

IEEE Std 1074-1997
(Revision of IEEE Std 1074-1995;
Replaces IEEE Std 1074.1-1995)

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. Horch, *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 Strysik
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 Gabrielididis	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 Rüsck
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.

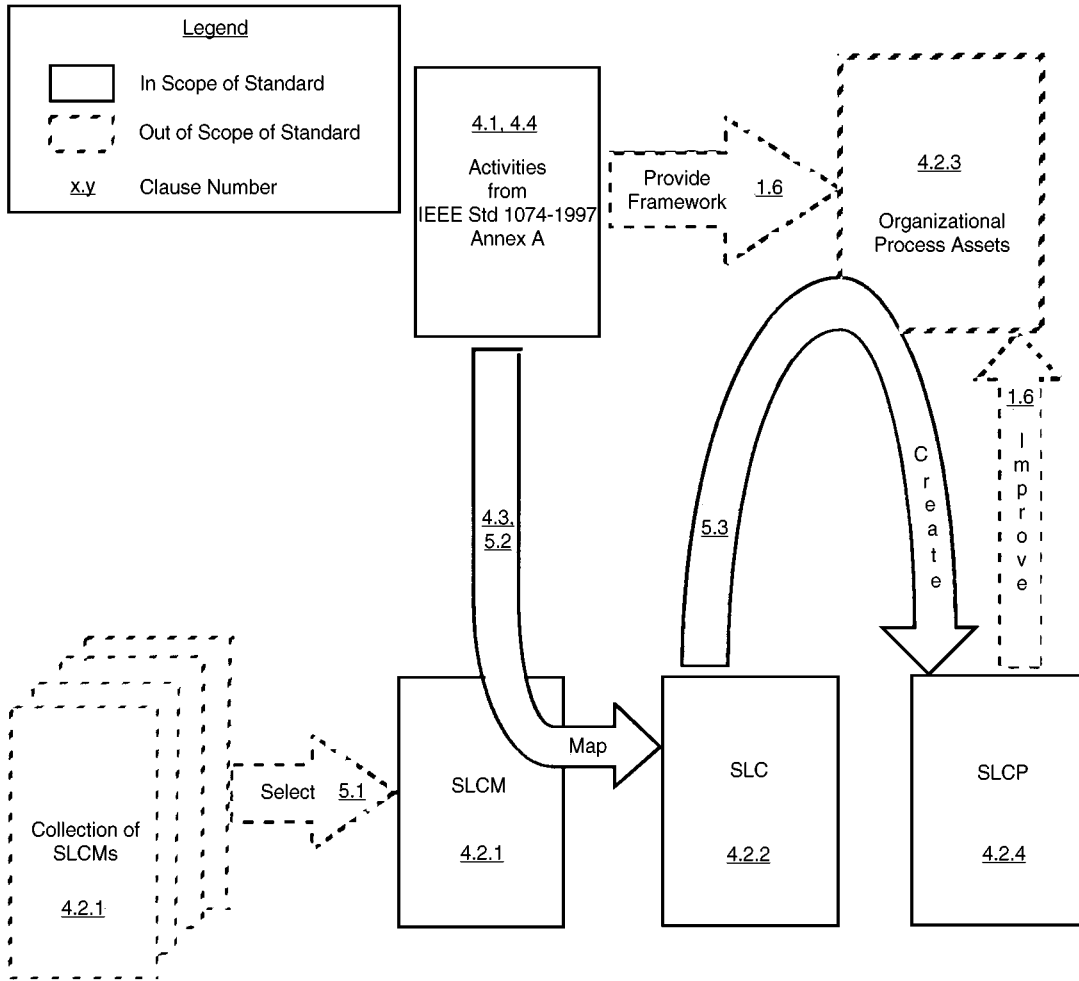


Figure 1—Developing an SLCP

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.

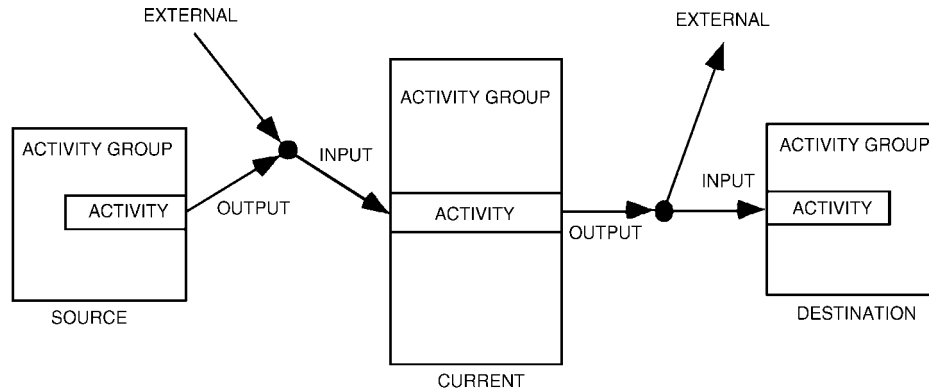


Figure 2—Information flow

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

A.1.2.6.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Training Planned Information	Project Monitoring and Control	Manage the Project (A.1.3.2)
	Maintenance	Identify Software Improvement Needs (A.4.3.1)
	Training	All Training Activities (A.5.4)

A.1.2.7 Plan Project Management

A.1.2.7.1 Input Information

Input Information	Source	
	Activity Group	Activity
Contractual Requirements	External	—
SLCP	Project Initiation	Create SLCP (A.1.1.1)
Project Estimates	Project Initiation	Perform Estimations (A.1.1.2)
Resource Allocations	Project Initiation	Allocate Project Resources (A.1.1.3)
Risk Management Reported Information	Project Monitoring and Control	Manage Risks (A.1.3.1)
Project Management Reported Information	Project Monitoring and Control	Manage the Project (A.1.3.2)
Preliminary Statement of Need	Concept Exploration	Identify Ideas or Needs (A.2.1.1)
Recommendations	Concept Exploration	Conduct Feasibility Studies (A.2.1.3)
Statement of Need	Concept Exploration	Refine and Finalize the Idea or Need
Imported Software Requirements	Software Importation	Identify Imported Software Requirements (A.2.3.1)

A.1.2.7.2 Description

Project management planning requires the collection and synthesis of a great deal of information into a coherent and organized SPMPI based on the SLCP. This Activity shall initially define, and subsequently update, the SPMPI using the Input Information. This Activity shall detail the project organization and assign responsibilities. Standards, methodologies, and tools for configuration management, quality, evaluation, training, documentation, and development shall be specified. This Activity shall apportion the project budget and staffing, and define schedules, using the applicable Input Information. It also shall define procedures for scheduling, tracking, and reporting. It shall address considerations such as regulatory approvals, required certifications, user involvement, subcontracting, and security.

This Activity shall include planning for support, problem reporting, risk management, and retirement. Support planning shall include methods for supporting the software in the operational environment. Problem Reporting and Resolution Planning Information (PR&RPI) shall include, at a minimum, a definition of the method for logging, routing, and handling problem reports; categories of severity; and the method for verifying problem resolution. Planning for managing risks includes identifying risk factors, analyzing those risks, and developing threshold conditions and contingency action plans. Retirement Planned Information shall address issues such as probable retirement date, archiving, replacement, and residual support issues.

As new or revised Input Information is received in this Activity, project plans shall be updated and further project planning shall be based upon these updated plans.

Additional guidance for SPMPIs can be found in IEEE Std 1058-1998 [B22] and IEEE Std 1220-1998 [B30].

Prior to the distribution of the SPMPI, 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.1.2.7.3 Output Information

Output Information	Destination	
	Activity Group	Activity
SPMPI	Most Activity Groups	Most Activities
PR&RPI	Project Monitoring and Control	Manage Risks (A.1.3.1)
		Manage the Project (A.1.3.2)
	Maintenance	Implement Problem Reporting Method (A.4.3.2)
		Reapply SLC (A.4.3.3)
Retirement Planned Information	Project Monitoring and Control	Manage the Project (A.1.3.2)
	Retirement	Notify User (A.4.4.1)
		Conduct Parallel Operations (If Applicable) (A.4.4.2)
		Retire System (A.4.4.3)
Support Planned Information	Project Monitoring and Control	Manage Risks (A.1.3.1)
		Manage the Project
	Operation and Support	Operation and Support Activities (A.4.2)

A.1.2.8 Plan Integration

Input Information	Source	
	Activity Group	Activity
Evaluation Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
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)
Software Detailed Design	Design	Perform Detailed Design (A.3.2.4)

A.1.2.8.1 Description

During the Plan Integration Activity, the Software Requirements and the Software Detailed Design are analyzed to determine the order for combining software components into an overall system. The SLCP, as defined in the SPMPI, shall be considered when planning integration. The integration methods shall be documented in the Integration Planned Information. The Integration Planned Information shall be coordinated with the Test Planned Information.

The integration planning information shall include the tools, techniques, and methodologies needed to perform the integrations. The planning shall include developing schedules, estimating resources, identifying special resources, staffing, and establishing exit or acceptance criteria.

Prior to the distribution of the Integration 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.1.2.8.2 Output Information

Output Information	Destination	
	Activity Group	Activity
Integration Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
	Project Monitoring and Control	Manage Risks (A.1.3.1)
		Manage the Project (A.1.3.2)
	Implementation	Perform Integration (A.3.3.3)

A.1.3 Project Monitoring and Control Activities

These Activities are used to track and manage the project. During the Project Monitoring and Control Activities, actual project performance is tracked, reported, and managed against the planned performance. Special consideration is given to the management of risk.

In addition, Project Monitoring and Control encompasses the collection and analysis of the software metrics of the project, the retention of project records, and the identification of SLCP Improvement Opportunities.

Project Monitoring and Control Activities are

- a) A.1.3.1, Manage Risks
- b) A.1.3.2, Manage the Project
- c) A.1.3.3, Identify SLCP Improvement Needs
- d) A.1.3.4, Retain Records
- e) A.1.3.5, Collect and Analyze Metric Data

A.1.3.1 Manage Risks**A.1.3.1.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Procurement/Lease Data	External	—
System Constraints	External	—
Historical Project Records	External	—
SLCP	Project Initiation	Create SLCP (A.1.1.1)
List of Activities Not Used	Project Initiation	Create SLCP (A.1.1.1)
Project Estimates	Project Initiation	Perform Estimations (A.1.1.2)
Estimation Assumptions	Project Initiation	Perform Estimations (A.1.1.2)
Resource Allocations	Project Initiation	Allocate Project Resources (A.1.1.3)
Evaluation Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
Transition Impact Statement (If Available)	Project Planning	Plan System Transition (A.1.2.3)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Support Planned Information	Project Planning	Plan Project Management (A.1.2.7)
PR&RPI	Project Planning	Plan Project Management (A.1.2.7)
Integration Planned Information	Project Planning	Plan Integration (A.1.2.8)
Project Management Reported Information	Project Monitoring and Control	Manage the Project (A.1.3.2)
Analysis Reported Information	Project Monitoring and Control	Collect and Analyze Metric Data (A.1.3.5)
Statement of Need	Concept Exploration	Refine and Finalize Idea or Need (A.2.1.4)
Imported Software Requirements	Software Importation	Identify Imported Software Requirements (A.2.3.1)
Software Interface Requirements	Requirements	Define Interface Requirements (A.3.1.2)
Software Requirements	Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Detailed Design	Design	Perform Detailed Design (A.3.2.4)
Evaluation Reported Information	Evaluation	Report Evaluation Results (A.5.1.7)

A.1.3.1.2 Description

This activity shall iteratively analyze and mitigate business, technical, managerial, economic, safety, schedule, and security risks. Factors that could impair or prevent the accomplishment of project objectives, or could require technical trade-offs for accomplishing the technical objectives of the project or product, shall be identified and analyzed. Technical factors can include such items as real-time performance, safety considerations, security considerations, implementation considerations, usability considerations, testability, and maintainability. Analytical approaches for technical risk assessment can include static and dynamic modeling and simulation, prototyping, independent reviews, and audits.

Cost, resource factors, earnings, liabilities, or any other economic measures involved in the project shall be identified and analyzed. The objective of this analysis is to identify potential economic opportunities, losses, and trade-offs. Analytical approaches for economic risk assessment can include financial analysis, such as return on investment and possible incentive and penalty contract clauses.

Operational support risk analysis shall determine the probability that the delivered software will meet the users' requirements. Operational support requirements such as interoperability, security, performance,

installability, and maintainability shall be considered. Both the completeness of, and the conformance to, these requirements shall be analyzed. The risks to the safety and reliability of the software, due to software requirements and requirement changes, shall be assessed.

Cost, resource, technical, and other requirements shall be evaluated for their impact on project schedules. This analysis should consider project interdependence and the effect of critical path analysis and resource leveling techniques.

Using the Input Information, this Activity shall also define alternative actions to reduce the cost or likelihood of risks occurring and actions to take in the event that a given risk materializes. Actions shall include resource planning and the establishment of trigger conditions that would invoke a contingency action. Contingency actions can include the consideration of revised requirements, delay, or the cancellation of the project. The threshold conditions that are determined shall be tracked against actual conditions. When a threshold condition is met, the contingency response shall be activated to address the risk.

Project Estimates and their corresponding Estimation Assumptions shall also be analyzed by the Manage Risks Activity. The results of the analyses that are conducted during this Activity shall be included in the Risk Management Reported Information.

Further information on risk management can be found in IEEE Std 1228-1994 [B31].

Prior to the distribution of the Risk Management Reported 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

Activity A.5.1.3, Conduct Audits, should be invoked.

