IEEE Standard for Software Configuration Management Plans

Sponsor

**Software Engineering Standards Subcommittee of the
Technical Committee on Software Engineering of the
IEEE Computer Society**

Approved September 28, 1990

IEEE Standards Board

Approved February 15, 1991

American National Standards Institute

Abstract: IEEE Std 828-1990, *IEEE Standard for Software Configuration Management Plans,*
establishes the minimum required contents of a Software Configuration Management Plan and
defines the specific activities to be addressed and their requirements for any portion of a software
product's life cycle.
Keywords: configuration control board, configuration items, software configuration
management, software configuration management activities.

Foreword

This standard is concerned with the activity of planning for software configuration management (SCM). SCM activities, whether planned or not, are performed on all software development projects; planning makes these activities more effective. Good planning results in a document that captures the planning information, makes the information the property of the project, communicates to all who are affected, and provides a basis for ongoing planning.

SCM is a formal engineering discipline that, as part of overall system configuration management, provides the methods and tools to identify and control the software throughout its development and use. SCM activities include the identification and establishment of baselines; the review, approval, and control of changes; the tracking and reporting of such changes; the audits and reviews of the evolving software product; and the control of interface documentation and project supplier SCM.

SCM is the means through which the integrity and traceability of the software system are recorded, communicated, and controlled during both development and maintenance. SCM also supports reduction of overall software life cycle cost by providing a foundation for product and project measurement.

SCM constitutes good engineering practice for all software projects, whether phased development, rapid prototyping, or ongoing maintenance. It enhances the reliability and quality of software by

- Providing a structure for identifying and controlling documentation, code, interfaces, and databases to support all life cycle phases
- Supporting a chosen development/maintenance methodology that fits the requirements, standards, policies, organization, and management philosophy
- Producing management and product information concerning the status of baselines, change control, tests, releases, audits, etc.

IEEE Std 828-1983 was originally prepared by a Working Group of the Software Engineering Standards Subcommittee of the Technical Committee on Software Engineering of the IEEE Computer Society and approved in June, 1983, by the IEEE Standards Board. This current standard is the first revision and has been completely rewritten to

- Update the standard to recognize current software engineering practices
- Be consistent with IEEE Std 1042-1987, IEEE Guide to Software Configuration Management, which is anticipated to be reviewed and revised as Std 828.1 to maintain this consistency
- Be more flexible and easier to use for all levels of expertise

The following individuals contributed to IEEE Std 828-1990 by attendance at two or more working sessions or substantial written commentary or both:

H. R. Berlack, *Co-Chair*   **M. Updike-Rumley,** *Co-Chair*

Editorial Committee

R. Frederick	D. L. Knirk	L. Roy

Working Group

B. Banerjee	A. Hartman	L. Siwiec
F. J. Buckley	R. Horner	M. Swain
B. Conger	W. M. Osborne	L. Tran
M. A. Daniels	B. F. Rospide	R. L. Van Tilburg
N. P. Ginex	D. P. Schwartz	A. M. Vaughan
	E. Showalter	

At the time that the IEEE Standards Board approved this revision, the Software Engineering Standards Subcommittee, which was the balloting committee that approved this document for submission to the IEEE Standards Board, included the following members:

John W. Horch, *Chairman*

K. A. Alley
R. L. Aurbach
M. Azuma
B. Banerjee
L. Beltracchi
M. Ben-Menachem
H. R. Berlack
W. J. Boll, Jr.
G. Bracon
K. L. Briggs
W. L. Bryan
F. J. Buckley
H. Carl
E. Chikofsky
F. A. Chimenti
S. V. Chiyyarata
T. S. Chow
P. H. Christensen
W. L. Chung
R. T. Close
F. Coallier
P. W. Daggett
M. A. Daniels
T. Daughtrey
P. I. Davis
P. A. Denny
B. Derganc
P. R. DeWeese
H. Dhama
K. Diggins
J. H. Dobbins
D. C. Doty
E. Dunaye
R. E. Dwyer
M. L. Eads
R. S. Euler
R. L. Evans
R. Fairley
J. Fendrich
A. M. Foley
J. J. Forman
J. Forster
M. Galinier
A. K. Geraci
Y. Gershkovitch
E. L. Gibbs
J. A. Gilmore

