IEEE
Std 828-1983

IEEE Standard for Software Configuration Management Plans

Sponsor

Software Engineering Technical Committee
of the
IEEE Computer Society

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

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

(This Foreword is not a part of IEEE Std 828-1983, IEEE Standard for Software Configuration Management Plans.)

Software Configuration Management (SCM) is a formal engineering discipline which provides software developers and users with the methods and tools to identify the software developed, establish baselines, control changes to these baselines, record and track status, and audit the product. SCM is the means through which the integrity and continuity of the software product are recorded, communicated, and controlled. The application of SCM, together with Hardware Configuration Management and overall Systems Configuration Management, benefits all phases of the life cycle of a system containing software components and is a matter of good engineering practice. Without SCM, the reliability and the quality of the software cannot be assured.

This standard assists in the preparation of SCM plans for all environments; for example, large-scale ADP systems, embedded computer systems, control processors, etc, and provides a standard against which such plans can be prepared and assessed. It is directed toward the development and maintenance of all software, including critical software, that is, software whose failure could impact safety or cause large financial or social losses.

There are three groups served by this standard: the users, the developers, and the public.

(1) It aids the users in obtaining and supporting a software product consistent with the users' evolving needs. It assists the users in obtaining a reasonable degree of confidence that the product is in the process of acquiring required attributes as the software is being developed and maintained.

(2) It aids the developers by providing an established standard against which a cost-effective solution to SCM needs can be planned and measured throughout the applicable portions of the software life cycle.

(3) It aids the public by enhancing the likelihood that a software product will perform as expected. The public has legal rights which preclude haphazard control of software development and maintenance. At some later date, the users and developers may be required to show that they acted in a reasonable and prudent manner to ensure that adequate controls were applied.

This standard is consistent with ANSI/IEEE Std 730-1981, IEEE Standard for Software Quality Assurance Plans and IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology. This standard may be applied in conjunction with those standards, or independently.

This standard was prepared by a working group of the Software Engineering Subcommittee of the Technical Committee on Software Engineering of the IEEE Computer Society. The working group had the following members:

R. Frederick, Chairman

H. R. Berlack
E. Bersoff
F. J. Buckley
R. Clingan
D. Dorazio
R. Evans
A. Ets
J. J. Forman
J. Fonj
N. P. Ginex
J. Granton
N. Hill
J. H. Johnston
C. Kolb
T. M. Kurihara
W. A. Mandeville
J. Miguel
J. Neilson
D. Paster
R. Phillips
J. Postak
R. M. Poston
P. B. Powell
S. Siegel
L. Starbuck
B. Taute
A. Terentiev
N. Thomas
R. L. Van Tilburg
J. Williamson

At the time the Software Engineering Standards Subcommittee approved the standard, it had the following members:

F. J. Buckley, Chairman

R. J. Abbott
A. F. Ackerman
R. L. Aurbach
J. P. Bateman
N. G. Biasi
E. G. Booch
M. A. Branstead
W. A. Bryan
D. C. Burt
H. C. Carney
M. Carney
C. L. Carpenter
J. V. Cellini
J. W. Center
E. R. Comer
J. D. Cooper
A. J. Cote, Jr
R. Cotter
J. W. Currier
P. W. Daggett
G. Darling
P. A. Denny
A. Dniestrowski
J. A. Dobbins
W. DuBlancia
R. E. Dwyer
M. L. Eads
J. W. Earls

R. A. Felker
J. W. Fendrich
J. Flourney
J. J. Forman
R. Fredrick
M. Galinier
A. Ganz
F. K. Gardner
D. Gelperin
M. J. Gooding
S. Gloss-Soler
R. W. Graves
B. W. Greene
J. J. Greene
J. W. Grigsby
R. M. Gross
R. T. Gustin
V. E. Haas
T. L. Hannan
L. R. Hestelton, III
C. P. Hollocker
S. Horvitz
H. B. Hoyle
S. Hwang
J. H. Ingram
J. M. Ives
R. M. Jacobs
G. Konomos

T. M. Kurihara
J. A. Kyle
D. V. LaRosa
R. C. Lane
G. N. Larson
A. C. Ma
A. J. Maher
A. K. Mahindru
W. A. Mandeville
P. C. Marriott
C. F. Martiny
G. T. Morun
W. G. Murch
S. Najafi
J. Nebb
G. R. Neidhart
D. J. Ostrom
D. E. Peercy
M. T. Perkins
D. J. Pfeiffer
R. M. Poston
M. A. Potter
P. B. Powell
S. J. Raddue
S. R. Rakitin
J. Rault
H. Reiche
D. J. Reifer