A.1.3.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Risk Management Reported Information	Project Planning	Plan Evaluations (A.1.2.1)
		Plan Project Management (A.1.2.7)
	Project Monitoring and Control	Manage the Project (A.1.3.2)
	Requirements	Define and Develop Software Requirements (A.3.1.1)
		Prioritize and Integrate Software Requirements (A.3.1.3)

A.1.3.2 Manage the Project**A.1.3.2.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Feedback Data	External	—
SLCP	Project Initiation	Create SLCP (A.1.1.1)
Project Estimates	Project Initiation	Perform Estimations (A.1.1.2)
Resource Allocations	Project Initiation	Allocate Project Resources (A.1.1.3)
Evaluation Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
SCMPI	Project Planning	Plan Configuration Management Program (A.1.2.2)
Transition Planned Information (If Available)	Project Planning	Plan System Transition (A.1.2.3)
Software Installation Planned Information	Project Planning	Plan Installation (A.1.2.4)
Documentation Planned Information	Project Planning	Plan Documentation (A.1.2.5)
Training Planned Information	Project Planning	Plan Training (A.1.2.6)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Retirement Planned Information	Project Planning	Plan Project Management (A.1.2.7)
Support Planned Information	Project Planning	Plan Project Management (A.1.2.7)
PR&RPI	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)
Analysis Reported Information	Project Monitoring and Control	Collect and Analyze Metric Data (A.1.3.5)
Selected Software Import Sources	Software Importation	Evaluate Software Import Sources (A.2.3.2)
Installation Reported Information	Installation	Install Software (A.4.1.2)
Software Improvement Recommendations	Maintenance	Identify Software Improvement Needs (A.4.3.1)
Evaluation Reported Information	Evaluation	Report Evaluation Results (A.5.1.7)
Status Reported Information	Software Configuration Management	Perform Status Accounting (A.5.2.3)

A.1.3.2.2 Description

This Activity shall manage the execution of all Activities in the SLCP, according to the plans set forth in the Project Planning Activities. The progress of the project shall be reviewed and measured against the established estimates and plans (e.g., estimated vs. actual cost, estimated vs. actual effort, and planned vs. actual progress). The Input Information shall be tracked and analyzed; any additional pertinent data shall be gathered and analyzed in order to enable the status of the project to be reported. Any Anomalies encountered shall also be reported. This Activity also encompasses the day-to-day management of the project that is needed to ensure successful project completion.

This Activity may invoke Activity A.5.1.1, Conduct Reviews, or Activity A.5.1.3, Conduct Audit, in order to verify compliance to the SLCP and/or Project Planning plans.

Prior to the distribution of the Project Management Reported Information, Activity A.5.1.1, Conduct Reviews, should be invoked.

A.1.3.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Project Management Reported Information	External	—
	Project Planning	Plan Project Management (A.1.2.7)
	Project Monitoring and Control	Manage Risks (A.1.3.1)
Identify SLCP Improvement Needs (A.1.3.3)		
Anomalies	Maintenance	Implement Problem Reporting Method (A.4.3.2)

A.1.3.3 Identify SLCP Improvement Needs

A.1.3.3.1 Input Information

Input Information	Source	
	Activity Group	Activity
Historical Project Records	External	—
SLCP	Project Initiation	Create SLCP (A.1.1.1)
Project Management Reported Information	Project Monitoring and Control	Manage the Project (A.1.3.2)
Analysis Reported Information	Project Monitoring and Control	Collect and Analyze Metric Data (A.1.3.5)
Software Improvement Recommendations	Maintenance	Identify Software Improvement Needs (A.4.3.1)
Post-Operation Review Reported Information	Retirement	Retire System (A.4.4.3)
Evaluation Reported Information	Evaluation	Report Evaluation Results (A.5.1.7)

A.1.3.3.2 Description

This activity shall analyze Project Management Reported Information, project metrics from Analysis Reported Information, Evaluation Reported Information, and the other inputs to determine instances in which SLCP improvements could be beneficial. These analyses could be accomplished by using techniques such as Pareto analysis, control charts, fishbone diagrams, and process capability measurements.

Historical Project Records might provide the historical information that is needed to analyze the information from the project.

Environment Improvement Needs shall describe the requested change and shall contain objective criteria to be used to determine if the implemented change produced a positive result. Environment Improvement Needs can point to improvement opportunities that are outside the scope of the project.

Further information on process improvement can be found in IEEE Std 1045-1992 [B21] and IEEE Std 1061-1998 [B24].

A.1.3.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Environment Improvement Needs	External	—
	Project Initiation	Create SLCP (A.1.1.1)
	Maintenance	Implement Problem Reporting Method (A.4.3.2)

A.1.3.4 Retain Records**A.1.3.4.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Information Retention Standards	External	—
Records	Originating Activity Group	Originating Activity

A.1.3.4.2 Description

This Activity accepts project records from each Activity Group. The Records shall be retained in accordance with pertinent planning information and any external Information Retention Standards. The Records become part of the Historical Project Records of the organization. Uses for these records can include project audits, future project planning, and corporate accounting.

Further information on record retention can be found in IEEE Std 1298-1992 [B33].

A.1.3.4.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Historical Project Records	External	—

A.1.3.5 Collect and Analyze Metric Data

A.1.3.5.1 Input Information

Input Information	Source	
	Activity Group	Activity
Customer Input Information	External	—
Support Personnel Reported Information	External	—
Historical Project Records	External	—
Metric Data	Originating Activity Group	Originating Activity
Defined Metrics	Project Initiation	Define Metrics (A.1.1.4)
Collection and Analysis Methods	Project Initiation	Define Metrics (A.1.1.4)
Evaluation Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
Correction Problem Reported Information	Maintenance	Implement Problem Reporting Method (A.4.3.2)
Enhancement Problem Reported Information	Maintenance	Implement Problem Reporting Method (A.4.3.2)
Report Log	Maintenance	Implement Problem Reporting Method (A.4.3.2)
Resolved Problem Reported Information	Maintenance	Reapply SLC (A.4.3.3)
Updated Report Log	Maintenance	Reapply SLC (A.4.3.3)
Evaluation Reported Information	Evaluation	Report Evaluation Results (A.5.1.7)

A.1.3.5.2 Description

This Activity collects and analyzes project-generated Metric Data, Evaluation Reported Information, Customer Input Information, and Support Personnel Reported Information, as defined in the Collection and Analysis Methods. The Customer Input Information should also be used to obtain a customer point-of-view of the project and to gauge the customer's satisfaction with the software. Historical Project Records can prove to be valuable in the analysis of the metric(s) for the purposes of comparison and for obtaining trend information.

Analysis Reported Information shall be generated that contains the resulting metric(s) and describes the metric(s) analysis.

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 the distribution of the Analysis Reported 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, Documentation Development

A.1.3.5.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Analysis Reported Information	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)
	Maintenance	Identify Software Improvement Needs (A.4.3.1)

A.2 Pre-Development Activity Groups

These are the Activity Groups that explore concepts and allocate system requirements before software development can begin.

A.2.1 Concept Exploration Activities

A development effort is initiated with the identification of an idea or need for a system to be developed, whether it is a new effort or a change to all or part of an existing application. The Concept Exploration Activity Group examines the requirements at the system level, thus producing a Statement of Need that initiates the System Allocation or Requirements Activity Group. The Concept Exploration Activity Group includes the identification of an idea or need, the evaluation and refinement of the idea or need, and, once boundaries are placed around it, the generation of a Statement of Need for developing a system.

Concept Exploration Activities are

- a) A.2.1.1, Identify Ideas or Needs
- b) A.2.1.2, Formulate Potential Approaches
- c) A.2.1.3, Conduct Feasibility Studies
- d) A.2.1.4, Refine and Finalize the Idea or Need

A.2.1.1 Identify Ideas or Needs**A.2.1.1.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Changing Software Requirements	External	—
Customer Requests	External	—
Ideas from Within the Development Organization	External	—
Marketing Information Sources	External	—
User Requests	External	—
Enhancement Problem Reported Information	Maintenance	Implement Problem Reporting Method (A.4.3.2)
Maintenance Recommendations	Maintenance	Reapply SLC (A.4.3.3)

A.2.1.1.2 Description

An idea or a need for a new or modified system is generated from one or more of the sources identified in the table above. Input Information to the Preliminary Statement of Need shall be documented, outlining the function and performance needs. Changing Software Requirements can come from legislation, regulations, national and international standards, maintenance, etc.

Prior to the distribution of the Preliminary Statement of Need, the Activity A.5.1.1, Conduct Reviews, may be invoked.

A.2.1.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Preliminary Statement of Need	Project Planning	Plan System Transition (If Applicable) (A.1.2.3)
		Plan Project Manager (A.1.2.7)
	Concept Exploration	Formulate Potential Approaches (A.2.1.2)
		Conduct Feasibility Studies (A.2.1.3)
		Refine and Finalize the Idea or Need (A.2.1.4)

A.2.1.2 Formulate Potential Approaches

A.2.1.2.1 Input Information

Input Information	Source	
	Activity Group	Activity
Development Resources and Budget	External	—
Market Availability Data	External	—
Resource Information	External	—
Preliminary Statement of Need	Concept Exploration	Identify Ideas or Needs (A.2.1.1)

A.2.1.2.2 Description

Using Resource Information, budget data, and the availability of third party or existing reusable software products, Potential Approaches shall be developed that are based upon the Preliminary Statement of Need and any data that is pertinent to the decision to develop or acquire the system. The Formulate Potential Approaches Activity shall also produce the Constraints and Benefits with regard to the development of the software. The Constraints and Benefits should include all aspects of the life cycle.

Prior to the distribution of the Constraints and Benefits and the Potential Approaches, the following Activities may be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control

A.2.1.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Constraints and Benefits	Concept Exploration	Conduct Feasibility Studies (A.2.1.3)
		Refine and Finalize the Idea or Need (A.2.1.4)
Potential Approaches	Concept Exploration	Conduct Feasibility Studies (A.2.1.3)
		Refine and Finalize the Idea or Need (A.2.1.4)

A.2.1.3 Conduct Feasibility Studies**A.2.1.3.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Preliminary Statement of Need	Concept Exploration	Identify Ideas or Needs (A.2.1.1)
Constraints and Benefits	Concept Exploration	Formulate Potential Approaches (A.2.1.2)
Potential Approaches	Concept Exploration	Formulate Potential Approaches (A.2.1.2)

A.2.1.3.2 Description

The feasibility study shall include the analysis of the idea or need, the Potential Approaches, and all life cycle Constraints and Benefits. Modeling and prototyping techniques might also be considered. In conducting the feasibility study, there could be a need to decide whether to make or buy the system, in part or in total. Justification for each Recommendation shall be fully documented and formally approved by all concerned organizations (including the user and the developer).

Prior to the distribution of the Recommendations, Activity A.5.1.1, Conduct Reviews, may be invoked.

A.2.1.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Recommendations	Project Planning	Plan System Transition (If Applicable) (A.1.2.3)
		Plan Project Management (A.1.2.7)
	Concept Exploration	Refine and Finalize the Idea or Need (A.2.1.4)
	System Allocation	Analyze Functions (A.2.2.1)

A.2.1.4 Refine and Finalize the Idea or Need

A.2.1.4.1 Input Information

Input Information	Source	
	Activity Group	Activity
Preliminary Statement of Need	Concept Exploration	Identify Ideas or Needs (A.2.1.1)
Constraints and Benefits	Concept Exploration	Formulate Potential Approaches (A.2.1.2)
Potential Approaches	Concept Exploration	Formulate Potential Approaches (A.2.1.2)
Recommendations	Concept Exploration	Conduct Feasibility Studies (A.2.1.3)
Transition Impact Statement (If Available)	Project Planning	Plan System Transition (A.1.2.3)

A.2.1.4.2 Description

The idea or need shall be refined by analyzing the Preliminary Statement of Need, the Potential Approaches, the Recommendations, and the Transition Impact Statement (If Available). An approach shall be selected and documented that refines the initial idea or need.

Based upon the refined ideas or needs, a Statement of Need shall be generated that identifies the software idea, need, or desire; the recommended approach for its implementation; and any data that is pertinent to a management decision concerning the initiation of the described development effort.

Prior to the distribution of the Statement of Need, the following Activities may 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.2.1.4.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Statement of Need	Project Initiation	Create SLCP (A.1.1.1)
		Perform Estimations (A.1.1.2)
		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)
	System Allocation	Analyze Functions (A.2.2.1)
		Develop System Architecture (A.2.2.2)

A.2.2 System Allocation Activities

The System Allocation Activity Group is the bridge between Concept Exploration and the definition of software requirements. This Activity Group maps the required functions to software and, when applicable, to hardware and people.

The Statement of Need forms the basis for the analysis of the system, thus resulting in system requirements. These requirements determine the inputs to the system, the processing to be applied to the inputs, and the required outputs. The software and hardware (if required) operational functions are also identified in these definitions.

The architecture of the system shall be developed through the System Allocation Activity Group. The system functions are derived from system requirements, and the hardware, software, and operational requirements are identified. These requirements are analyzed to produce System Functional Software Requirements and System Functional Human and Hardware Requirements (If Applicable). The hardware, software, and operational interfaces shall be defined and closely monitored. No hardware requirements analysis is discussed in this document; it is beyond the scope of this standard.

System Allocation Activities are

- a) A.2.2.1, Analyze Functions
- b) A.2.2.2, Develop System Architecture
- c) A.2.2.3, Decompose System Requirements

A.2.2.1 Analyze Functions

A.2.2.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
Recommendations	Concept Exploration	Conduct Feasibility Studies (A.2.1.3)
Statement of Need	Concept Exploration	Refine and Finalize the Idea or Need (A.2.1.4)

A.2.2.1.2 Description

The Statement of Need and Recommendations for solution shall be analyzed to identify the functions of the total system. Once the functions have been identified, they are delineated in the Functional Description of the System and are used to develop the system architecture and to identify the software and (if applicable) hardware functions.

Prior to the distribution of the Functional Description of the System, the following Activities shall be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control

A.2.2.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Functional Description of the System	System Allocation	Develop System Architecture (A.2.2.2)
		Decompose System Requirements (A.2.2.3)
	Requirements	Define Interface Requirements (A.3.1.2)

A.2.2.2 Develop System Architecture

A.2.2.2.1 Input Information

Input Information	Source	
	Activity Group	Activity
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Statement of Need	Concept Exploration	Refine and Finalize the Idea or Need (A.2.1.4)
Functional Description of the System	System Allocation	Analyze Functions (A.2.2.1)

A.2.2.2.2 Description

The Statement of Need and the Functional Description of the System shall be transformed into the System Architecture using the methodology, standards, and tools that are established by the organization. The System Architecture becomes the basis for the Design Activity Group and for the determination of the software functions and the hardware functions, if any.

A.2.2.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
System Architecture	System Allocation	Decompose System Requirements (A.2.2.3)
	Design	Perform Architectural Design (A.3.2.1)

A.2.2.3 Decompose System Requirements

A.2.2.3.1 Input Information

Input Information	Source	
	Activity Group	Activity
Functional Description of the System	System Allocation	Analyze Functions (A.2.2.1)
System Architecture	System Allocation	Develop System Architecture (A.2.2.2)

A.2.2.3.2 Description

The system functions that are documented in the Functional Description of the System shall be divided according to the System Architecture in order to form software requirements, human and hardware requirements (if applicable), and the System Interface Requirements (if available). The System Interface Requirements define the interfaces that are external to the system and the interfaces between configuration items that comprise the system. Note that any hardware requirements go to an external destination because they are beyond the scope of this standard. The decomposition of the system could result in requirements for more than one project. Each software project shall be managed individually.

