IEEE Standard for Developing a Software Project Life Cycle Process

IEEE Computer Society

Sponsored by the
Software Engineering Standards Committee

1074™

IEEE
3 Park Avenue
New York, NY 10016-5997, USA

28 July 2006

IEEE Std 1074™-2006
(Revision of
IEEE Std 1074-1997)

IEEE Standard for Developing a Software Project Life Cycle Process

Sponsored by the

Software Engineering Standards Committee of the IEEE Computer Society

Approved 30 March 2006

IEEE-SA Standards Board

Abstract: This standard provides a process for creating a software project life cycle process (SPLCP). It is primarily directed at the process architect for a given software project.
Keywords: software project life cycle, software project life cycle model, software project life cycle process

---

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print: ISBN 0-7381-4956-X SH95532
PDF: ISBN 0-7381-4957-8 SS95532

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.
The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

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.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

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 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. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

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. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Introduction

This introduction is not part of IEEE Std 1074-2006, IEEE Standard for Developing a Software Project Life Cycle Process.

Background

IEEE Std 1074 is a standard for the generation of the process governing software development and maintenance for a project. This standard requires selection of a user's software project life cycle model (SPLCM) based on the organization's mission, vision, goals, and resources. It is not intended to define or imply a software project life cycle (SPLC) of its own nor does it presume or suggest any particular SPLCM. This standard describes the individual activities that are to be mapped within the selected model and provides examples of mapping onto typical SPLCMs. However, this standard is not an instructional guide. In addition to providing for the generation of a project process, this standard may also be used to develop organizational processes to support software development and maintenance or to develop special, single-function processes within a project.

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

The activities listed in this standard are not executable processes. They are components of processes and not intended to stand alone. The activities are generic and do not imply a sequential order. They have been administratively grouped for convenience and may be likened to a dictionary where words and meanings are arranged to allow the user to quickly locate a desired activity and its components.

This standard provides activities to be addressed in a software life cycle. It allows the user great flexibility in the manner in which activities are mapped onto the selected model and software project life cycle (SPLC) while preserving a normative standard to which to conform.

Utilization of these activities maximizes the benefits to the user when the use of this standard is initiated early in the software project's life cycle. A project that has proceeded past the initialization phase when this standard is invoked should gradually move into conformance to this standard.

This standard was written for any organization 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 in a larger system. This standard is also useful for projects that do not span the full software life cycle (i.e., developing a software specification or designing, writing, and verifying software that is based upon specifications developed by another organization, company, or previous project).

This standard allows for continuing harmonization with IEEE/EIA 12207.0 [B3][a] and its successors. The standard may be used to develop the primary and supporting life cycle processes specified in IEEE/EIA 12207. This standard supports the development of organizational standard processes and the selection of a standard, defined project process that is tailored from an organization's set of standard processes (organizational process definition) and integrated project management for each individual project.[b]

---

[a] The numbers in brackets correspond to the numbers of the bibliographic references listed in Annex F.

[b] Process improvement models that meet these criteria include the SEI CMMI® model. This information is given for the convenience of users and does not constitute an endorsement by the IEEE of these models. Equivalent models may be used if they can be shown to lead to the same results.

History

Since this standard's original publication, considerable worldwide attention has been paid to software project life cycle processes (SPLCPs). Use of and comments on IEEE Std 1074-1991/1995/1997, and other quality system and life cycle standards activity, have been carefully considered in preparing this substantive revision of this standard.

The 1995 version was a minor revision to correct specific errors found in the 1991 version.

The 1997 version saw the following changes:

— Activities were rearranged into more logical groupings (called activity groups) such as placing all planning activities into the new Project Planning Activity Group, collecting all project initiation activities, and collecting and expanding all review activities.

— The term *process* as 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 misinterpreting the collections as actual *processes* and trying to execute them as such. The term *activity group* is intended to 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.

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

— The focus of the standard was more clearly centered on a single process for a given project.

— The term *compliance* was changed to *conformance* to reflect international standards usage.

— Recognition of the importance of release management led to the addition of release management activities.

— The emerging importance of, and increased attention to, software security led to the addition of two activities: determine security objectives and confirm security accreditation.

Notice to users

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

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 or patent applications for which a license may be required to implement an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Participants

At the time this standard was completed, the Software Life Cycle Processes Working Group had the following membership:

Keith R. Middleton, *Chair*
Tom E. Starai, *Vice Chair*
John W. Horch, *Editor*

Bar Biszick-Lockwood
Ronald F. Dean, Sr.
Christof Ebert
Carol A. Long
Michael T. McCaffrey
John Napier
Cathie L. Neel
Robert J. Schaaf
Bob Shillato
Jan Van Moll
Charlene C. Walrad

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

Dave Bachman
Marisa Burt
Geoffrey Darnton
Peter Hantos
Bernard Holmes
Susan K. Land
Denis Meredith
James W. Moore
David J. Schultz

The following members of the balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Edward A. Addy
T. Scott Ankrum
Butch Anton
Ali Al Awazi
Adam J. Bagby
William D. Bartholomew
Edward E. Bartlett
Richard E. Biehl
Bar Biszick-Lockwood
Juris Borzovs
David G. Bowen
Curtis Browne
Dino Butorac
Juan C. Carreon
Norbert N. Carte
Danila Chernetsov
Keith Chow
S. M. Claassen
James P. Conigliaro
Paul R. Croll
Sylvia V. Daiqui
Geoffrey Darnton
Ronald F. Dean, Sr.
Sean Dougherty
Scott P. Duncan
Sourav K. Dutta
Kenneth D. Echeberry
Timothy Ehrler
William Eventoff
Harriet Feldman
Andrew C. Fieldsend
Ronald M. Fluegge
Andre F. Fournier
Michael D. Geipel
Gregg C. Giesler
Colin M. Glanville
John Garth Glynn
Lewis Gray
Randall C. Groves
Scott A. Gudgel
John Harauz
Mark O. Henley
Rutger A. Heunks
Bernard Homes
John W. Horch
Dennis Horwitz
Piotr Karocki
Robert B. Kelsey
Mark J. Knight
Thomas M. Kurihara
George W. Kyle
Susan K. Land
Kenneth Lang, Jr.
J. Dennis Lawrence
David J. Leciston
Solomon Lee
Yeou Song Lee
Daniel G. Levesque
Daniel A. Lindberg
Murray R. Little, Jr.
Carol A. Long
William Lumpkins
Faramarz Maghsoodlou
Richard A. Martin
L. J. Tajerian Martinez
Richard A. McBride
Michael T. McCaffrey
Mark F. McGranaghan
Francisco J. Melendez
Gary L. Michel
Keith R. Middleton
James W. Moore
John Napier
Cathie L. Neel
Michael S. Newman
Ciro A. Noronha
Mark C. Paulk
Miroslav Pavlovic
Howard W. Penrose
Robert A. Peterson
Donivan R. Porterfield
Cam K. Posani
Vikram Punj
Gerald M. Radack
Annette D. Reilly
Pascal Richer
Robert A. Robinson
James E. Sanders, Jr.
Helmut H. Sandmayr
Gary A. Savard
Robert J. Schaaf
Terry A. Schmidt
David J. Schultz
William J. Selph
Lynn J. Simms
Carl A. Singer
David Singleton
James M. Sivak
Mitchell W. Smith

Luca Spotorno
Friedrich Stallinger
Thomas E. Starai
Doug Steele
Gerald J. Stueve
K. S. Subrahmanyam
Douglas H. Thiele
John A. Thywissen
Winfried Tiedge
Thomas A. Tullia
Mark-Rene Uchida
Irene P. Viana
Richard J. Walker
Charlene C. Walrad
John W. Walz
Paul R. Work
Oren Yuen
Janusz Zalewski

When the IEEE-SA Standards Board approved this standard on 30 March 2006, it had the following membership:

Steve M. Mills, *Chair*
Richard H. Hulett, *Vice Chair*
Judith Gorman, *Secretary*

Mark D. Bowman
Dennis B. Brophy
Joseph Bruder
Richard Cox
Bob Davis
Julian Forster*
Joanna N. Guenin
Mark S. Halpin
Raymond Hapeman
William B. Hopf
Lowell G. Johnson
Herman Koch
Joseph L. Koepfinger*
David J. Law
Daleep C. Mohla
Paul Nikolich
T. W. Olsen
Glenn Parsons
Ronald C. Petersen
Gary S. Robinson
Frank Stone
Malcolm V. Thaden
Richard L. Townsend
Joe D. Watson
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish K. Aggarwal, *NRC Representative*
Richard DeBlasio, DOE Representative
Alan H. Cookson, NIST Representative

Jennie Steinhagen
IEEE Standards Project Manager, Document Development

Angela Ortiz
IEEE Standards Program Manager, Technical Program Development

Contents

IEEE Standard for Developing a Software Project Life Cycle Process

1. Overview

1.1 Scope

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

This methodology begins with the selection of an appropriate software project life cycle model (SPLCM) for use on the specific project. It continues through the definition of the software project life cycle (SPLC), using the selected SPLCM, the activities provided in Annex A, and the portion of the software life cycle that is relevant to the project. The methodology concludes with the augmentation of the software life cycle with organizational process assets (OPAs) to create the SPLCP.

The activities that are provided in Annex A cover the entire life cycle of a software system, from concept exploration through the eventual retirement of the software system. This standard does not address nonsoftware activities, such as contracting, purchasing, or hardware development. It also does not mandate the use of a specific SPLCM, nor does it provide a selection of, or a tutorial on, SPLCMs. This standard presumes that the process architect is already familiar with a variety of SPLCMs, 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 SPLCP is developed. 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 an application of this standard is the software project life cycle process (SPLCP) required for a specific software project.

1.4 Intended audiences

This standard is written to provide direction and guidance to process architects and other project personnel concerned with the implementation or performance of project processes.

1.4.1 Process architect

The primary audience for this standard is the process architect. The process architect is the person or organizational entity with the responsibility and authority to develop and maintain SPLCPs. This standard presumes that the process architect is already familiar with a variety of software project life cycle models (SPLCMs), with the criteria for choosing among them and with the criteria for determining the attributes and constraints of the desired end product and the project environment that affects this selection.

The process architect is expected to have the following expertise:

a) An understanding of the activities in Annex A

b) Knowledge of the OPAs

c) Knowledge of the standards listed in Annex F

d) Involvement in any and all software process improvement activities

1.4.2 Other interested parties

This standard may also be of use to the performers of the activities presented in Annex A (i.e., the project team).

1.5 Conformance

This standard can be applied to any SPLCM. Examples of some SPLCMs are contained in the informative Annex D. In order to conform to this standard, the development of the SPLCP shall conform to Clause 4; and the required activities identified in Annex A shall be included in the software project's SPLCP.

Many SPLCs do not address the entire software life cycle, but instead only address a subset of the software life cycle. If a SPLC uses only selected part(s) of the software life cycle, conformance to this standard may be achieved by including all the activities that impact the selected part(s) and the activities identified as *required* in Annex A. For example, if a SPLC is addressing only the designing and coding of a software artifact, all the activities in A.3.2 and A.3.3 and all the required activities shall be included. In this example, the remaining activities are optional and may be included in the SPLC as needed. If specific software development methodologies and techniques are used that combine the functionality of two or more activities, the activities not used, or combined, shall be noted and justified in the software project planned information for the project.

1.6 Relationship to other key standards

No standard lives isolated from its associated standards. This standard is related to ISO 9001 [B29] and IEEE/EIA 12207.0 [B3].

1.6.1 Relationship to ISO 9001 [B29]

The ISO 9001 family of standards 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 assure that the developed SPLCPs satisfy specific requirements of applicable clauses. Application of an SPLCP to ISO 9001 can be facilitated through the usage of ISO/IEC 90003 [B33].

1.6.2 Relationship to IEEE/EIA 12207.0 [B3]

IEEE/EIA 12207.0 establishes a common framework for the life cycle of software in terms of the processes that can be employed to

— Acquire, supply, develop, operate, and maintain software

— Manage, control, and improve the processes

— Provide the basis for world trade in software.

IEEE/EIA 12207.0 places requirements upon the characteristics of a designated set of life cycle processes, but does not specify the detailed implementation of those processes.

IEEE Std 1074 complements the application of IEEE/EIA 12207.0 and is written for a process architect, i.e., the individual or group responsible for establishing the software project life cycle process (SPLCP) to be followed on a particular project. The process architect may use IEEE Std 1074 in developing project-specific processes complying with the requirements of IEEE/EIA 12207.0.