L. B. Robertson
R. E. Sandborgh
M. J. Schindler
N. F. Schneidewind
R. W. Scholten
L. W. Seagren
C. L. Shermer
D. J. Simkins
J. E. Skeen, Jr
J. M. Smith
H. N. Sneed
T. Q. Stevenson
D. B. Tabor
K. C. Tai
B. Taute
J. R. Thompson
P. U. Thompson
G. D. Tice, Jr
T. L. Tillmanns
H. J. Trochesset
C. L. Troyanowski
W. S. Turner, III
E. A. Ulbrich, Jr
R. L. Van Tilburg
A. H. Weigel
P. J. D. Weyman
W. M. Wong
A. W. Yonda
P. F. Zoll

Special representatives to the Software Engineering Standards Subcommittee were:

H.R. Berlack	*Electronic Industries Association*
J. Milandin	*ANS Z1*
W. Perry	*Data Processing Management Association*
R. Pritchett	*EDP Auditors Association*
T. L. Regulinski	*IEEE Reliability Society*

At the time the IEEE Standards Board approved this standard on June 23, 1983 it had the following members:

James H. Beall, *Chairman*

Edward Chelotti, *Vice Chairman*

Sava I. Sherr, *Secretary*

J. J. Archambault
John T. Boettger
J. V. Bonucchi
Rene Castenschiold
Edward J. Cohen
Len S. Corey
Donald C. Fleckenstein
Jay Forster

Donald H. Heirman
Irvin N. Howell
Joseph L. Koepfinger*
Irving Kolodny
George Konomos
John E. May
Donald T. Michael*

John P. Riganati
Frank L. Rose
Robert W. Seelbach
Jay A. Stewart
Clifford O. Swanson
Robert E. Weiler
W. B. Wilkens
Charles J. Wylie

*Member emeritus

Contents

SECTION PAGE

1. Scope and References 7
   1.1 Scope 7
   1.2 References 7

2. Definitions and Acronyms 7
   2.1 Definitions 7
   2.2 Acronyms 7

3. Software Configuration Management Plans 7
   3.1 Introduction (Section 1 of the Plan) 8
      3.1.1 Purpose (1.1 of the Plan) 8
      3.1.2 Scope (1.2 of the Plan) 8
      3.1.3 Definitions and Acronyms (1.3 of the Plan) 8
      3.1.4 References (1.4 of the Plan) 8
   3.2 Management (Section 2 of the Plan) 8
      3.2.1 Organization (2.1 of the Plan) 8
      3.2.2 SCM Responsibilities (2.2 of the Plan) 8
      3.2.3 Interface Control (2.3 of the Plan) 9
      3.2.4 SCMP Implementation (2.4 of the Plan) 9
      3.2.5 Applicable Policies, Directives, and Procedures (2.5 of the Plan) 9
   3.3 SCM Activities (Section 3 of the Plan) 9
      3.3.1 Configuration Identification (3.1 of the Plan) 9
      3.3.2 Configuration Control (3.2 of the Plan) 9
      3.3.3 Configuration Status Accounting (3.3 of the Plan) 10
      3.3.4 Audits and Reviews (3.4 of the Plan) 11
   3.4 Tools, Techniques, and Methodologies (Section 4 of the Plan) 11
   3.5 Supplier Control (Section 5 of the Plan) 11
   3.6 Records Collection and Retention (Section 6 of the Plan) 11

IEEE Standard for Software Configuration Management Plans

1. Scope and References

1.1 Scope.

This standard provides minimum requirements for preparation and content of Software Configuration Management (SCM) Plans. SCM Plans document the methods to be used for identifying software product items, controlling and implementing changes, and recording and reporting change implementation status.

This standard applies to the entire life cycle of critical software; for example, where failure could impact safety or cause large financial or social losses. For noncritical software, or for software already developed, a subset of the requirements may be applied.

This standard identifies those essential items that shall appear in all Software Configuration Management Plans. In addition to these items, the users of this standard are encouraged to incorporate additional items into the plan, as appropriate, to satisfy unique configuration management needs, or to modify the contents of specific sections to fully describe the scope and magnitude of the software configuration management effort. Where this standard is invoked for a project engaged in producing several software items, the applicability of the standard shall be specified for each of the software product items encompassed by the project.