N. P. Ginex
S. Gloss-Soler
M. Goldberg
J. Gonzalez-Sanz
J. K. Grau
D. A. Gustafson
U. Hartmann
C. M. Hay
C. P. Hollocker
R. S. Hurst
L. Ihlenfeldt
F. Jay
D. Johnson III
M. S. Karasik
R. H. Karpinski
A. N. Kasad
R. A. Kessler
P. P. Klopfenstein
D. M. Knepper, Sr.
D. L. Knirk
S. Koenig
R. Kosinski
J. Krupinski
T. Kurihara
R. A. Lane
G. N. Larsen
F. C. Lim
B. Livson
I. S. Ludin
A. J. Maher
K. C. Majumdar
H. A. Malec
P. C. Marriott
R. J. Martin
T. Matsubara
I. Mazza
J. P. McArdle
M. C. McElvany
P. E. McKenney
G. A. Meldrum
J. Mersky
C. H. Modell
R. E. Monahan
R. C. Natale
G. Neidhart
D. E. Nickle
W. M. Osborne

D. E. Peercy
W. E. Perry
D. J. Pfeiffer
R. M. Poston
I. C. Pyle
J. Rault
M. Razy
S. T. Redwine, Jr.
B. F. Rospide
F. A. Ruhlman
D. A. Rutherford
S. R. Schach
W. A. Schnoege
R. G. Schueppert
D. J. Schultz
G. Schumacher
D. P. Schwartz
L. W. Seagren
T. Sgarlatti
G. P. Shabe
R. W. Shillato
E. Showalter
D. M. Siefert
M. W. Smith
R. Staunton
R. N. Sulgrove
W. G. Sutcliffe
C. Swain
B. J. Taute
P. U. Thompson
S. Trauth
D. L. Ulery
M. Updike-Rumley
R. L. Van Tilburg
A. M. Vaughan
H. E. Verne
D. R. Wallace
J. W. Walz
A. H. Weigel
A. T. Williams
A. F. Wilson
T. J. Wojcik
P. A. T. Wolfgang
D. L. Wood
A. W. Yonda
N. C. Yopconka
P. F. Zoll

When the IEEE Standards Board approved this standard on September 28, 1990, it had the following membership:

Marco W. Migliaro, *Chairman* **James M. Daly,** *Vice Chairman*

Andrew G. Salem, *Secretary*

Dennis Bodson
Paul L. Borrill
Fletcher J. Buckley
Allen L. Clapp
Stephen R. Dillon
Donald C. Fleckenstein
Jay Forster*
Thomas L. Hannan

Kenneth D. Hendrix
John W. Horch
Joseph L. Koepfinger*
Irving Kolodny
Michael A. Lawler
Donald J. Loughry
John E. May, Jr.

Lawrence V. McCall
L. Bruce McClung
Donald T. Michael*
Stig Nilsson
Roy T. Oishi
Gary S. Robinson
Terrance R. Whittemore
Donald W. Zipse

*Member Emeritus

Contents

IEEE Standard for Software Configuration Management Plans

1. Introduction to the Standard

1.1 Scope. IEEE Std 828-1990 establishes the minimum required contents of a Software Configuration Management (SCM) Plan (the Plan). It is supplemented by IEEE Std 1042-1987 [4][1], which provides approaches to good software configuration management planning. 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. It also applies to noncritical software and to software already developed. The application of this standard is not restricted to any form, class, or type of software.

The Plan documents what SCM activities are to be done, how they are to be done, who is responsible for doing specific activities, when they are to happen, and what resources are required. It can address SCM activities over any portion of a software product's life cycle.

The content of the Plan is identified in Section 2 of this standard. The required information is indicated by the words "shall" and "required." Additional optional information is also identified as appropriate. The user of this standard, however, is expected to expand and supplement the minimum requirements as necessary for the development environment, specific industry, organization, and project. Tailoring of a plan in conformance with this standard is described in Section 3.

The primary users of this standard are assumed to be those planning SCM activities or performing SCM audits.

In considering adoption of this standard, regulatory bodies should be aware that specific application of this standard may already be covered by one or more IEEE Standards documents relating to quality assurance, definitions, or other matters (see [2]). It is not the purpose of IEEE Std 828-1990 to supersede, revise, or amend existing standards directed to specific industries or applications.