Further information on system requirements can be found in IEEE Std 1233, 1998 Edition [B32].

Prior to the distribution of Software Functional Requirements and System Interface Requirements, 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.2.2.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
System Functional Human and Hardware Requirements (If Applicable)	External	—
System Functional Software Requirements	Project Initiation	Perform Estimations (A.1.1.2)
		Allocate Project Resources (A.1.1.3)
	Requirements	Define and Develop Software Requirements (A.3.1.1)
		Define Interface Requirements (A.3.1.2)
System Interface Requirements (If Available)	External	—
	Requirements	Define and Develop Software Requirements (A.3.1.1)
		Define Interface Requirements (A.3.1.2)

A.2.3 Software Importation Activities

Some or all of the software requirements may best be satisfied by reusing existing software or by acquiring software from outside the project. This software may or may not belong to the developing organization. Imported Software can consist of code libraries, device drivers, various utilities, or even fully functional systems that are to be integrated into the current development project. Software Importation Activities provide the means to extract the software requirements that will be satisfied through importation, to evaluate candidate sources from which the imported software might be obtained, to determine the method of importation, and to import the software, including documentation, into the project.

Software Importation Activities are

- a) A.2.3.1, Identify Imported Software Requirements
- b) A.2.3.2, Evaluate Software Import Sources (If Applicable)
- c) A.2.3.3, Define Software Import Method (If Applicable)
- d) A.2.3.4, Import Software (If Applicable)

A.2.3.1 Identify Imported Software Requirements

A.2.3.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Software Requirements	Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)

A.2.3.1.2 Description

The Identify Imported Software Requirements Activity extracts those Software Requirements that can best be satisfied with existing or acquired software. The resulting requirements for imported software (Imported Software Requirements) cover all categories of requirements, including schedule and budget constraints.

Further information related to this Activity can be found in IEEE Std 1062, 1998 Edition [B25].

Prior to the distribution of the Imported Software Requirements, Activity A.5.1.1, Conduct Reviews, shall be invoked.

A.2.3.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Imported Software Requirements	Project Planning	Project Planning Activities (A.1.2)
	Project Monitoring and Control	Manage Risks (A.1.3.1)
	Software Importation	Evaluate Software Import Sources (A.2.3.2)
	Design	All Design Activities (A.3.2)
	Evaluation	Conduct Reviews (A.5.1.1)
Create Test Data (A.5.1.5)		

A.2.3.2 Evaluate Software Import Sources (If Applicable)

A.2.3.2.1 Input Information

Input Information	Source	
	Activity Group	Activity
Imported Software Requirements	Software Importation	Identify Imported Software Requirements (A.2.3.1)

A.2.3.2.2 Description

The Evaluate Software Import Sources Activity applies when software is to be imported for use on the project. This Activity is the mechanism to determine if the Imported Software Requirements are to be satisfied using software from another project within the organization, including items from a reuse library, or if the requirements are to be satisfied by a source outside the organization. Software outside the organization can include public domain software, freeware, shareware, subcontracted development, or purchased commercial software. The available sources shall be evaluated with respect to the compliance of the available software with the requirements, availability, schedule, cost, and the software quality program of the source. The effects on overall project budget, cost, and risk shall be considered in this evaluation and shall be communicated to project management.

For the Selected Software Import Sources, Candidate Software Import Methods by which the software will actually be acquired shall be determined. For example, in the case of software that is to be purchased off the shelf, methods could include site licensing, limited individual licenses, bulk purchase, etc. In the case of software that is to be contractually acquired, methods could include turn-key development, development in the target system's physical project location, various forms of test conduct, contractual clauses dealing with quality programs and configuration management, etc.

Further information related to this Activity can be found in IEEE Std 1063-1987 [B26].

A.2.3.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Selected Software Import Sources	Project Monitoring and Control	Manage the Project (A.1.3.2)
	Software Importation	Decline Software Import Method (A.2.3.3)
		Import Software (A.2.3.4)
Candidate Software Import Methods	Software Importation	Define Software Import Method (A.2.3.3)

A.2.3.3 Define Software Import Method (If Applicable)**A.2.3.3.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Selected Software Import Sources	Software Importation	Evaluate Software Import Sources (A.2.3.2)
Candidate Software Import Methods	Software Importation	Evaluation Software Import Sources (A.2.3.2)

A.2.3.3.2 Description

The Define Software Import Method Activity applies when software is to be imported for use on the project. Using the listed Input Information, this Activity shall select the most appropriate methods by which the Selected Software Import Sources will provide the imported software. Consideration should be given to the integration of the software importation with the overall project schedule, configuration management, budget and personnel resource requirements, imported software testing requirements, etc.

Further information related to this Activity can be found in IEEE Std 1062, 1998 Edition [B25].

A.2.3.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Selected Software Import Methods	Software Importation	Import Software (A.2.3.4)

A.2.3.4 Import Software (If Applicable)**A.2.3.4.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Selected Software Import Sources	Software Importation	Evaluate Software Import Sources (A.2.3.2)
Selected Software Import Methods	Software Importation	Define Software Import Method (A.2.3.3)

A.2.3.4.2 Description

The Import Software Activity applies when software is to be imported for use on the project. This Activity brings the imported components into the software project in a controlled manner that ensures their orderly integration into the total software system. The imported software shall be integrated into the design as well as the implementation.

Further information related to this Activity can be found in IEEE Std 1062, 1998 Edition [B25].

Prior to the distribution of the Imported Software, the following Activity Groups shall be invoked:

- a) A.5.1.6, Execute Tests
- b) A.5.2.2, Perform Configuration Control
- c) A.5.4.1, Develop Training Materials

A.2.3.4.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Imported Software	Implementation	Perform Integration (A.3.3.3)
	Evaluation	Execute Tests (A.5.1.6)
Imported Software Documentation	Design	Perform Detailed Design (A.3.2.4)
	Evaluation	Conduct Reviews (A.5.1.1)
	Documentation Development	Implement Documentation (A.5.3.1)
	Training	Develop Training Materials (A.5.4.1)

A.3 Development Activity Groups

These are the Activity Groups that are performed during the development of a software product.

A.3.1 Requirements Activities

This Activity Group includes those Activities that are directed toward the development of software requirements. In the development of a system that contains hardware, human, and software components, the Requirements Activity Group follows the development of total system requirements, and the functional allocation of those system requirements to hardware, humans, and software.

Requirements Activities are

- a) A.3.1.1, Define and Develop Software Requirements
- b) A.3.1.2, Define Interface Requirements
- c) A.3.1.3, Prioritize and Integrate Software Requirements

A.3.1.1 Define and Develop Software Requirements

A.3.1.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
Installation Support Requirements	External	—
System Constraints	External	—
SPMPI	Project Initiation	Plan Project Management (A.1.2.7)
Risk Management Reported Information	Project Monitoring and Control	Manage Risks (A.1.3.1)
System Functional Software Requirements	System Allocation	Decompose System Requirements (A.2.2.3)
System Interface Requirements (If Available)	System Allocation	Decompose System Requirements (A.2.2.3)

A.3.1.1.2 Description

The first Activity in this Activity Group, defining the software requirements, is usually iterative in nature. Whether the software development constitutes the entire project or is part of a system (e.g., hardware, humans, and software), software requirements, including constraints, shall be generated from Input Information documents and the results of modeling, prototyping, or other techniques.

Using the above Input Information, the developer shall analyze the software functional and performance requirements to determine traceability, clarity, validity, testability, safety, and any other project-specific characteristics. The use of a comprehensive methodology is recommended to ensure that requirements are complete and consistent. Techniques such as structured analysis, modeling, prototyping, or transaction analysis are helpful in this Activity. When needed, the requirements for a data base shall be included in the requirements.

The Preliminary Software Requirements and Installation Requirements determination shall include the consideration of System Constraints such as timing, sizing, language, marketing restrictions, and technology.

Further information related to this Activity can be found in IEEE Std 830-1998 [B7].

Prior to the distribution of the Preliminary Software Requirements and Installation Requirements, 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.3.1.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Preliminary Software Requirements	Project Planning	Plan Evaluations (A.1.2.1)
	Requirements	Define Interface Requirements (A.3.1.2)
		Prioritize and Integrate Software Requirements (A.3.1.3)
Installation Requirements	Project Planning	Plan Installation (A.1.2.4)

A.3.1.2 Define Interface Requirements**A.3.1.2.1 Input Information**

Input Information	Source	
	Activity Group	Activity
System Constraints	External	—
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Functional Description of the System	System Allocation	Analyze Functions (A.2.2.1)
System Functional Software Requirements	System Allocation	Decompose System Requirements (A.2.2.3)
Preliminary Software Requirements	Requirements	Define and Develop Software Requirements (A.3.1.1)
System Interface Requirements (If Available)	System Allocation	Decompose System Requirements (A.2.2.3)

A.3.1.2.2 Description

All user, software, and hardware interfaces shall be defined using the applicable Input Information. These interfaces shall be defined either as requirements or as constraints, and shall be reviewed by all involved parties.

The user interface is critical in determining the usability of the system. The user interface definition shall specify not only the information flow between the user and the system, but also the manner in which a user goes about using the system.

The Software Interface Requirements shall specify all software interfaces that are required to support the development and execution of the software system. Software interfaces can be affected by System Constraints including operating system, data base management system, language compiler, tools, utilities, network protocol drivers, and hardware interfaces.

Further information related to this Activity can be found in IEEE Std 830-1998 [B7] and IEEE Std 1175-1991 [B27].

Prior to the distribution of the Software Interface Requirements, 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.3.1.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Software Interface Requirements	Project Monitoring and Control	Manage Risks (A.1.3.1)
	Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)

A.3.1.3 Prioritize and Integrate Software Requirements

A.3.1.3.1 Input Information

Input Information	Source	
	Activity Group	Activity
Risk Management Reported Information	Project Monitoring and Control	Manage Risks (A.1.3.1)
Preliminary Software Requirements	Requirements	Define and Develop Software Requirements (A.3.1.1)
Software Interface Requirements	Requirements	Define Interface Requirements (A.3.1.2)

A.3.1.3.2 Description

The functional and performance requirements shall be reviewed, and a prioritized list of requirements shall be produced that addresses any trade-offs that may be needed. The organization of the emerging Software Requirements shall be reviewed and revised as necessary. While completing the requirements, a particular design shall not be imposed (i.e., design decisions are made in the Design Activity Group). The Software Requirements shall describe the functional, interface, and performance requirements, and shall also define operational support environments.

Further information related to this Activity can be found in IEEE Std 830-1998 [B7].

Prior to the distribution of the Software Requirements, 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.3.1.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Software Requirements	Project Initiation	Allocate Project Resources (A.1.1.3)
		Define Metrics (A.1.1.4)
	Project Planning	Plan Evaluations (A.1.2.1)
		Plan Training (A.1.2.6)
		Plan Integration (A.1.2.8)
	Project Monitoring and Control	Manage Risks (A.1.3.1)
	Software Importation	Identify Imported Software Requirements (A.2.3.1)
	Design	Design Activities (A.3.2)
	Implementation	Create Operating Documentation (A.3.3.2)
	Evaluation	Conduct Reviews (A.5.1.1)
		Create Traceability Matrix (A.5.1.2)
		Develop Test Procedures (A.5.1.4)
		Create Test Data (A.5.1.5)

A.3.2 Design Activities

The objective of the Design Activity Group is to develop a coherent, well-organized representation of the software system that meets the Software Requirements. At the architectural design level, the focus is on the software components that comprise the software system, and the structure and interfacing of those components. At the detailed design level, the emphasis is on the data structures and algorithms for each software component.

The Perform Architectural Design and Perform Detailed Design Activities are usually carried out in sequence because detailed design is largely dependent on the architectural design. They differ from each other in the level of design detail. Other Design Activity Group Activities can be carried out in parallel with these Activities.

Design Activities are

- a) A.3.2.1, Perform Architectural Design
- b) A.3.2.2, Design Data Base (If Applicable)
- c) A.3.2.3, Design Interfaces
- d) A.3.2.4, Perform Detailed Design

A.3.2.1 Perform Architectural Design

A.3.2.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
System Architecture	System Allocation	Develop System Architecture (A.2.2.2)
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)

A.3.2.1.2 Description

The Perform Architectural Design Activity transforms the Software Requirements and the System Architecture into high-level design concepts. During this Activity, the software components that constitute the software system and their structures are identified. Purchased software and the contents of the software libraries can influence the architectural design. Techniques such as modeling and prototyping could be used to evaluate alternative designs.

By the end of the Perform Architectural Design Activity, the description of the design of each software component shall have been completed. The data, relationships, and constraints shall be specified. All internal interfaces (among components) shall be defined. This Activity shall create the Software Architectural Design.

Prior to the distribution of the Software Architectural Design, 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.3.2.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Software Architectural Design	Design	Perform Detailed Design (A.3.2.4)

A.3.2.2 Design Data Base (If applicable)

A.3.2.2.1 Input Information

Input Information	Source	
	Activity Group	Activity
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)

A.3.2.2.2 Description

The Design Data Base Activity applies when a data base is to be created or modified as a part of the project. This Activity shall specify the information structure that is outlined in the Software Requirements and its characteristics within the software system. The Design Data Base Activity involves three separate but related steps: conceptual data base design, logical data base design, and physical data base design. It does not involve designing or developing the Data Base Management System.

Techniques such as data dictionary, data base optimization, and data modeling should be considered. Requirements are molded into an external schema that describes data entities, attributes, relationships, and constraints. The various external schemata are integrated into a single conceptual schema. The conceptual schema is then mapped into an implementation-dependent logical schema. Finally, the physical data structures and access paths are defined. The result of this Activity is the generation of the Data Base Design.

Prior to the distribution of the Data Base Design, 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.3.2.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Data Base Design	Design	Perform Detailed Design (A.3.2.4)

A.3.2.3 Design Interfaces**A.3.2.3.1 Input Information**

Input Information	Source	
	Activity Group	Activity
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)

A.3.2.3.2 Description

The Design Interfaces Activity shall be concerned with the user, software, and hardware interfaces of the software system that is contained in the Software Requirements. This Activity shall consolidate these interface descriptions into a single Interface Design for the software system.