Examples are incorporated into the text of this standard to enhance clarity and to promote understanding. Examples are either explicitly identified as such, or can be recognized by the use of the verb *may*. Examples shall not be construed as mandatory implementations.

1.2 References.

The standards listed here should be considered when applying this standard. The latest revisions shall apply.

(1) ANSI/IEEE Std 730-1981, IEEE Standard for Software Quality Assurance Plans[1]

(2) IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology[2]

(3) IEEE Std 829-1983, IEEE Standard for Software Test Documentation

2. Definitions and Acronyms

2.1 Definitions. The definition listed here establishes meaning in the context of this standard. Other definitions can be found in IEEE Std 729-1983 [2].[3] See specifically: **baseline, configuration item, configuration management, configuration control, configuration control board, configuration audit, configuration identification, configuration status accounting,** and **software library.**

interface control. The process of: (1) Identifying all functional and physical characteristics relevant to the interfacing of two or more configuration items provided by one or more organizations. (2) Ensuring that proposed changes to these characteristics are evaluated and approved prior to implementation.

2.2 Acronyms. The following acronyms are referred to within the text of this standard:

CCB	Configuration Control Board
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan

3. Software Configuration Management Plans

The organization or person responsible for Software Configuration Management shall prepare a Software Configuration Management Plan (hereafter referred to as the Plan) that includes the sections and subsections listed below.

[1] ANSI standards are available from the Sales Department, American National Standards Institute, 1430 Broadway, New York, NY 10018

[2] IEEE standards are available from IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854.

[3] The numbers in brackets correspond to the references in 1.2.

The sections and subsections shall be ordered in the described sequence (if there is no information pertinent to a section or subsection, the following information shall appear below the section or subsection heading: **There is no pertinent information for this section** together with the appropriate reasons for the exclusion).

(1) Introduction
  (a) Purpose
  (b) Scope
  (c) Definitions and acronyms
  (d) References
(2) Management
  (a) Organization
  (b) SCM responsibilities
  (c) Interface control
  (d) SCMP implementation
  (e) Applicable policies, directives and procedures
(3) SCM activities
  (a) Configuration identification
  (b) Configuration control
  (c) Configuration status accounting
  (d) Audits and reviews
(4) Tools, techniques, and methodologies
(5) Supplier control
(6) Records collection and retention

Additional sections may be added at the end, as required. Some of the material may appear in other documents. If so, reference to those documents shall be made in the body of the plan.

The cover page of the plan shall identify the plan and the project to which the plan pertains. As a minimum, the plan shall be signed by the chief operating-officer of each unit having responsibilities defined within the SCMP.

Detailed requirements for each portion of the Plan are described in 3.1 through 3.6 of this standard.

3.1 Introduction (Section 1 of the Plan). This section shall provide an overview of the plan.

3.1.1 Purpose (1.1 of the Plan). This subsection shall delineate the specific purpose of the particular Software Configuration Management Plan.

3.1.2 Scope (1.2 of the Plan). This subsection shall identify the software items to be produced and used, the organizations, the activities, and the phases of the software life cycle to which the plan applies.

3.1.3 Definitions and Acronyms (1.3 of the Plan). This subsection shall define or provide a reference to the definitions of all terms and acronyms required to properly interpret the SCMP.

3.1.4 References (1.4 of the Plan). This subsection shall:

(1) Provide a complete list of all documents referenced elsewhere in the SCMP.

(2) Identify each document by title, report number, if applicable, date, and publishing organization.

(3) Specify the sources from which the referenced documents can be obtained.

3.2 Management (Section 2 of the Plan). This section shall describe the organization, and associated responsibilities.

3.2.1 Organization (2.1 of the Plan). This subsection shall describe the organizational structure that influences the configuration management of the software during the development and the operation and maintenance phases. This shall:

(1) Describe each major element of the organization together with the delegated responsibilities. Organizational dependence or independence of the elements responsible for SCM from those responsible for software development and use shall be clearly described or depicted.

(2) Include an organizational chart or list for the project which illustrates the structure for program/project/system management.

(3) Describe the organization responsible for SCM during the operation and maintenance phase.

(4) Describe the interface between the developing organization and the using organization, if any, with particular emphasis on the transfer of SCM functions in the operation and maintenance phase.