1.2 References. This standard may be used in conjunction with the following IEEE Software Engineering Standards:

[1] IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.[2]

[2] IEEE Std 730.1-1989, IEEE Standard for Software Quality Assurance Plans (ANSI).

[3] IEEE Std 1042-1987, IEEE Guide to Software Configuration Management (ANSI).

---

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

[2]IEEE publications may be obtained from the IEEE Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331.

Table 1
SCM Classes of Information

Class of Information	Description	Section in Standard	Section in Plan
Introduction	Describes the Plan's purpose, scope of application, key terms, and references	2.1	1
SCM Management	(Who?) Identifies the responsibilities and authorities for accomplishing the planned activities	2.2	2
SCM Activities	(What?) Identifies all activities to be performed in applying to the project	2.3	3
SCM Schedules	(When?) Identifies the required coordination of SCM activities with the other activities in the project	2.4	4
SCM Resources	(How?) Identifies tools and physical and human resources required for execution of the Plan	2.5	5
SCM Plan Maintenance	Identifies how the Plan will be kept current while in effect	2.6	6

1.3 Definitions and Acronyms. The definitions below describe specific terms as used within the context of this standard.

control point (project control point). A project agreed on point in time or times when specified agreements or controls are applied to the software configuration items being developed, e.g., an approved baseline or release of a specified document/code.

release. The formal notification and distribution of an approved version.

Additional terms that are relevant are defined in IEEE Std 610.12-1990 [1], and are as follows: baseline, component, configuration, configuration audit, configuration control, configuration control board, configuration identification, configuration item, configuration management, configuration status accounting, interface, interface control, software, software library, software life cycle, unit, version.

The following acronyms appear within the text of this standard:

CCB Configuration Control Board
CI   Configuration Item
SCM   Software Configuration Management

The term "the Plan" is used throughout this standard to refer to the Software Configuration Management Plan.

2. The Software Configuration Management Plan

SCM planning information shall be partitioned into the six classes described in Table 1. The referenced sections of the standard provide the reader with detailed requirements for each class of information.

SCM planning information may be presented in any format, sequence, or location that is meaningful to the intended users of the Plan with the following restrictions:

(1) A document with the title "Software Configuration Management Plan" shall exist either in stand-alone form or embedded in another project document.

(2) This document shall contain all SCM planning information either by inclusion or by reference to other locations, such as other documents or automated systems.

(3) A format for this document shall be defined.

The writer of the Plan shall use the sequence of sections specified in Table 1 unless a different format has been defined in the Introduction of the Plan (see 2.1).

2.1 Introduction. Introduction information provides a simplified overview of the SCM activities so that those approving, those performing, and those interacting with SCM can obtain a clear understanding of the Plan. The

7

Introduction shall include four topics: the purpose of the Plan, the scope, the definition of key terms, and references.

The purpose shall briefly address why the Plan exists and who the intended audience is.

The scope shall address SCM applicability, limitations, and assumptions on which the Plan is based. The following items shall be included:

(1) Overview description of the software development project

(2) Identification of the software CI(s) to which SCM will be applied

(3) Identification of other software to be included as part of the Plan (e.g., support or test software)

(4) Relationship of SCM to the hardware or system configuration management activities for the project

(5) The degree of formality, depth of control, and portion of the software life cycle for applying SCM on this project

(6) Limitations, such as time constraints, that apply to the Plan

(7) Assumptions that might have an impact on the cost, schedule, or ability to perform defined SCM activities (e.g., assumptions of the degree of customer participation in SCM activities or the availability of automated aids)

Key terms shall be defined as they apply to the Plan in order to establish a common terminology among all users of the Plan.

All references in the Plan to policies, directives, procedures, standards, terminology, and related documents shall be uniquely identified to enable retrieval by users of the Plan.

2.2 SCM Management. SCM management information describes the allocation of responsibilities and authorities for SCM activities to organizations and individuals within the project structure.

SCM management information shall include three topics: the project organization(s) within which SCM is to apply; the SCM responsibilities of these organizations; and references to the SCM policies and directives that apply to this project.