Prior to the distribution of the Interface Design, 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.3.2.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Interface Design	Design	Perform Detailed Design (A.3.2.4)

A.3.2.4 Perform Detailed Design

A.3.2.4.1 Input Information

Input Information	Source	
	Activity Group	Activity
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Imported Software Requirements	Software Importation	Identify Imported Software Requirements (A.2.3.1)
Imported Software Documentation	Software Importation	Import Software (A.2.3.4)
Software Requirements	Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Architectural Design	Design	Perform Architectural Design (A.3.2.1)
Data Base Design (If Available)	Design	Design Data Base (If Applicable) (A.3.2.2)
Interface Design	Design	Design Interfaces (A.3.2.3)

A.3.2.4.2 Description

In this Activity, design alternatives shall be chosen for implementing the functions that are specified for each software component. By the end of this Activity, the data structure, algorithm, and control information of each software component shall be specified. The Software Detailed Design contains the consolidated data for all of the above Input Information. The details of the interfaces shall be identified within the Software Detailed Design.

For further information on this topic, see IEEE Std 1016-1998 [B15] and IEEE Std 1016.1-1993 [B16].

Prior to the distribution of the Software Detailed Design, 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.3.2.4.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Software Detailed Design	Project Planning	Plan Evaluations (A.1.2.1)
		Plan Integration (A.1.2.8)
	Project Monitoring and Control	Manage Risks (A.1.3.1)
	Implementation	Create Executable Code (A.3.3.1)
		Create Operating Documentation (A.3.3.2)
	Evaluation	Develop Test Procedures (A.5.1.4)
		Create Test Data (A.5.1.5)
	Training	Develop Training Materials (A.5.4.1)

A.3.3 Implementation Activities

The Activities included in the Implementation Activity Group result in the transformation of the Detailed Design representation of a software product into a programming language realization. This Activity Group produces the Executable Code, the Data Base (if applicable), and the documentation that constitutes the physical manifestation of the design. In addition, the code and data base are integrated. Care must also be taken during the Implementation Activity Group to apply the appropriate coding standards.

The code and data base, along with the documentation that was produced within previous Activity Groups, are the first complete representation of the software product.

Implementation Activities are

- a) A.3.3.1, Create Executable Code
- b) A.3.3.2, Create Operating Documentation
- c) A.3.3.3, Perform Integration

A.3.3.1 Create Executable Code

A.3.3.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Software Detailed Design	Design	Perform Detailed Design (A.3.2.4)

A.3.3.1.2 Description

The Source Code, including suitable comments, shall be generated using the SLCP, as found in the SPMPI and the Software Detailed Design. If the Source Code is required for integration, it shall be made available to Activity A.3.3.3, Perform Integration. If the Source Code is going to be used to create test data, it shall be made available to Activity A.5.1.5, Create Test Data.

The code shall be grouped into processable units. (This will be dictated by the selected language and design information.) All units shall be transformed into Executable Code and debugged. Syntactically incorrect code, as identified by the transform output, shall be reworked until the Source Code can be processed free of syntactical errors.

Further information related to this Activity can be found in IEEE Std 1008-1987 [B12].

Prior to the distribution of the Source Code, Executable Code, and Data Base, the following Activities shall be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.1.6, Execute Tests
- c) A.5.2.2, Perform Configuration Control

A.3.3.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Source Code (When Required)	Implementation	Perform Integration (A.3.3.3)
Source Code (When Required)	Evaluation	Create Test Data (A.5.1.5)
Executable Code	Implementation	Perform Integration (A.3.3.3)
	Evaluation	Execute Tests (A.5.1.6)
Data Base (If Available)	Implementation	Perform Integration (A.3.3.3)
	Evaluation	Create Test Data (A.5.1.5)

A.3.3.2 Create Operating Documentation

A.3.3.2.1 Input Information

Input Information	Source	
	Activity Group	Activity
Documentation Planned Information	Project Planning	Plan Documentation (A.1.2.5)
Software Requirements	Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Detailed Design	Design	Perform Detailed Design (A.3.2.4)

A.3.3.2.2 Description

This Activity shall produce the software project's operating documentation from the Software Detailed Design and the Software Interface Requirements, in accordance with the Documentation Planned Information. The Operating Documentation is required for installing, operating, and supporting the system throughout the life cycle.

For further information, IEEE Std 1063-1987 [B26] can be used.

Prior to the distribution of the Operating Documentation, 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.3.3.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Operating Documentation	Project Planning	Plan Installation (A.1.2.4)
	Installation	Distribute Software (A.4.1.1)

A.3.3.3 Perform Integration

A.3.3.3.1 Input Information

Input Information	Source	
	Activity Group	Activity
System Components	External	—
SPMPI	Project Initiation	Plan Project Management (A.1.2.7)
Integration Planned Information	Project Planning	Plan Integration (A.1.2.8)
Imported Software	Software Importation	Import Software (A.2.3.4)
Source Code (When Required)	Implementation	Create Executable Code (A.3.3.1)
Executable Code	Implementation	Create Executable Code (A.3.3.1)
Tested Software	Evaluation	Execute Tests (A.5.1.6)
Data Base (If Available)	Implementation	Create Executable Code (A.3.3.1)
Stubs and Drivers (If Available)	Evaluation	Create Test Data (A.5.1.5)

A.3.3.3.2 Description

This Activity shall integrate the Data Base, Source Code (if required), Executable Code, and Stubs and Drivers, as specified in the Integration Planned Information, into the Integrated Software. Other necessary Executable Code, from the SLCP as defined in the SPMPI, shall also be integrated. If a system includes both hardware and software components, the system integration could be included as part of this Activity.

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

- a) A.5.1.1, Conduct Reviews
- b) A.5.1.6, Execute Tests
- c) A.5.2.2, Perform Configuration Control

A.3.3.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Integrated Software	Evaluation	Execute Tests (A.5.1.6)

A.4 Post-Development Activity Groups

These are the Activity Groups that are performed to install, operate, support, maintain, and retire a software product.

A.4.1 Installation Activities

Installation consists of the transportation and installation of a software system from the development environment to the target environment(s). It includes the necessary software modifications, checkout in the target environment(s), and customer acceptance. If a problem arises, it shall be identified and reported. If necessary and possible, a temporary “work-around” may be applied.

In the Installation Activity Group, the software to be delivered is installed, operationally checked out, and monitored. This effort culminates in formal customer acceptance. The scheduling of turnover and customer acceptance is defined in the SPMPI.

Installation Activities are

- a) A.4.1.1, Distribute Software
- b) A.4.1.2, Install Software
- c) A.4.1.3, Accept Software in Operational Environment

A.4.1.1 Distribute Software

A.4.1.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
Data Base Data	External	—
Software Installation Planned Information	Project Planning	Plan Installation (A.1.2.4)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Operating Documentation	Implementation	Create Operating Documentation (A.3.3.2)
Tested Software	Evaluation	Execute Tests (A.5.1.6)

A.4.1.1.2 Description

During this Activity, the Tested Software, Data Base Data, Operating Documentation, and Software Installation Planned Information shall be packaged onto their respective media as designated in the SPMPI. The Packaged Software is distributed to the appropriate site(s) for installation efforts. The Packaged Installation Planned Information is distributed, as appropriate, to the site(s) to instruct the installation efforts. The Packaged Operating Documentation shall be available for the operation of the system.

Prior to the distribution of the Packaged Information, Software, and Documentation, the following Activities shall be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.1.3, Conduct Audits
- c) A.5.2.2, Perform Configuration Control
- d) A.5.3.1, Implement Documentation

A.4.1.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Packaged Installation Planned Information	Installation	Install Software (A.4.1.2)
Packaged Software	Installation	Install Software (A.4.1.2)
Packaged Operating Documentation	Installation	Install Software (A.4.1.2)
	Operation and Support	Operate the System (A.4.2.1)

A.4.1.2 Install Software**A.4.1.2.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Data Base Data	External	—
Packaged Installation Planned Information	Installation	Distribute Software (A.4.1.1)
Packaged Operating Documentation	Installation	Distribute Software (A.4.1.1)
Packaged Software	Installation	Distribute Software (A.4.1.1)

A.4.1.2.2 Description

The Packaged Software, and any required Data Base Data, shall be installed in the target environment according to the procedures in the Packaged Installation Planned Information. This could include tailoring by the customer. The Installation Reported Information shall document the installation and any problems that are encountered.

A.4.1.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Installation Reported Information	Project Monitoring and Control	Manage the Project (A.1.3.2)
Installed Software	Installation	Accept Software in Operational Environment (A.4.1.3)

A.4.1.3 Accept Software in Operational Environment**A.4.1.3.1 Input Information**

Input Information	Source	
	Activity Group	Activity
User Acceptance Planned Information	External	—
Installed Software	Installation	Install Software (A.4.1.2)
Evaluation Reported Information	Evaluation	Report Evaluation Results (A.5.1.7)

A.4.1.3.2 Description

The software acceptance shall consist of an analysis of the Evaluation Reported Information, according to the User Acceptance Planned Information, to ensure that the Installed Software performs as expected. When the results of the analysis satisfy the requirements of the User Acceptance Planned Information, the Installed Software System is accepted by the user.

Prior to the completion of the acceptance of the software in the operational environment, the following Activities should be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control

A.4.1.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Customer Acceptance	External	—
Installed Software System	Operation and Support	Operate the System (A.4.2.1)
	Retirement	Conduct Parallel Operations (If Applicable) (A.4.4.2)

A.4.2 Operation and Support Activities

The Operation and Support Activity Group involves user operation of the system and ongoing support. Support includes providing technical assistance, consulting with the user, and recording user support requests by maintaining a Support Request Log. Thus, the Operation and Support Activity Group can trigger maintenance activities via the ongoing Project Monitoring and Control Activity Group, which will provide information that re-enters the SLCP.

Operation and Support Activities are

- a) A.4.2.1, Operate the System
- b) A.4.2.2, Provide Technical Assistance and Consulting
- c) A.4.2.3, Maintain Support Request Log

A.4.2.1 Operate the System

A.4.2.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
Feedback Data	External	—
Support Planned Information	Project Planning	Plan Project Management (A.1.2.7)
Packaged Operating Documentation	Installation	Distribute Software (A.4.1.1)
Installed Software System	Installation	Accept Software in Operational Environment (A.4.1.3)

A.4.2.1.2 Description

During this Activity, the Installed Software System shall be utilized in the intended environment and in accordance with the operating instructions. Feedback Data are collected for product and documentation improvement and system tuning. The user shall analyze the Feedback Data and identify Anomalies (which may include desired enhancements). Anomalies are then reported.

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

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control

A.4.2.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Operation Logs	External	—
Anomalies	Maintenance	Implement Problem Reporting Method (A.4.3.2)

A.4.2.2 Provide Technical Assistance and Consulting**A.4.2.2.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Request for Support	External	—
Support Planned Information	Project Planning	Plan Project Management (A.1.2.7)

A.4.2.2.2 Description

This Activity applies after the user has accepted the software. The support function shall include providing responses to the user's technical questions or problems. A Support Response is generated to the Maintain Support Request Log so that feedback can be provided to other Activity Groups.

A.4.2.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Support Response	External	—
	Operation and Support	Maintain Support Request Log (A.4.2.3)

A.4.2.3 Maintain Support Request Log**A.4.2.3.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Support Planned Information	Project Planning	Plan Project Management (A.1.2.7)
Support Response	Operation and Support	Provide Technical Assistance and Consulting (A.4.2.2)

A.4.2.3.2 Description

This Activity shall record support requests in the Support Request Log. The methodology regarding management of this Activity shall be as identified in the Support Planned Information. Anomalies that are reported shall be reported to the Maintenance Activity Group.

Prior to the release of the Support Request Log, Activity A.5.1.1, Conduct Reviews, shall be invoked.

A.4.2.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Anomalies	Maintenance	Implement Problem Reporting Method (A.4.3.2)
Support Request Log	Evaluation	Conduct Reviews (A.5.1.1)

A.4.3 Maintenance Activities

The Maintenance Activity Group is concerned with the identification of enhancements and the resolution of software errors, faults, and failures. The requirement for software maintenance initiates SLCP changes. The SLCP is remapped and executed, thereby treating the Maintenance Activity Group as iterations of development.

Maintenance Activities are

- a) A.4.3.1, Identify Software Improvement Needs
- b) A.4.3.2, Implement Problem Reporting Method
- c) A.4.3.3, Reapply SLC

A.4.3.1 Identify Software Improvement Needs

A.4.3.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
Evaluation Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
Training Planned Information	Project Planning	Plan Training (A.1.2.6)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Analysis Reported Information	Project Monitoring and Control	Collect and Analyze Metric Data (A.1.3.5)
Post-Operation Review Reported Information	Retirement	Retire System (A.4.4.3)
Evaluation Reported Information	Evaluation	Report Evaluation Results (A.5.1.7)

A.4.3.1.2 Description

This Activity identifies lessons learned and needs for software improvements, and outputs the Software Improvement Recommendations in accordance with the SPMPI. This is accomplished by using the Input Information. These recommendations shall include their impact on the quality of the software that is delivered. In addition, applicable tools, techniques, and methods for the implementation of these recommendations should be identified.

Further information related to this Activity can be found in IEEE Std 1219-1998 [B29].

A.4.3.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Software Improvement Recommendations	External	—
	Project Monitoring and Control	Manage the Project (A.1.3.2)
	Project Monitoring and Control	Identify SLCP Improvement Needs (A.1.3.3)
	Maintenance	Implement Problem Reporting Method (A.4.3.2)

A.4.3.2 Implement Problem Reporting Method**A.4.3.2.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Anomalies	External	—
	Creating Activity Group	Creating Activity
PR&RPI	Project Planning	Plan Project Management (A.1.2.7)
Environment Improvement Needs	Project Monitoring and Control	Identify SLCP Improvement Needs (A.1.3.3)
Software Improvement Recommendations	Maintenance	Identify Software Improvement Needs (A.4.3.1)
Evaluation Reported Information	Evaluation	Report Evaluation Results (A.5.1.7)
Controlled Item	Software Configuration Management	Perform Configuration Control (A.5.2.2)

A.4.3.2.2 Description

This Activity accepts Anomalies from any source and prepares a problem report. The problem report shall contain information as specified in the PR&RPI. Possible problem solutions can be suggested by the problem reporter. Problems can be resolved through corrections or enhancements (as defined in the PR&RPI). Corrections are documented in the Correction Problem Reported Information for further consideration. Enhancements may be documented in the Enhancement Problem Reported Information, and are possible candidates for new projects. A Report Log shall be maintained to ensure that all problems are tracked until they are resolved and the resolution has been approved.

