements review (SRR)
(2) Preliminary design review (PDR)
(3) Critical design review (CDR)
(4) Software verification and validation plan review (SVVPR)
(5) Functional audit (FA)
(6) Physical audit (PA)
(7) In-process audit (IPA)
(8) Managerial review (MR)

A2.

A2.1 The goal of this Appendix is to provide one possible mapping of specific process application needs identified in that standard, to candidate processes for meeting those needs. Selection of a specific process should be made as local, contractual, or project specific conditions dictate.

For example:

(1) The software inspection process may be used to meet the software verification and validation plan review application need.
(2) The management review process may be used to meet the managerial review application need.

Table A1
Critical Software Examination

ANSI/IEEE Std 730-1984 Requirements	SRR	PDR	CDR	SVVPR	FCA	PCA	IPA	MR
Management review			X					X
Technical review	X	X	X	X				
Software inspection	X	X	X	X				
Walkthrough	*	*	*	*				
Audit					X	X	X	**

* As common practice in many companies, walkthroughs are allowed wherever software inspections might be applied. It is suggested however, that for critical software as presented in ANSI/IEEE Std 730-1984, only more formal/rigorous techniques be used. One significant difference is disallowing the author from presenting the work.

** The audit process may be tailored in its application or administration to meet the need for a managerial review.

A particular process need not be prescribed for all applications. For example, using the same process for all the SRR, PDR, and CDR might not be cost effective or might unwisely be biased toward either technical or business concerns.

A2.2 Similarly, a combination of distinct processes may be applied to meet the needs of a specific application. An example is providing adequate examination of an SRR for a very large and complex system. SRR concerns could be subdivided into three categories:

(1) Customer satisfaction with functionality and performance with consideration for technical and business feasibility
(2) Technical completeness and correctness of requirements
(3) Fitness-for-use of the specifications document(s) as viewed by development, test, and maintenance organizations

In this example, it may be appropriate to use a management review to examine the first category, a technical review for the second category, and a software inspection to verify fitness-for-use of the document itself.

Table A1 shows possible mappings of review and audit processes to the applications referenced in ANSI/IEEE Std 730-1984 [A1].

A3.

The following references apply to Appendixes A through C:

[A1] ANSI/IEEE Std 730-1984, IEEE Standard for Software Quality Assurance Plans.

[A2] ANSI/IEEE Std 1016-1987, IEEE Recommended Practice for Software Design Descriptions.

Appendix B

Guide to Specific Review Applications

This Appendix is added to guide in the application of review processes, and to delineate some specific applications.

Although review processes defined by this standard have distinct objectives, Table B1 is provided to further accent process distinctions.

In addition, the following review process applications are presented:

(1) Software requirements review (SRR)
(2) Preliminary design review (PDR)
(3) Critical design review (CDR)
(4) Software verification and validation plan review (SVVPR)
(5) Managerial reviews (MR)

B1. Software Requirements Review (SRR)

The SRR is held to ensure the adequacy of the requirements stated in the software requirements specification. See ANSI/IEEE Std 730-1984 [A1].

The SRR is an evaluation of the software requirements specification (SRS) to ensure adequacy, technical feasibility, and completeness. Moreover, the SRR is held with user participation to examine the SRS to verify that is is unambiguous, verifiable, consistent, modifiable, traceable, and usable during the operation and maintenance phase. All organizational elements (for example, software design, software test, software quality assurance, systems engineering, marketing, manufacturing, and field service) that are directly affected by the requirements should participate.

The review checklist for the software requirements review may include but is not limited to the following:

(1) Traceability and completeness of the requirement from the next higher level specification (such as a system specification or user requirements specification)
(2) Adequacy of rationale for any derived requirements
(3) Compatibility of external (hardware and software) interfaces
(4) Adequacy of requirements for the man-machine interface
(5) Availability of constants and tables for calculations
(6) Testability of the functional requirements
(7) Adequacy and completeness of the verification and acceptance requirements
(8) Conformance to requirements specification standards
(9) Adequacy and feasibility of performance requirements
(10) Adequacy of consideration for human factors

B2. Preliminary Design Review (PDR)

B2.1 The PDR is held to evaluate the technical adequacy of the preliminary design of the software as depicted in a preliminary version of the Software Design Description. See ANSI/IEEE Std 730-1984 [A1].