2.2.1 Organization. The organizational context, both technical and managerial, within which the planned SCM activities are to

be implemented shall be described. The Plan shall identify the following:

(1) All organizational units that participate in or are responsible for any SCM activity on the project

(2) The functional roles of these organizational units within the project structure

(3) Relationships between organizational units

Organizational units may consist of a vendor and customer, a prime contractor and subcontractors, or different groups within one organization. Organization charts, supplemented by statements of function and relationships, can be an effective way of presenting this information.

2.2.2 SCM Responsibilities. The allocation of SCM activities to organizational units shall be specified. For each activity listed within SCM activities (see 2.3), the name of the organizational unit or job title to perform this activity shall be provided. A matrix that relates the organizations defined above to the SCM functions, activities, and tasks can be useful for documenting the SCM responsibilities.

For any review board or special organization established for performing SCM activities on this project, the Plan shall describe its

(1) Purpose and objectives

(2) Membership and affiliations

(3) Period of effectivity

(4) Scope of authority

(5) Operational procedures

2.2.3 Applicable Policies, Directives, and Procedures. Any external constraints placed on the Plan by other policies, directives, and procedures shall be identified. For each, its impact and effect on the Plan shall be stated.

2.3 SCM Activities. SCM activities information identifies all functions and tasks required to manage the configuration of the software system as specified in the scope of the Plan. Both technical and managerial SCM activities shall be identified. General project activities that have SCM implications shall be described from the SCM perspective.

SCM activities are traditionally grouped into four functions: configuration identification, configuration control, status accounting, and configuration audits and reviews. The information requirements for each function are identified in 2.3.1 through 2.3.4.
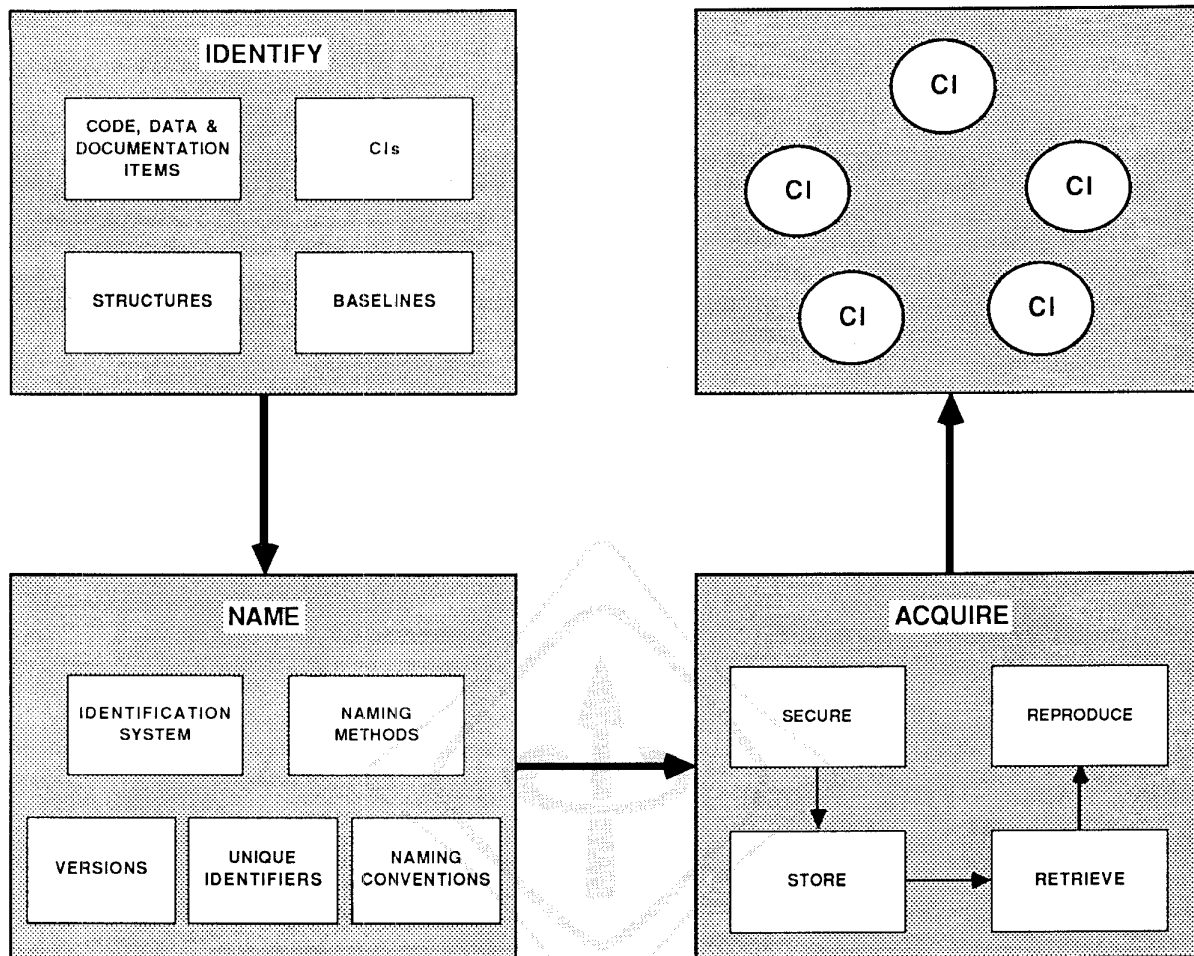
Fig 1
Configuration Identification Processes