1.6.3 Relationship to process improvement models

This standard supports the development of organizational standard processes and the selection of a standard, defined project process that is tailored from an organization's set of standard processes (organizational process definition) and integrated project management for each individual project.[1]

1.6.4 Relationship to software engineering standards

Annex F contains a list of software engineering standards that may provide insight into, or implementation suggestions for, the performance of the activities in Annex A. These and other standards may be used, but their inclusion in Annex F does not imply that conformance to them is required as part of conformance to this standard.

1.7 Relationship to process improvement

This standard can be integrated into an organization's process improvement program by using this standard as the framework for the OPAs.

Building the OPAs around this standard's structure of activities, input information, and output information can

a) Minimize the effort to develop an SPLCP

b) Facilitate the reuse of existing OPAs

c) Lead to improvement of the OPAs by incorporating lessons learned from the use of the OPAs in projects

A project's SPLCP, in part or as a whole, can become part of the organization's project assets for use by future projects.

[1] Process improvement models that meet these criteria include the SEI CMMI® model. This information is given for the convenience of users and does not constitute an endorsement by the IEEE of these models. Equivalent models may be used if they can be shown to lead to the same results.

1.8 Organization of this document

1.8.1 Clauses

Clause 1, Clause 2, and Clause 3 contain introductory information. Clause 4 provides a brief discussion of key concepts beneficial to the understanding and use of this standard. Clause 4 provides the requirements for the creation of an SPLCP. Requirements for the content of a SPLCP are presented in Annex A, which is normative. Annex B through Annex F 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	Introduction
Clause 2	Definitions
Clause 3	Key concepts
Clause 4	Implementing this standard
Annex A (normative)	Activities
Annex B (informative)	Mapping example
Annex C (informative)	Information mapping template
Annex D (informative)	Sample SPLCMs
Annex E (informative)	Glossary
Annex F (informative)	Bibliography

1.8.2 Activity grouping

The available components of the SPLCP are the 69 activities, grouped into 17 activity groups, in Annex A. The full set of 69 activities covers the entire software life cycle 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	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	Software Requirements Design Implementation
Post-Development	A.4	Installation Operation and Support Maintenance Retirement
Support	A.5	Evaluation Software Configuration Management Documentation Development Training

The Support Section (A.5) includes the activity groups that are necessary to assure the successful completion of a project, but are considered as supporting activities rather than activities directly oriented to the development effort.

The activity groups in the Support Section contain two types of activities:

a) Those that are performed discretely and are, therefore, mapped onto an SPLCM.

b) Those that are invoked (see 4.3.3) by other activities

1.8.3 Activity sequencing

This standard offers users a set of activities, grouped administratively. The orders of the sections, activity groups within sections, and activities within activity groups are for convenience only. No required, nor defined, sequence or timing of the performance of the activities, activity groups, or sections is intended, nor should any be implied.

These activities are not processes themselves. They are not intended to be executed sequentially and are not requirements for specific supporting documents. Users of this standard map these activities into a selected model in order to form processes that are appropriate for a using organization or project.

2. Definitions and acronyms

2.1 Definitions

For the purposes of this document, the following terms and definitions apply. The glossary in Annex F and *The Authoritative Dictionary of IEEE Standards Terms* [B5][2] should be referenced for terms not defined in this clause.

2.1.1 software project life cycle (SPLC): The portion of the entire software life cycle applicable to a specific project. It is the sequence of activities created by mapping the activities of IEEE Std 1074 onto a selected software project life cycle model (SPLCM).

2.1.2 software project life cycle model (SPLCM): The framework selected by each using organization on which to map the activities of IEEE Std 1074 to produce the software project life cycle (SPLC).

2.1.3 software project life cycle process (SPLCP): The project-specific description of the process developed by adding the organizational process assets (OPAs) to the software project life cycle (SPLC) and the OPAs.

2.1.4 Support Activity Group: An activity group that is necessary to assure the successful completion of a project, but consists of supporting activities rather than activities directly oriented to the development effort.

2.1.5 threat modeling: A systematic exploration technique to expose any circumstance or event having the potential to cause harm to a system in the form of destruction, disclosure, modification of data, and/or denial of service. It results in a vulnerability assessment.

[2]The numbers in brackets correspond to the numbers of the bibliographic references listed in Annex F.

2.1.6 security accreditation: Formal declaration by management that an information technology (IT) system is approved to operate in a particular security mode using a prescribed set of safeguards at an acceptable level of risk. ALSO: An independent accreditation body's certification that an IT system meets a predetermined security standard.

2.2 Acronyms

PR&RPI	problem reporting and resolution planned information
SCMPI	software configuration management planned information
SPLC	software project life cycle
SPLCM	software project life cycle model
SPLCP	software project life cycle process
SPMPI	software project management planned information
SRMPI	software release management planned information

3. Key concepts

3.1 Activities

An activity is a defined body of work to be performed, including its required input information 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 available input information has been processed and all applicable output information has been generated.

3.1.1 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: Discussion of the value-added actions to be performed to accomplish the transformation

c) Output information: A list of the information required to be generated by the transformation and the destination(s) of that information.

3.1.2 Entry and exit criteria

To *enter*, or start, an activity, at least one element of the specified input information shall be present. To *exit*, or complete, an activity, all input information from the activities appropriate to the selected life cycle 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 SPLCM.

3.1.3 Required activities

Most activities in the Project Management Section of activities group and some activities of the Support Section are required. Depending on the SPLCM chosen, activities in the Pre-Development, Development, and Post-Development sections of activity groups may or may not be required.

3.1.4 Organizational structure

This standard does not presume nor 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 or that an organizational entity's involvement be concentrated in only one activity

group. This standard does, however, presume that persons or organizations will be assigned accountability for the performance of the activities and for the quality of the input information and output information sets.

3.2 Elements of the software project life cycle process (SPLCP)

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

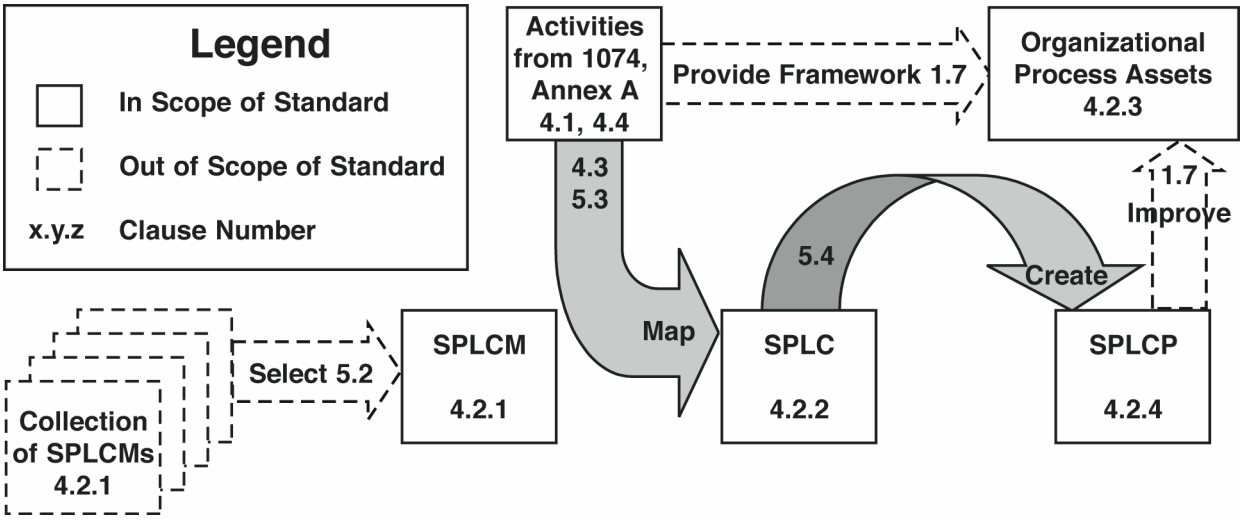

Figure 1—Developing an SPLCP

3.2.1 Software project life cycle model (SPLCM)

The SPLCM is the framework onto which the activities of this standard will be mapped to produce the software project life cycle (SPLC). To use this standard, a SPLCM shall be selected for a project. This selection is based on project attributes and organizational capabilities.

Annex D provides examples of typical models that might be considered.

3.2.2 Software project life cycle (SPLC)

The SPLC is the executable sequence of the activities to be performed during a project. The SPLC is developed by mapping the required and optional activities, provided in Annex A, onto the SPLCM selected for the project.

3.2.3 Organizational process assets (OPAs)

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

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

3.2.4 Software project life cycle process (SPLCP)

The SPLCP is developed by augmenting the software project life cycle (SPLC) with the OPAs selected for the project. It provides the specific approach to be used for the project.

3.3 Mapping

Mapping establishes the executable sequence of activities onto a selected software project life cycle model (SPLCM). Activities can be mapped in three ways: instance, iteration, and invocation.

3.3.1 Instance

An activity is mapped as an instance if it takes all of its available specified inputs, processes them, and produces all of its applicable specified outputs. It is mapped once and appears as a single event in the SPLC. The Allocate Project Resources Activity (A.1.1.3) could be an example of a single instance mapping.

3.3.2 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 available input information is processed and all applicable output information is created. The Manage the Project Activity (A.1.3.2) could require multiple iterations.

3.3.3 Invocation

In addition to the activities that are discretely mapped, there are activity groups of the Support Section 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 specific function and then return to the invoking activity.

The following example is taken from the Plan Project Management Activity (A.1.2.7), 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 SPMPI[1], the following activities shall[2] be invoked[3]:

a) Conduct Reviews (A.5.1.1)[4]

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)"

[1] This is the specified output information on which the invoked activities are to be performed. In other words, not all of this activity's output information is required to be documented, controlled, and evaluated, just the SPMPI.
[2] Required invocations are indicated with *shall*; others are recommended (*should*) or optional (*may*). The *should* and *may* invocations are to be considered based on the needs of the project.
[3] *Invoked* means to initiate a parallel task that is necessary to complete the required invoked activities and return here, before this activity can be considered complete.
[4] This is the 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 the invoking Plan Project Management Activity (A.1.2.7).

3.4 Input information and output information

Figure 2 depicts the conceptual flow of input information and output information into and out from an activity, respectively.

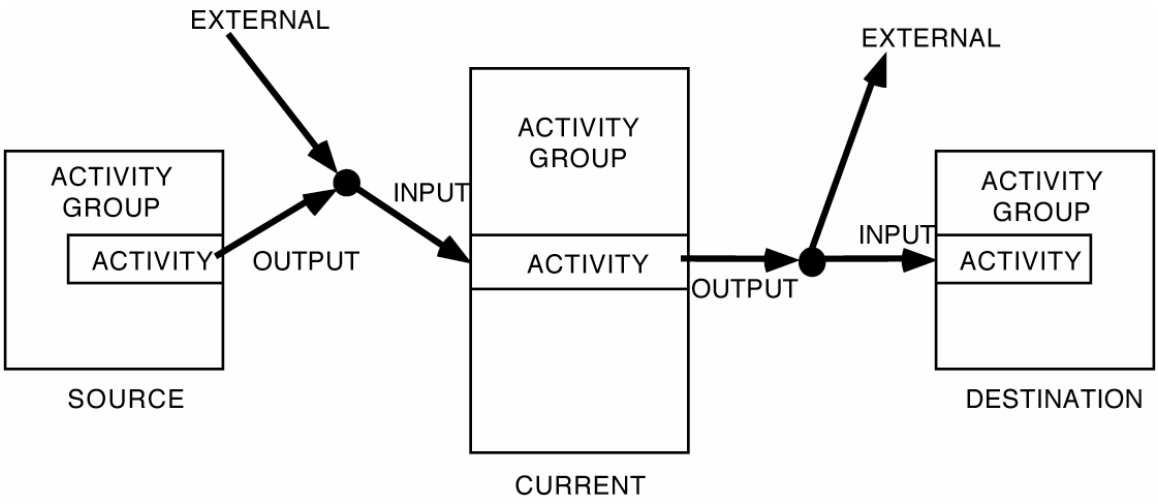

Figure 2—Information flow

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 in Annex A.

3.4.1 Conventions

The input information and output information for each activity are listed in a two-column format in the Annex A tables. The input information or output information name is listed in the left-hand column. The information source or destination activity is shown in the right-hand column.

As a convention of this standard, input information and output information names are capitalized in an activity's description.

3.4.2 External information

External information sources and destinations are outside the purview of this standard.

External input information sources might or might 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 column.

