

IEEE Standard for Developing Software Life Cycle Processes

Sponsor

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

Approved 9 December 1997

IEEE Standards Board

Abstract: A process for creating a software life cycle process is provided. Although this standard is directed primarily at the process architect, it is useful to any organization that is responsible for managing and performing software projects.

Keywords: software life cycle, software life cycle model, software life cycle process

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Copyright © 1998 by the Institute of Electrical and Electronics Engineers, Inc.
Copyright © 1998 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 2 December 1998. Printed in the United States of America.

Print: ISBN 1-55937-993-6 SH94600
PDF: ISBN 0-7381-0532-5 SS94600

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

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) 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 that 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 societies and Standards Coordinating 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-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (508) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

(This introduction is not part of IEEE Std 1074-1997, IEEE Standard for Developing Software Life Cycle Processes.)

This introduction is intended to provide the reader with some background into the rationale used to develop this standard. This information is being provided to aid in the understanding and usage of this standard. This introduction is nonbinding.

Background

This is a standard for the generation of the process that governs software development and maintenance for a project. This standard requires the definition of a user's software life cycle and shows mapping into typical software life cycles. It is not intended to define or imply a software life cycle of its own.

This standard applies to the management and support activities that continue throughout the entire life cycle, as well as all aspects of the software life cycle from concept exploration through retirement.

The utilization of these Activities maximizes the benefits to the user when the use of this standard is initiated early in the software life cycle. Software that has proceeded past the initialization phase when this standard is invoked should gradually move into compliance with this standard.

This standard was written for any organization that is responsible for managing and conducting software projects. It will be useful to project managers, software developers, quality assurance organizations, purchasers, users, and maintainers. It can be used where software is the total system or where software is embedded into a larger system.

This standard allows for continuing harmonization with IEEE/EIA 12207.0-1996 and EIA/IEEE J-STD-016-1995 and their successors.

Terminology

The word *shall* and the imperative verb form identify the mandatory material within this standard. The words *should* and *may* identify optional material. As with other IEEE Software Engineering Standards, the terminology in this document is based on IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology. To avoid inconsistency when the Glossary is revised, the definitions are not repeated in this document. New terms and modified definitions are included.

History

Since the original publication of this standard, considerable worldwide attention has been paid to software life cycle processes. Use of IEEE Std 1074-1991, IEEE Std 1074-1995, and other quality system and life cycle standards activity has been carefully considered in preparing this substantive revision of this standard. (The 1995 version of this standard was a minor revision to correct specific errors found in the 1991 version.)

The following changes are among those that are included in this current version:

- Activities are rearranged into more logical groupings (called Activity Groups), such as placing all planning Activities into the new Project Planning Activities Activity Group, collecting all Project Initiation Activities, and collecting and expanding all Review Activities.
- The term “Process,” as was used in earlier versions of this Standard, was replaced with the term “Activity Group” to identify collections of Activities. Some users of this standard were misinterpret-

ing the collections as actual “processes” and were trying to execute them as such. The term “Activity Groups” should eliminate this misconception.

- The importance of risk management led to the addition of a new Activity, Manage Risks.
- The recognition that software can be acquired from other sources, for use in the system being developed, led to the addition of the Software Importation Activity Group.

Participants

This standard was developed by a working group consisting of the following members who attended two or more meetings, provided text, or submitted comments on more than two drafts of this standard:

David J. Schultz, *Chair*

Susan M. Burgess, *Configuration Manager*

Dennis E. Nickle, *Vice Chair*

John W. Horsch, *Editor*

David W. Burnett
Ron Dean
Jean A. Gilmore
Arthur Godin

Daniel Gray
Lynn Ihlenfeldt
Robert J. Kierzyk

Pat Marcinko
Keith Middleton
Robert W. Shillato
Diane Switzer

The following individuals also contributed to the development of this standard by attending one meeting or providing comments on one or two drafts:

Alan Braaten
W. Larry Campbell
Bostjan K. Derganc
Dorothy Deutch
Leo Egan
Michael Frehse
John Garth Glynn

Sam Godfrey
Rob Harker
John Jenkins
Denis Meredith
Noritoshi Murakami
Christopher Neubert
John Pellegrin

David Pepper
James Shimp
David Smith
John Swearingen
Allan Willey
Natalie Yopconka
Janusz Zalewski

The following persons were on the balloting committee:

Jeremy A. Adams	Mark Henley	Peter T. Poon
Syed Ali	John W. Horch	Margaretha W. Price
Mikhail Auguston	Jerry Huller	Lawrence S. Przybylski
Leo Beltracchi	Peter L. Hung	Kenneth R. Ptack
H. Ronald Berlack	Fabrizio Imelio	Ann Reedy
Richard E. Biehl	George Jackelen	Annette D. Reilly
William J. Boll	John O. Jenkins	Dennis Rilling
Alan L. Bridges	Frank V. Jorgensen	Patricia Rodriguez
M. Scott Buck	Vladan V. Jovanovic	Andrew P. Sage
David W. Burnett	William S. Junk	Helmut Sandmayr
Edward R. Byrne	George X. Kambic	Stephen R. Schach
Leslie Chambers	Diana Kang	Norman Schneidewind
Keith Chan	Myron S. Karasik	David J. Schultz
Theo Clarke	Ron S. Kenett	Gregory D. Schumacher
Sylvain Clermont	Judy Kerner	Robert W. Shillato
Francois Coallier	Robert J. Kierzyk	Carl A. Singer
Virgil Lee Cooper	Dwayne L. Knirk	James M. Sivak
Geoff Cozens	Shaye Koenig	Alfred R. Sorkowitz
Gregory T. Daich	Thomas M. Kurihara	Donald W. Sova
Bostjan K. Derganc	John B. Lane	Luca Spotorno
Perry R. DeWeese	J. Dennis Lawrence	Julia Stesney
Sherman Eagles	Michael Lines	Fred J. Strauss
Leo Egan	David Maibor	Christine Brown Stryzik
Richard L. Evans	Robert Martin	Robert N. Sulgrove
William Eventoff	Tomoo Matsubara	Toru Takeshita
Jonathan H. Fairclough	Sue McGrath	Patricia A. Trellue
John W. Fendrich	Bret Michael	Leonard L. Tripp
Jon J. Fineman	Alan Miller	T. H. Tse
Jay Forster	James W. Moore	Margaret C. Updike
Simon Gabrihelidis	R. Muralidharan	Theodore J. Urbanowicz
Hiranmay Ghosh	Pavol Navrat	Glenn D. Venables
Marilyn Ginsberg-Finner	Dennis E. Nickle	Udo Voges
John Garth Glynn	Myrna L. Olson	Ronald L. Wade
Lawrence M. Gunther	Mike Ottewill	Dolores Wallace
David A. Gustafson	Gerald L. Ourada	John W. Walz
John Harauz	Indradeb P. Pal	Scott A. Whitmire
Rob Harker	Mark Paulk	Paul A. T. Wolfgang
Carol J. Harkness	Warren L. Persons	Natalie C. Yopconka
William Hefley	John G. Phippen	Weider D. Yu
Manfred Hein	Alex Polack	Janusz Zalewski
Mark Heinrich		Geraldine Zimmerman