(5) Specifically cover the organizational relationships with the Configuration Control Board in the development and the operation, and maintenance phases.

3.2.2 SCM Responsibilities (2.2 of the Plan). This subsection shall describe:

(1) The organizational responsibilities for each SCM task; for example, identification, control, status accounting, and reviews and audits.

(2) The relationships with software quality-assurance, software development, and other functional organizations required to ensure delivery of the approved final product configuration.

(3) The responsibilities of the users and developer/maintenance activity in the review, audit, and approval process during each phase of the life cycle indicated in 1.2 of the Plan.

(4) Any SCM responsibilities of the representatives from each organization participating in the product development.

(5) The overall responsibility of the Configuration Control Board.

(6) Any unusual responsibilities such as special approval requirements necessary to meet SCM requirements.

3.2.3 Interface Control (2.3 of the Plan). This subsection shall describe the methods to be utilized to:

(1) Identify interface specifications and control documents.

(2) Process changes to released interface specifications and documents.

(3) Provide follow-up on action items scheduled to be accomplished on items pertaining to SCM.

(4) Maintain status of interface specifications and control documents.

(5) Control the interface between software and the hardware on which it is running.

3.2.4 SCMP Implementation (2.4 of the Plan). This subsection shall establish the major milestones for implementation of the SCMP.

Example milestones include the establishment of:

(1) The configuration control board

(2) Each configuration baseline

(3) Schedules and procedures for SCM reviews and audits

(4) Configuration management of related software development, test, and support tools

3.2.5 Applicable Policies, Directives, and Procedures (2.5 of the Plan). This subsection shall:

(1) Identify all applicable SCM policies, directives, and procedures which are to be implemented as part of this plan. The degree of implementation of each shall be stated

(2) Describe any SCM policies, directives, and procedures that are to be written and implemented for this project

Examples of material which may be covered by policies, directives, and procedures are:

(a) Identification of levels of software in a hierarchical tree

(b) Program and module naming conventions

(c) Version level designations

(d) Software product identification methods

(e) Identification of specifications, test plans and procedures, programming manuals, and other documents

(f) Media identification and file management identification

(g) Document release process

(h) Turnover or release of software products to a library function

(i) Processing of problem reports, change requests, and change orders

(j) Structure and operation of configuration control boards

(k) Release, and acceptance of software products

(l) Operation of software library systems to include methods of preparing, storing, and updating modules

(m)Auditing of SCM activities

(n) Problem report, change request or change order documentation requirements describing purpose and impact of a configuration change, or both

(o) Level of testing required prior to entry of software into configuration management

(p) Level of quality assurance; for example, verification against development standards, required prior to entry of software into configuration management

3.3 SCM Activities (Section 3 of the Plan). This section shall describe how the following requirements for SCM shall be satisfied:

(1) Configuration identification

(2) Configuration control

(3) Configuration status accounting and reporting

(4) Configuration audits and reviews

3.3.1 Configuration Identification (3.1 of the Plan). This subsection shall:

3.3.1.1 Identify the software project baselines (that is, the initial approved configuration identifications) and correlate them to the specific life-cycle phases defined in 1.2 of the Plan. For each baseline, the following shall be described:

3.3.1.1.1 The items which form each baseline (for example, software requirements specification, deliverable software, etc).

3.3.1.1.2 The review and approval events, and the acceptance criteria associated with establishing each baseline.

3.3.1.1.3 The users' and developers' participation in establishing baselines.

The following are example baselines which may be established as indicated:

(1) *Functional Baseline.* The agreement between the developers and the users that defines all the system level functions and the system test criteria.

(2) *Allocated Baseline.* The agreement between the developers and the users that identifies all the software requirements to include design constraints and user-required standards.

(3) *Product Baseline.* The agreement between the developer and the user that defines the exact version of the software product which is to be accepted. Elements of this baseline definition might include the following:

(a) Product name and nomenclature

(b) Product identification number

(c) For each new version release, the version release number, a description of the new changes, the change release vehicle, the changes to any support software, and the changes to the associated documentation

(d) Installation instructions

(e) Known faults and failures

(f) Software media and media identification

3.3.1.2 Delineate the project titling, labeling, numbering, and cataloging procedures for all software code and documentation.

As an example, for code:

3.3.1.2.1 The compilation date may be indicated as a part of the identification for each delivered module.

3.3.1.2.2 The sequence numbering of all source lines of code in a module may be structured so that future changes to any module can be properly noted.