3.4.3 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 "Retain Records" (A.1.3.4), the various input information to Retain Records is collected under the term "Records." The corresponding "Activity group" and "Activity" columns refer simply to "Originating Activity Group" and "Originating Activity." The information for the Collect and Analyze Metric Data Activity (A.1.3.5) is received in the same way.

3.4.4 Information versus documents

This standard prescribes the activities of the software life cycle, not the form of the products of that life cycle. Therefore, when this standard refers to the creation of documentation, it does not require any specific

form. The information resulting from the execution of the activities is expected to be collected in whatever manner and form are consistent with the selected SPLCM and OPAs. Annex C provides a template for the assignment of output information into project-specific documents.

4. Implementing the standard

This clause describes the way in which implementation of this standard is to be approached. As stated in 1.4.1, the process architect has primary responsibility for creating and maintaining the software project life cycle process (SPLCP). This responsibility is implemented in five steps, as described below, and is performed as the Develop SPLCP Activity (A.1.1.1). An example of implementing this standard appears in Annex B.

4.1 Establish the requirements for a software project life cycle process (SPLCP)

The process architect shall assure that requirements for an SPLCP are established. Relevant stakeholders are identified, and their needs and expectations are transformed into a feasible set of requirements that are acceptable to the stakeholders.

4.2 Select software project life cycle model (SPLCM)

Initially, the process architect shall identify the SPLCM to which the activities will be mapped. This step encompasses locating, evaluating, selecting, and acquiring a SPLCM. It is possible for an organization to have multiple SPLCMs, but only one model is to be selected for a given project.

The process architect shall perform the following five steps to evaluate and select a SPLCM:

a) Identify all the SPLCMs available to the software project.

b) Identify the attributes that apply to the desired end system and the project environment.

c) Identify any constraints that might be imposed on the selection of the SPLCM.

d) Evaluate the various SPLCMs based on past experience and organizational capabilities.

e) Select the SPLCM that will best satisfy the project attributes and constraints.

4.3 Develop software project life cycle (SPLC)

The required activities identified in Annex A, plus all others applicable to the project, shall be mapped onto the SPLCM. Note that the selected SPLCM, or the project itself, could benefit from or require activities that are not included in Annex A. Additional activities are acceptable in the SPLC. However, note that not mapping the activities identified in Annex A as "required" will result in an SPLC and, therefore, an SPLCP that are not conformant to this standard. The steps in mapping follow.

4.3.1 Place the activities in executable sequence

The order in which activities will be performed is determined by four major considerations:

a) The selected SPLCM will dictate an initial ordering of activities; and, as mapping progresses, the actual order in which activities will be performed will be established.

b) Schedule constraints might require the overlapping of activities in the SPLCM and, thus, impact ordering. In this case, activities might be mapped for parallel execution rather than serial execution.

c) Selection and ordering of activities might 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 might require, as inputs, one or more of the outputs of the first.

d) If a selected activity is necessary to complete the mapping to the SPLCM, but one or more of its inputs come from activities not included for this SPLC, the source of the unavailable input shall be considered external. If an output is created for an activity that has not been selected and mapped, its destination for the output shall be considered external.

4.3.2 Develop and justify List of Activities Not Used

All activities that do not apply to this project shall be identified, along with the justification for their exclusion, in the List of Activities Not Used.

4.3.3 Verify the map

The process architect shall assure that the appropriate activities are fully mapped onto the selected SPLCM and that the resulting SPLC contains all of the activities 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.

4.4 Establish software project life cycle process (SPLCP)

The preceding steps develop the software project life cycle (SPLC). As the next step, the available OPAs shall be applied to the SPLC activities, and known constraints shall be reconciled. The output information generated by each activity shall be assigned to the appropriate document(s). (Annex C can be used for assistance in the assignment of information to documents.) The result is the established SPLCP.

4.5 Validate software project life cycle process (SPLCP)

The SPLCP shall be validated against the set of requirements identified in 4.14.

Annex A

(normative)

Software project life cycle (SPLC) activities

The activities included in this annex are to be mapped, as described in Clause 4, to the selected software project life cycle model (SPLCM) in order to develop the project-specific life cycle process. Additional activities that are required by the project may also be mapped.

A.1 Project Management Section of activity groups

The activity groups of the Project Management Section include activities that initiate, monitor, and control a software project throughout its life cycle. The activities listed are not processes. They shall be mapped onto a software project life cycle model (SPLCM) or joined together by a process architect to become processes appropriately tailored and relevant to an organization or project.

A.1.1 Project Initiation Activity Group

The activities of the Project Initiation Activity Group create and update the infrastructure of a software development or maintenance project. They build the base for the full SPLCP.

The activities of the Project Initiation Activity Group are

a) Develop SPLCP (A.1.1.1)
b) Perform Estimations (A.1.1.2)
c) Allocate Project Resources (A.1.1.3)
d) Define Metrics (A.1.1.4)
e) Determine Security Objectives (A.1.1.5)

A.1.1.1 Develop SPLCP (Required)

A.1.1.1.1 Input information

Input information	Source activity
Attributes	External
Available SPLCMs	External
Constraints	External
Contractual Requirements	External
Historical Records	External
IEEE Std 1074	External
OPAs	External
Environmental Improvement Needs	Identify SPLCP Improvement Needs (A.1.3.3)
Quality Policy and Quality Objectives	External
Statement of Need	Refine and Finalize the Idea or Need (A.2.1.4)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Maintenance Recommendations	Reapply SPLCP (A.4.3.3)

A.1.1.1.2 Description

Using the input information, the process architect shall develop the SPLCP as described in the five steps of Clause 4. Any required activities not used shall be included in the List of Activities Not Used. Exclusion of any required activity, however, will preclude conformance to this standard.

Additional guidance on this topic is provided by IEEE Std 1490™ [B27].

Prior to the distribution of the SPLCP, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

d) Develop Training Materials (A.5.4.1)

A.1.1.1.3 Output information

Output information	Destination activity
SPLCP	Perform Estimations (A.1.1.2)
	Allocate Project Resources (A.1.1.3)
	Define Metrics (A.1.1.4)
	Determine Security Objectives (A.1.1.5)
	Plan Documentation (A.1.2.5)
	Plan Training (A.1.2.6)
	Plan Project Management (A.1.2.7)
	Plan Release Management (A.1.2.9)
	Manage Risks (A.1.3.1)
	Manage the Project (A.1.3.2)
	Identify SPLCP Improvement Needs (A.1.3.3)
List of Activities Not Used	Manage Risks (A.1.3.1)

A.1.1.2 Perform Estimations (Required)

A.1.1.2.1 Input information

Input information	Source activity
Historical Project Records	External
SPLCP	Develop SPLCP (A.1.1.1)
Security Objectives	Determine Security Objectives (A.1.1.5)
Statement of Need	Refine and Finalize the Idea or Need (A.2.1.4)
System Functional Software Requirements	Allocate 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 SPLC. In addition, target computer resource usage shall be estimated.

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

Prior to the distribution of Project Estimates, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.1.1.2.3 Output information

Output information	Destination activity
Project Estimates	Allocate Project Resources (A.1.1.3)
	Plan Project Management (A.1.2.7)
	Manage Risks (A.1.3.1)
	Manage the Project (A.1.3.2)
Estimation Assumptions	Manage Risks (A.1.3.1)

A.1.1.3 Allocate Project Resources (Required)

A.1.1.3.1 Input information

Input information	Source activity
Historical Project Records	External
Resources	External
SPLCP	Develop SPLCP (A.1.1.1)
Project Estimates	Perform Estimations (A.1.1.2)
Security Objectives	Determine Security Objectives (A.1.1.5)
Statement of Need	Refine and Finalize the Idea or Need (A.2.1.4)
System Functional Software Requirements	Allocate System Requirements (A.2.2.3)
Software 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 SPLC. Resources that are to be allocated include budget, personnel, equipment, space, and computer resources.

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

A.1.1.3.3 Output information

Output information	Destination activity
Resource Allocations	Plan Project Management (A.1.2.7)
	Manage Risks (A.1.3.1)
	Manage the Project (A.1.3.2)

A.1.1.4 Define Metrics (Required)

A.1.1.4.1 Input information

Input information	Source activity
SPLCP	Develop SPLCP (A.1.1.1)
Security Objectives	Determine Security Objectives (A.1.1.5)
Evaluation Planned Information	Plan Evaluations (A.1.2.1)
SPMPI	Plan Project Management (A.1.2.7)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)

A.1.1.4.2 Description

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

Additional guidance on this topic is provided by IEEE Std 982.1™ [B10], IEEE Std 1044™ [B15], IEEE Std 1045™ [B16], and IEEE Std 1061™ [B18].

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

A.1.1.4.3 Output information

Output information	Destination activity
Defined Metrics	Plan Evaluations (A.1.2.1)
	Collect and Analyze Metric Data (A.1.3.5)
Collection and Analysis Methods	Plan Evaluations (A.1.2.1)
	Collect and Analyze Metric Data (A.1.3.5)

A.1.1.5 Determine Security Objectives (Required)

A.1.1.5.1 Input information

Input information	Source activity
Projected Value, Criticality and Data Sensitivity Assessment	External
Historical Security Records	External
Security Classification Policy	External
Validated Industry Protection Profile	External

Input information	Source activity
Internal Acceptability Standards (if required)	External
Security Stakeholders	External
OPAs	External
SPLCP	Develop SPLCP (A.1.1.1)
Preliminary Statement of Need	Identify Ideas or Needs (A.2.1.1)
Statement of Need	Refine and Finalize the Idea or Need (A.2.1.4)

A.1.1.5.2 Description

The Determine Security Objectives Activity shall determine the appropriate security assurance levels, if any, that are to be applied to the project and the product to mitigate organizational security risk. Security assurance levels shall be derived by estimating the potential likelihood and business impact of product security failure based on historical precedence and the security status of interdependent systems, in light of the high-level assessment of the proposed product's estimated business value, operational criticality, and data sensitivity. The level of security assurance shall be applied to achieve an acceptable security risk for the project, product, and specified categories of information.

First, OPAs related to security shall be collected and evaluated for relevance based on the Preliminary Statement of Need. Sources of requirements to be considered may include current organizational security, privacy and liability policy, contractor, vendor service level and partnership agreements, and the prevailing security technology and rules applied to users and groups to protect against unauthorized data access and modification. Second, opportunity for liability shall be explored through analysis of Historical Security Records, including analysis of security breach history and cost of impacts, the current security status of proposed interfacing systems, legal recommendations, and the results of security audits for similar systems. Finally, the Preliminary Statement of Need and the Projected Value, Criticality, and Data Sensitivity Assessment are combined and evaluated against the accumulated body of knowledge to determine the appropriate level of security that needs to be applied to the project. Taken together, this information will indicate an appropriate security level for the class of product under development.

Upon receipt of the Statement of Need, the Security Objectives shall be refined. If the product is required to meet industry accreditation standards, then the applicable Validated Industry Protection Profile shall be identified and applied.

If a Validated Industry Protection Profile does not exist or if such formal accreditation is not required, internal acceptability standards regarding security shall be applied. The security level identified for the product shall suggest the security level and budget also needed for the project. Therefore, policies and procedures shall be identified that are required to be applied to the project team and development environment to ensure the protection of proprietary project information and the responsible management of project assets.

Finally, Security Stakeholders shall be identified, and planning shall begin for security documentation, which may include specialized documentation for external organizational units such as legal, asset management, internal audit, maintenance and support, and business continuity.

Additional guidance on this topic is provided by ISO 15408 [B30].

Prior to the distribution of the Security Objectives, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.1.1.5.3 Output information

Output information	Destination activity
Security Objectives	Most activities of the Project Planning Activity Group (A.1.2)
	Perform Estimations (A.1.1.2)
	Allocate Project Resources (A.1.1.3)
	Define Metrics (A.1.1.4)
	Formulate Potential Approaches (A.2.1.2)
	Develop System Architecture (A.2.2.2)
	Identify Software Improvement Needs (A.4.3.1)
	Implement Problem Reporting Method (A.4.3.2)
	Reapply SPLCP (A.4.3.3)
	Retire System (A.4.4.3)
	Confirm Security Accreditation (A.5.1.8)

A.1.2 Project Planning Activity Group

The activities of the Project Planning Activity Group 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 or at each development iteration.