When the IEEE Standards Board approved this standard on 9 December 1997, it had the following membership:

Donald C. Loughry, *Chair*

Richard J. Holleman, *Vice Chair*

Andrew G. Salem, *Secretary*

Clyde R. Camp	Lowell Johnson	Louis-François Pau
Stephen L. Diamond	Robert Kennelly	Gerald H. Peterson
Harold E. Epstein	E. G. "Al" Kiener	John W. Pope
Donald C. Fleckenstein	Joseph L. Koepfinger*	Jose R. Ramos
Jay Forster*	Stephen R. Lambert	Ronald H. Reimer
Thomas F. Garrity	Lawrence V. McCall	Ingo Rüsch
Donald N. Heirman	L. Bruce McClung	John S. Ryan
Jim Isaak	Marco W. Migliaro	Chee Kiow Tan
Ben C. Johnson		Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
Alan H. Cookson

Contents

1. Overview.....	1
1.1 Scope.....	1
1.2 Purpose.....	1
1.3 Product of standard	2
1.4 Intended audiences.....	2
1.5 Relationship to other key standards	2
1.6 Relationship to process improvement	3
1.7 Organization of this standard	3
2. References.....	4
3. Definitions and acronyms	4
3.1 Definitions.....	4
3.2 Acronyms.....	5
4. Key concepts.....	6
4.1 Activities.....	6
4.2 Elements of the SLCP	6
4.3 Mapping	8
4.4 Input Information and Output Information	9
5. Implementation of the standard	10
5.1 Select an SLCM	10
5.2 Create an SLC	10
5.3 Establish an SLCP.....	11
Annex A (normative) Activities	12
A.1 Project Management Activity Groups	12
A.1.1 Project Initiation Activities	12
A.1.1.1 Create SLCP	12
A.1.1.2 Perform Estimations	13
A.1.1.3 Allocate Project Resources	14
A.1.1.4 Define Metrics	15
A.1.2 Project Planning Activities.....	15
A.1.2.1 Plan Evaluations	16
A.1.2.2 Plan Configuration Management.....	17
A.1.2.3 Plan System Transition (If Applicable)	18
A.1.2.4 Plan Installation	19
A.1.2.5 Plan Documentation.....	20
A.1.2.6 Plan Training	21
A.1.2.7 Plan Project Management	22
A.1.2.8 Plan Integration.....	23
A.1.3 Project Monitoring and Control Activities.....	24
A.1.3.1 Manage Risks.....	25
A.1.3.2 Manage the Project	27

A.1.3.3	Identify SLCP Improvement Needs.....	28
A.1.3.4	Retain Records.....	29
A.1.3.5	Collect and Analyze Metric Data	30
A.2	Pre-Development Activity Groups	31
A.2.1	Concept Exploration Activities.....	31
A.2.1.1	Identify Ideas or Needs.....	31
A.2.1.2	Formulate Potential Approaches.....	32
A.2.1.3	Conduct Feasibility Studies	33
A.2.1.4	Refine and Finalize the Idea or Need	34
A.2.2	System Allocation Activities	34
A.2.2.1	Analyze Functions	35
A.2.2.2	Develop System Architecture	36
A.2.2.3	Decompose System Requirements	36
A.2.3	Software Importation Activities.....	37
A.2.3.1	Identify Imported Software Requirements	37
A.2.3.2	Evaluate Software Import Sources (If Applicable)	38
A.2.3.3	Define Software Import Method (If Applicable).....	39
A.2.3.4	Import Software (If Applicable).....	39
A.3	Development Activity Groups	40
A.3.1	Requirements Activities.....	40
A.3.1.1	Define and Develop Software Requirements	40
A.3.1.2	Define Interface Requirements.....	41
A.3.1.3	Prioritize and Integrate Software Requirements.....	42
A.3.2	Design Activities.....	43
A.3.2.1	Perform Architectural Design.....	44
A.3.2.2	Design Data Base (If applicable).....	44
A.3.2.3	Design Interfaces	45
A.3.2.4	Perform Detailed Design	46
A.3.3	Implementation Activities.....	47
A.3.3.1	Create Executable Code.....	47
A.3.3.2	Create Operating Documentation	48
A.3.3.3	Perform Integration.....	49
A.4	Post-Development Activity Groups.....	49
A.4.1	Installation Activities	49
A.4.1.1	Distribute Software.....	50
A.4.1.2	Install Software.....	51
A.4.1.3	Accept Software in Operational Environment.....	51
A.4.2	Operation and Support Activities.....	52
A.4.2.1	Operate the System	52
A.4.2.2	Provide Technical Assistance and Consulting.....	53
A.4.2.3	Maintain Support Request Log.....	53
A.4.3	Maintenance Activities	54
A.4.3.1	Identify Software Improvement Needs.....	54
A.4.3.2	Implement Problem Reporting Method	55
A.4.3.3	Reapply SLC.....	56

A.4.4 Retirement Activities	57
A.4.4.1 Notify User	57
A.4.4.2 Conduct Parallel Operations (If Applicable)	58
A.4.4.3 Retire System.....	58
A.5 Integral Activity Groups	59
A.5.1 Evaluation Activities.....	59
A.5.1.1 Conduct Reviews	60
A.5.1.2 Create Traceability Matrix.....	62
A.5.1.3 Conduct Audits	62
A.5.1.4 Develop Test Procedures	63
A.5.1.5 Create Test Data	64
A.5.1.6 Execute Tests	65
A.5.1.7 Report Evaluation Results	66
A.5.2 Software Configuration Management Activities	67
A.5.2.1 Develop Configuration Identification.....	67
A.5.2.2 Perform Configuration Control.....	68
A.5.2.3 Perform Status Accounting.....	69
A.5.3 Documentation Development Activities.....	70
A.5.3.1 Implement Documentation	70
A.5.3.2 Produce and Distribute Documentation.....	71
A.5.4 Training Activities	71
A.5.4.1 Develop Training Materials.....	72
A.5.4.2 Validate the Training Program	72
A.5.4.3 Implement the Training Program.....	73
Annex B (informative) Mapping example	74
B.1 Step 1: Select SLCM.....	74
B.2 Step 2: Compare Activities to SLC.....	76
B.3 Step 3: Develop and justify List of Activities Not Used	77
B.4 Step 4: List Activities and Invocations	78
B.5 Step 5: Place Activities in executable sequence	79
B.6 Step 6: Verify information flow.....	80
B.7 Step 7: Map information into deliverable documents.....	81
B.8 Step 8: Add OPAs.....	81
B.9 Step 9: Add project planning information	81

IEEE Standard for Developing Software Life Cycle Processes

1. Overview

This clause presents an overview of this standard.