This Activity shall also analyze the problem including the Controlled Item, the problem report, and the Report Log to make the following determinations:

- a) What the Anomalies are
- b) Source and cause of product or process problem
- c) Product(s) or process(es) presumed to contain the error, including documentation
- d) Problem severity
- e) Course of corrective action
- f) Impact on customer, cost, schedule, and risk

Anomalies that originate from outside the scope of this standard are noted as resolved within this Activity and are forwarded for appropriate action to the responsible authority.

Further information related to this Activity can be found in IEEE Std 1044-1993 [B19] and IEEE Std 1219-1998 [B29].

Prior to the distribution of the Problem Reported Information or the Report Log, Activity A.5.2.2, Perform Configuration Control, should be invoked.

A.4.3.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Out of Scope Anomalies	External	—
Report Log	Project Monitoring and Control	Collect and Analyze Metric Data (A.1.3.5)
	Maintenance	Reapply SLC (A.4.3.3)
Enhancement Problem Reported Information	Project Monitoring and Control	Collect and Analyze Metric Data (A.1.3.5)
	Concept Exploration	Identify Ideas or Needs (A.2.1.1)
	Maintenance	Reapply SLC (A.4.3.3)
Correction Problem Reported Information	Project Monitoring and Control	Collect and Analyze Metric Data (A.1.3.5)
	Maintenance	Reapply SLC (A.4.3.3)

A.4.3.3 Reapply SLC

A.4.3.3.1 Input Information

Input Information	Source	
	Activity Group	Activity
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
PR&RPI	Project Planning	Plan Project Management (A.1.2.7)
Enhancement Problem Reported Information	Maintenance	Implement Problem Reporting Method (A.4.3.2)
Correction Problem Reported Information	Maintenance	Implement Problem Reporting Method (A.4.3.2)
Report Log	Maintenance	Implement Problem Reporting Method (A.4.3.2)

A.4.3.3.2 Description

The information that is provided by the Correction Problem Reported Information, Enhancement Problem Reported Information, and current SPMPI shall result in the generation of Maintenance Recommendations. These Maintenance Recommendations will then enter the SLCP at the Concept Exploration Activity Group in order to improve the quality of the software system. When the estimate is greater than a predefined threshold of person-days, it may be appropriate to plan a separate project to complete the recommendations. In this case, the Maintenance Recommendations will go to External.

This Activity shall monitor the problem correction efforts that are performed by the responsible Activity Group, shall determine (according to the Enhancement and Correction Problem Reported Information) that the implementation of the solution by the responsible Activity Group has been completed, and shall then record the resolution of the problem in the Resolved Problem Reported Information. The Resolved Problem Reported Information shall be distributed as specified in the SPMPI. The Resolved Problem Reported Information should be made available to the Activity Group or to the external source that reported the problem.

The Report Log should be updated to reflect the corrective action taken.

A.4.3.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Maintenance Recommendations	External	—
	Project Initiation	Create SLCP (A.1.1.1)
	Concept Exploration	Identify Ideas or Needs (A.2.1.1)
Resolved Problem Reported Information	External	—
	Creating Activity Group	Creating Activity
	Project Monitoring and Control	Collect and Analyze Metric Data (A.1.3.5)
	Evaluation	Conduct Reviews (A.5.1.1)
		Create Test Data (A.5.1.5)
Report Evaluation Results (A.5.1.7)		
Updated Report Log	Project Monitoring and Control	Collect and Analyze Metric Data (A.1.3.5)

A.4.4 Retirement Activities

The Retirement Activity Group involves the removal of an existing system from its active support or use, either by ceasing its operation or support, or by replacing it with a new system or an upgraded version of the existing system.

Retirement Activities are

- a) A.4.4.1, Notify User
- b) A.4.4.2, Conduct Parallel Operations (If Applicable)
- c) A.4.4.3, Retire System

A.4.4.1 Notify User**A.4.4.1.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Retirement Planned Information	Project Planning	Plan Project Management (A.1.2.7)

A.4.4.1.2 Description

This Activity shall be the formal notification to any user (including both internal and external customers) of an operating software system that is to be removed from active support or use. This notification can take any of several forms, as appropriate for the individual environment. It is important that all users of the outgoing system are made aware that it will become unsupported. The actual dates of the removal of support are to be clearly specified and must allow time for current users to make whatever arrangements are necessary to respond to this notification. Included in the user notification should be one or more of the following:

- a) A description of the replacement system, including its date of availability
- b) A statement as to why the system is not being supported
- c) A description of possible other support

Prior to the distribution of the Official Notification, Activity A.5.3.1, Implement Documentation, shall be invoked.

A.4.4.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Official Notification	External	—
	Project Monitoring and Control	Retain Records (A.1.3.4)

A.4.4.2 Conduct Parallel Operations (If Applicable)

A.4.4.2.1 Input Information

Input Information	Source	
	Activity Group	Activity
Transition Planned Information (for the replacing system)	External	—
Retirement Planned Information	Project Planning	Plan Project Management (A.1.2.7)
Installed Software System	Installation	Accept Software in Operational Environment (A.4.1.3)

A.4.4.2.2 Description

If the outgoing system is being replaced by a new system, this Activity may apply. This Activity shall involve a period of dual operation that utilizes the retiring system for official results, while completing the preparation of the new system for formal operation. It is a period of user training on the new system and validation of the new system. The Retirement Planned Information, as well as the Transition Planned Information, can be used to provide information to conduct parallel operations for the replacing system.

Prior to the distribution of the Parallel Operations Log, the following Activities shall be invoked:

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

A.4.4.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Parallel Operations Log	Project Monitoring and Control	Retain Records (A.1.3.4)

A.4.4.3 Retire System

A.4.4.3.1 Input Information

Input Information	Source	
	Activity Group	Activity
Retirement Planned Information	Project Planning	Plan Project Management (A.1.2.7)

A.4.4.3.2 Description

This Activity shall consist of the actual removal and archiving of the retiring system from regular usage according to the Retirement Planned Information. It could be spread over a period of time and take the form of a phased removal, or it could be the simple removal of the entire system from the active software library. Prior to retirement, users shall be notified of the event. Any preparations for the use of a replacement system should have been completed. The Post-Operation Review Reported Information is generated at this time. The Retire System Activity shall be documented in an Archive Reported Information.

Prior to the distribution of the Post-Operation Review Reported Information or Archive Reported 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.4.4.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Archive Reported Information	External	—
Post-Operation Review Reported Information	External	—
	Project Monitoring and Control	Identify SLCP Improvement Needs (A.1.3.3)
		Retain Records (A.1.3.4)
Maintenance	Identify Software Improvement Needs (A.4.3.1)	

A.5 Integral Activity Groups

These are the Activities that are needed to successfully complete project Activities. These Activities are utilized to ensure the completion and quality of project functions.

A.5.1 Evaluation Activities

Evaluation Activities are those Activities performed during the SLCP that are designed to uncover defects in the product or the processes that are used to develop the product. This includes review and audit activities, traceability analysis, test preparation and execution, and the reporting of the results of all the Evaluation Activities.

Because exacting details of these Evaluation Activities can be found in other IEEE software standards, many of the traditional evaluation functions of software development are not specifically called out in this standard. They are placed into more generic groupings. For example, performing in-process reviews, process improvement reviews, etc., are grouped under the generic Activity of “Conduct Reviews.” This clause also discusses other topics such as traceability, testing, auditing, and evaluation reporting.

Each Evaluation Activity needs to be applied to each of its Instances in the SLCP. Consider, for example, an SLCP that has six phases, with a requirement for an in-process review at the end of each phase. The “Conduct Reviews” Activity would be mapped for each Instance of a completed phase. Figure A.1 depicts this situation.

Evaluation Activities are

- a) A.5.1.1, Conduct Reviews
- b) A.5.1.2, Create Traceability Matrix
- c) A.5.1.3, Conduct Audits
- d) A.5.1.4, Develop Test Procedures
- e) A.5.1.5, Create Test Data
- f) A.5.1.6, Execute Tests
- g) A.5.1.7, Report Evaluation Results

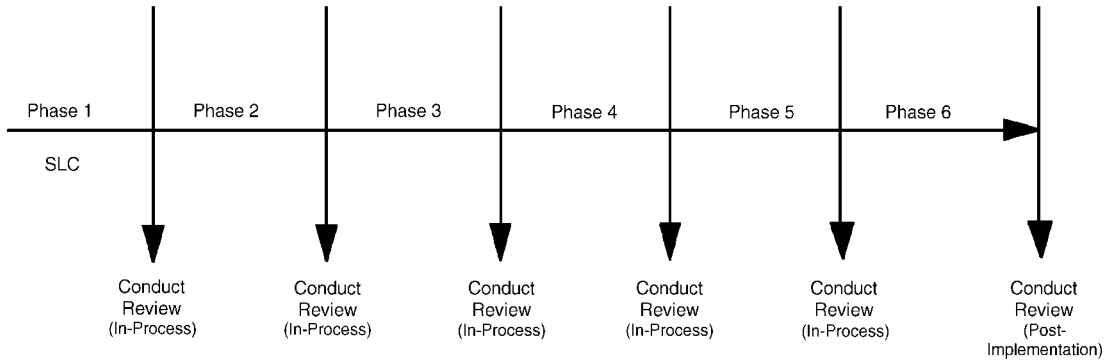


Figure A.1—Mapping reviews

A.5.1.1 Conduct Reviews

A.5.1.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
Review Standards and Guidelines	External	—
Item to be Reviewed	Creating Activity Group	Creating Activity
Evaluation Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Imported Software Requirements	Software Importation	Identify Imported Software Requirements (A.2.3.1)
Imported Software Documentation	Software Importation	Import Software (A.2.3.4)
Support Request Log	Operation and Support	Maintain Support Request Log (A.4.2.3)
Resolved Problem Reported Information	Maintenance	Reapply SLC (A.4.3.3)
Traceability Matrix	Evaluation	Create Traceability Matrix (A.5.1.2)
Audit Results Information	Evaluation	Conduct Audits (A.5.1.3)

A.5.1.1.2 Description

Reviews are to be performed throughout the life cycle. They fall into the following four broad categories:

- a) *In-Process Reviews*—This type of review shall be held to remove defects from software requirements, preliminary designs, detailed designs, code, and documentation. The reviews are sometimes referred to as peer reviews or technical reviews. They can be formal and structured, following a strict set of rules, roles, and procedures, or informal. They can utilize traceability analysis, walk-through, and inspection techniques. Using these reviews, the functional and performance requirements shall also be reviewed constantly throughout the life cycle to ensure they are being fully addressed in the work products of each phase. Traceability Analysis Reported Information and In-Process Review Results are produced as a result of these various reviews.
- b) *Management Reviews*—A review of the products and quality system shall be held at periodic intervals to determine if there is a need to implement corrective action and contingency plans, continue the effort, or cancel the effort. The progress of the effort is reviewed and measured against project milestones that are established in the SPMPI. Each review shall also reconfirm the need for each requirement and its system allocation. If there are changes, System Allocation Change Reported Information shall be generated. Since these reviews are usually held at or near the end of an SLCP phase, they are also referred to as phase-end reviews. Management Status Reported Information is produced after these reviews.
- c) *Process Improvement Reviews*—These reviews shall be held to evaluate metrics from the development effort in order to determine if processes need to be modified to prevent or reduce quality related problems in the future of the effort or in new efforts. The reviews can be part of the development schedule or they can be ad-hoc (i.e., driven by the results of one of the other types of reviews). Process Improvement Recommendations are generated as a result of this type of review.
- d) *Post-Implementation Review*—This review shall be held after the completion, or cancellation, of a development effort. It shall compare all planning information with the actual results, and shall use the resulting analysis to determine any improvements needed in such areas as resource utilization, return on investment, quality system, etc. Post-Implementation Review Reported Information is generated at this time.

Further information related to this Activity can be found in IEEE Std 730-1998 [B3], IEEE Std 1012-1998 [B13], IEEE Std 1028-1997 [B17], and IEEE Std 1059-1993 [B23].

Prior to the distribution of the Output Information or Archive Reported Information, Activity A.5.2.2, Perform Configuration Control, may be invoked.

A.5.1.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
In-Process Review Results	Evaluation	Report Evaluation Results (A.5.1.7)
Post-Implementation Review Reported Information	Evaluation	Report Evaluation Results (A.5.1.7)
Process Improvement Recommendations	Evaluation	Report Evaluation Results (A.5.1.7)
Management Status Reported Information	Evaluation	Report Evaluation Results (A.5.1.7)
Traceability Analysis Reported Information	Evaluation	Report Evaluation Results (A.5.1.7)
System Allocation Change Reported Information	Software Configuration Management	Perform Configuration Control (A.5.2.2)

A.5.1.2 Create Traceability Matrix

A.5.1.2.1 Input Information

Input Information	Source	
	Activity Group	Activity
Project-Specific Technical Requirements	External	—
Software Requirements	Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)

A.5.1.2.2 Description

A traceability matrix shall be developed showing, as a minimum, each requirement, the source of the requirement, the life cycle phases that are utilized by this project, and an associated requirement item identification. This shall allow the matrix to be reviewed during each in-process or management review in order to ensure that each requirement is addressed by the output products of each phase. The matrix will allow phase-to-phase and end-to-end review. A reviewer will be able to trace requirements through the development life cycle, forwards or backwards.

Further information related to this Activity can be found in IEEE Std 1012-1998 [B13] and IEEE Std 1059-1993 [B23].

Prior to the distribution of the Traceability Matrix, 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.5.1.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Traceability Matrix	Evaluation	Conduct Reviews (A.5.1.1)
	Software Configuration Management	Develop Configuration Identification (A.5.2.1)

A.5.1.3 Conduct Audits

A.5.1.3.1 Input Information

Input Information	Source	
	Activity Group	Activity
Evaluation Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Auditable Products and Processes	Creating Activity Group	Creating Activity

A.5.1.3.2 Description

Audits shall be performed by independent examiner(s) on software products or processes. The purpose is to assess the compliance of the products or processes with specification requirements, various SLCP plans, standards, the quality system, and any contractual or other agreed-upon requirements. The audits are performed in accordance with the Evaluation Planned Information. Audit results, items of noncompliance, and recommendations are reported in the Audit Results Information. Audits may be conducted in concert with in-process, management, and process improvement reviews.