The activities of the Project Planning Activity Group are

a) Plan Evaluations (A.1.2.1)

b) Plan Configuration Management (A.1.2.2)

c) Plan System Transition (if applicable) (A.1.2.3)

d) Plan Installation (A.1.2.4)

e) Plan Documentation (A.1.2.5)

f) Plan Training (A.1.2.6)

g) Plan Project Management (A.1.2.7)

h) Plan Integration (A.1.2.8)

i) Plan Release Management (A.1.2.9)

A.1.2.1 Plan Evaluations (Required)

A.1.2.1.1 Input information

Input information	Source activity
Defined Metrics	Define Metrics (A.1.1.4)
Collection and Analysis Methods	Define Metrics (A.1.1.4)
SPMPI	Plan Project Management (A.1.2.7)
Integration Planned Information	Plan Integration (A.1.2.8)
Risk Management Reported Information	Manage Risks (A.1.3.1)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)

Input information	Source activity
Preliminary Software Requirements	Define and Develop Software Requirements (A.3.1.1)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Detailed Design	Perform Detailed Design (A.3.2.4)
Configuration Identification	Develop Configuration Identification (A.5.2.1)

A.1.2.1.2 Description

The Plan Evaluations Activity shall identify and describe the evaluation tasks that are necessary to assure 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, testing, and the appropriate level of security accreditation (internal or industry). Reviews that are to be planned include peer, management, technical, operational, process improvement, security, and post-implementation. More information on reviews can be found in the Conduct Reviews Activity (A.5.1.1).

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 the Conduct Audits Activity (A.5.1.3).

Test planning shall be used to define the test strategy, the basic test environment, and the test structure that are needed to support the required levels of testing. Each planned test shall identify the items to be tested, the corresponding requirements, and the pass-fail criteria. Test planning shall also identify the test coverage criteria. Test planning shall be coordinated with the Plan Integration Activity (A.1.2.8).

The evaluation planning information shall include the evaluation team's 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, technical, and security risks. This planning shall be documented in the Evaluation Planned Information.

Additional guidance on this topic is provided by IEEE Std 730™ [B6], IEEE Std 828™ [B7], IEEE Std 829™ [B8], IEEE Std 982.1 [B10], IEEE Std 1008™ [B11], IEEE Std 1012™ [B12], IEEE Std 1028™ [B14], IEEE Std 1044 [B15], and IEEE Std 1045 [B16].

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

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.1.2.1.3 Output information

Output information	Destination activity
Evaluation Planned Information	Define Metrics (A.1.1.4)
	Plan Integration (A.1.2.8)
	Manage Risks (A.1.3.1)
	Manage the Project (A.1.3.2)
	Collect and Analyze Metric Data (A.1.3.5)
	Identify Software Improvement Needs (A.4.3.1)
	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)
	Confirm Security Accreditation (A.5.1.8)

A.1.2.2 Plan Configuration Management (Required)

A.1.2.2.1 Input information

Input information	Source activity
Deliverable List	External
Security Objectives	Determine Security Objectives (A.1.1.5)
SPMPI	Plan Project Management (A.1.2.7)
SRMPI	Plan Release Management (A.1.2.9)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)

A.1.2.2.2 Description

The Plan Configuration Management 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.

Software configuration management includes the evaluation, coordination, approval or disapproval, and implementation of changes to product components (e.g., code, documentation) after a baseline has been established.

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

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

Additional guidance on this topic is provided by IEEE Std 828 [B7].

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

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.1.2.2.3 Output information

Output information	Destination activity
SCMPI	Plan System Transition (if applicable) (A.1.2.3)
	Plan Release Management (A.1.2.9)
	Manage the Project (A.1.3.2)
	Retain Records (A.1.3.4)
	All activities of the Software Configuration Management Activity Group (A.5.2)

A.1.2.3 Plan System Transition

A.1.2.3.1 Input information

Input information	Source activity
Retirement Planned Information (for the system being replaced)	External
Security Objectives	Determine Security Objectives (A.1.1.5)
SCMPI	Plan Configuration Management (A.1.2.2)
SRMPI	Plan Release Management (A.1.2.9)
Preliminary Statement of Need	Identify Ideas or Needs (A.2.1.1)
Recommendations	Conduct Feasibility Studies (A.2.1.3)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)

A.1.2.3.2 Description

The Plan System Transition Activity is applicable only when an existing system (automated or manual) is being replaced with a new or revised 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, including an evaluation of transition security risk and mitigation strategy.

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 and security controls and reporting procedures as well as the risks and contingencies. Special attention should be given to minimizing operational risks, especially those that might compromise security. This planning shall be documented in the Transition Planned Information.

The transition planning activities shall include a review and revision of the SCMPI in respect of these plans, tools, and controls.

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

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.1.2.3.3 Output information

Output information	Destination activity
Transition Planned Information	Plan Installation (A.1.2.4)
	Manage the Project (A.1.3.2)
Transition Impact Statement	Manage Risks (A.1.3.1)
	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
Security Objectives	Determine Security Objectives (A.1.1.5)
Transition Planned Information	Plan System Transition (A.1.2.3)
SPMPI	Plan Project Management (A.1.2.7)
SRMPI	Plan Release Management (A.1.2.9)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)
Installation Requirements	Define and Develop Software Requirements (A.3.1.1)
Operating Documentation	Create Operating Documentation (A.3.3.2)

A.1.2.4.2 Description

The tasks to be performed during the Plan Installation Activity 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) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

d) Develop Training Materials (A.5.4.1)

A.1.2.4.3 Output information

Output information	Destination activity
Software Installation Planned Information	Manage the Project (A.1.3.2)
	Distribute Software (A.4.1.1)

A.1.2.5 Plan Documentation (Required)

A.1.2.5.1 Input information

Input information	Source activity
Contractual Requirements	External
SPLCP	Develop SPLCP (A.1.1.1)
Security Objectives	Determine Security Objectives (A.1.1.5)
SPMPI	Plan Project Management (A.1.2.7)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)

A.1.2.5.2 Description

In the Plan Documentation Activity, information such as the SCMPI, product descriptions, organizational security requirements, 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, the documentation security classifications and access levels, 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 on this topic is provided by IEEE Std 1063™ [B20].

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

a) Conduct Reviews (A.5.1.1)

b) Implement Documentation (A.5.3.1)

The Perform Configuration Control Activity (A.5.2.2) should also be invoked.

A.1.2.5.3 Output information

Output information	Destination activity
Documentation Planned Information	Manage the Project (A.1.3.2)
	Retain Records (A.1.3.4)
	Create Operating Documentation (A.3.3.2)
	All activities of the Documentation Development Activity Group (A.5.3)

A.1.2.6 Plan Training

A.1.2.6.1 Input information

Input information	Source activity
Applicable Information	External
Skills Inventory	External
SPLCP	Develop SPLCP (A.1.1.1)
Security Objectives	Determine Security Objectives (A.1.1.5)
SPMPI	Plan Project Management (A.1.2.7)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Training Feedback	Validate the Training Program (A.5.4.2)
	Implement the Training Program (A.5.4.3)

A.1.2.6.2 Description

The Plan Training 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 on this topic is provided by Australian Standard (AS) 3563.1 [B1].

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

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.1.2.6.3 Output information

Output information	Destination activity
Training Planned Information	Manage the Project (A.1.3.2)
	Identify Software Improvement Needs (A.4.3.1)
	All activities of the Training Activity Group (A.5.4)

A.1.2.7 Plan Project Management (Required)

A.1.2.7.1 Input information

Input information	Source activity
Contractual Requirements	External
Historical Project Records	External
SPLCP	Develop SPLCP (A.1.1.1)
Project Estimates	Perform Estimations (A.1.1.2)
Resource Allocations	Allocate Project Resources (A.1.1.3)
Security Objectives	Determine Security Objectives (A.1.1.5)
Risk Management Reported Information	Manage Risks (A.1.3.1)
Project Management Reported Information	Manage the Project (A.1.3.2)
Preliminary Statement of Need	Identify Ideas or Needs (A.2.1.1)
Recommendations	Conduct Feasibility Studies (A.2.1.3)
Statement of Need	Refine and Finalize the Idea or Need (A.2.1.4)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)

A.1.2.7.2 Description

The Plan Project Management Activity requires the collection and synthesis of a great deal of information into a coherent and organized SPMPI based on the SPLCP. 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, security, 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 enterprise planning, target operational environment, regulatory approvals, required certifications, user involvement, subcontracting, and security.

This activity shall include planning for support, problem reporting, risk management, security compliance, and retirement.

Support planning shall include methods for supporting the software in the operational environment, including recommended methods for secure backup, recovery, and archiving.

Problem Reporting and Resolution Planned 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.

Security Planned Information shall address ensuring appropriate security control at all times over organizational assets and information and the responsible integration of the product into the target security infrastructure. This shall include determining security levels and enforcement methods to be applied to the project, its environments, and its channels of communication. It may include managing data privacy and auditability risk, coordinating with security authorities and stakeholders, and determining the product's role in organizational intrusion or failure response.

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 on this topic is provided by IEEE Std 1058™ [B17], IEEE Std 1220™ [B23], and IEEE Std 1490 [B27].

Prior to the distribution of the SPMPI, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.1.2.7.3 Output information

Output information	Destination activity
SPMPI	Most activities
PR&RPI	Manage Risks (A.1.3.1)
	Manage the Project (A.1.3.2)
	Implement Problem Reporting Method (A.4.3.2)
	Reapply SPLCP (A.4.3.3)
Retirement Planned Information	Manage the Project (A.1.3.2)
	Notify User (A.4.4.1)
	Conduct Parallel Operations (if applicable) (A.4.4.2)
	Retire System (A.4.4.3)
Security Planned Information	Manage Risks (A.1.3.1)
	Manage the Project (A.1.3.2)
Support Planned Information	Manage Risks (A.1.3.1)
	Manage the Project (A.1.3.2)
	All activities of the Operation and Support Activity Group (A.4.2)

A.1.2.8 Plan Integration

A.1.2.8.1 Input information

Input information	Source activity
Evaluation Planned Information	Plan Evaluations (A.1.2.1)
SPMPI	Plan Project Management (A.1.2.7)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)

Input information	Source activity
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Detailed Design	Perform Detailed Design (A.3.2.4)

A.1.2.8.2 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 SPLCP, as defined in the SPMPI, and the security requirements embodied in the Software Requirements and Software Detailed Design, 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 Evaluation 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) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.1.2.8.3 Output information

Output information	Destination activity
Integration Planned Information	Plan Evaluations (A.1.2.1)
	Manage Risks (A.1.3.1)
	Manage the Project (A.1.3.2)
	Perform Integration (A.3.3.3)

A.1.2.9 Plan Release Management

A.1.2.9.1 Input information

Input information	Source activity
Deliverable List	External
SPLCP	Develop SPLCP (A.1.1.1)
SCMPI	Plan Configuration Management (A.1.2.2)
SPMPI	Plan Project Management (A.1.2.7)
Configuration Identification	Develop Configuration Identification (A.5.2.1)

A.1.2.9.2 Description

The Plan Release Management Activity shall plan and document specific release management organizations and responsibilities, procedures, tools, techniques, and methodologies in the software release management planned information (SRMPI). The SRMPI shall also describe how and when such procedures are to be performed. This planning activity will be coordinated with the Plan Configuration Management Activity (A.1.2.2).

Overall software release management objectives, including release frequency, release milestones, and release media, are derived using internal guidelines as well as contractual or other agreed-upon requirements from the SPMPI.

Items that are to be released include code, user and/or operations documentation, release notes, and other materials for the user. The release management planning shall include developing schedules, estimating resources, identifying special resources, staffing, and defining management controls and reporting procedures.

Release management planning shall include build procedures and naming conventions, branching models, packaging requirements, and delivery media.

Prior to the distribution of the SRMPI, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.1.2.9.3 Output information

Output information	Destination activity
SRMPI	Plan Configuration Management (A.1.2.2)
	Plan System Transition (A.1.2.3)
	Plan Installation (A.1.2.4)
	Manage the Project (A.1.3.2)
	Retain Records (A.1.3.4)
	Manage Software Releases (A.3.3.4)
	All activities of the Software Configuration Management Activity Group (A.5.2)

A.1.3 Project Monitoring and Control Activity Group

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

In addition, the Project Monitoring and Control Activity Group encompasses the collection and analysis of the software metrics of the project, the retention of project records, and the identification of SPLCP improvement opportunities.