1.1 Scope

This standard provides a process for creating a software life cycle process (SLCP). It is primarily directed at the process architect for a given software project. It is the function of the process architect to create the SLCP.

This methodology begins with the selection of an appropriate software life cycle model (SLCM) for use on the specific project. It continues through the creation of the software life cycle (SLC), using the selected SLCM and the Activities provided in Annex A. The methodology concludes with the augmentation of the SLC with Organizational Process Assets (OPAs) to create the SLCP.

The Activities that are provided in Annex A cover the entire life cycle of a software project, from concept exploration through the eventual retirement of the software system. This standard does not address non-software activities, such as contracting, purchasing, or hardware development. It also does not mandate the use of a specific SLCM, nor does it provide a selection of, or a tutorial on, SLCMs. This standard presumes that the process architect is already familiar with a variety of SLCMs, with the criteria for choosing among them, and with the criteria for determining the attributes and constraints of the desired end system and the development environment that affects this selection. Finally, this standard does not prescribe how to perform the software Activities in Annex A.

1.2 Purpose

This standard defines the process by which an SLCP is created. It is useful to any organization that is responsible for managing and performing software projects. It can be used where software is the total system or where software is part of a larger system.

1.3 Product of standard

The product of this standard is the SLCP that is required for a specific software project. The SLCP is based on the following:

- a) An SLCM that is selected for the project
- b) The Activities that are provided in Annex A
- c) The OPAs that are selected for the project

While this standard describes the creation of a single, overall SLCP that is to be used for a project, the user of this standard should recognize that an SLCP can itself include lower-level SLCPs. This is the same concept as in configuration management, in which a particular Configuration Item can include subordinate Configuration Items. This standard applies equally to the development of SLCPs at any level.

1.4 Intended audiences

This standard is written to provide direction and guidance to those individuals who are responsible for determining the implementation of this standard's Activities.

1.4.1 Process architect

The primary audience for this standard is the process architect. The process architect is expected to have

- a) The authority to develop SLCPs
- b) A knowledge of the OPAs
- c) A knowledge of SLCMs
- d) An understanding of the Activities that are presented in Annex A of this standard

1.4.2 Other interested parties

This standard also can be of use to the performers of the Activities that are presented in Annex A.

1.5 Relationship to other key standards

No standard exists isolated from its associated standards. This standard is related to ISO 9001 : 1994 [B38]¹ and IEEE/EIA 12207.0-1996 [B35].

1.5.1 Relationship to ISO 9001 : 1994 [B38] and ISO 9000-3 : 1994 [B39]

ISO 9001 : 1994 [B38], as interpreted by the guidance in Clause 5.1 of ISO 9000-3 : 1994 [B39], recommends organizing a software development project in accordance with a selected life cycle model. It is intended that a conforming application of this standard would satisfy this recommendation; however, it would be the responsibility of the applier to ensure that the created SLCPs satisfy the specific requirements of ISO 9001 : 1994 [B38] and other applicable standards.

1.5.2 Relationship to IEEE/EIA 12207.0-1996 [B35]

Clause 5.1.2.2 of IEEE/EIA 12207.0-1996 [B35] requires an acquirer to “determine which processes, activities, and tasks of (IEEE/EIA 12207.0-1996 [B35]) are appropriate for the project and tailor them accordingly.” Clause 5.2.4.2 of IEEE/EIA 12207.0-1996 [B35] requires a supplier to “define or select a software

¹The numbers in brackets preceded by the letter B correspond to those of the bibliography in Annex D.

life cycle model” and map the processes, activities, and tasks of IEEE/EIA 12207.0-1996 [B35] onto that model. Clause 5.3.1.1 places a similar requirement upon a developer in some situations. It is intended that a conforming application of this standard would satisfy any of these requirements; however, it would be the responsibility of the applier to ensure that the created SLCPs satisfy the other specific requirements of IEEE/EIA 12207.0-1996 [B35] and other applicable standards.

1.6 Relationship to process improvement

While process improvement is outside the scope of this standard, this standard can be integrated into an organization’s process improvement program through its use as the framework for the OPAs.

Building the OPAs around this standard’s structure of Activities and Input and Output Information can

- a) Minimize the effort needed to create an SLCP
- b) Facilitate the reuse of existing OPAs
- c) Lead to the improvement of the OPAs by incorporating lessons that were learned from the use of the OPAs in projects

The SLCP for a project, in part or as a whole, can become part of the OPAs for use by future projects.

1.7 Organization of this standard

Clauses 1, 2, and 3 of this standard contain required, introductory information. Clause 4 provides a brief discussion of the key concepts that are beneficial to the understanding and use of this standard. Clause 5 provides the requirements for the creation of an SLCP. Requirements for the content of an SLCP are presented in Annex A, which is normative. Annexes B, C, and D are informative and include useful information, but no requirements. Table 1 presents the organization of this standard.

Table 1—Organization of this standard

Element	Title
Clause 1	Overview
Clause 2	References
Clause 3	Definitions and acronyms
Clause 4	Key concepts
Clause 5	Implementation of the standard
Annex A (normative)	Activities
Annex B (informative)	Mapping example
Annex C (informative)	Information mapping template
Annex D (informative)	Bibliography

The components of the SLCP consist of 65 Activities. These Activities are included in Annex A, and are organized into 17 Activity Groups. The Activities cover the entire life cycle of a software project, from concept exploration through the eventual retirement of the software system. The Activity Groups are further grouped into five sections, as shown in Table 2.

Table 2—Activity grouping

Section Title	Clause	Activity Groups
Project Management	A.1	Project Initiation Project Planning Project Monitoring and Control
Pre-Development	A.2	Concept Exploration System Allocation Software Importation
Development	A.3	Requirements Design Implementation
Post-Development	A.4	Installation Operation and Support Maintenance Retirement
Integral	A.5	Evaluation Software Configuration Management Documentation Development Training

The Integral section (Clause A.5) includes those Activity Groups that are necessary to ensure the successful completion of a project, but are considered as support Activities, rather than those Activities that are directly oriented to the development effort. The Integral Activity Groups contain the following two types of Activities:

- a) Activities that are performed discretely and are therefore mapped into an SLCM
- b) Activities that are invoked (see 4.3.3) by other Activities

2. References

No other publications are required for the use of this standard. A list of standards, which can be consulted for additional guidance, is given in Annex D. Although this standard does not require adherence to any other standard, a knowledge of the principles and concepts that are described in the standards listed in Annex D can be helpful.

3. Definitions and acronyms

This clause defines the terms and identifies the acronyms that are used within the context of this standard.

3.1 Definitions

The definitions listed in this subclause establish meanings within the context of this standard. Definitions of the other terms that are used in this document can be found in IEEE Std 610.12-1990 [B2].

3.1.1 Activity: A defined body of work to be performed, including its required Input and Output Information. *See also:* **Activity Group**.