Due to their high risk nature, the requirements for interface control and subcontractor/vendor control activities are identified separately in 2.3.5 and 2.3.6.

2.3.1 Configuration Identification. Configuration identification activities shall identify, name, and describe the documented physical and functional characteristics of the code, specifications, design, and data elements to be controlled for the project. The documents are acquired for configuration control. Controlled items may be intermediate and final outputs (such as executable code, source code, user documentation, program listings, data bases, test cases, test plans, specifications, and management plans) and elements of the support environment (such as

compilers, operating systems, programming tools, and test beds).

The Plan shall identify the project configuration items (CI) and their structures at each project control point. The Plan states how each CI and its versions are to be uniquely named and describes the activities performed to define, track, store, and retrieve CIs. The following sections specify information required for configuration identification (see Fig 1).

2.3.1.1 Identifying Configuration Items. The Plan shall record the items to be controlled, the project CIs, and their definitions as they evolve or are selected. The Plan shall also describe how the list of items and the structures are to be maintained for the project.

As a minimum, all CIs that are to be delivered shall be listed.

Appropriate baselines shall be defined at control points within the project life cycle in terms of the following:

(1) The event that creates the baseline
(2) The items that are to be controlled in the baseline
(3) The procedures used to establish and change the baseline
(4) The authority required to approve changes to the approved baselined documents

A means of identifying changes and associating them with the affected CIs and the related baseline shall be specified.

2.3.1.2 Naming Configuration Items. The Plan shall specify an identification system for assigning unique identifiers to each item to be controlled. It shall also specify how different versions of each are to be uniquely identified. Identification methods could include naming conventions and version numbers and letters.

The Plan shall describe the methods for naming controlled items for purposes of storage, retrieval, tracking, reproduction, and distribution. Activities may include version marking, labeling of documentation and executable software, serialization and altered item marking for executable code or data embedded on a microchip, and identification of physical packaging.

Subcontracted software, vendor proprietary software, and support software may require special identification schemes and labeling.

2.3.1.3 Acquiring Configuration Items. The Plan shall identify the controlled software libraries for the project and describe how the code, documentation, and data of the identified baselines are to be physically placed under control in the appropriate library. For each library the format, location, documentation requirements, receiving and inspection requirements, and access control procedures shall be specified.

The Plan shall specify procedures for the actual storage of documents and magnetic media, including the physical marking and labeling of items. Data retention periods and disaster prevention and recovery procedures may also be described.

Procedures shall describe how to retrieve and reproduce controlled items from library storage. These activities include verification of marking and labeling, tracking of controlled copies, and protection of proprietary and security information.

2.3.2 Configuration Control. Configuration control activities request, evaluate, approve or disapprove, and implement changes to baselined CIs. Changes encompass both error correction and enhancement. The degree of formality necessary for the change process depends on the project baseline affected and on the impact of the change within the configuration structure.