Specifically, the need is to apply a review process to assess the progress, consistency, and technical adequacy of the selected design approach.

All organizational elements (for example, software test, software quality assurance, systems engineering, marketing, manufacturing,and field service) that are directly affected by the software design description should participate.

The following items may be included in the review checklist for the preliminary design review (PDR):

(1) Compatibility of the interfaces between the software, hardware, and end users.
(2) Fulfillment of the requirements contained in the SRS.
(3) Conformance to design documentation standards.

B2.2 Concerns that may be addressed by the checklist for preliminary design review

Table B1

Category and Attributes	Management Review	Technical Review	Software Inspection	Walkthrough
Objective	Ensure progress. Recommend corrective action. Ensure proper allocation of resources.	Evaluate conformance to specifications and plans. Ensure change integrity.	Detect and identify defects. Verify resolution.	Detect defects. Examine alternatives. Forum for learning.
Delegated Controls				
Decision making	Management team charts course of action. Decisions are made at the meeting or as a result of recommendations.	Review team petitions. Management or technical leadership to act on recommendations.	Team chooses from predefined product dispositions. Defects must be removed.	All decisions made by producer. Change is the prerogative of the producer.
Change verification	Change verification left to other project controls.	Leader verifies as part of review report.	Moderator verifies rework.	Change verification left to other project controls.
Group Dynamics				
Recommended size	Two or more persons.	Three or more persons.	Three to six persons.	Two to seven persons.
Attendance	Management, technical leadership, and peer mix.	Technical leadership and peer mix.	College of peers meet with documented attendance.	Technical leadership and peer mix.
Leadership	Usually the responsible manager.	Usually the lead engineer.	Trained moderator.	Usually producer.
Procedures				
Matrial volume	Moderate to high, depending on the specific *statement of objectives* for the meeting.	Moderate to high, depending on the specific *statement of objectives* for the meeting.	Relatively low.	Relatively low.
Presenter	Project representative.	Software element representative.	Presentation by *reader* other than producer.	Usually the producer.
Data collection	As required by applicable policies, standards, or plans.	Not a formal project requirement. May be done locally.	Formally required.	Not a formal project requirement. May be done locally.
Outputs				
Reports	Management review report.	Technical review reports.	Defect list and summary. Inspection report.	Walkthrough report.
Data base entries	Any schedule changes must be entered into the project tracking database.	No formal data base required.	Defect counts, characteristics, severity, and meeting attributes are kept.	No formal data base requirement.

(PDR) include (but are not limited to):

(1) Software interfaces
(2) Hardware interfaces
(3) Baseline for configuration
(4) Functional overview
(5) Design alternative selection rationale
(6) Standard operating environments
(7) Human factors requirements
(8) Documentation requirements
(9) Testing considerations
(10) Manufacturability
(11) Maintainability
(12) Assumptions
(13) Risks

More information is available in ANSI/IEEE Std 1016-1987 [A2].

B3. Critical Design Review (CDR)

The CDR is held to determine the acceptability of the detailed software designs as depicted in the software design description in satisfying the requirements of the software requirements specification. See ANSI/IEEE Std 730-1984 [A1].

The CDR is an evaluation of the completed software design description (SDD) to evaluate its technical adequacy, completeness, and correctness. Other concerns include its compatibility with the other software and hardware (where the product is required to interact), and the technical, cost, and schedule risks of the product design.

All other organizational elements that impose requirements or are affected by the design should participate (for example, software design, software test, and systems engineering).