3.1.2 Activity Group: A set of related Activities. *See also:* **Activity**.

3.1.3 constraint: A restriction on software life cycle process (SLCP) development. *See also:* **software life cycle process (SLCP)**.

3.1.4 external: An Input Information source or Output Information destination that is outside the scope of this standard and, therefore, may or may not exist.

3.1.5 Instance: The mapping of an Activity that processes all of its Input Information and generates all of its Output Information. *Contrast with:* **Invocation; Iteration.** *See also:* **mapping.**

3.1.6 Integral Activity Group: An Activity Group that is needed to complete project Activities, but is outside the management and development Activity Groups.

3.1.7 Invocation: The mapping of a parallel initiation of Activities of an Integral Activity Group that perform a distinct function and return to the initiating Activity. *Contrast with:* **Instance; Iteration.** *See also:* **mapping.**

3.1.8 Iteration: The mapping of any execution of an Activity where at least some Input Information is processed and some Output Information is created. One or more Iterations comprise an Instance. *Contrast with:* **Instance; Invocation.** *See also:* **mapping.**

3.1.9 mapping: Establishing a sequence of the Activities in this standard according to a selected software life cycle model (SLCM). *See also:* **Instance; Invocation; Iteration; software life cycle model (SLCM).**

3.1.10 Organizational Process Asset (OPA): An artifact that defines some portion of an organization's software project environment.

3.1.11 process architect: The person or group that has primary responsibility for creating and maintaining the software life cycle process (SLCP). *See also:* **software life cycle process (SLCP).**

3.1.12 product: Any output of the software development Activities (e.g., document, code, or model). *See also:* **Activity.**

3.1.13 software life cycle (SLC): The project-specific sequence of Activities that is created by mapping the Activities of this standard onto a selected software life cycle model (SLCM). *Contrast with:* **software life cycle model (SLCM); software life cycle process (SLCP).**

3.1.14 software life cycle model (SLCM): The framework, selected by each using organization, on which to map the Activities of this standard to produce the software life cycle (SLC). *Contrast with:* **software life cycle (SLC); software life cycle process (SLCP).**

3.1.15 software life cycle process (SLCP): The project-specific description of the process that is based on a project's software life cycle (SLC) and the Organizational Process Assets (OPA). *Contrast with:* **software life cycle (SLC); software life cycle model (SLCM).** *See also:* **Organizational Process Asset (OPA).**

3.2 Acronyms

The following acronyms appear within the text of this standard:

CI	Configuration Item
OPA	Organizational Process Asset
PR&RPI	Problem Reporting and Resolution Planned Information
SCMPI	Software Configuration Management Planned Information
SLC	software life cycle
SLCM	software life cycle model
SLCP	software life cycle process
SPMPI	Software Project Management Planned Information

4. Key concepts

This clause provides an explanation of the key concepts that are used throughout this standard.

4.1 Activities

An Activity is a defined body of work that is to be performed, including its required Input and Output Information. Thus, it is a description of the required transformation of Input Information into Output Information. The performance of an Activity is complete when all Input Information has been processed and all Output Information has been generated.

4.1.1 Format

An Activity consists of three parts:

- a) *Input Information*—A list of the required information to be transformed and its source(s)
- b) *Description*—A discussion of the value-added actions to be performed in order to accomplish the transformation
- c) *Output Information*—A list of the information that is required to be generated by the transformation, and its destination(s)

4.1.2 Entry and exit criteria

To “enter,” or start, an Activity, at least one element of the specified Input Information must be present. To “exit,” or complete, an Activity, all Input Information shall have been processed and all Output Information shall be generated. Each project is expected to determine information flow requirements during the mapping of Activities to the SLCM.

4.1.3 “If Applicable” Activities

Activities are categorized as either mandatory or “If Applicable.” “If Applicable” Activities are marked “If Applicable” in the Activity title. All other Activities are mandatory. Each “If Applicable” Activity contains an explanation of the cases to which it will apply. For example, A.3.2.2, Design Data Base (If Applicable), applies when a data base is to be created as a part of the project. When an “If Applicable” Activity is used, its Output Information becomes “Available” for use by other Activities.

4.1.4 Organizational structure

This standard does not presume or dictate an organizational structure for a software project. Therefore, it is neither implied nor required that Activities within an Activity Group be performed by the same organizational entity, nor that an organizational entity’s involvement be concentrated in only one Activity Group. This standard does, however, presume that persons will be assigned accountability for the performance of the Activities and for the quality of the Input and Output Information sets.

4.2 Elements of the SLCP

Figure 1 depicts the key concepts involved in the development of an SLCP.