For each project software library identified according to 2.3.1.3, the Plan shall describe the change controls imposed on the baselined CIs. The Plan shall define the following sequence of specific steps:

(1) Identification and documentation of the need for a change
(2) Analysis and evaluation of a change request
(3) Approval or disapproval of a request
(4) Verification, implementation, and release of a change

The Plan shall identify the records to be used for tracking and documenting this sequence of steps for each change. Any differences in handling changes based on the origin of the request shall be explicitly documented.

2.3.2.1 Requesting Changes. The Plan shall specify the procedures for requesting a change to a baselined CI and the information to be documented for the request. As a minimum, the information recorded for a proposed change shall contain the following:

(1) The name(s) and version(s) of the CIs where the problem appears
(2) Originator's name and organization
(3) Date of request
(4) Indication of urgency
(5) The need for the change
(6) Description of the requested change

Additional information, such as priority or classification, may be included to clarify the significance of the request and to assist in its analysis and evaluation. Other information, such as change request number, status, and disposition, may be recorded for change tracking.

2.3.2.2 Evaluating Changes. The Plan shall specify the analysis required to determine the impact of the proposed change and the procedures for reviewing the results of the

analysis. Changes should be evaluated according to their effect on the deliverable and their impact on project resources.

2.3.2.3 Approving or Disapproving Changes. The Plan shall identify each configuration control board (CCB) and its level of authority for approving proposed changes. A CCB may be an individual or a group. Multiple levels of CCBs may be specified, depending upon the degree of system or project complexity and upon the project baseline involved. When multiple CCBs are used, the Plan shall specify how the proper level is determined for a change request, including any variations during the project life cycle.

For any CCB utilized, the Plan shall indicate its level of authority and its responsibilities as defined in 2.2.2.

2.3.2.4 Implementing Changes. The Plan shall specify the activities for verifying and implementing an approved change. The information recorded for the completion of a change shall contain the following as a minimum:

(1) The associated change request(s)
(2) The names and versions of the affected items
(3) Verification date and responsible party
(4) Release or installation date and responsible party
(5) The identifier of the new version

Additional information, such as software fault metrics or identification of the supporting software used to implement the change, may be included.

The Plan shall also specify activities for release planning and control, i.e., coordinating multiple changes, reconfiguring the CIs, and delivering a new baseline.

2.3.3 Configuration Status Accounting. Configuration status accounting activities record and report the status of project CIs.

The Plan shall include information on the following:

(1) What data elements are to be tracked and reported for baselines and changes
(2) What types of status accounting reports are to be generated and their frequency
(3) How information is to be collected, stored, processed, and reported
(4) How access to the status data is to be controlled

If an automated system is used for any status accounting activity, its function shall be described or referenced.

The following minimum data elements shall be tracked and reported for each CI: its initial approved version, the status of requested changes, and the implementation status of approved changes. The level of detail and specific data required may vary according to the information needs of the project and the customer.

2.3.4 Configuration Audits and Reviews. Configuration audits determine to what extent the actual CI reflects the required physical and functional characteristics. Configuration reviews are management tools for establishing a baseline.

The Plan shall identify the configuration audits and reviews to be held for the project. At a minimum, a configuration audit shall be performed on a CI prior to its release.

For each planned configuration audit or review, the Plan shall define the following:

(1) Its objective
(2) The CIs under audit or review
(3) The schedule of audit or review tasks
(4) The procedures for conducting the audit or review
(5) The participants by job title
(6) Documentation required to be available for review or to support the audit or review
(7) The procedure for recording any deficiencies and reporting corrective actions
(8) The approval criteria and the specific action(s) to occur upon approval

2.3.5 Interface Control. Interface control activities coordinate changes to the project CIs with changes to interfacing items outside the scope of the Plan. Hardware, system software and support software, as well as other projects and deliverables, should be examined for potential interfacing effects on the project.

The Plan shall identify the external items to which the project software interfaces. For each interface the Plan shall define the following:

(1) The nature of the interface
(2) The affected organizations
(3) How the interface code, documentation, and data are to be controlled
(4) How the interface control documents are approved and released into a specified baseline

For any CCB established to control interfaces, the Plan shall identify its responsibilities and procedures as specified in 2.2.2.