The following items may be included in the review checklist for the critical design review:

(1) The compatibility of the detailed design with the SRS
(2) Available data in the form of logic diagrams, algorithms, storage allocation charts, and detailed design representations (for example, flow chart, program design language) to establish design integrity
(3) Compatibility and completeness of interface design requirements
(4) Adequacy and completeness of algorithms and equations
(5) Correctness and suitability of logic descriptions that may be warranted
(6) Technical accuracy and currency of all available test documentation and its compatibility with the test requirements of the SRS
(7) The requirements for the support and test software and hardware to be used in the development of the product
(8) The inclusion in the final design of function flow, timing, sizing, storage requirements, memory maps, data base, and other performance factors

B4. Software Verification and Validation Plan Review (SVVPR)

The software verification and validation plan review is held to evaluate the adequacy and completeness of the verification and validation methods described in the SVVP. See ANSI/EEE Std 730-1984 [A1].

The SVVPR is an evaluation of the completed software verification and validation plan (SVVP). Since the SVVP may be developed incrementally, multiple reviews may be required. These reviews are held to ensure that the verification and validation methods described in the SVVP are adequate and will provide complete evaluation data.

All organizational elements that impose requirements or are impacted by the SVVP should participate. These groups may include system engineering, software marketing, software design, software test, software quality assurance, customers, and users.

The following items may be included as review requirements for the software verification and validation plan review:

(1) All verification and validation methods, along with completion criteria to assure traceability to, and compatibility with, the functional and performance requirements expressed in the SRS.
(2) The methods to be used to verify and validate that the product satisfies the requirements of the SRS; these should include review process applications (for example, technical review, inspection, or walkthrough), demonstration, analysis, and test.

(3) Reports that adequately document the results of all reviews, audits, and tests based on the requirements listed in the SVVP.
(4) Adequate descriptions of the software configuration to be tested including test support software and hardware.
(5) Test plans, test procedures, test cases, and test data to ensure that all specification requirements are tested and that test instructions are clear and concise.
(6) Test procedures and test cases to ensure that test inputs and success criteria are adequately defined.
(7) A test plan and test case schedule identifying tests to be done, when, and by whom.
(8) Conformance to test documentation standards.

B5. Managerial Reviews

These reviews are held periodically to assess the execution of (the SQA) plan. These reviews shall be held by an organizational element(s) independent of the unit being audited, or by a qualified third party. See ANSI/IEEE Std 730-1984 [A1].

The planned frequency and structure of the (SQA Plan) managerial reviews should be stated in the SQA Plan. They should be conducted at the direction of the program manager.

Although the application of the *Management Review Process* meets this review requirement, an alternative is to perform the *Quality Systems Audit* (as presented in Appendix C) to the extent possible given the particular degree of independence and authority.

Appendix C

A Guide to Specific Audit Applications

To guide in the application of the audit process, and to delineate some specific applications, this Appendix has been added. The following audit process applications are presented:

(1) Functional audit (FA)
(2) Physical audit (PA)
(3) In-process audit (IPA)
(4) Quality systems audit (QSA)

Where self-examination (that is, application of the audit process by an element(s) not organizationally independent of the area audited) is desired, the general audit process is applied, with the independence requirement waived.

C1. The Functional Audit (FA)

The objective of the functional audit (FA) is to provide an independent evaluation of software products, verifying that its configuration items' actual functionality and performance is consistent with the requirement specifications. Specifically, this audit is held prior to the software delivery to verify that all requirements specified in the Software Requirements Specification have been met. See ANSI/IEEE Std 730-1984 [A1].

C1.1 Central to the FA is the identification of software elements to be audited by:

1) Nomenclature
2) Specification identification number
3) Configuration item number

C1.2 In addition, materials to be provided by the audited organization include:

(1) The software requirements specification
(2) Object copy of code
(3) Current listing of waivers against specific configuration items
(4) When available, the status of test programs to test configured items with automatic test equipment
(5) The software verification and validation report
(6) All in-process audit reports for items under audit (see C3)
(7) Any other test documentation (for example, plans, specifications, procedures, and reports) for the configuration item
(8) Listing of all successfully accomplished functional testing
(9) Listing of all planned, yet unexecuted, functional testing

C1.3 The following activities are required to conduct a functional audit:

(1) An audit of the formal test documentation shall be performed against test data. Results shall be checked for completeness and accuracy, and all deficiencies documented.
(2) An audit of the software verification and validation report shall be made to validate that the reports are accurate and completely describe the testing effort. All approved changes (problem reporting and corrective action) shall be reviewed to ensure that they have been technically incorporated and verified.
(3) Updates to previously delivered documents shall be reviewed to ensure accuracy and consistency.
(4) Preliminary design review and critical design review outputs shall be sampled to ensure that all findings have been incorporated and completed.