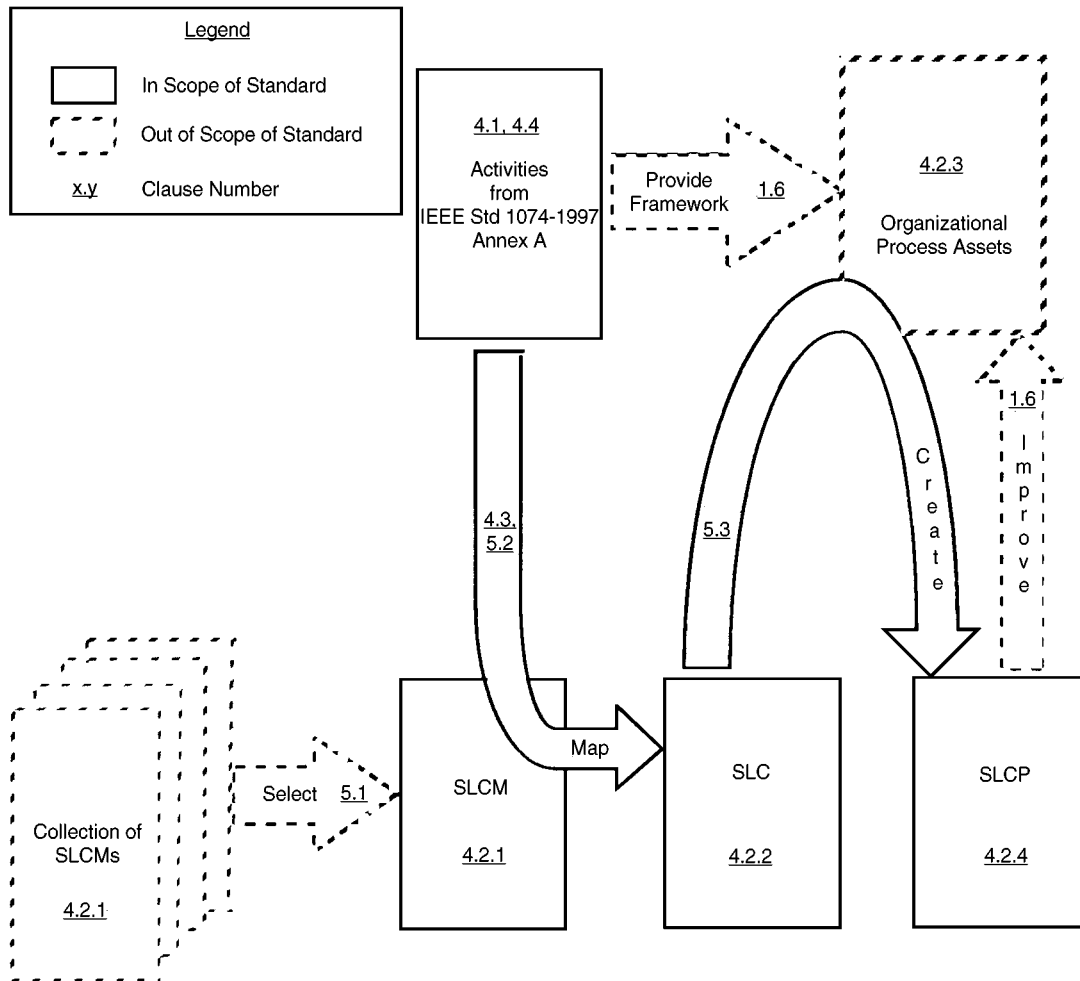


Figure 1—Developing an SLCP

4.2.1 SLCM

The SLCM is the framework on which the Activities of this standard will be mapped to produce the SLC for a project. To use this standard, a SLCM shall be selected for a project. This selection is based on project attributes and organizational capabilities.

This standard does not provide a collection of SLCMs. Providing such a collection of SLCMs is outside the scope of this standard.

4.2.2 SLC

The SLC is the executable sequence of Activities that are to be performed during a project. The SLC is created by mapping the Activities provided in Annex A of this standard onto the SLCM selected for the project.

4.2.3 OPAs

OPAs are the artifacts that define the environment of an organization for software projects. These artifacts are selected and adapted for a particular project.

The content of the Process Assets collection of an organization will vary from organization to organization. Definition of the collection of OPAs is the responsibility of the using organization. It is recommended, however, that the organization consider including assets such as policies, standards, procedures, existing SLCPs, metrics, tools, methodologies, etc.

4.2.4 SLCP

The SLCP is created by augmenting the SLC with the OPAs that are selected for the project. It provides the specific approach to be used for the project.

4.3 Mapping

Mapping establishes the executable sequence of the Activities in this standard onto a selected SLCM. Activities can be mapped in three ways: Instance, Iteration, and Invocation.

4.3.1 Instance

An Activity is mapped as an Instance if it takes all of its specified inputs, processes them, and produces all of its specified outputs. It is mapped once, and appears as a single event in the SLC. Activity A.1.1.3, Allocate Project Resources, could be an example of a single Instance mapping.

4.3.2 Iteration

An Activity is mapped as an Iteration if at least some Input Information is processed and some Output Information is created. Iterations are mapped until all Input Information is processed and all Output Information is created. Activity A.1.3.2, Manage the Project, could require multiple Iterations.

4.3.3 Invocation

In addition to the Activities that are discretely mapped, there are groups of Activities that are invoked in parallel from many Activities. An Activity is invoked to further process specific information before that information is considered complete and permitted to be output by the creating Activity. When invoked, these Activities perform a distinct function and then return to the invoking Activity.

The following example is taken from Activity A.1.2.7, Plan Project Management, with notes added. In this example, the Software Project Management Planned Information (SPMPI) shall be “sent” to the three Activities listed.

“Prior to distribution of the SPMPI², the following Activities shall be invoked³:

- a) A.5.1.1, Conduct Reviews⁴
- b) A.5.2.2, Perform Configuration Control
- c) A.5.3.1, Implement Documentation”

²The specified Output Information on which the invoked Activities are to be performed. That is, not all of the Output Information of this Activity is required to be documented, controlled, and evaluated, just the SPMPI.

³Initiate a parallel task that is necessary to complete the required invoked Activities and return here, before this Activity can be considered complete.

⁴The Activity to which Output Information is sent. In this example, the SPMPI shall be “sent” to the three named Activities. The evaluated, controlled, and documented information is then returned to Activity A.1.2.7, Plan Project Management.

4.4 Input Information and Output Information

The Input and Output tables show the flow of Information among the Activities in Annex A. Where Information flows among Activities, it can be traced from its original Activity to the receiving Activity through the Input Information and Output Information tables.

Figure 2 depicts the conceptual flow of Input Information and Output Information into and out from an Activity, respectively.

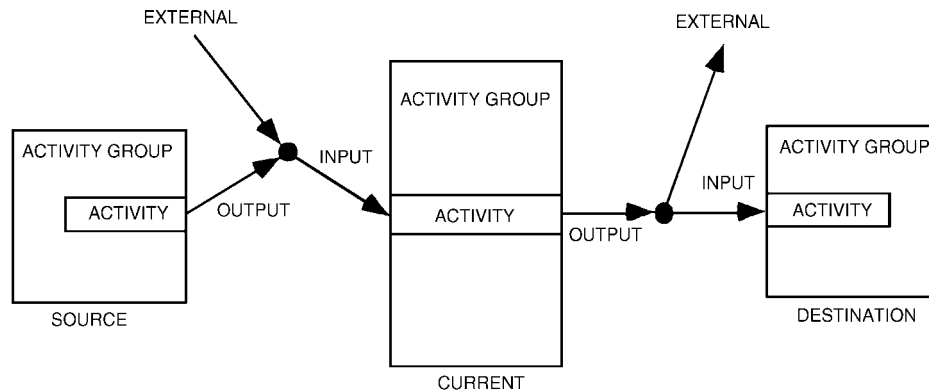


Figure 2—Information flow

4.4.1 Conventions

The Input Information and Output Information for each Activity are listed (in Annex A) in a three-column format. The Input or Output Information name is listed in the left-hand column. The source or destination of the Information (both Activity Group and Activity) is shown in the two right-hand columns.

As a convention of this document, Input and Output Information names are capitalized in the description of an Activity.

4.4.2 External Information

External Information sources and destinations are outside the scope of this standard.

External Input Information sources may or may not exist. If an External Input does not exist, the processing listed for it is not required for completion of the Activity. When an External Input does exist, it shall be used.

External Output Information destinations will receive the information sent, if they exist. No assumption about the use of the Output Information by external destinations is made by this standard.

External sources and destinations are denoted by “External” in the Activity Group column, and a blank in the Activity column of the affected Input or Output table.

4.4.3 Generic Information

In most cases, the Input Information and Output Information columns of the tables designate the specific Information that enters or exits the Activity. However, since many Activities have Output Information whose destination is A.1.3.4, Retain Records, the various Input Information to Retain Records is collected under the term “Records.” The corresponding Activity Group and Activity columns refer simply to the originating Activity Group and originating Activity. A.1.3.5, Metric Data, is received in the same way.