Two specific types of audits, functional and physical configuration audits, can be performed.

Further information related to this Activity can be found in IEEE Std 730-1998 [B3], IEEE Std 1012-1998 [B13], IEEE Std 1028-1997 [B17], and IEEE Std 1059-1993 [B23].

A.5.1.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Audit Results Information	Creating Activity Group	Creating Activity
	Evaluation	Conduct Reviews (A.5.1.1)
		Report Evaluation Results (A.5.1.7)

A.5.1.4 Develop Test Procedures**A.5.1.4.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Evaluation Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
Software Requirements	Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Detailed Design	Design	Perform Detailed Design (A.3.2.4)

A.5.1.4.2 Description

Test Procedures for each level of testing (i.e., unit/module/component, integration, acceptance, regression, and system) shall be developed in order to refine the test approach from the Evaluation Planned Information to the item-specific test procedures used for test execution. The Test Procedures shall define what type of tests are to be conducted (i.e., white box, black box, destructive, noninvasive, etc.), what is to be tested, the data to be used in testing, the expected results, the test environment components, and the procedures to be followed in testing. Information from the Software Requirements, the Software Detailed Design, and the Evaluation Planned Information is used to generate the Test Procedures.

Further information related to this Activity can be found in IEEE Std 829-1998 [B6], IEEE Std 1008-1987 [B12], IEEE Std 1012-1998 [B13], and IEEE Std 1059-1993 [B23].

Prior to the distribution of the Test Procedures, 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.5.1.4.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Test Procedures	Evaluation	Create Test Data (A.5.1.5)
		Execute Tests (A.5.1.6)

A.5.1.5 Create Test Data

A.5.1.5.1 Input Information

Input Information	Source	
	Activity Group	Activity
Evaluation Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
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)
Software Detailed Design	Design	Perform Detailed Design (A.3.2.4)
Source Code (When Required)	Implementation	Create Executable Code (A.3.3.1)
Data Base (If Available)	Implementation	Create Executable Code (A.3.3.1)
Resolved Problem Reported Information	Maintenance	Reapply SLC (A.4.3.3)
Test Procedures	Evaluation	Develop Test Procedures (A.5.1.4)

A.5.1.5.2 Description

Using the Software Requirements, the Software Detailed Design, and the Source Code (when required), Test Data shall be generated for unit/module/component, integration, acceptance, regression, and system tests. In the case of regression testing, defect scenarios and data from previously failed tests and users in the field are also used and integrated into the regression test data. For each type of test, the Evaluation Planned Information describes the test environment. Test Procedures define the type of test data to be used. To support the testing effort, test Stubs and Drivers may be generated at this time for each item to be tested. The Stubs and Drivers allow the execution of software tests on an individual or integrated basis.

Further information can be found in IEEE Std 829-1998 [B6] and IEEE Std 1008-1987 [B12].

A.5.1.5.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Stubs and Drivers (If Available)	Implementation	Perform Integration (A.3.3.3)
	Evaluation	Execute Tests (A.5.1.6)
Test Data	Evaluation	Execute Tests (A.5.1.6)

A.5.1.6 Execute Tests**A.5.1.6.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Test Environment Components	External	—
Evaluation Planned Information	Project Planning	Plan Evaluations (A.1.2.1)
Imported Software	Software Importation	Import Software (A.2.3.4)
Executable Code	Implementation	Create Executable Code (A.3.3.1)
Integrated Software	Implementation	Perform Integration (A.3.3.3)
Test Procedures	Evaluation	Develop Test Procedures (A.5.1.4)
Test Data	Evaluation	Create Test Data (A.5.1.5)
Stubs and Drivers (If Available)	Evaluation	Create Test Data (A.5.1.5)

A.5.1.6.2 Description

This Activity shall configure the Test Environment Components as required by the Test Procedures. Tests shall be conducted on Executable Code units/modules/components, Integrated Software, and the full system using Test Data and the associated Test Procedures, in accordance with the Evaluation Planned Information.

This Activity could be iterative, with several Instances performed during the life of the software. Not all Input Information and Output Information are required for a given Iteration. The presence of any Input Information is sufficient as an entry criterion, and the creation of any Output Information is sufficient as an exit criterion.

Based on a comparison of the actual results with the expected results, and according to the pass-fail criteria in the Evaluation Planned Information, a pass-fail determination shall be made and recorded in a test log. Each Anomaly that occurs during test execution that requires further investigation shall be reported. The impact on the validity of the test should also be noted.

Test Summary Reported Information shall summarize the results of a test based on its Test Procedures and test log. Tested Software is that software that has successfully passed all tests at the appropriate level and has met the specified criteria and requirements. Tested Software may then be further integrated with other software or sent for installation.

Further information related to this Activity can be found in IEEE Std 829-1998 [B6] and IEEE Std 1008-1987 [B12].

Prior to the distribution of the Test Summary Reported 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.5.1.6.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Test Summary Reported Information	External	—
	Evaluation	Report Evaluation Results (A.5.1.7)
Tested Software	Implementation	Perform Integration (A.3.3.3)
	Installation	Distribute Software (A.4.1.1)
Anomalies	Maintenance	Implement Problem Reporting Methods (A.4.3.2)
	Evaluation	Report Evaluation Results (A.5.1.7)

A.5.1.7 Report Evaluation Results

A.5.1.7.1 Input Information

Input Information	Source	
	Activity Group	Activity
Basis or Bases for Evaluation	External	—
	Creating Activity Group	Creating Activity
Anomalies	External	—
	Creating Activity Group	Creating Activity
	Evaluation	Execute Tests (A.5.1.6)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Resolved Problem Reported Information	Maintenance	Reapply SLC (A.4.3.3)
In-Process Review Results	Evaluation	Conduct Reviews (A.5.1.1)
Post-Implementation Review Reported Information	Evaluation	Conduct Reviews (A.5.1.1)
Process Improvement Recommendations	Evaluation	Conduct Reviews (A.5.1.1)
Management Status Reported Information	Evaluation	Conduct Reviews (A.5.1.1)
Traceability Analysis Reported Information	Evaluation	Conduct Reviews (A.5.1.1)
Audit Results Information	Evaluation	Conduct Audits (A.5.1.3)
Test Summary Reported Information	Evaluation	Execute Tests (A.5.1.6)

A.5.1.7.2 Description

This Activity shall gather the information, recommendations, and data supplied by the Input Information, and shall formulate the results as specified in the SPMPI. The results shall be provided in the Evaluation Reported Information. Anomalies that are identified during the performance of these tasks shall be reported.

Prior to the distribution of the Evaluation Reported Information, Activity A.5.1.1, Conduct Reviews, may be invoked.

A.5.1.7.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Evaluation Reported Information	Creating Activity Group	Creating Activity
	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)
		Collect and Analyze Metric Data (A.1.3.5)
	Installation	Accept Software in Operational Environment (A.4.1.3)
Maintenance	Identify Software Improvement Needs (A.4.3.1)	
	Implement Problem Reporting Method (A.4.3.2)	
Anomalies	Maintenance	Implement Problem Reporting Method (A.4.3.2)

A.5.2 Software Configuration Management Activities

Software Configuration Management identifies the items in a software development project and provides both for control of the identified items and for the generation of Status Reported Information for management visibility and accountability throughout the SLC. Items to be managed are those that are defined in the SCMPI. Examples to be considered for inclusion in the SCMPI are code, documentation, plans, and specifications. Configuration audits, if required by the project, should be addressed in the Evaluation Activity Group. The Software Configuration Management approach for a given project should be compatible with the configuration management approach that is being used on associated systems.

Configuration Activities are

- a) A.5.2.1, Develop Configuration Identification
- b) A.5.2.2, Perform Configuration Control
- c) A.5.2.3, Perform Status Accounting

A.5.2.1 Develop Configuration Identification**A.5.2.1.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Deliverable List	External	—
SCMPI	Project Planning	Plan Configuration Management (A.1.2.2)
SPMPI	Project Planning	Plan the Project (A.1.2.7)
Traceability Matrix	Evaluation	Create Traceability Matrix (A.5.1.2)

A.5.2.1.2 Description

This Activity shall define the software Configuration Identification including project baseline definition, titling, labeling, and numbering to reflect the structure of the product for tracking. The SCMPI identifies those Configuration Items that are to be addressed by the Configuration Identification. The identification shall support the software throughout the SLC, and shall be documented in the SCMPI. The Configuration Identification shall also define the documentation that is required in order to record the functional and physical characteristics of each Configuration Item.

A series of baselines, based on the Traceability Matrix, shall be established as the product moves from the initial idea to the maintenance phase as required by the SPMPI.

Further information related to this Activity can be found in IEEE Std 828-1998 [B5] and IEEE Std 1028-1997 [B17].

Prior to the distribution of the Configuration Identification, 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.5.2.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Configuration Identification	Software Configuration Management	Perform Configuration Control (A.5.2.2)
		Perform Status Accounting (A.5.2.3)

A.5.2.2 Perform Configuration Control

A.5.2.2.1 Input Information

Input Information	Source	
	Activity Group	Activity
Items to be Controlled	Creating Activity Group	Creating Activity
Proposed Change	Creating Activity Group	Creating Activity
SCMPI	Project Planning	Plan Configuration Management (A.1.2.2)
System Allocation Change Reported Information	Evaluation	Conduct Reviews (A.5.1.1)
Configuration Identification	Software Configuration Management	Develop Configuration Identification (A.5.2.1)

A.5.2.2.2 Description

This Activity controls the configuration of products according to the SCMPI and the Configuration Identification. Changes to controlled products shall be tracked to ensure that the configuration of the product is known at all times. All items specified in the SCMPI are subject to this change management discipline.

Changes to Controlled Items shall be allowed only with the approval of the responsible authority. This can result in the establishment of a formal software configuration control board. Controlled Items shall be maintained in a software library.

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

A.5.2.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Controlled Item	Creating Activity Group	Creating Activity
	Maintenance	Implement Problem Reporting Method (A.4.3.2)
Change Status	Software Configuration Management	Perform Status Accounting (A.5.2.3)

A.5.2.3 Perform Status Accounting**A.5.2.3.1 Input Information**

Input Information	Source	
	Activity Group	Activity
SCMPI	Project Planning	Plan Configuration Management (A.1.2.2)
Configuration Identification	Software Configuration Management	Develop Configuration Identification (A.5.2.1)
Change Status	Software Configuration Management	Perform Configuration Control (A.5.2.2)

A.5.2.3.2 Description

This Activity shall receive Configuration Identification and Change Status and shall create and update the Status Reported Information to reflect the status and history of Controlled Items. The history of changes to each Controlled Item shall be maintained throughout the SLC as required by the SCMPI.

Status Reported Information may include such data as the number of changes to date for the project, the number of releases, and the latest version and revision identifiers.

Each baseline shall be established as required by the SCMPI, and all subsequent changes shall be tracked relative to it.

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 Status Reported Information, the following Activities shall be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control

A.5.2.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Controlled Item	Creating Activity Group	Creating Activity
Status Reported Information	External	—
	Project Monitoring and Control	Manage the Project (A.1.3.2)

A.5.3 Documentation Development Activities

The Documentation Development Activity Group for software development and usage is the set of Activities that plan, design, implement, edit, produce, distribute, and maintain those documents that are needed by developers and users. The purpose of the Documentation Development Activity Group is to provide timely software documentation to those who need it, based on Input Information from the invoking Activity Groups.

This Activity Group covers both product-oriented and procedure-oriented documentation for internal and external users. Examples of internal users include those who plan, design, implement, or test software. External users can include those who install, operate, apply, or maintain the software.

The Documentation Development Activity Group occurs over various phases of the SLCP, depending on the individual document and the timing of its development. Typically, there will be multiple documents, each at a different stage of development.

Documentation Activities are

- a) A.5.3.1, Implement Documentation
- b) A.5.3.2, Produce and Distribute Documentation

A.5.3.1 Implement Documentation

A.5.3.1.1 Input Information

Input Information	Source	
	Activity group	Activity
Input Information for Document	Creating Activity Group	Creating Activity
Documentation Planned Information	Project Planning	Plan Documentation (A.1.2.5)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Imported Software Documentation	Software Importation	Import Software (A.2.3.4)

A.5.3.1.2 Description

This Activity includes the design, preparation, and maintenance of documentation. Those documents that are identified in the Documentation Planned Information shall be formulated in terms of audience, approach, content, structure, and graphics. Arrangements may be made with word or text processing and graphics facilities for their support.

Input Information shall be used to produce the document, including any related graphics.

Following a documentation review, any changes shall be incorporated to produce a technically correct document. Organizational format, style, and production rules shall be applied to produce a final document.

Prior to the distribution of the Document, the following Activities should be invoked:

- a) A.5.1.1, Conduct Reviews
- b) A.5.2.2, Perform Configuration Control

A.5.3.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Document	Documentation Development	Produce and Distribute Documentation (A.5.3.2)

A.5.3.2 Produce and Distribute Documentation**A.5.3.2.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Documentation Planned Information	Project Planning	Plan Documentation (A.1.2.5)
Document	Documentation Development	Implement Documentation (A.5.3.1)

A.5.3.2.2 Description

This Activity shall provide the intended audience with the needed information that is collected in the document, as specified in the Documentation Planned Information. Document production and distribution can involve electronic file management, paper document reproduction and distribution, or other media handling techniques.

A.5.3.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Published Document	External	—
	Creating Activity Group	Creating Activity
	Project Monitoring and Control	Retain Records (A.1.3.4)

A.5.4 Training Activities

The development of quality software products is largely dependent upon knowledgeable and skilled people. These include the developer's technical staff and management. Customer personnel may also need to be trained to install, operate, and maintain the software. It is essential that the Training Planned Information be completed early in the SLC, prior to the time when personnel would be expected to apply required expertise to the project. Plans for customer training should be prepared and reviewed with the customer.

Training Activities are

- a) A.5.4.1, Develop Training Materials
- b) A.5.4.2, Validate the Training Program
- c) A.5.4.3, Implement the Training Program

A.5.4.1 Develop Training Materials

A.5.4.1.1 Input Information

Input Information	Source	
	Activity Group	Activity
Applicable Information	External	—
	Creating Activity Group	Creating Activity
Training Planned Information	Project Planning	Plan Training (A.1.2.6)
SPMPI	Project Planning	Plan Project Management (A.1.2.7)
Imported Software Documentation	Software Importation	Import Software (A.2.3.4)
Software Detailed Design	Design	Perform Detailed Design (A.3.2.4)

A.5.4.1.2 Description

This Activity shall consist of the identification and review of all available materials and Input Information that is pertinent to the training objectives. Included in the Develop Training Materials Activity shall be the development of the substance of the training, training manual, and materials that are to be used in presenting the training, such as outlines, text, exercises, case studies, visuals, and models. Instructors shall review the training materials and develop the actual presentations that are to be based on the developed materials. Instructors are expected to be competent in up-to-date educational methods and effective presentation techniques.