The audit team shall compare the code with the documented software requirements as stated in the current specifications document, to determine that the code addresses all (and only) the documented requirements.

C1.4 This audit is concerned not only with functionality, but also with performance as follows: Testing accomplished with appropriate test documentation and validated data shall sufficiently establish configuration item performance. For performance parameters that can not be verified completely during testing, simulations or other analysis shall be accomplished to establish confidence that the configuration item does meet performance criteria when in operation as stated in its specifications.

The FA report should include an evaluation of each software element(s) examined, and provides this evaluation within the standard audit report framework. The evaluation shall take the form of

(1) *Approval.* If the software element(s) had been observed as having all required functions and shows evidence of meeting minimum performance standards.
(2) *Contingent Approval.* If the evaluation of the software element(s) would result in approval given successful completion of specific, well-defined (observable) corrective action.
(3) *Disapproval.* If the software element(s) were observed to be seriously inadequate.

C2. The Physical Audit (PA)

The objective of the physical sudit (PA) is to provide an independent evaluation of a software product configuration item to confirm that components in the *as-built version* map to its specifications. Specifically, this audit is held to verify that the software and its documentation are internally consistent and are ready for delivery. See ANSI/IEEE Std 730-1984 [A1].

C2.1 Central to the PA is the identification of software elements to be audited by

(1) Nomenclature
(2) Specification identification number
(3) Configuration item number

C2.2 In addition, materials to be provided by the audited organization include

(1) Current listing of waivers against specific configuration items
(2) Requirements and architectural and detailed design specifications
(3) Listing of approved changes against the configuration with outstanding changes identified
(4) Acceptance test documentation (for examplee, plans, specifications, procedures, and reports) for the configuration item
(5) Appropriate customer deliverable documentation (for example, operating manuals, and maintenance manuals)
(6) Approved nomenclature, markings, and nameplates
(7) Software version description document
(8) FA reports

In addition, the audited organization shall identify any differences between the configuration of the system in the development environment, and the specific configuration of the system under examination. Moreover, they shall certify or otherwise demonstrate that any differences do not degrade the functional or performance characteristics of the system.

C2.3 To provide evidence of adequate control of system content and consistency, the audit team should as a minimum, examine

(1) The system specification document(s) for format and completeness
(2) Functional (FA) reports for discrepancies and actions taken
(3) Design descriptions (at least on a sampling basis) for proper symbols, labels, references, and data descriptions
(4) Design descriptions at various levels of abstraction, cross-checking for consistency
(5) Manuals (for example, user's manuals, programmer's manuals, and operator's manuals,) for format completeness and conformance with data item descriptions. Formal verification or acceptance of these manuals shall be waived until all other system testing is completed to ensure that procedural contents are correct.
(6) The software element(s) delivery media (and media controls) to ensure proper transfer or transmittal.

The audit team shall examine the system for vendor-provided or supplier-developed software elements, and seek evidence that any such software elements were produced under reasonable and prudent quality controls.

C2.4 The audit team shall examine the engineering release and change control function (and associated documentation) for evidence that it can

(1) Identify the composition of any configuration in terms of its subordinate units
(2) Determine, given a unit identifier, the parent element

(3) Describe the composition of a configuration item or its identifier with respect to other configuration items or identifiers
(4) Identify specifications relating to a given configuration item
(5) Identify outstanding changes against a given configuration item
(6) Identify changes and retain records of earlier configurations

C2.5 The PA report should include an evaluation of all software elements as a whole, and shall provide this evaluation within the standard audit report framework. The evaluation shall take the form of
(1) Approval. When the system of software elements has been observed as having all deliverable components accounted for, and of compatible issue.
(2) Contingent Approval. When the evaluation of the system of software elements results in approval given successful completion of specific, well-defined (observable) corrective action.
(3) Disapproval. When the system of software elements was observed to be seriously inadequate.