4.4.4 Information vs. documents

This standard prescribes the Activities of the SLC, not the products of that life cycle. Therefore, this standard does not require the creation of specific documents. The information that results from the execution of the Activities is expected to be collected in whatever manner and form are consistent with the selected SLCM and OPAs.

5. Implementation of the standard

This clause presents a description of the way in which implementation of this standard is to be approached. The process architect has primary responsibility for creating and maintaining the SLCP. This responsibility is implemented in three steps, as described below, and is performed as Activity A.1.1.1, Create SLCP, in Annex A. A sample implementation of this standard appears in Annex B.

5.1 Select an SLCM

Initially, the process architect shall identify the SLCM to which the Activities will be mapped. This step encompasses locating, evaluating, selecting, and acquiring an SLCM. It is possible for an organization to have multiple SLCMs; however, only one model is to be selected for a project.

The process architect shall follow the following five steps in order to evaluate and select an SLCM:

- a) Identify all the SLCMs that are available to the development project.
- b) Identify the attributes that apply to the desired end system and the development environment.
- c) Identify any constraints that might be imposed on the selection.
- d) Evaluate the various SLCMs based on past experience and organizational capabilities.
- e) Select the SLCM that will best satisfy the project attributes and constraints.

5.2 Create an SLC

The Activities identified in Annex A shall be mapped onto the SLCM. Note that the selected SLCM, or the project itself, could include or require Activities that are not included in Annex A. Additional Activities are acceptable in the SLC. It should be noted, however, that failure to map one or more of the mandatory Activities in Annex A will result in an SLC and, therefore, an SLCP that are not compliant with this standard. The components of mapping are as follows.

5.2.1 Place the Activities in executable sequence

The order in which Activities will be performed will be determined by three major factors:

- a) The selected SLCM will dictate an initial ordering of Activities. As mapping progresses, the actual order in which Activities will be performed will be established.
- b) Schedule constraints may require the overlapping of Activities in the SLCM and may thus impact the ordering. In this case, Activities may be mapped for parallel execution rather than for serial execution.
- c) The ordering of Activities may be impacted by the entry and exit criteria of associated Activities. The availability of Output Information from one Activity could affect the start of another Activity. The second Activity may require, as inputs, one or more of the outputs of the first Activity. For example, no software design of any kind can be done unless some minimum information is available about software requirements. Another example is that no Evaluation Activities can be performed unless there is some output product upon which to work.

5.2.2 Develop and justify a List of Activities Not Used

All “If Applicable” Activities that do not apply to this project shall be identified and explained in the List of Activities Not Used.

5.2.3 Verify the map

The process architect shall ensure that the Activities are fully mapped onto the selected SLCM and that the resulting SLC contains all of the Activities that are necessary to successfully complete a software project.

The process architect shall also verify that the information flow into and out of the Activities will support the relative order into which they have been mapped.

5.3 Establish an SLCP

The preceding steps develop the SLC. As the next step, the available OPAs shall be applied to the SLC Activities, and the known constraints shall be reconciled. The Output Information that is generated by each Activity shall be assigned to the appropriate document(s). (The Information mapping template in Annex C can be used for assistance in the assignment of information to documents.) The result is the established SLCP.

Annex A

(normative)

Activities

This annex contains the mandatory and “If Applicable” Activities that are to be mapped onto the selected SLCM.

A.1 Project Management Activity Groups

These Activity Groups initiate, monitor, and control a software project throughout its life cycle.

A.1.1 Project Initiation Activities

Project Initiation Activities are those Activities that create and update the infrastructure of a software development or maintenance project. They build the base for the full SLCP.

Project Initiation Activities are

- a) A.1.1.1, Create Software Life Cycle Process
- b) A.1.1.2, Perform Estimations
- c) A.1.1.3, Allocate Project Resources
- d) A.1.1.4, Define Metrics

A.1.1.1 Create SLCP

A.1.1.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
Attributes	External	—
Available SLCMs	External	—
Constraints	External	—
Contractual Requirements	External	—
IEEE Std 1074-1997	External	—
OPAs	External	—
Environmental Improvement Needs	Project Monitoring and Control	Identify SLCP Improvement Needs (A.1.3.3)
Statement of Need	Concept Exploration	Refine and Finalize the Idea or Need (A.2.1.4)
Maintenance Recommendations	Maintenance	Reapply SLC (A.4.3.3)

A.1.1.1.2 Description

Using the Input Information, the process architect shall create the SLCP as described in the three steps of Clause 5 of this standard. Any mandatory Activities not used shall be included in the List of Activities Not Used. Exclusion of any mandatory Activity, however, will preclude compliance with this standard.

Prior to the distribution of the SLCP, the following Activities shall be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control
- c) A.5.3.1, Implement Documentation
- d) A.5.4.1, Develop Training Materials

A.1.1.1.3 Output information

Output Information	Destination	
	Activity Group	Activity
SLCP	Project Initiation	Perform Estimations (A.1.1.2)
		Allocate Project Resources (A.1.1.3)
		Define Metrics (A.1.1.4)
	Project Planning	Plan Documentation (A.1.2.5)
		Plan Training (A.1.2.6)
		Plan Project Management (A.1.2.7)
	Project Monitoring and Control	Manage Risks (A.1.3.1)
		Manage the Project (A.1.3.2)
		Identify SLCP Improvement Needs (A.1.3.3)
List of Activities Not Used	Project Monitoring and Control	Manage Risks (A.1.3.1)

A.1.1.2 Perform Estimations

A.1.1.2.1 Input Information

Input Information	Source	
	Activity Group	Activity
Historical Project Records	External	—
SLCP	Project Initiation	Create SLCP (A.1.1.1)
Statement of Need	Concept Exploration	Refine and Finalize the Idea or Need (A.2.1.4)
System Functional Software Requirements	System Allocation	Decompose System Requirements (A.2.2.3)

A.1.1.2.2 Description

Based on the project requirements that are documented in the Statement of Need and the System Functional Software Requirements, size estimates of work products to be created (both deliverable and nondeliverable) shall be derived. The work products shall be decomposed to the level of granularity that is needed to plan and track the project. Based on these size estimates, effort and cost estimates shall be created for all of the Activities of the SLC. In addition, target computer resource usage shall be estimated.

Historical Project Records shall be used as the basis of estimation, when available. All Estimation Assumptions that were made in deriving the estimates shall be specified. Project Estimates should be reaffirmed and revised throughout the SLCP.