The activities of the Project Monitoring and Control Activity Group are

a) Manage Risks (A.1.3.1)

b) Manage the Project (A.1.3.2)

c) Identify SPLCP Improvement Needs (A.1.3.3)

d) Retain Records (A.1.3.4)

e) Collect and Analyze Metric Data (A.1.3.5)

f) Close Project (A.1.3.6)

A.1.3.1 Manage Risks (Required)

A.1.3.1.1 Input information

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

A.1.3.1.2 Description

The Manage Risks 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 tradeoffs 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, and independent evaluations.

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 tradeoffs. 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 this activity. The results of the analyses that are conducted during this activity shall be included in the Risk Management Reported Information.

Additional guidance on this topic is provided by IEEE Std 1228™ [B24] and IEEE Std 1540™ [B28].

Prior to the distribution of the Risk Management Reported Information, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

The Conduct Audits Activity (A.5.1.3) should be invoked.

A.1.3.1.3 Output information

Output information	Destination activity
Risk Management Reported Information	Plan Evaluations (A.1.2.1)
	Plan Project Management (A.1.2.7)
	Manage the Project (A.1.3.2)
	Define and Develop Software Requirements (A.3.1.1)
	Prioritize and Integrate Software Requirements (A.3.1.3)
	Confirm Security Accreditation (A.5.1.8)

A.1.3.2 Manage the Project (Required)

A.1.3.2.1 Input information

Input information	Source activity
Feedback Data	External
SPLCP	Develop SPLCP (A.1.1.1)
Project Estimates	Perform Estimations (A.1.1.2)
Resource Allocations	Allocate Project Resources (A.1.1.3)
Evaluation Planned Information	Plan Evaluations (A.1.2.1)
SCMPI	Plan Configuration Management (A.1.2.2)
Transition Planned Information	Plan System Transition (A.1.2.3)

Input information	Source activity
Software Installation Planned Information	Plan Installation (A.1.2.4)
Documentation Planned Information	Plan Documentation (A.1.2.5)
Training Planned Information	Plan Training (A.1.2.6)
Security Planned Information	Plan Project Management (A.1.2.7)
SPMPI	Plan Project Management (A.1.2.7)
Retirement Planned Information	Plan Project Management (A.1.2.7)
Support Planned Information	Plan Project Management (A.1.2.7)
PR&RPI	Plan Project Management (A.1.2.7)
Integration Planned Information	Plan Integration (A.1.2.8)
SRMPI	Plan Release Management (A.1.2.9)
Risk Management Reported Information	Manage Risks (A.1.3.1)
Analysis Reported Information	Collect and Analyze Metric Data (A.1.3.5)
Selected Software Import Sources	Evaluate Software Import Sources (A.2.3.2)
Installation Reported Information	Install Software (A.4.1.2)
Software Improvement Recommendations	Identify Software Improvement Needs (A.4.3.1)
Evaluation Reported Information	Report Evaluation Results (A.5.1.7)
Status Reported Information	Perform Status Accounting (A.5.2.3)

A.1.3.2.2 Description

The Manage the Project Activity shall manage the execution of all activities in the SPLCP, according to the plans set forth in the Project Planning Activity Group (A.1.2). The progress of the project shall be reviewed and measured against the established estimates and plans (e.g., estimated versus actual cost, estimated versus actual effort, and planned versus 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 assure successful project completion.

This activity may invoke the Conduct Reviews Activity (A.5.1.1) or the Conduct Audits Activity (A.5.1.3) in order to verify compliance to the SPLCP and/or project planning plans.

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

A.1.3.2.3 Output information

Output information	Destination activity
Project Management Reported Information	External
	Plan Project Management (A.1.2.7)
	Manage Risks (A.1.3.1)
	Identify SPLCP Improvement Needs (A.1.3.3)
	Close Project (A.1.3.6)
Anomalies	Implement Problem Reporting Method (A.4.3.2)

A.1.3.3 Identify SPLCP Improvement Needs (Required)

A.1.3.3.1 Input information

Input information	Source activity
Historical Project Records	External
SPLCP	Develop SPLCP (A.1.1.1)
Project Management Reported Information	Manage the Project (A.1.3.2)
Analysis Reported Information	Collect and Analyze Metric Data (A.1.3.5)
Software Improvement Recommendations	Identify Software Improvement Needs (A.4.3.1)
Post-Operation Review Reported Information	Retire System (A.4.4.3)
Evaluation Reported Information	Report Evaluation Results (A.5.1.7)

A.1.3.3.2 Description

The Identify SPLCP Improvement Needs Activity shall analyze Project Management Reported Information, project metrics from Analysis Reported Information, Evaluation Reported Information, and the other input information to determine instances in which SPLCP 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 whether the implemented change produced a positive result. Environment Improvement Needs can point to improvement opportunities that are outside the scope of the project.

Additional guidance on this topic is provided by IEEE Std 1045 [B16] and IEEE Std 1061 [B18].

A.1.3.3.3 Output information

Output information	Destination activity
Environment Improvement Needs	External
	Develop SPLCP (A.1.1.1)
	Implement Problem Reporting Method (A.4.3.2)
	Close Project (A.1.3.6)

A.1.3.4 Retain Records (Required)

A.1.3.4.1 Input information

Input information	Source activity
Information Retention Standards	External
Records	Originating Activity

A.1.3.4.2 Description

The Retain Records 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.

Additional guidance on this topic is provided by AS 3563.1 [B1].

A.1.3.4.3 Output information

Output information	Destination activity
Historical Project Records	External
	Close Project (A.1.3.6)

A.1.3.5 Collect and Analyze Metric Data (Required)

A.1.3.5.1 Input information

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

A.1.3.5.2 Description

The Collect and Analyze Metric Data 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.

Additional guidance on this topic is provided by IEEE Std 982.1 [B10], IEEE Std 1044 [B15], IEEE Std 1045 [B16], IEEE Std 1061 [B18], IEEE Std 1220 [B23], ISO/IEC 14598 [B30], and ISO/IEC 15939 [B32].

Prior to the distribution of the Analysis Reported Information, the following activities should be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.1.3.5.3 Output information

Output information	Destination activity
Analysis Reported Information	Manage Risks (A.1.3.1)
	Manage the Project (A.1.3.2)
	Identify SPLCP Improvement Needs (A.1.3.3)
	Identify Software Improvement Needs (A.4.3.1)

A.1.3.6 Close Project (Required)

A.1.3.6.1 Input information

Input information	Source activity
Stakeholder Input	External
Customer Acceptance	Accept Software In Operational Environment (A.4.1.3)
Environment Improvement Needs	Identify SPLCP Improvement Needs (A.1.3.3)
Maintenance Recommendations	Reapply SPLCP (A.4.3.3)
Historical Project Records	Retain Records (A.1.3.4)
Project Management Reported Information	Manage the Project (A.1.3.2)
Correction Problem Reported Information	Implement Problem Reporting Method (A.4.3.2)
Enhancement Problem Reported Information	Implement Problem Reporting Method (A.4.3.2)
Report Log	Implement Problem Reporting Method (A.4.3.2)
Resolved Problem Reported Information	Reapply SPLCP (A.4.3.3)
Updated Report Log	Reapply SPLCP (A.4.3.3)

A.1.3.6.2 Description

The Close Project Activity generates, gathers, verifies, and distributes information required to formally conclude a project. Conclusion may be the result of the successful completion of the project or its premature termination. The activity includes tasks to assure administrative closure as well as tasks to assure retention of relevant historical project information.

At the time of project closure, salient information about the project's history is collected and preserved in the Project Archival Information. This shall include any Project Management Reported Information, such as metric data, status reports, evaluation reported information or software improvement needs, as well as retained Historical Project Records generated under A.1.3.4, such as documentation, software, and data deliverables.

Additional guidance on this topic is provided by IEEE Std 1490 [B27].

Prior to distribution of the Project Archival Information, the Implement Documentation Activity (A.5.3.1) should be invoked.

The Conduct Reviews Activity (A.5.1.1) may be invoked.

A.1.3.6.3 Output information

Output information	Destination activity
Project Archival Information	Retain Records (A.1.3.4)

A.2 Pre-Development Section of activity groups

The activity groups of the Pre-Development Section include the activities that explore and allocate system requirements before software development can begin. The activities listed are not processes. They are required to be mapped onto a software project life cycle model (SPLCM) or joined together by a process architect to become processes appropriately tailored and relevant to an organization or project.

A.2.1 Concept Exploration Activity Group

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 Activity Group or the Software 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.

The activities of the Concept Exploration Activity Group are

a) Identify Ideas or Needs (A.2.1.1)

b) Formulate Potential Approaches (A.2.1.2)

c) Conduct Feasibility Studies (A.2.1.3)

d) Refine and Finalize the Idea or Need (A.2.1.4)

A.2.1.1 Identify Ideas or Needs (Required)

A.2.1.1.1 Input information

Input information	Source activity
Changing Software Requirements	External
Customer Requests	External
Enhancement Problem Reported Information (from an earlier project)	External
Enterprise Constraints	External
Ideas from Within the Development Organization	External
Maintenance Recommendations (from an earlier project)	External
Marketing Information Sources	External
User Requests	External

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 functional and nonfunctional needs. Changing Software Requirements can come from legislation, regulations, national and international standards, maintenance, etc.

Known enterprise constraints, such as those imposed by the target environment, enterprise policies, acquisition plans, and/or interoperability and version compatibility constraints will also be documented.

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

A.2.1.1.3 Output information

Output information	Destination activity
Preliminary Statement of Need	Determine Security Objectives (A.1.1.5)
	Plan System Transition (if applicable) (A.1.2.3)
	Plan Project Management (A.1.2.7)
	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 (Required)

A.2.1.2.1 Input information

Input information	Source activity
Development Resources and Budget	External
Market Availability Data	External
Resource Information	External
Security Objectives	Determine Security Objectives (A.1.1.5)
Preliminary Statement of Need	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 each of the Potential Approaches. The Constraints and Benefits should include all aspects of the life cycle.

Additional guidance on this topic is provided by IEEE Std 830™ [B9] and IEEE Std 1220 [B23].

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

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

A.2.1.2.3 Output information

Output information	Destination activity
Constraints and Benefits	Conduct Feasibility Studies (A.2.1.3)
	Refine and Finalize the Idea or Need (A.2.1.4)
Potential Approaches	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 (Required)

A.2.1.3.1 Input information

Input information	Source activity
Preliminary Statement of Need	Identify Ideas or Needs (A.2.1.1)
Constraints and Benefits	Formulate Potential Approaches (A.2.1.2)
Potential Approaches	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, the Conduct Reviews Activity (A.5.1.1) may be invoked.

A.2.1.3.3 Output information

Output information	Destination activity
Recommendations	Plan System Transition (if applicable) (A.1.2.3)
	Plan Project Management (A.1.2.7)
	Refine and Finalize the Idea or Need (A.2.1.4)
	Analyze System Functions (A.2.2.1)

A.2.1.4 Refine and Finalize the Idea or Need (Required)

A.2.1.4.1 Input information

Input information	Source activity
Preliminary Statement of Need	Identify Ideas or Needs (A.2.1.1)
Constraints and Benefits	Formulate Potential Approaches (A.2.1.2)
Potential Approaches	Formulate Potential Approaches (A.2.1.2)
Recommendations	Conduct Feasibility Studies (A.2.1.3)
Transition Impact Statement	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. 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) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.2.1.4.3 Output information

Output information	Destination activity
Statement of Need	Develop SPLCP (A.1.1.1)
	Perform Estimations (A.1.1.2)
	Allocate Project Resources (A.1.1.3)
	Determine Security Objectives (A.1.1.5)
	Plan Project Management (A.1.2.7)
	Manage Risks (A.1.3.1)
	Analyze System Functions (A.2.2.1)
	Develop System Architecture (A.2.2.2)

A.2.2 System Allocation Activity Group

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 human or hardware requirements analyses are discussed in this document; they are beyond the scope of this standard.

The activities of the System Allocation Activity Group are

a) Analyze System Functions (A.2.2.1)

b) Develop System Architecture (A.2.2.2)

c) Allocate System Requirements (A.2.2.3)

A.2.2.1 Analyze System Functions

A.2.2.1.1 Input information

Input information	Source activity
Recommendations	Conduct Feasibility Studies (A.2.1.3)
Statement of Need	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. Nonfunctional requirements and known constraints are analyzed to identify their impact on functional requirements. 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) human and hardware functions.