Prior to the distribution of the Training Manual and Training Materials, 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.5.4.1.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Training Manual	Training	Validate the Training Program (A.5.4.2)
Training Materials	Training	Validate the Training Program (A.5.4.2)
Prepared Presentations	Training	Validate the Training Program (A.5.4.2)

A.5.4.2 Validate the Training Program

A.5.4.2.1 Input Information

Input Information	Source	
	Activity Group	Activity
Training Planned Information	Project Planning	Plan Training (A.1.2.6)
Training Manual	Training	Develop Training Materials (A.5.4.1)
Training Materials	Training	Develop Training Materials (A.5.4.1)
Prepared Presentations	Training	Develop Training Materials (A.5.4.1)

A.5.4.2.2 Description

This Activity shall consist of competent instructors who present the training to a class of evaluators using the preliminary training manual and materials. The evaluators shall assess the training presentation and materials in detail. The purpose is to evaluate the effectiveness of the delivery and to validate the material presented. Lessons learned in the test of the training program shall be incorporated into the material prior to a general offering. All training manuals and materials shall be evaluated and, if necessary, updated at this time.

Prior to the distribution of the Updated Training Manuals and Materials, 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 be also invoked.

A.5.4.2.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Training Feedback	Project Planning	Plan Training (A.1.2.6)
Updated Training Manual	Training	Implement the Training Program (A.5.4.3)
Updated Training Materials	Training	Implement the Training Program (A.5.4.3)

A.5.4.3 Implement the Training Program**A.5.4.3.1 Input Information**

Input Information	Source	
	Activity Group	Activity
Staff Participants	External	—
Students	External	—
Training Planned Information	Project Planning	Plan Training (A.1.2.6)
Updated Training Manual	Training	Validate the Training Program (A.5.4.2)
Updated Training Materials	Training	Validate the Training Program (A.5.4.2)

A.5.4.3.2 Description

This Activity shall ensure the provision of all necessary materials, the arrangement of the locations and facilities for training, and the delivery of the training. Included in this Activity shall be the enrolling of students and the monitoring of the course effectiveness.

Lessons learned and the information that is needed for updating the materials for the next training cycle shall be fed back into the Training Activity Group.

A.5.4.3.3 Output Information

Output Information	Destination	
	Activity Group	Activity
Updated Skills Inventory	External	—
Trained Personnel	Creating Activity Group	Creating Activity
Training Feedback	Project Planning	Plan Training (A.1.2.6)

Annex B

(informative)

Mapping example

This annex provides an example of mapping the Activities of this standard onto a selected SLCM.

The purpose of this example is to show the mapping process without constraining the reader to any specific methodologies or tools.

For purposes of illustration, an oversimplified, four-phase waterfall SLCM has been selected. It is understood that any SLCM could be interactive and iterative in the real world, which would cause the expansion of the mapped SLC to reflect multiple Instances of Activities.

At each step, constraints should be identified that impact the development of the SLCP. In the example that follows, common constraints require consideration while verifying information flows, mapping information into deliverable documents, and adding actual dates and times to the SLCP.

Once the SLCP is in place, the experience of the project might dictate necessary modifications to the SLCP. In this case, some or all of steps 1–8 may need to be repeated.

B.1 Step 1: Select SLCM

This first step, as specified by Clause 5 of this standard, is to identify the SLCM to which the Activities will be mapped. This step could mean that the whole process of locating, evaluating, selecting, and acquiring an SLCM shall be performed (see 5.1). For this example, a four-phase waterfall SLCM is selected. During this step, the process architect reviews the SLCM to ensure that it is appropriate for the specific project.

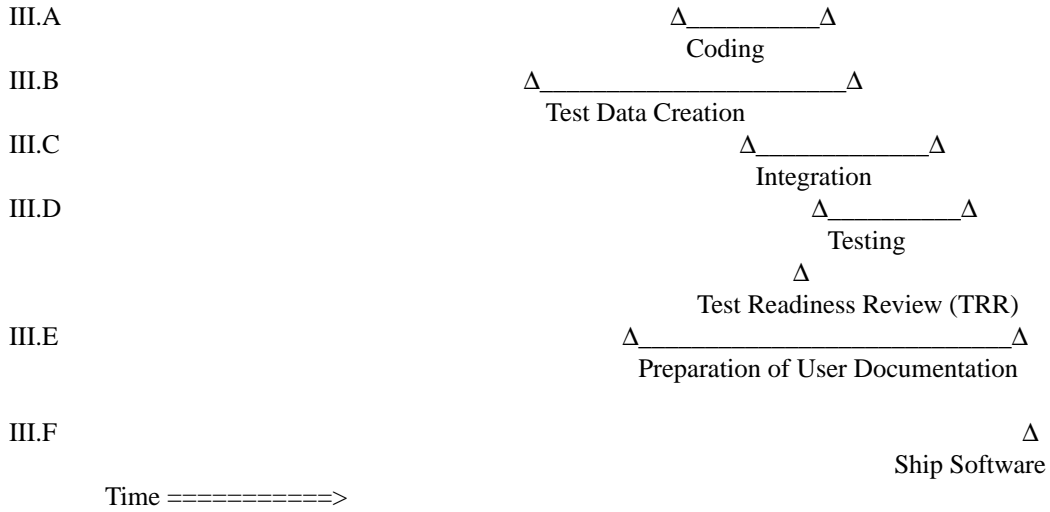
I. Project Initiation/Concept Development

- I.A Δ
 Project Kick-off
- I.B Δ_____Δ
 Project Plan
- I.C Δ_____Δ
 System Level Concepts
- I.D Δ_____Δ
 User (System) Requirements
- I.E Δ_____Δ
 System Design (Hardware/Software Allocation)

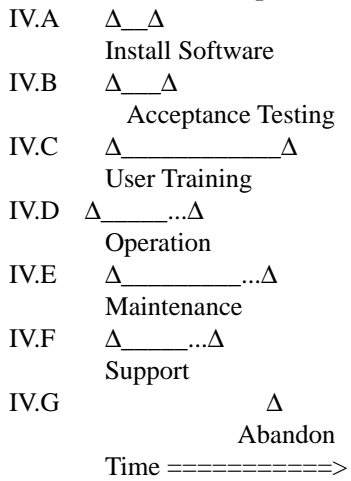
II. Definition and Design

- II.A Δ_____Δ
 Define Software Requirements
- II.B Δ_____Δ
 Define Software Design
- II.C Δ_____Δ
 Design User Interface
- II.D Δ_____Δ
 Plan Testing

III. System Development



IV. Installation and Operation



B.2 Step 2: Compare Activities to SLCM

Having selected an SLCM, perform a detailed mapping of the Activities of this standard against the SLCM. This involves the matching of the Activities against the requirements of the SLCM. This step provides a checklist to ensure that all Activities are mapped and that all SLCM requirements are covered by one or more Activities. Activities that are not mapped will be noted in step 3 (Clause B.3).

NOTE—The Define Software Requirements (II.A) sub-phase, within the Definition and Design phase, will be used to illustrate mapping details.

For this example, the following matrix can be generated:

	IA	IB	IC	ID	IE	II.A	II.B	II.C	II.D	III.A	III.B	III.C	III.D	III.E	III.F	IVA	IV.B	IV.C	IV.D	IV.E	IV.F	IV.G	
A.1.1.1	X																						
A.1.1.2		X																					
A.1.1.3		X																					
A.1.1.4		X							X														
A.1.2.1		X																					
A.1.2.2		X																					
A.1.2.3																							
A.1.2.4		X																					
A.1.2.5		X																					
A.1.2.6		X																					
A.1.2.7		X																					
A.1.2.8		X																					
A.1.3.1		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
A.1.3.2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
A.1.3.3		X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	
A.1.3.4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
A.1.3.5			X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	
A.2.1.1	X		X	X	X																		
A.2.1.2	X		X	X	X																		
A.2.1.3			X	X	X																		
A.2.1.4			X	X	X																		
A.2.2.1				X	X																		
A.2.2.2				X	X																		
A.2.2.3					X																		
A.2.3.1				X		X																	
A.2.3.2																							
A.2.3.3																							
A.2.3.4																							
A.3.1.1						X																	
A.3.1.2						X																	
A.3.1.3						X																	
A.3.2.1							X																
A.3.2.2																							
A.3.2.3							X	X															
A.3.2.4							X																
A.3.3.1										X													
A.3.3.2														X									

	IA	IB	IC	ID	IE	II.A	II.B	II.C	II.D	III.A	III.B	III.C	III.D	III.E	III.F	IV.A	IV.B	IV.C	IV.D	IV.E	IV.F	IV.G
A.3.3.3												X										
A.4.1.1															X							
A.4.1.2																X						
A.4.1.3																	X					
A.4.2.1																			X			
A.4.2.2																				X	X	
A.4.2.3																					X	
A.4.3.1																			X	X	X	
A.4.3.2		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
A.4.3.3																				X		
A.4.4.1																			X		X	X
A.4.4.2																			X		X	X
A.4.4.3																						X
A.5.1.1		X	X	X	X	X	X							X								
A.5.1.2					X	X	X	X	X	X	X											
A.5.1.3					X										X						X	
A.5.1.4									X	X							X					
A.5.1.5									X	X								X				
A.5.1.6													X				X					
A.5.1.7					X				X	X		X		X		X					X	
A.5.2.1		X			X						X											
A.5.2.2		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
A.5.2.3		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
A.5.3.1		X	X	X	X	X	X	X						X								
A.5.3.2														X								
A.5.4.1																						
A.5.4.2														X				X				
A.5.4.3														X				X				

B.3 Step 3: Develop and justify List of Activities Not Used

The following Activities are not used:

- a) A.1.2.3—No system transition is needed or planned for.
- b) A.2.3.2, A.2.3.3, and A.2.3.4—No imported software will be used.
- c) A.3.2.2—No data base design will be needed. The existing structure will be utilized.

B.4 Step 4: List Activities and Invocations

List each Activity and Invocation for all the pieces of the SLCM. This listing will be used in the next step to develop the initial ordering. In the example, the Activities and Invocations that are contained in sub-phase II.A are listed in numerical order.

The initial listing would look like this:

II. Definition and Design

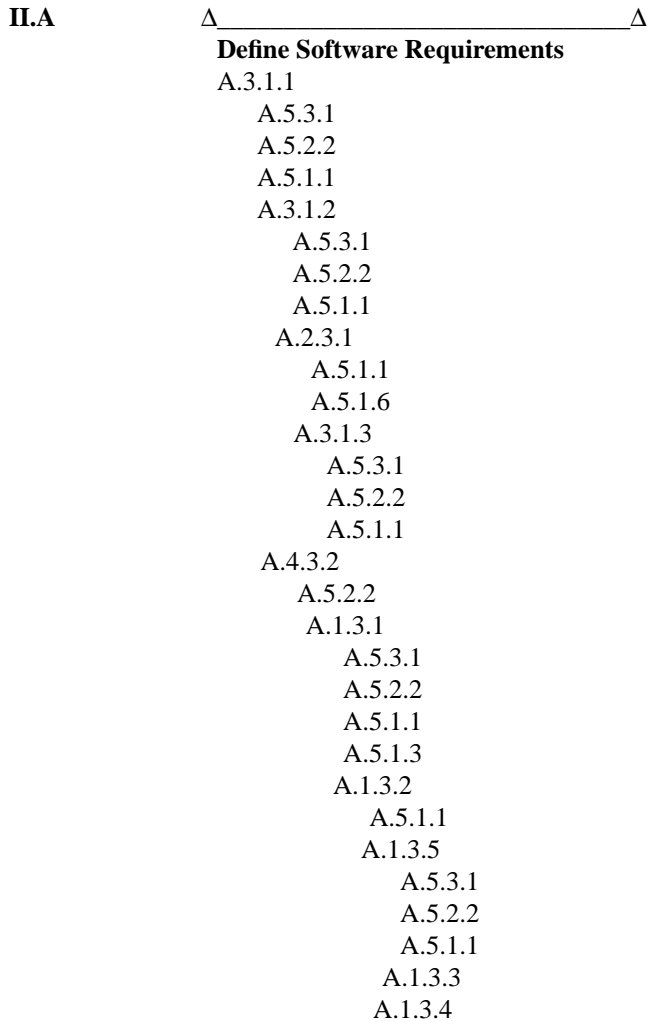
II.A	Δ		Δ
		Define Software Requirements	
		A.1.3.1	
		A.5.1.1	
		A.5.1.3	
		A.5.2.2	
		A.5.3.1	
		A.1.3.2	
		A.5.1.1	
		A.1.3.3	
		A.1.3.4	
		A.1.3.5	
		A.5.1.1	
		A.5.2.2	
		A.5.3.1	
		A.2.3.1	
		A.5.1.1	
		A.5.1.6	
		A.3.1.1	
		A.5.1.1	
		A.5.2.2	
		A.5.3.1	
		A.3.1.2	
		A.5.1.1	
		A.5.2.2	
		A.5.3.1	
		A.3.1.3	
		A.5.1.1	
		A.5.2.2	
		A.5.3.1	
		A.4.3.2	
		A.5.2.2	

B.5 Step 5: Place Activities in executable sequence

This step further organizes the Activities within the SLCM sub-phases to refine executable order relationships.

The result, for sub-phase II.A, would look like this:

II. Definition and Design



B.6 Step 6: Verify information flow

The Input and Output Information tables in this standard specify the information that is to be used and generated by each Activity. This step verifies that the information flow into and out of the Activities will support the relative order into which they have been mapped. While it is unlikely that this will cause a major rearrangement or modification of the mapping from step 5 (Clause B.5), it is a necessary check to be sure that all information will be available to the Activities that need it, when they need it.

A check of the information flow for this example could result in remapping. In this example, A.2.3.1, Identify Imported Software Requirements, was moved after A.3.1.3, Prioritize and Integrate Software Requirements, as A.2.3.1 needs the output of A.3.1.3. Additionally, A.1.3.5, Collect and Analyze Metric Data, was moved before A.1.3.1, Manage Risks, as A.1.3.1 uses the output of A.1.3.5.