Prior to the distribution of project estimates, the following Activities shall be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control
- c) A.5.3.1, Implement Documentation

A.1.1.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Project Estimates	Project Initiation	Allocate Project Resources (A.1.1.3)
	Project Planning	Plan Project Management (A.1.2.7)
	Project Monitoring and Control	Manage Risks (A.1.3.1)
		Manage the Project (A.1.3.2)
Estimation Assumptions	Project Monitoring and Control	Manage Risks (A.1.3.1)

A.1.1.3 Allocate Project Resources

A.1.1.3.1 Input Information

Input Information	Source	
	Activity Group	Activity
Historical Project Records	External	—
Resources	External	—
SLCP	Project Initiation	Create SLCP (A.1.1.1)
Project Estimates	Project Initiation	Perform Estimations (A.1.1.2)
Statement of Need	Concept Exploration	Refine and Finalize the Idea or Need (A.2.1.4)
System Functional Software Requirements	System Allocation	Decompose System Requirements (A.2.2.3)
Software Requirements	Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)

A.1.1.3.2 Description

Resource Allocations shall be identified at the Activity level of the SLC. Resources that are to be allocated include budget, personnel, equipment, space, and computer resources.

Historical Project Records, if available, and the Statement of Need can provide valuable insight into Resource Allocations.

A.1.1.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Resource Allocations	Project Planning	Plan Project Management (A.1.2.7)
	Project Monitoring and Control	Manage Risks (A.1.3.1)
		Manage the Project (A.1.3.2)

A.1.1.4 Define Metrics**A.1.1.4.1 Input Information**

Input Information	Source	
	Activity Group	Activity
SLCP	Project Initiation	Create SLCP (A.1.1.1)
Evaluation Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Software Requirements	Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)

A.1.1.4.2 Description

The metrics for the project, based on the SLC, SPMPI, and Software Requirements, shall be defined. Metrics shall be applied to the products of the project, and to the processes that affect the project, throughout the SLC. For each Defined Metric, Collection and Analysis Methods shall be specified.

Further information related to this Activity can be found in IEEE Std 982.1-1988 [B8], IEEE Std 982.2-1988 [B9], IEEE Std 1044-1993 [B19], IEEE Std 1045-1992 [B21], and IEEE Std 1061-1998 [B24].

Prior to distributing the Defined Metrics, Activity A.5.1.1, Conduct Reviews, should be invoked.

A.1.1.4.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Defined Metrics	Project Planning	Plan Evaluations (A.1.2.1)
	Project Monitoring and Control	Collect and Analyze Metric Data (A.1.3.5)
Collection and Analysis Methods	Project Planning	Plan Evaluations (A.1.2.1)
	Project Monitoring and Control	Collect and Analyze Metric Data (A.1.3.5)

A.1.2 Project Planning Activities

Project Planning Activities address the planning for all project management, including contingencies. These Activities can be done as needed (mapped in several Iterations), e.g., at every phase review.

Project Planning Activities are

- a) A.1.2.1, Plan Evaluations
- b) A.1.2.2, Plan Configuration Management
- c) A.1.2.3, Plan System Transition (If Applicable)
- d) A.1.2.4, Plan Installation
- e) A.1.2.5, Plan Documentation
- f) A.1.2.6, Plan Training
- g) A.1.2.7, Plan Project Management
- h) A.1.2.8, Plan Integration

A.1.2.1 Plan Evaluations

A.1.2.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
Defined Metrics	Project Initiation	Define Metrics (A.1.1.4)
Collection and Analysis Methods	Project Initiation	Define Metrics (A.1.1.4)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Integration Planned Information	Project Planning	Plan Integration (A.1.2.8)
Risk Management Reported Information	Project Monitoring and Control	Manage Risks (A.1.3.1)
Imported Software Requirements	Software Importation	Identify Imported Software Requirements (A.2.3.1)
Preliminary Software Requirements	Requirements	Define and Develop Software Requirements (A.3.1.1)
Software Requirements	Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Detailed Design	Design	Perform Detailed Design (A.3.2.4)

A.1.2.1.2 Description

This Activity shall identify and describe the evaluation tasks that are necessary to ensure that the software product and development efforts meet their goals, as specified in the SPMPI and their requirements. Evaluation methods that are to be considered in this planning Activity include audits, reviews, and testing. The Activities and Activity Output Information that are to be evaluated shall be identified; and the evaluation method, purpose, and scope of the evaluation for each of those Activities and Activity Output Information shall be defined. The size, complexity, and criticality of the software should dictate the minimum reviews, audits, and testing.

Reviews that are to be planned include peer reviews, management reviews, technical reviews, operational reviews, process improvement reviews, and post-implementation reviews. More information on reviews can be found in Activity A.5.1.1, Conduct Reviews.

Audits shall be planned to provide an independent examination of software products and processes in order to assess their compliance with requirements and standards. More information on audits can be found in Activity A.5.1.3, Conduct Audits.

Test planning shall be used to define the generic levels of testing and the basic test environment and structure that are needed to support the required levels of testing. The types of testing to be planned include unit/module/component, integration, acceptance, regression, and system tests. Each planned test shall identify the items to be tested, the requirements to be tested, and the test pass-or-fail criteria. Test planning shall also identify the test coverage criteria. Test planning shall be coordinated with Activity A.1.2.8, Integration Planning.

The evaluation planning information shall include the evaluation teams' organization and responsibilities, and the tools, techniques, and methodologies that will be used to perform the evaluations. The planning shall include developing schedules, estimating resources, identifying special resources, staffing, and establishing exit or acceptance criteria. Evaluation planning shall also define the management controls and reporting procedures, as well as the risks and contingencies. Special attention should be given to minimizing business and technical risks. This planning shall be documented in the Evaluation Planned Information.

Further information on planning evaluations can be found in IEEE Std 730-1998 [B3], IEEE Std 828-1998 [B5], IEEE Std 829-1998 [B6], IEEE Std 982.1-1988 [B8], IEEE Std 982.2-1988 [B9], IEEE Std 1008-1987 [B12], IEEE Std 1012-1998 [B13], IEEE Std 1028-1997 [B17], IEEE Std 1042-1987 [B18], IEEE Std 1044-1993 [B19], IEEE Std 1045-1992 [B21], and IEEE Std 1059-1993 [B23].

Prior to the distribution of the Evaluation Planned Information, the following Activities shall be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control
- c) A.5.3.1, Implement Documentation

A.1.2.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Evaluation Planned Information	Project Initiation	Define Metrics (A.1.1.4)
	Project Planning	Plan Integration (A.1.2.8)
	Project Monitoring and Control	Manage Risks (A.1.3.1)
		Manage the Project (A.1.3.2)
		Collect and Analyze Metric Data (A.1.3.5)
	Maintenance	Identify Software Improvement Needs (A.4.3.1)
	Evaluation	Conduct Reviews (A.5.1.1)
		Conduct Audits (A.5.1.3)
		Develop Test Procedures (A.5.1.4)
		Create Test Data (A.5.1.5)
		Execute Tests (A.5.1.6)

A.1.2.2 Plan Configuration Management

A.1.2.2.1 Input Information

Input Information	Source	
	Activity group	Activity
Deliverable List	External	—
SPMPI	Project Initiation	Plan Project Management (A.1.2.7)
Imported Software Requirements	Software Importation	Identify Imported Software Requirements (A.2.3.1)

A.1.2.2.2 Description

This Activity shall plan and document specific software configuration management organizations and responsibilities, procedures, tools, techniques, and methodologies in the Software Configuration Management Planned Information (SCMPI). The SCMPI shall also describe how and when such procedures are to be performed.