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

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

A.2.2.1.3 Output information

Output information	Destination activity
Functional Description of the System	Develop System Architecture (A.2.2.2)
	Allocate System Requirements (A.2.2.3)
	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
Known Security Vulnerabilities	External
Applicable Threat Models	External
Available Protection Methods	External
Security Objectives	Determine Security Objectives (A.1.1.5)
SPMPI	Plan Project Management (A.1.2.7)
Statement of Need	Refine and Finalize the Idea or Need (A.2.1.4)
Functional Description of the System	Analyze System 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. Threat modeling or other exploratory techniques shall be applied to explore known and potential security vulnerabilities and their impacts. The architecture shall be adjusted accordingly to mitigate threats not

covered by available protection methods that will insure secure operation, failover, and recovery. The System Architecture becomes the basis for the determination of the software functions, security controls, and the hardware functions, if any.

Additional guidance on this topic is provided by IEEE Std 1471™ [B26].

A.2.2.2.3 Output information

Output information	Destination activity
System Architecture	Allocate System Requirements (A.2.2.3)
	Perform Architectural Design (A.3.2.1)
	Confirm Security Accreditation (A.5.1.8)
Security Requirements	Define and Develop Software Requirements (A.3.1.1)
	Define Interface Requirements (A.3.1.2)

A.2.2.3 Allocate System Requirements

A.2.2.3.1 Input information

Input information	Source activity
Functional Description of the System	Analyze System Functions (A.2.2.1)
System Architecture	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. 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 allocation of the system functions could result in requirements for more than one project. Each software project shall be managed individually.

Additional guidance on this topic is provided by IEEE Std 830 [B9], IEEE Std 1220 [B23], and IEEE Std 1233™ [B25].

Prior to the distribution of Software Requirements and System Interface Requirements, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

d) Develop Training Materials (A.5.4.1)

A.2.2.3.3 Output information

Output information	Destination activity
System Human and Hardware Requirements	External
System Software Functional Requirements	Perform Estimations (A.1.1.2)
	Allocate Project Resources (A.1.1.3)
	Define and Develop Software Requirements (A.3.1.1)
	Define Interface Requirements (A.3.1.2)
System Interface Requirements	External
	Define and Develop Software Requirements (A.3.1.1)
	Define Interface Requirements (A.3.1.2)

A.2.3 Software Importation Activity Group

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. The activities of the Software Importation Activity Group 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.

The activities of the Software Importation Activity Group are

a) Identify Imported Software Requirements (A.2.3.1)

b) Evaluate Software Import Sources (A.2.3.2)

c) Define Software Import Method (A.2.3.3)

d) Import Software (A.2.3.4)

A.2.3.1 Identify Imported Software Requirements

A.2.3.1.1 Input information

Input information	Source activity
SPMPI	Plan Project Management (A.1.2.7)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)

A.2.3.1.2 Description

The Identify Imported Software Requirements Activity extracts the 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.

Additional guidance on this topic is provided by IEEE Std 1062™ [B19].

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

A.2.3.1.3 Output information

Output information	Destination activity
Imported Software Requirements	Most activities of the Project Planning Activity Group (A.1.2)
	Manage Risks (A.1.3.1)
	Evaluate Software Import Sources (A.2.3.2)
	All activities of the Design Activity Group (A.3.2)
	Conduct Reviews (A.5.1.1)
	Create Test Data (A.5.1.5)
	Confirm Security Accreditation (A.5.1.8)

A.2.3.2 Evaluate Software Import Sources

A.2.3.2.1 Input information

Input information	Source activity
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)

A.2.3.2.2 Description

The Evaluate Software Import Sources 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, schedule, cost, and the software quality and security programs of the source. The effects on the target operating environment, 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 turnkey 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.

Additional guidance on this topic is provided by IEEE Std 1063 [B20].

A.2.3.2.3 Output information

Output information	Destination activity
Selected Software Import Sources	Manage the Project (A.1.3.2)
	Define Software Import Method (A.2.3.3)
	Import Software (A.2.3.4)
Candidate Software Import Methods	Define Software Import Method (A.2.3.3)

A.2.3.3 Define Software Import Method

A.2.3.3.1 Input information

Input information	Source activity
Selected Software Import Sources	Evaluate Software Import Sources (A.2.3.2)
Candidate Software Import Methods	Evaluate Software Import Sources (A.2.3.2)

A.2.3.3.2 Description

Using the listed input information, the Define Software Import Method 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.

Additional guidance on this topic is provided by IEEE Std 1062 [B19].

A.2.3.3.3 Output information

Output information	Destination activity
Selected Software Import Methods	Import Software (A.2.3.4)

A.2.3.4 Import Software

A.2.3.4.1 Input information

Input information	Source activity
Selected Software Import Sources	Evaluate Software Import Sources (A.2.3.2)
Selected Software Import Methods	Define Software Import Method (A.2.3.3)

A.2.3.4.2 Description

The Import Software Activity brings the imported components into the software project in a controlled manner that assures their orderly integration into the total software system. The imported software shall be integrated into the design as well as the implementation.

Additional guidance on this topic is provided by IEEE Std 1062 [B19].

Prior to the distribution of the Imported Software, the following activity groups shall be invoked:

a) Execute Tests (A.5.1.6)

b) Perform Configuration Control (A.5.2.2)

c) Develop Training Materials (A.5.4.1)

A.2.3.4.3 Output information

Output information	Destination activity
Imported Software	Perform Integration (A.3.3.3)
	Execute Tests (A.5.1.6)
Imported Software Documentation	Perform Detailed Design (A.3.2.4)
	Conduct Reviews (A.5.1.1)
	Confirm Security Accreditation (A.5.1.8)
	Implement Documentation (A.5.3.1)
	Develop Training Materials (A.5.4.1)

A.3 Development Section of activity groups

The activity groups of the Development Section include the activities performed during the development and enhancement of a software project. The activities listed are not processes. They are required to be mapped onto a software project life cycle model (SPLCM) or joined together by a process architect to become processes appropriately tailored and relevant to an organization or project.

A.3.1 Software Requirements Activity Group

The Software Requirements Activity Group includes the activities that are directed toward the development of software requirements. In the development of a system that contains hardware, human, and software components, the Software Requirements Activity Group refines the software requirements allocated from the total system requirements.

Information discovered during these activities may trigger a need to revisit the System Allocation Activity Group (A.2.2).

The activities of the Software Requirements Activity Group are

a) Define and Develop Software Requirements (A.3.1.1)

b) Define Interface Requirements (A.3.1.2)

c) Prioritize and Integrate Software Requirements (A.3.1.3)

A.3.1.1 Define and Develop Software Requirements

A.3.1.1.1 Input information

Input information	Source activity
Installation Support Requirements	External
System Constraints	External
SPMPI	Plan Project Management (A.1.2.7)
Risk Management Reported information	Manage Risks (A.1.3.1)
System Architecture	Develop System Architecture (A.2.2.2)
System Software Functional Requirements	Allocate System Requirements (A.2.2.3)
System Interface Requirements	Allocate System Requirements (A.2.2.3)

A.3.1.1.2 Description

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 the input information and the results of modeling, prototyping, or other techniques.

Using the input information for the Define and Develop Software Requirements Activity, the developer shall analyze the software functional and nonfunctional requirements to determine traceability, clarity, validity, testability, and any other project-specific characteristics. The use of a consistent methodology is recommended to assure that requirements are complete and consistent. When needed, the requirements for a database 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.

Additional guidance on this topic is provided by IEEE Std 830 [B9].

Prior to the distribution of the Preliminary Software Requirements and Installation Requirements, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.3.1.1.3 Output information

Output information	Destination activity
Preliminary Software Requirements	Plan Evaluations (A.1.2.1)
	Define Interface Requirements (A.3.1.2)
	Prioritize and Integrate Software Requirements (A.3.1.3)
Installation Requirements	Plan Installation (A.1.2.4)

A.3.1.2 Define Interface Requirements

A.3.1.2.1 Input information

Input information	Source activity
System Constraints	External
SPMPI	Plan Project Management (A.1.2.7)
Functional Description of the System	Analyze System Functions (A.2.2.1)
System Architecture	Develop System Architecture (A.2.2.2)
System Software Functional Requirements	Allocate System Requirements (A.2.2.3)
System Interface Requirements	Allocate System Requirements (A.2.2.3)
Preliminary Software Requirements	Define and Develop Software Requirements (A.3.1.1)

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 they 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, database management system, language compiler, tools, utilities, network protocol drivers, and hardware interfaces.

Additional guidance on this topic is provided by IEEE Std 830 [B9] and IEEE Std 1175™ [B21].

Prior to the distribution of the Software Interface Requirements, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.3.1.2.3 Output information

Output information	Destination activity
Software Interface Requirements	Manage Risks (A.1.3.1)
	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
Risk Management Reported Information	Manage Risks (A.1.3.1)
Preliminary Software Requirements	Define and Develop Software Requirements (A.3.1.1)
Software Interface Requirements	Define Interface Requirements (A.3.1.2)

A.3.1.3.2 Description

The functional and nonfunctional requirements shall be reviewed, and a prioritized list of requirements shall be produced that addresses any tradeoffs 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; see A.3.2). The Software Requirements shall describe the functional, interface, and nonfunctional requirements, and they shall also define operational and security support environments.

Additional guidance on this topic is provided by IEEE Std 830 [B9].

Prior to the distribution of the Software Requirements, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.3.1.3.3 Output information

Output information	Destination activity
Software Requirements	Develop SPLCP (A.1.1.1)
	Allocate Project Resources (A.1.1.3)
	Define Metrics (A.1.1.4)
	Plan Evaluations (A.1.2.1)
	Plan Training (A.1.2.6)
	Plan Integration (A.1.2.8)
	Manage Risks (A.1.3.1)
	Identify Imported Software Requirements (A.2.3.1)
	All activities of the Design Activity Group (A.3.2)
	Create Operating Documentation (A.3.3.2)
	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)
	Confirm Security Accreditation (A.5.1.8)

A.3.2 Design Activity Group

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 constitute the software system and on 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 activities of the Design Activity Group can be carried out in parallel with these activities.

The activities of the Design Activity Group are

a) Perform Architectural Design (A.3.2.1)

b) Design Database (if applicable) (A.3.2.2)

c) Design Interfaces (A.3.2.3)

d) Perform Detailed Design (A.3.2.4)

A.3.2.1 Perform Architectural Design

A.3.2.1.1 Input information

Input information	Source activity
SPMPI	Plan Project Management (A.1.2.7)
System Architecture	Develop System Architecture (A.2.2.2)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)
Software 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. Principles supporting "design for testability" should be taken into account when making design decisions.

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, security controls, and constraints shall be specified. All internal interfaces (among components) shall be defined. This activity shall create the Software Architectural Design.

Additional guidance on this topic is provided by IEEE Std 1220 [B23] and IEEE Std 1471 [B26].

Prior to the distribution of the Software Architectural Design, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.3.2.1.3 Output information

Output information	Destination activity
Software Architectural Design	Perform Detailed Design (A.3.2.4)

A.3.2.2 Design Database

A.3.2.2.1 Input information

Input information	Source activity
SPMPI	Plan Project Management (A.1.2.7)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)

A.3.2.2.2 Description

The Design Database Activity shall specify the information structure that is outlined in the Software Requirements and its characteristics within the software system. This activity involves three separate but related steps: conceptual database design, logical database design, and physical database design. It does not involve designing or developing the database management system.

Techniques such as data dictionary creation, database optimization, and data and data security modeling should be considered. Requirements are molded into an external schema that describes data entities, attributes, relationships, protections, and constraints. The various external schemata are integrated into a single conceptual schema. The conceptual schema is then mapped onto 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 Database Design.

Prior to the distribution of the Database Design, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.3.2.2.3 Output information

Output information	Destination activity
Database 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
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)
Software 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) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.3.2.3.3 Output information

Output information	Destination activity
Interface 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
SPMPI	Plan Project Management (A.1.2.7)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)
Imported Software Documentation	Import Software (A.2.3.4)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Architectural Design	Perform Architectural Design (A.3.2.1)
Database Design	Design Database (if applicable) (A.3.2.2)
Interface Design	Design Interfaces (A.3.2.3)

A.3.2.4.2 Description

In the Perform Detailed Design 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 this activity's input information. The details of the interfaces shall be identified within the Software Detailed Design. Principles supporting "design for testability" should be taken into account when making design decisions.

Additional guidance on this topic is provided by IEEE Std 1016™ [B13].