2.3.6 Subcontractor/Vendor Control.

Subcontractor/vendor control activities incorporate items developed outside the project environment into the project CIs. Included are software developed by contract and software acquired in its finished form. Special attention should be directed to these SCM activities due to the added organizational and legal relationships.

For both subcontracted and acquired software, the Plan shall define the activities to incorporate the externally developed items into the project CIs and to coordinate changes to these items with their development organizations.

For subcontracted software, the Plan shall describe the following:

(1) What SCM requirements, including an SCM Plan, are to be part of the subcontractor's agreement

(2) How the subcontractor will be monitored for compliance

(3) What configuration audits and reviews of subcontractor items will be held

(4) How external code, documentation, and data will be tested, verified, accepted, and merged with the project software

(5) How proprietary items will be handled for security of information and traceability of ownership (e.g., copyright and royalties)

(6) How changes are to be processed, including the subcontractor's participation

For acquired software, the Plan shall describe how the software will be received, tested, and placed under SCM; how changes to the supplier's software are to be processed; and whether and how the supplier will participate in the project's change management process. Acquired software can come from a vendor, a subcontractor, a customer, another project, or other source.

2.4 SCM Schedules.

SCM schedule information establishes the sequence and coordination for the identified SCM activities and for all events affecting the Plan's implementation.

The Plan shall state the sequence and dependencies among all SCM activities and the relationship of key SCM activities to project milestones or events. The schedule shall cover the duration of the Plan and contain all major milestones of the project related to SCM activities. SCM milestones shall include establishment of a configuration baseline, implementation of change control procedures, and the start and completion dates for a configuration audit.

Schedule information shall be expressed as absolute dates, as dates relative to either SCM or project milestones, or as a simple sequence of events. Graphic representation can be particularly appropriate for conveying this information.

2.5 SCM Resources.

SCM resource information identifies the software tools, techniques, equipment, personnel, and training necessary for the implementation of the specified SCM activities.

SCM can be performed by a combination of software tools and manual procedures. Tools can be SCM-specific or embedded in general project aids; they can be standard organizational resources or ones specially acquired or built for this project. Tools can be applied to library structure and access control; documentation development and tracking; code control; baseline system generation; change processing, communication and authorization; change/problem tracking and status reporting; archiving, retention, and retrieval of controlled items; or the SCM planning process itself.

For each type of SCM activity identified, the Plan shall specify what tools, techniques, equipment, personnel, and training are required and how each resource will be provided or obtained.

For each software tool, whether developed within the project or brought in from outside the project, the Plan shall describe or reference its functions and shall identify the configuration controls to be placed on the tool.

2.6 SCM Plan Maintenance.

SCM plan maintenance information identifies the activities and responsibilities necessary to ensure continued SCM planning during the life cycle of the project. The Plan shall state the following:

(1) Who is responsible for monitoring the Plan
(2) How frequently updates are to be performed
(3) How changes to the Plan are to be evaluated and approved
(4) How changes to the Plan are to be made and communicated

The Plan should be reviewed at the start of each project software phase, changed accordingly, and approved and distributed to the project team.

If the Plan has been constructed with detailed procedures documented elsewhere in appendixes or references, different maintenance mechanisms for those procedures may be appropriate.

3. Tailoring of the Plan

This standard permits significant flexibility in preparing an SCM Plan. A successful Plan reflects its project environment. It should be written in terms familiar to its users and should be consistent with the development and procurement processes of the project.

To conform to the requirements set forth in other applicable standards or to accommodate local practices, a Plan may be tailored upward, to add information, or tailored to use a specified format. The Plan may also be tailored downward, omitting information required by this standard, when specific standard requirements are identified as not applicable to this project.

3.1 Upward Tailoring. Some information requirements applicable to a particular project may not be stated in this standard due to its scope of establishing the minimum required contents of an SCM Plan. If additional requirements are applicable to the project, the Plan shall so state these additions as part of the Introduction and indicate the reason for their insertion. A cost-benefits analysis should be completed for each additional requirement. Requirements that are additional should be agreed on by all affected project functions and the parties responsible for approval of the plan.

3.2 Downward Tailoring. Some information requirements stated in this standard may not apply to a particular project due to the project's limited scope, low complexity, or unusual environment. If a requirement is not applicable to the project, the Plan shall so state this deletion as part of the Introduction and indicate the reason for removal. Requirements that are inapplicable should be agreed upon by all affected project functions and all parties responsible for approval of the Plan.