C2.6 The audit team shall examine the engineering release and change control function (and associated documentation) for evidence that it can
(1) Identify the composition of any configuration in terms of its subordinate units
(2) Determine, given a unit identifier, the parent element
(3) Describe the composition of a configuration item or its identifier with respect to other configuration items' identifiers
(4) Identify specifications relating to a given configuration item

C3. In-Process Audits

In-process audits are held during the design and development phases prior to the functional audit to verify the consistency of the design.

The objective is to verify the consistency of the product as it evolves through the development process by determining, on a sampling basis, that
(1) Hardware and software interfaces are consistent with design requirements in the software requirements specification (SRS)
(2) The code is fully tested by the SVVP to ensure that the functional requirements of the SRS are satisfied
(3) The design of the product, as the software design description (SDD) is evolving, satisfies the functional requirements of the SRS
(4) The code is consistent with the SDD

Moreover, in-process audits can be applied to determine compliance with applicable process standards (for example, auditing of the inspection process, on a sampling basis, to determine if inspection process rules are being followed).

Input material should include formal process descriptions and a product sample evidencing process execution.

In-process audit procedures are unique to the process and product examined. Specific procedures should be established locally, using the general steps found in Section 8.

The results of all in-process audits shall be documented in in-process audit reports, which identify all discrepancies found, and provide an overall opinion accordingly. The opinion should take the form of approval, contingent approval, or disapproval. Significant compliance deviations are noted and must be followed up to ensure their correction or waiver. This occurs during or before the functional audit.

C4. The Quality Systems Audit (QSA)

The purpose of a quality systems audit, or quality program evaluation, is to provide an independent assessment of the compliance to a software quality assurance plan.

C4.1 Specifically, the QSA objective is to determine, based on observable (verifiable) evidence, that
(1) The software quality program documentation established by the development organization addresses as a minimum, the basic elements of ANSI/IEEE Std 730-1984 [A1], or other appropriate standards.

(2) The software development organization follows its documented software quality program.

The SQA plan must incorporate all objective organizational performance criteria; internal standards and procedures; process requirements mandated by law, contract, or other policy; and conform to the ANSI/IEEE 730-1984 [A1] or other appropriate standards for software quality assurance.

C4.2 Central to the QSA is the identification of quality program documentation, and its availability to the audit team.

The following activities are required:

(1) A preaudit conference with audited organization personnel is held to familiarize them with audit goals and procedures
(2) Examination of quality program documents and related items for completeness, traceability, and consistency
(3) Perform selective compliance testing to objectively ascertain that SQA procedures and standards are implemented correctly
(4) Interview staff
(5) Perform in-process audits as appropriate (for example, observing inspections and their documented evidence)
(6) Examine reports from any recent functional and physical configuration audits.

The QSA report should include an evaluation of the quality program as a whole, and should provide this evaluation within the standard audit report framework.

The evaluation (following the general format of Section 8) shall take the form of approval, contingent approval, or disapproval for each major program element(s) identified in the audit plan.

Acknowledgments

The following organizations supported the development of this standard:

A. Frank Ackerman Associates
AT&T Bell Laboratories
AT&T Information Systems
AT&T Technologies
Bell Canada
Bell Communications Research
Burroughs Corporation
C.S. Draper Laboratory
CAP GEMINI/DASD
Coldframe, Inc
Combustion Engineering
Computer Sciences Corporation
DP Auditing Report
Grumman-CTEC
IBM Corp
Intermetrics, Inc
Martin-Marietta
McDonnell Douglas Corporation
Motorola
Naval Weapons Center
NCR Corporation
Northern Telecom, Ltd
Northern Telecom, Inc
OASAS, Ltd
Pacific Gas and Electric
Pansophic Systems, Inc
Pinpoint Retail Systems, Inc
Quality Assurance Institute
RCA
RGE Software Methodologies, Inc
Sperry
US Department of Transportation
University of California, Sacramento
Worldwide Service Technologies, Inc