II. Definition and Design

II.A	Δ	Δ
	Define Software Requirements	
	A.3.1.1	
	A.5.3.1	
	A.5.2.2	
	A.5.1.1	
	A.3.1.2	
	A.5.3.1	
	A.5.2.2	
	A.5.1.1	
	A.2.3.1	
	A.5.1.1	
	A.5.1.6	
	A.3.1.3	
	A.5.3.1	
	A.5.2.2	
	A.5.1.1	
	A.4.3.2	
	A.5.2.2	
	A.1.3.1	
	A.5.3.1	
	A.5.2.2	
	A.5.1.1	
	A.5.1.3	
	A.1.3.2	
	A.5.1.1	
	A.1.3.5	
	A.5.3.1	
	A.5.2.2	
	A.5.1.1	
	A.1.3.3	
	A.1.3.4	

B.7 Step 7: Map information into deliverable documents

Each SLCM requires, and defines the formatted content of, its own set of output products. These products are, for the most part, the specific documents that the SLCM delivers. Note that the term “document” does not imply any particular medium. This step compares the Output Information that is generated by each Activity with the SLCM-required document(s) into which it must go.

Once again, the order of the mapping, this time from step 6 (Clause B.6), might have to be modified. If a particular document, as specified by the selected SLCM, is to be created at a particular point in the development schedule, all of the Activities that contribute information to be recorded in that document must have had an opportunity to generate it.

For this project, the mapping of deliverable documents will occur as follows:

- a) A.3.1.1, A.3.1.2, A.3.1.3, A.1.4.1, and A.5.1.2 will be delivered as a Software Requirements Specification.
- b) A.1.3.1, A.1.3.2, A.1.3.5, and A.1.3.3 will be delivered as a Project Management Report.
- c) A.4.3.2 will be delivered as Problem Reports.
- d) No other documentation will be generated.

The information was checked in the mapping of sub-phase II.A to ensure that all required information from it would be available for the SLCM-defined documentation that was detailed in the previous paragraph. No changes to sub-phase II.A were necessary.

B.8 Step 8: Add OPAs

OPAs are now added to the fully mapped Activities and deliverable documents at the appropriate points in the SLCP. Adding the OPAs expands the SLCP beyond the minimum set of Activities that are specified in the standard, and produces a fully robust SLCP for the development project. For this example, the project will use organizational standards for the content and format of the Software Requirements Specification and Project Management Report. The methodology and tools that will be used to create the Software Requirements Specification are also decided.

B.9 Step 9: Add project planning information

Throughout these steps, the project manager could be adding planning specifics to the evolving SLCP. Additions are normally identified during the continuing audit and review process.

Annex C

(informative)

Information mapping template

This annex provides an information mapping template that is designed to assist project managers in identifying project-critical deliverables and ensuring their completion as needed.

This template can be used to assist in the project-specific mapping of information into the required project documentation.

Table C.1—Information mapping template

Activity Group or Activity name	Clause	Output Information	Mapped deliverables
PROJECT MANAGEMENT ACTIVITIES	A.1.1		
Project Initiation Activities	A.1.1		
Create SLCP	A.1.1.1	SLCP List of Activities Not Used	
Perform Estimation	A.1.1.2	Project Estimates Estimation Assumptions	
Allocate Project Resources	A.1.1.3	Resource Allocations	
Define Metrics	A.1.1.4	Defined Metrics Collection and Analysis Methods	
Project Planning Activities	A.1.2		
Plan Evaluations	A.1.2.1	Evaluation Planned Information	
Plan Configuration Management	A.1.2.2	SCMPI	
Plan System Transition (If Applicable)	A.1.2.3	Transition Planned Information Transition Impact Statement	
Plan Installation	A.1.2.4	Software Installation Planned Information	
Plan Documentation	A.1.2.5	Documentation Planned Information	
Plan Training	A.1.2.6	Training Planned Information	
Plan Project Management	A.1.2.7	PR&RPI Retirement Planned Information SPMPI Support Planned Information	
Plan Integration	A.1.2.8	Integration Planned Information	
Project Monitoring and Control Activities	A.1.3		
Manage Risks	A.1.3.1	Risk Management Reported Information	
Manage the Project	A.1.3.2	Project Management Reported Information Anomalies	
Identify SLCP Improvement Needs	A.1.3.3	Environment Improvement Needs	
Retain Records	A.1.3.4	Historical Project Records	
Collect and Analyze Metric Data	A.1.3.5	Analysis Reported Information	

Table C.1—Information mapping template (continued)

Activity Group or Activity name	Clause	Output Information	Mapped deliverables
PRE-DEVELOPMENT ACTIVITY GROUPS	A.2		
Concept Exploration Activities	A.2.1		
Identify Ideas or Needs	A.2.1.1	Preliminary Statement of Need	
Formulate Potential Approaches	A.2.1.2	Constraints and Benefits	
		Potential Approaches	
Conduct Feasibility Studies	A.2.1.3	Recommendations	
Refine and Finalize the Idea or Need	A.2.1.4	Statement of Need	
System Allocation Activities	A.2.2		
Analyze Functions	A.2.2.1	Functional Description of the System	
Develop System Architecture	A.2.2.2	System Architecture	
Decompose System Requirements	A.2.2.3	System Functional Hardware Requirements (If Applicable)	
		System Functional Software Requirements	
		System Interface Requirements (If Applicable)	
Software Importation Activities	A.2.3		
Identify Imported Software Requirements	A.2.3.1	Imported Software Requirements	
Evaluate Software Import Sources (If Applicable)	A.2.3.2	Selected Software Import Source	
		Candidate Software Import Methods	
Define Software Import Method (If Applicable)	A.2.3.3	Selected Software Import Method	
Import Software (If Applicable)	A.2.3.4	Imported Software	
		Imported Software Documentation	
DEVELOPMENT ACTIVITY GROUPS	A.3		
Requirements Activities	A.3.1		
Define and Develop Software Requirements	A.3.1.1	Preliminary Software Requirements	
		Installation Requirements	
Define Interface Requirements	A.3.1.2	Software Interface Requirements	
Prioritize and Integrate Software Requirements	A.3.1.3	Software Requirements	
Design	A.3.2		
Perform Architectural Design	A.3.2.1	Software Architectural Design	
Design Data Base (If Applicable)	A.3.2.2	Data Base Design	
Design Interfaces	A.3.2.3	Interface Design	
Perform Detailed Design	A.3.2.4	Software Detailed Design	
Implementation Activities	A.3.3		
Create Executable Code	A.3.3.1	Data Base (If Applicable)	
		Source Code	
		Executable Code	
		Source Code (If Required)	
Create Operating Documentation	A.3.3.2	Operating Documentation	
Perform Integration	A.3.3.3	Integrated Software	

Table C.1—Information mapping template (continued)

Activity Group or Activity name	Clause	Output Information	Mapped deliverables
POST-DEVELOPMENT ACTIVITY GROUPS	A.4		
Installation Activities	A.4.1		
Distribute Software	A.4.1.1	Packaged Installation Planned Information	
		Packaged Software	
		Packaged Operating Documentation	
Install Software	A.4.1.2	Installation Reported Information	
		Installed Software	
Accept Software in Operational Environment	A.4.1.3	Customer Acceptance	
		Installed Software System	
Operation and Support Activities	A.4.2		
Operate the System	A.4.2.1	Operation Logs	
		Anomalies	
Provide Technical Assistance and Consulting	A.4.2.2	Support Response	
Maintain Support Request Log	A.4.2.3	Anomalies	
		Support Request Log	
Maintenance Activities	A.4.3		
Identify Software Improvement Needs	A.4.3.1	Software Improvement Recommendations	
Implement Problem Reporting Method	A.4.3.2	Out of Scope Anomalies	
		Report Log	
		Enhancement Problem Reported Information	
		Correction Problem Reported Information	
Reapply SLC	A.4.3.3	Maintenance Recommendations	
		Resolved Problem Reported Information	
		Updated Report Log	
Retirement Activities	A.4.4		
Notify User	A.4.4.1	Official Notification	
Conduct Parallel Operations (If Applicable)	A.4.4.2	Parallel Operations Log	
Retire System	A.4.4.3	Archive Reported Information	
		Post-Operation Review Reported Information	

Table C.1—Information mapping template (continued)

Activity Group or Activity name	Clause	Output Information	Mapped deliverables
INTEGRAL ACTIVITY GROUPS	A.5		
Evaluation Activities	A.5.1		
Conduct Reviews	A.5.1.1	In-Process Review Results	
		Post-Implementation Review Reported Information	
		Process Improvement Recommendations	
		Management Status Reported Information	
		System Allocation Change Reported Information	
Create Traceability Matrix	A.5.1.2	Traceability Matrix	
Conduct Audits	A.5.1.3	Audit Results Information	
Develop Test Procedures	A.5.1.4	Test Procedures	
Create Test Data	A.5.1.5	Stubs and Drivers (If Applicable)	
		Test Data	
Execute Tests	A.5.1.6	Test Summary Reported Information	
		Tested Software	
		Anomalies	
Report Evaluation Results	A.5.1.7	Evaluation Reported Information	
		Anomalies	
Software Configuration Management Activities	A.5.2		
Develop Configuration Identification	A.5.2.1	Configuration Identification	
Perform Configuration Control	A.5.2.2	Change Status	
		Controlled Item	
Perform Status Accounting	A.5.2.3	Status Reported Information	
Documentation Development Activities	A.5.3		
Implement Documentation	A.5.3.1	Document	
Produce and Distribute Documentation	A.5.3.2	Published Document	
Training Activities	A.5.4		
Develop Training Materials	A.5.4.1	Training Manual	
		Training Materials	
		Prepared Presentations	
Validate the Training Program	A.5.4.2	Training Feedback	
		Updated Training Manual	
		Updated Training Materials	
Implement the Training Program	A.5.4.3	Updated Skills Inventory	
		Trained Personnel	
		Training Feedback	

Annex D

(informative)

Bibliography

This Informative Annex provides a listing of potentially helpful software engineering standards. The standards listed below, and any subsequent standards should be consulted when using this document. Compliance with this standard, however, neither requires nor implies compliance with the listed standards.

[B1] EIA/IEEE J-STD-016-1995, EIA/IEEE Interim Standard for Information Technology—Software Life Cycle Processes—Software Development: Acquirer–Supplier Agreement.⁵

[B2] IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.⁶

[B3] IEEE Std 730-1998, IEEE Standard for Software Quality Assurance Plans.

[B4] IEEE Std 730.1-1995, IEEE Guide for Software Quality Assurance Planning.

[B5] IEEE Std 828-1998, IEEE Standard for Software Configuration Management Plans.

[B6] IEEE Std 829-1998, IEEE Standard for Software Test Documentation.

[B7] IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

[B8] IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Procedure Reliable Software.

[B9] IEEE Std 982.2-1988, IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Procedure Reliable Software.

[B10] IEEE Std 990-1987 (Reaff 1992), IEEE Recommended Practice for Ada as a Program Design Language.

[B11] IEEE Std 1002-1987 (Reaff 1992), IEEE Standard Taxonomy for Software Engineering Standards.

[B12] IEEE Std 1008-1987 (Reaff 1993), IEEE Standard for Software Unit Testing.

[B13] IEEE Std 1012-1998, IEEE Standard for Software Verification and Validation.

[B14] IEEE Std 1012a-1998, IEEE Standard for Software Verification and Validation: Content Map to IEEE/EIA 12207.1-1997.

[B15] IEEE Std 1016-1998, IEEE Recommended Practice for Software Design Descriptions.

[B16] IEEE Std 1016.1-1993, IEEE Guide to Software Design Descriptions.

[B17] IEEE Std 1028-1997, IEEE Standard for Software Reviews.

[B18] IEEE Std 1042-1987 (Reaff 1993), IEEE Guide to Software Configuration Management.

⁵This document is available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (<http://www.standards.ieee.org/>).

⁶IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (<http://www.standards.ieee.org/>).

- [B19] IEEE Std 1044-1993, IEEE Standard for Classification of Software Anomalies.
- [B20] IEEE Std 1044.1-1995, IEEE Guide to Classification for Software Anomalies
- [B21] IEEE Std 1045-1992, IEEE Standard for Software Productivity Metrics.
- [B22] IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans.
- [B23] IEEE Std 1059-1993, IEEE Guide for Software Verification and Validation Plans.
- [B24] IEEE Std 1061-1998, IEEE Standard for a Software Quality Metrics Methodology.
- [B25] IEEE Std 1062, 1998 Edition, IEEE Recommended Practice for Software Acquisition.
- [B26] IEEE Std 1063-1987 (Reaff 1993), IEEE Standard for Software User Documentation.
- [B27] IEEE Std 1175-1991, IEEE Standard Reference Model for Computing System Tool Interconnections.
- [B28] IEEE Std 1209-1992, IEEE Recommended Practice for the Evaluation and Selection of CASE Tools.
- [B29] IEEE Std 1219-1998, IEEE Standard for Software Maintenance.
- [B30] IEEE Std 1220-1998, IEEE Standard for Application and Management of the Systems Engineering Process.
- [B31] IEEE Std 1228-1994, IEEE Standard for Software Safety Plans.
- [B32] IEEE Std 1233, 1998 Edition, IEEE Guide for Developing System Requirements Specifications.
- [B33] IEEE Std 1298-1992 (AS 3563.1-1991), IEEE Standard for Software Quality Management System, Part 1: Requirements.
- [B34] IEEE Std 1348-1995, IEEE Recommended Practice for the Adoption of Computer-Aided Software Engineering (CASE) Tools.
- [B35] IEEE/EIA 12207.0-1996, IEEE/EIA Standard—Industry Implementation of ISO/IEC 12207: 1995, Standard for Information Technology—Software life cycle processes.
- [B36] IEEE/EIA 12207.1-1997, IEEE/EIA Guide for Information Technology--Software life cycle processes—Life cycle data.
- [B37] IEEE/EIA 12207.2-1997, IEEE/EIA Guide for Information Technology—Software life cycle processes—Implementation considerations.

[B38] ISO 9001 : 1994, Quality systems — Model for quality assurance in design, development, production, installation and servicing.⁷

[B39] ISO 9003 : 1994, Quality systems — Model for quality assurance in final inspection and test.

⁷ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembe, CH-1211, Genève 20, Switzerland/Suisse. ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.