The Plan shall omit none of the six major classes of information. Detailed information may be omitted as indicated above but within the limits of the consistency criteria stated in Section 4.

If certain information has not been decided on or is unavailable at the time the Plan is initially approved, the Plan shall mark those areas or sections as "to be determined" and shall indicate, as part of Plan maintenance, information on how and when further information will be provided.

3.3 Format. The information may be presented in the Plan in any sequence or presentation style deemed suitable for the Plan's users. To achieve consistency and convenience within a single organization or industry segment, a standard format for SCM plans is desirable and appropriate. To customize this standard for a particular group of users, a supplement to the standard specifying Plan structure and standard terminology may be used.

4. Conformance to the Standard

An SCM Plan shall satisfy the following criteria in order to conform with this standard.

4.1 Minimum Information. The Plan shall include the six classes of SCM information identified in Section 2: Introduction, Management, Activities, Schedules, Resources, and Plan Maintenance. Within each class, all of the required information stated in Section 2 of this standard, as indicated by the words "shall" and "required," shall be documented within the Plan. If certain required information is not applicable, the reasons shall be so stated. If a sequence of information other than the sequence of this standard is used, an explicit cross reference

between the Plan and the standard shall be provided.

4.2 Presentation Format. One document, section title, or such reference shall exist that is specifically labeled "Software Configuration Management Plan." Within this document, each of the six classes of information shall be included. While the information may be provided in a number of presentation styles, the requirement is to provide all Plan information and references in a single document.

4.3 Consistency Criteria. The documented information shall satisfy the following consistency criteria:

(1) All activities defined in the Plan (see 2.3.1 to 2.3.6) shall be assigned to an organizational unit (see 2.2.2).

(2) All activities defined shall have resources identified to accomplish the activities (see 2.5).

(3) All CIs identified in the Plan (see 2.3.1) shall have defined processes for baseline establishment and change control (see 2.3.2).

4.4 Conformance Declaration. If the preceding criteria are met, then the conformance of any SCM planning documentation with this standard may be stated accordingly: "This SCM Plan conforms with the requirements of IEEE Std 828-1990."

Appendix
Cross Reference to IEEE Std 1042-1987

(This Appendix is not part of IEEE Std 828-1990, IEEE Standard for Software Configuration Management Plans, but is included for information only.)

Section in IEEE Std 828-1990	Section in IEEE Std 1042-1987
1. Introduction to the Standard	1. Introduction 2. SCM Disciplines in Software Management
2. The SCM Plan	3. Software Configuration Management Plans
2.1 Introduction	3.1 Introduction
2.2 SCM Management	3.2 Management
2.3 SCM Activities	3.3 SCM Activities
2.3.1 Configuration Identification	3.3.1 Configuration Identification
2.3.2 Configuration Control	3.3.2 Configuration Control
2.3.3 Configuration Status Accounting	3.3.3 Configuration Status Accounting
2.3.4 Configuration Audits and Reviews	3.3.4 Audits and Reviews
2.3.5 Interface Control	3.2.3 Interface Control
2.3.6 Subcontractor/Vendor Control	3.5 Supplier Control
2.4 Schedules	3.2.4 SCM Plan Implementation
2.5 Resources	3.4 Tools, Techniques and Methodologies
2.6 SCM Plan Maintenance	2.5 The Planning of SCM
3. Tailoring of the Plan	2.5 The Planning of SCM
4. Conformance to the Standard	2.5 The Planning of SCM

Acknowledgments

The following organizations supported working group members in the development of this standard. This support does not constitute or imply approval or endorsement of this standard.

Babcock & Wilcox

Battelle Northwest Laboratories

BTG Inc.

Compass Corporation

CTA Inc.

DSC Communications

Eaton/AIL

General Electric–MSD

Jet Propulsion Laboratory

Lockheed Missiles & Space Company

Lockheed Sanders Inc.

MACTEC

McDonnell Douglas

Mitre Corporation

Motorola/Computer X

National Institute of Standards and Technology

Northrop Corporation

Programming Environments Inc.

Rockwell Telecommunications

Texas Instruments

Unisys Corporation