3.3.2 Configuration Control (3.2 of the Plan). This subsection shall:

3.3.2.1 Describe the level of authority for change approval to be used in each of the life cycle phases identified in 1.2 of the Plan.

3.3.2.2 Define the methods to be used to process change proposals to established configurations. As a part of this, this section shall:

(1) Identify the routing of change proposals during each of the software life cycle phases identified in 1.2 of the Plan. This may be provided in chart form with narrative support.

(2) Describe the methods of implementing approved change proposals (to include changes in source and object code, and documentation).

(3) Describe the procedures for software library control including those procedures which provide for:

(a) Access control

(b) Read and write protection for applicable baselines

(c) Member protection

(d) Member identification

(e) Archive maintenance

(f) Change history

(g) Disaster recovery

(4) If patches must be used to change object code, describe the methods for identification and control

3.3.2.3 For each CCB and other change management bodies:

(1) Define the role of each; for example, change review authority

(2) Specify their authority and responsibility

(3) Identify the chairperson and the membership in the organizations, if the organizations have been formed

(4) State how the chairperson and the members (and alternates) are to be appointed, if the organizations have not yet been formed

(5) State the relationships of the developers and the users to the CCB(s)

3.3.2.4 State the methods to be used for configuration control of interfaces with programs/projects beyond the scope of this SCMP. If the software changes are required to be reviewed by other boards or teams prior to or in addition to the CCB(s), this subsection shall describe these boards (or teams, or both) and their relationship to the CCB(s) and to each other.

3.3.2.5 State the control procedures for associated special software products, such as nonreleased software, off-the-shelf software, user-furnished software, and in-house support software.

3.3.3 Configuration Status Accounting (3.3 of the Plan). This subsection shall:

(1) Delineate how information on the status of configuration items is to be collected, verified, stored, processed, and reported

(2) Identify the periodic reports to be provided, and their distribution

(3) State what dynamic inquiry capabilities, if any, are to be provided

(4) Describe the means to be used to implement any special status accounting requirements specified by the user

Some examples of information normally desired is as follows:

(a) Status of specifications

(b) Status of proposed changes

(c) Reports of approved changes

(d) Status of product versions or revisions

(e) Reports of the implementation of installed updates or releases

(f) Status of user-furnished property; for example, user-furnished operating systems

3.3.4 Audits and Reviews (3.4 of the Plan). This subsection shall:

(1) Define the SCM role in audits and reviews to be performed at specified points in the software life cycle defined in 1.2 of the SCMP

(2) Identify the configuration items to be covered at each of these audits and reviews

(3) State the procedures to be used for the identification and resolution of problems occurring during these audits and reviews

3.4 Tools, Techniques, and Methodologies (Section 4 of the Plan). This section shall identify, state the purposes, and describe (within the developers' scope of proprietary rights) the use of the specific software tools, techniques, and methodologies to be employed to s' ɔport SCM on the specific project. This shall include the tools, techniques, and methodologies used to:

(1) Identify software media and media documentation

(2) Bring documentation and media under SCM control and formally release it to a user

As examples, it may:

(a) Provide a description of the tools, methodologies, and techniques to be used for source and object control within the software libraries, to include a description of the Database Management System, if used

(b) State how the software library tools, methodologies, and techniques are to be used to process software products for release

(3) Document the status of changes made to software and associated documentation. It shall further define the tools, methodologies, and techniques to be used to prepare reports for various levels of management, such as the project manager, CCB, SCM, and the user.

3.5 Supplier Control (Section 5 of the Plan). This section shall state the provisions for assuring that vendor-provided and subcontractor-developed software meet established SCM requirements. As a part of this, this section shall:

(1) Indicate the proposed methods for control of subcontractors and vendors insofar as it impacts on the execution of this SCMP

(2) Explain the methods to be used:

(a) To determine the SCM capability of subcontractors and vendors

(b) To monitor their adherence to the requirements of this SCMP

As a minimum, the supplier shall be required to prepare and implement a SCM plan in accordance with this standard.

3.6 Records Collection and Retention (Section 6 of the Plan). This section shall:

(1) Identify the SCM documentation to be retained

(2) State the methods and facilities to be used to assemble, safeguard, and maintain this documentation. As part of this, identify any off-site backup facilities to be used

(3) Designate the retention period