Overall software configuration management objectives are derived using internal guidelines as well as contractual or other agreed-upon requirements from the SPMPI. The software configuration management approach should be compatible with the approaches that are being used on associated systems.

Items that are to be managed should include code, documentation, plans, specifications, and other work products. The configuration identification defined in Activity A.5.2.1, Develop Configuration Identification, should be included in the planned information once it is developed.

The configuration management planning shall include developing schedules, estimating resources, identifying special resources, and staffing, defining management controls and reporting procedures, and the risks and contingencies.

Further information related to this Activity can be found in IEEE Std 828-1998 [B5] and IEEE Std 1042-1987 [B18].

Prior to the distribution of the SCMPI, the following Activities shall be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control
- c) A.5.3.1, Implement Documentation

A.1.2.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
SCMPI	Project Monitoring and Control	Manage the Project (A.1.3.2)
		Retain Records (A.1.3.4)
	Software Configuration Management	All Software Configuration Management Activities (A.5.2)

A.1.2.3 Plan System Transition (If Applicable)

A.1.2.3.1 Input Information

Input Information	Source	
	Activity Group	Activity
Retirement Planned Information (for the system being replaced)	External	—
Preliminary Statement of Need	Concept Exploration	Identify Ideas or Needs (A.2.1.1)
Recommendations	Concept Exploration	Conduct Feasibility Studies (A.2.1.3)
Imported Software Requirements	Software Importation	Identify Imported Software Requirements (A.2.3.1)

A.1.2.3.2 Description

This Activity is applicable only when an existing system (automated or manual) is being replaced with a new system. The transition shall be planned and documented in accordance with the Retirement Planned Information of the system being replaced, the Preliminary Statement of Need, and the recommended solutions. Transition strategies and tools shall be part of the Transition Planned Information. A Transition Impact Statement shall also be produced.

The transition planning information shall include the transition team's organization and responsibilities, as well as the tools, techniques, and methodologies that are needed to perform the transition.

The planning shall include developing schedules, estimating resources, identifying special resources, and staffing. Transition planning shall also define management controls and reporting procedures, as well as the risks and contingencies. Special attention should be given to minimizing operational risks. This planning shall be documented in the Transition Planned Information.

Prior to the distribution of the Transition Planned Information, the following Activities should be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control
- c) A.5.3.1, Implement Documentation

A.1.2.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Transition Planned Information	Project Planning	Plan Installation (A.1.2.4)
	Project Monitoring and Control	Manage the Project (A.1.3.2)
Transition Impact Statement	Project Monitoring and Control	Manage Risks (A.1.3.1)
	Concept Exploration	Refine and Finalize the Idea or Need (A.2.1.4)

A.1.2.4 Plan Installation

A.1.2.4.1 Input Information

Input Information	Source	
	Activity Group	Activity
Transition Planned Information (If Available)	Project Planning	Plan System Transition (A.1.2.3)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Imported Software Requirements	Software Importation	Identify Imported Software Requirements (A.2.3.1)
Installation Requirements	Requirements	Define and Develop Software Requirements (A.3.1.1)
Operating Documentation	Implementation	Create Operating Documentation (A.3.3.2)

A.1.2.4.2 Description

The tasks to be performed during installation shall be described in the Software Installation Planned Information. The Installation Requirements and the other Input Information are analyzed in order to guide the development of the Software Installation Planned Information. This Planned Information, the associated documentation, and the developed software are used to install the software product.

The Software Installation Planned Information shall include the required hardware and other constraints (e.g., minimum memory requirements, color monitor), detailed instructions for the installer, and any additional steps that are required prior to the operation of the system (e.g., registering the software). The type of software to be installed, and the expected level of expertise of the installer, shall be considered when writing installation instructions.

In some cases, the installation planning shall include defining the order of installation at several sites. It could also define one or more configurable options that are to be handled in the installation process.

Prior to the distribution of the Software Installation Planned Information, the following Activities shall be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control
- c) A.5.3.1, Implement Documentation
- d) A.5.4.1, Develop Training Materials

A.1.2.4.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Software Installation Planned Information	Project Monitoring and Control	Manage the Project (A.1.3.2)
	Installation	Distribute Software (A.4.1.1)

A.1.2.5 Plan Documentation

A.1.2.5.1 Input Information

Input Information	Source	
	Activity Group	Activity
Contractual Requirements	External	—
SLCP	Project Initiation	Create SLCP (A.1.1.1)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Imported Software Requirements	Software Importation	Identify Imported Software Requirements (A.2.3.1)

A.1.2.5.2 Description

In this Activity, information such as the SCMPI, product descriptions, schedules, and resource constraints shall be assimilated to create a consistent and disciplined approach to achieving the required documentation. The approach shall identify the required documents, the document production and delivery schedules, and the documentation standards. Responsible organizations, information sources, and intended audiences shall be defined for each document. The approach shall be documented in the Documentation Planned Information. The Documentation Planned Information shall include resource allocations for this Activity.

Additional guidance for the development of user documentation can be found in IEEE Std 1063-1987 [B26].

Prior to the distribution of the Documentation Planned Information, the following Activities shall be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.3.1, Implement Documentation

Activity A.5.2.2, Perform Configuration Control, should also be invoked.

A.1.2.5.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Documentation Planned Information	Project Monitoring and Control	Manage the Project (A.1.3.2)
		Retain Records (A.1.3.4)
	Implementation	Create Operating Documentation (A.3.3.2)
	Documentation Development	All Document Development Activities (A.5.3)

A.1.2.6 Plan Training**A.1.2.6.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Applicable Information	External	—
Skills Inventory	External	—
SLCP	Project Initiation	Create SLCP (A.1.1.1)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Imported Software Requirements	Software Importation	Identify Imported Software Requirements (A.2.3.1)
Software Requirements	Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Training Feedback	Training	Validate the Training Program (A.5.4.2)
		Implement the Training Program (A.5.4.3)

A.1.2.6.2 Description

This Activity shall identify the needs for different types of training and the categories of people that require training for each need. Customer and project information shall be reviewed, along with existing personnel inventories. Both internal (e.g., project team, sales force) and external (e.g., customers, users, dealers) training needs shall be identified. Responsible organizations, information sources, and the intended audiences shall be defined for each type of training. Training tools, techniques, and methodologies shall be specified.

The planning shall include developing schedules, estimating resources, identifying special resources, staffing, and establishing exit or acceptance criteria. This planning shall be documented in the Training Planned Information.

Additional guidance for training can be found in IEEE Std 1298-1992 [B33].

Prior to the distribution of the Training Planned Information, the following Activities shall be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control
- c) A.5.3.1, Implement Documentation