Prior to the distribution of the Software Detailed Design, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.3.2.4.3 Output information

Output information	Destination activity
Software Detailed Design	Plan Evaluations (A.1.2.1)
	Plan Integration (A.1.2.8)
	Manage Risks (A.1.3.1)
	Create Executable Code (A.3.3.1)
	Create Operating Documentation (A.3.3.2)
	Develop Test Procedures (A.5.1.4)
	Create Test Data (A.5.1.5)
	Confirm Security Accreditation (A.5.1.8)
	Develop Training Materials (A.5.4.1)

A.3.3 Implementation Activity Group

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 Database (if applicable), and the documentation that constitutes the

physical manifestation of the design. In addition, the code and database are integrated. Care should also be taken during the activities of the Implementation Activity Group to apply the appropriate coding standards.

The code and database, along with the documentation that was produced within activities of previous activity groups, are the first complete representation of the software product.

The activities of the Implementation Activity Group are

a) Create Executable Code (A.3.3.1)
b) Create Operating Documentation (A.3.3.2)
c) Perform Integration (A.3.3.3)
d) Manage Software Releases (A.3.3.4)

A.3.3.1 Create Executable Code

A.3.3.1.1 Input information

Input information	Source activity
Software Coding Standards	External
SPMPI	Plan Project Management (A.1.2.7)
Software Detailed 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 SPLCP, as found in the SPMPI, and the Software Detailed Design. Coding standards, if available, shall be applied. If the Source Code is required for integration, it shall be made available to the Perform Integration Activity (A.3.3.3). If the Source Code is going to be used to create test data, it shall be made available to Create Test Data Activity (A.5.1.5).

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.

Critical software demands that coding constructs conform to programming standards.

Additional guidance on this topic is provided by IEEE Std 1008 [B11].

Prior to the distribution of the Source Code, Executable Code, and Database, the following activities shall be invoked:

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

A.3.3.1.3 Output information

Output information	Destination activity
Source Code (when required)	Perform Integration (A.3.3.3)
Source Code (when required)	Create Test Data (A.5.1.5)
Executable Code	Perform Integration (A.3.3.3)
	Execute Tests (A.5.1.6)
	Confirm Security Accreditation (A.5.1.8)
Database	Perform Integration (A.3.3.3)
	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
Documentation Planned Information	Plan Documentation (A.1.2.5)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Detailed Design	Perform Detailed Design (A.3.2.4)

A.3.3.2.2 Description

The Create Operating Documentation 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.

Additional guidance on this topic is provided by IEEE Std 1063 [B20].

Prior to the distribution of the Operating Documentation, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.3.3.2.3 Output information

Output information	Destination activity
Operating Documentation	Plan Installation (A.1.2.4)
	Distribute Software (A.4.1.1)

A.3.3.3 Perform Integration

A.3.3.3.1 Input information

Input information	Source activity
System Components	External
SPMPI	Plan Project Management (A.1.2.7)
Integration Planned Information	Plan Integration (A.1.2.8)
Imported Software	Import Software (A.2.3.4)
Source Code (when required)	Create Executable Code (A.3.3.1)
Executable Code	Create Executable Code (A.3.3.1)
Database	Create Executable Code (A.3.3.1)
Stubs and Drivers	Develop Test Procedures (A.5.1.4)
Tested Software	Execute Tests (A.5.1.6)

A.3.3.3.2 Description

The Perform Integration Activity shall integrate the Database, Source Code (when required), Executable Code, and Stubs and Drivers, as specified in the Integration Planned Information, into the Integrated Software. Other necessary Executable Code, from the SPLCP 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) Conduct Reviews (A.5.1.1)

b) Execute Tests (A.5.1.6)

c) Perform Configuration Control (A.5.2.2)

A.3.3.3.3 Output information

Output information	Destination activity
Integrated Software	Execute Tests (A.5.1.6)

A.3.3.4 Manage Software Releases

A.3.3.4.1 Input information

Input information	Source activity
OPAs	External
SRMPI	Plan Release Management (A.1.2.9)
Configuration Identification	Develop Configuration Identification (A.5.2.1)

A.3.3.4.2 Description

The Manage Software Release Activity includes the building, naming, packaging, and release of a particular version of a software product, including its associated release notes, user documentation, etc.

Software building compiles the correct versions of software items, using the appropriate configuration information, into version-identified (named) executable object code that can be delivered to testing, to a system build activity, or to customers, depending on the purpose of the release. Software packaging includes arranging the executable code and documentation into the appropriate directory structures. Software release includes delivery to a predesignated destination of the packaged software, making it available for media manufacturing (for shrink-wrapped software products) or for other modes of distribution and installation.

Prior to the release of the software package, the following activities shall be invoked:

a) Execute Tests (A.5.1.6)

b) Report Evaluation Results (A.5.1.7)

c) Develop Configuration Identification (A.5.2.1)

d) Perform Configuration Control (A.5.2.2)

A.3.3.4.3 Output information

Output information	Destination activity
Released Product Package	All activities of the Installation Activity Group (A.4.1)

A.4 Post-Development Section of activity groups

The activity groups of the Post-Development Section include the activities that install, operate, support, maintain, and retire a software product. The activities listed are not processes. They shall be mapped onto a software project life cycle model (SPLCM) or joined together by a process architect to become processes appropriately tailored and relevant to an organization or project.

A.4.1 Installation Activity Group

The Installation Activity Group 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.

The activities of the Installation Activity Group are

a) Distribute Software (A.4.1.1)

b) Install Software (A.4.1.2)

c) Accept Software in Operational Environment (A.4.1.3)

A.4.1.1 Distribute Software

A.4.1.1.1 Input information

Input information	Source activity
Database Data	External
Software Installation Planned Information	Plan Installation (A.1.2.4)
SPMPI	Plan Project Management (A.1.2.7)
Operating Documentation	Create Operating Documentation (A.3.3.2)
Released Product Package (electronic files)	Manage Software Releases (A.3.3.4)
Tested Software	Execute Tests (A.5.1.6)

A.4.1.1.2 Description

During the Distribute Software Activity, the Tested Software, Database 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 Installation Planned Information, Packaged Software, and Packaged Operating Documentation, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Conduct Audits (A.5.1.3)

c) Perform Configuration Control (A.5.2.2)

d) Implement Documentation (A.5.3.1)

A.4.1.1.3 Output information

Output information	Destination activity
Packaged Installation Planned Information	Install Software (A.4.1.2)
Packaged Software	Install Software (A.4.1.2)
Packaged Operating Documentation	Install Software (A.4.1.2)
	Operate the System (A.4.2.1)

A.4.1.2 Install Software

A.4.1.2.1 Input information

Input information	Source activity
Database Data	External
Packaged Installation Planned Information	Distribute Software (A.4.1.1)
Packaged Operating Documentation	Distribute Software (A.4.1.1)
Packaged Software	Distribute Software (A.4.1.1)
Released Product Package (electronic files)	Manage Software Releases (A.3.3.4)

A.4.1.2.2 Description

The security capabilities of the operating environment shall be baselined, and the availability of required resources shall be verified prior to installation. The Packaged Software and any required Database 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. The security of the Installed Software shall be verified.

A.4.1.2.3 Output information

Output information	Destination activity
Installation Reported Information	Manage the Project (A.1.3.2)
	Accept Software in Operational Environment (A.4.1.3)
Installed Software	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
User Acceptance Planned Information	External
Released product package (electronic files)	Manage Software Releases (A.3.3.4)
Installation Reported Information	Install Software (A.4.1.2)
Installed Software	Install Software (A.4.1.2)
Evaluation Reported Information	Report Evaluation Results (A.5.1.7)
Security Accreditation	Confirm Security Accreditation (A.5.1.8)

A.4.1.3.2 Description

The Accept Software in Operational Environment Activity shall consist of an analysis of the Evaluation Reported Information and the Installation Reported Information according to the User Acceptance Planned Information to assure 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.

Once the software has been accepted, the software development completion project information is collected, placed in the Historical Project Records, and saved for input to future projects.

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

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

A.4.1.3.3 Output information

Output information	Destination activity
Customer Acceptance	External
	Close Project (A.1.3.6)

Output information	Destination activity
Historical Project Records	External
Installed Software System	Operate the System (A.4.2.1)
	Conduct Parallel Operations (A.4.4.2)

A.4.2 Operation and Support Activity Group

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 reenters the SPLCP.

The activities of the Operation and Support Activity Group are

a) Operate the System (A.4.2.1)

b) Provide Technical Assistance and Consulting (A.4.2.2)

c) Maintain Support Request Log (A.4.2.3)

A.4.2.1 Operate the System

A.4.2.1.1 Input information

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

A.4.2.1.2 Description

During the Operate the System 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) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

A.4.2.1.3 Output information

Output information	Destination activity
Operation Logs	External
Anomalies	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
Request for Support	External
Support Planned Information	Plan Project Management (A.1.2.7)

A.4.2.2.2 Description

The Provide Technical Assistance and Consulting 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 Activity (A.4.2.3) so that feedback can be provided to other activity groups.

A.4.2.2.3 Output information

Output information	Destination activity
Support Response	External
	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
Support Planned Information	Plan Project Management (A.1.2.7)
Support Response	Provide Technical Assistance and Consulting (A.4.2.2)

A.4.2.3.2 Description

The Maintain Support Request Log 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 suspected of affecting corporate security shall be escalated, thoroughly explored, and, if warranted, immediately addressed.

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

A.4.2.3.3 Output information

Output information	Destination activity
Anomalies	Implement Problem Reporting Method (A.4.3.2)
Support Request Log	Conduct Reviews (A.5.1.1)

A.4.3 Maintenance Activity Group

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 SPLCP changes.

The SPLCP is remapped and executed, thereby treating the Maintenance Activity Group as iterations of development.

The activities of the Maintenance Activity Group are

a) Identify Software Improvement Needs (A.4.3.1)
b) Implement Problem Reporting Method (A.4.3.2)
c) Reapply SPLCP (A.4.3.3)

A.4.3.1 Identify Software Improvement Needs

A.4.3.1.1 Input information

Input information	Source activity
Security Objectives	Determine Security Objectives (A.1.1.5)
Evaluation Planned Information	Plan Evaluations (A.1.2.1)
Training Planned Information	Plan Training (A.1.2.6)
SPMPI	Plan Project Management (A.1.2.7)
Analysis Reported Information	Collect and Analyze Metric Data (A.1.3.5)
Post-Operation Review Reported Information	Retire System (A.4.4.3)
Evaluation Reported Information	Report Evaluation Results (A.5.1.7)

A.4.3.1.2 Description

The Identify Software Improvement Needs Activity identifies lessons learned and needs for software improvements, and it outputs the Software Improvement Recommendations in accordance with the SPMPI. This is accomplished by using this activity's 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.

Additional guidance on this topic is provided by IEEE Std 1219™ [B22].

A.4.3.1.3 Output information

Output information	Destination activity
Software Improvement Recommendations	External
	Manage the Project (A.1.3.2)
	Identify SPLCP Improvement Needs (A.1.3.3)
	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
Anomalies	External
	Creating activity
Security Objectives	Determine Security Objectives (A.1.1.5)

Input information	Source activity
PR&RPI	Plan Project Management (A.1.2.7)
Environment Improvement Needs	Identify SPLCP Improvement Needs (A.1.3.3)
Software Improvement Recommendations	Identify Software Improvement Needs (A.4.3.1)
Evaluation Reported Information	Report Evaluation Results (A.5.1.7)
Controlled Item	Perform Configuration Control (A.5.2.2)

A.4.3.2.2 Description

The Implement Problem Reporting Method 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 assure 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) Anomalies

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.

Additional guidance on this topic is provided by IEEE Std 1044 [B15] and IEEE Std 1219 [B22].

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

A.4.3.2.3 Output information

Output information	Destination activity
Out-of-Scope Anomalies	External
Report Log	Collect and Analyze Metric Data (A.1.3.5)
	Close Project (A.1.3.6)
	Reapply SPLCP (A.4.3.3)
Enhancement Problem Reported Information	External
	Collect and Analyze Metric Data (A.1.3.5)
	Close Project (A.1.3.6)
	Reapply SPLCP (A.4.3.3)

Output information	Destination activity
Correction Problem Reported Information	Collect and Analyze Metric Data (A.1.3.5)
	Close Project (A.1.3.6)
	Reapply SPLCP (A.4.3.3)

A.4.3.3 Reapply SPLCP

A.4.3.3.1 Input information

Input information	Source activity
Security Objectives	Determine Security Objectives (A.1.1.5)
SPMPI	Plan Project Management (A.1.2.7)
PR&RPI	Plan Project Management (A.1.2.7)
Enhancement Problem Reported Information	Implement Problem Reporting Method (A.4.3.2)
Correction Problem Reported Information	Implement Problem Reporting Method (A.4.3.2)
Report Log	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 be entered into the SPLCP at the Concept Exploration Activity Group (A.2.1) 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 destinations.

The Reapply SPLCP Activity shall monitor the problem correction efforts that are performed by the responsible activity group, shall determine (according to the Enhancement Problem Reported Information 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 Updated Report Log should reflect the corrective action taken.

A.4.3.3.3 Output information

Output information	Destination activity
Maintenance Recommendations	External
	Develop SPLCP (A.1.1.1)
	Close Project (A.1.3.6)
Resolved Problem Reported Information	External
	Creating activity
	Collect and Analyze Metric Data (A.1.3.5)
	Close Project (A.1.3.6)
	Conduct Reviews (A.5.1.1)
	Create Test Data (A.5.1.5)
	Report Evaluation Results (A.5.1.7)

Output information	Destination activity
Updated Report Log	Collect and Analyze Metric Data (A.1.3.5)
	Close Project (A.1.3.6)

A.4.4 Retirement Activity Group

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.

The activities of the Retirement Activity Group are

a) Notify User (A.4.4.1)

b) Conduct Parallel Operations (if applicable) (A.4.4.2)

c) Retire System (A.4.4.3)

A.4.4.1 Notify User

A.4.4.1.1 Input information

Input information	Source activity
Retirement Planned Information	Plan Project Management (A.1.2.7)

A.4.4.1.2 Description

The Notify User 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 shall 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 about why the system is not being supported

c) A description of possible other support

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

A.4.4.1.3 Output information

Output information	Destination activity
Official Notification	External
	Retain Records (A.1.3.4)

A.4.4.2 Conduct Parallel Operations

A.4.4.2.1 Input information

Input information	Source activity
Transition Planned Information (for the replacing system)	External
Retirement Planned Information	Plan Project Management (A.1.2.7)
Installed Software System	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, the Conduct Parallel Operations 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) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Develop Training Materials (A.5.4.1)

A.4.4.2.3 Output information

Output information	Destination activity
Parallel Operations Log	Retain Records (A.1.3.4)

A.4.4.3 Retire System

A.4.4.3.1 Input information

Input information	Source activity
Security Objectives	Determine Security Objectives (A.1.1.5)
Retirement Planned Information	Plan Project Management (A.1.2.7)

A.4.4.3.2 Description

The Retire System 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) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.4.4.3.3 Output information

Output information	Destination activity
Archive Reported Information	External
Post-Operation Review Reported Information	External
	Identify SPLCP Improvement Needs (A.1.3.3)
	Retain Records (A.1.3.4)
	Identify Software Improvement Needs (A.4.3.1)

A.5 Support Section of activity groups

The activity groups of the Support Section include the activities that are needed to successfully complete project activities. These activities are utilized to assure the completion and quality of project functions. The activities listed are not processes. They shall be mapped onto a software project life cycle model (SPLCM) or joined together by a process architect to become processes appropriately tailored and relevant to an organization or project.

A.5.1 Evaluation Activity Group

The Evaluation Activity Group includes the activities performed during the SPLCP 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 activities in the Evaluation Activity Group.

Because exacting details of the activities of the Evaluation Activity Group 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 activity of the Evaluation Activity Group needs to be applied to each of its instances in the SPLCP. Consider, for example, an SPLCP 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.

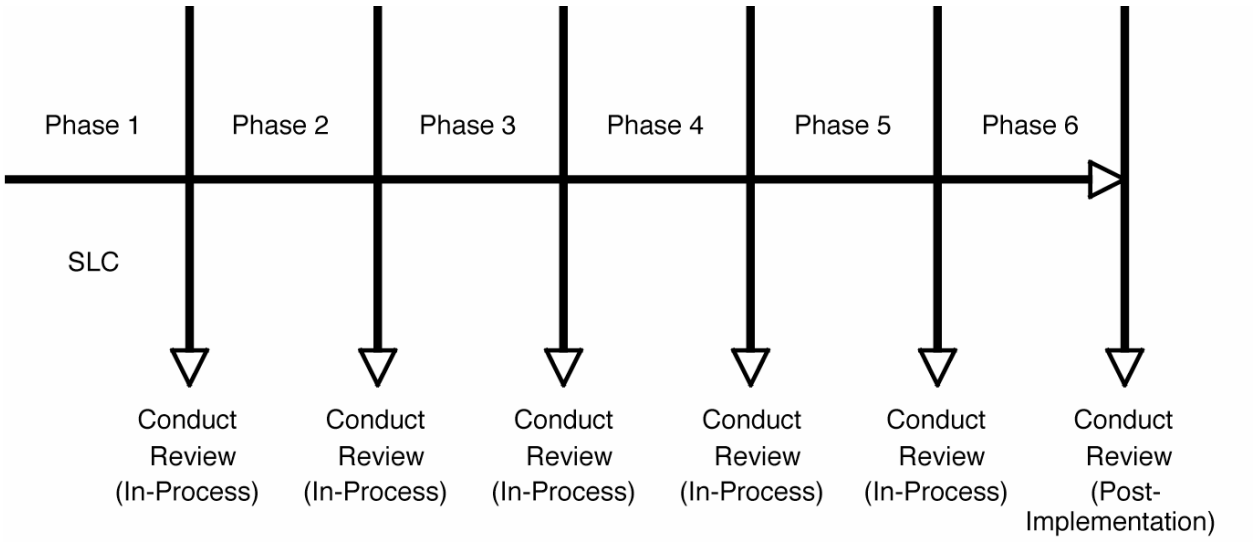

Figure A.1—Mapping reviews

The activities of the Evaluation Activity Group are

a) Conduct Reviews (A.5.1.1)
b) Create Traceability Matrix (A.5.1.2)
c) Conduct Audits (A.5.1.3)
d) Develop Test Procedures (A.5.1.4)
e) Create Test Data (A.5.1.5)
f) Execute Tests (A.5.1.6)
g) Report Evaluation Results (A.5.1.7)
h) Confirm Security Accreditation (A.5.1.8)

A.5.1.1 Conduct Reviews (Required)

A.5.1.1.1 Input information

Input information	Source activity
Review Standards and Guidelines	External
Item to be Reviewed	Creating activity
Evaluation Planned Information	Plan Evaluations (A.1.2.1)
SPMPI	Plan Project Management (A.1.2.7)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)
Imported Software Documentation	Import Software (A.2.3.4)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Support Request Log	Maintain Support Request Log (A.4.2.3)
Resolved Problem Reported Information	Reapply SPLCP (A.4.3.3)
Traceability Matrix	Create Traceability Matrix (A.5.1.2)
Audit Results Information	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 five broad categories:

a) *In-process reviews:* These reviews shall be held to remove defects from software requirements, preliminary designs, detailed designs, code, and documentation. These 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, nonfunctional, and security requirements shall also be reviewed constantly throughout the life cycle to assure 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:* These reviews of the products and the security and quality systems 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 SPLCP phase, they are also referred to as *phase-end reviews*. Management Status Reported Information is produced after these reviews.

c) *Operational readiness reviews*: These reviews shall be held, usually close to the release date, to verify that all requirements have been successfully met to assure that the enterprise is fully prepared to accept and support the product in operation.

d) *Process improvement reviews:* These reviews shall be held to evaluate metrics from the development effort in order to determine whether 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.

e) *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.

Additional guidance on this topic is provided by IEEE Std 730 [B6], IEEE Std 1012 [B12], IEEE Std 1028 [B14].

Prior to the distribution of this activity's output information or the Archive Reported Information from the Retire System Activity (A.4.4.3), the Perform Configuration Control Activity (A.5.2.2) may be invoked.

A.5.1.1.3 Output information

Output information	Destination activity
In-Process Review Results	Report Evaluation Results (A.5.1.7)
Post-Implementation Review Reported Information	Report Evaluation Results (A.5.1.7)
Process Improvement Recommendations	Report Evaluation Results (A.5.1.7)
Management Status Reported Information	Report Evaluation Results (A.5.1.7)
Traceability Analysis Reported Information	Report Evaluation Results (A.5.1.7)
System Allocation Change Reported Information	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
Project-Specific Technical Requirements	External
Software 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, the testing to be applied to each requirement, and the associated requirement item identification. This shall allow the matrix to be reviewed during each in-process or management review in order to assure 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, forward or backward.

The traceability matrix should also be used to assist in impact assessment as requirements change.

In addition, traceability between project documentation and the information created by each activity is to be maintained. This traceability is also to be reviewed.

Additional guidance on this topic is provided by IEEE Std 1012 [B12].

Prior to the distribution of the Traceability Matrix, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.5.1.2.3 Output information

Output information	Destination activity
Traceability Matrix	Conduct Reviews (A.5.1.1)
	Develop Configuration Identification (A.5.2.1)

A.5.1.3 Conduct Audits

A.5.1.3.1 Input information

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

A.5.1.3.2 Description

Audits shall be performed by examiners who are organizationally independent of the project. The purpose of the Conduct Audits Activity is to assess the compliance of the products or processes with specification requirements, various SPLCP 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.

Additional guidance on this topic is provided by IEEE Std 730 [B6], IEEE Std 1012 [B12], and IEEE Std 1028 [B14].

A.5.1.3.3 Output information

Output information	Destination activity
Audit Results Information	Creating activity
	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
Evaluation Planned Information	Plan Evaluations (A.1.2.1)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Detailed Design	Perform Detailed Design (A.3.2.4)

A.5.1.4.2 Description

Test Procedures for each level of testing (e.g., unit/module/component, integration, acceptance, regression, 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 the type of tests to be conducted (e.g., white box, black box, destructive, noninvasive), the items 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. 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.

Additional guidance on this topic is provided by IEEE Std 829 [B8], IEEE Std 1008 [B11], IEEE Std 1012 [B12], and IEEE Std 1220 [B23].

Prior to the distribution of the Test Procedures, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.5.1.4.3 Output information

Output information	Destination activity
Test Procedures	Create Test Data (A.5.1.5)
	Execute Tests (A.5.1.6)
Stubs and Drivers	Perform Integration (A.3.3.3)
	Execute Tests (A.5.1.6)

A.5.1.5 Create Test Data

A.5.1.5.1 Input information

Input information	Source activity
Evaluation Planned Information	Plan Evaluations (A.1.2.1)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Detailed Design	Perform Detailed Design (A.3.2.4)
Source Code (when required)	Create Executable Code (A.3.3.1)
Database	Create Executable Code (A.3.3.1)
Resolved Problem Reported Information	Reapply SPLCP (A.4.3.3)
Test Procedures	Develop Test Procedures (A.5.1.4)

A.5.1.5.2 Description

Test Procedures define the type of data to be used in testing. Using the Software Requirements, the Software Detailed Design, and the Source Code (when required), Test Data shall be generated for all defined tests. In the case of regression testing, defect scenarios and data from previously failed tests and feedback from 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.

Additional guidance on this topic is found in IEEE Std 829 [B8] and IEEE Std 1008 [B11].

A.5.1.5.3 Output information

Output information	Destination activity
Test Data	Execute Tests (A.5.1.6)

A.5.1.6 Execute Tests

A.5.1.6.1 Input information

Input information	Source activity
Test Environment Components	External
Evaluation Planned Information	Plan Evaluations (A.1.2.1)
Imported Software	Import Software (A.2.3.4)
Executable Code	Create Executable Code (A.3.3.1)
Integrated Software	Perform Integration (A.3.3.3)

Input information	Source activity
Test Procedures	Develop Test Procedures (A.5.1.4)
Test Data	Create Test Data (A.5.1.5)
Stubs and Drivers	Develop Test Procedures (A.5.1.4)

A.5.1.6.2 Description

The Execute Tests 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. Any Anomalies that occur during test execution that require 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 the 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.

Additional guidance on this topic is provided by IEEE Std 829 [B8] and IEEE Std 1008 [B11].

Prior to the distribution of the Test Summary Reported Information, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

c) Implement Documentation (A.5.3.1)

A.5.1.6.3 Output information

Output information	Destination activity
Test Summary Reported Information	External
	Report Evaluation Results (A.5.1.7)
Tested Software	Perform Integration (A.3.3.3)
	Distribute Software (A.4.1.1)
Anomalies	Implement Problem Reporting Method (A.4.3.2)
	Report Evaluation Results (A.5.1.7)

A.5.1.7 Report Evaluation Results (Required)

A.5.1.7.1 Input information

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

A.5.1.7.2 Description

The Report Evaluation Results 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, the Conduct Reviews Activity (A.5.1.1) may be invoked.

A.5.1.7.3 Output information

Output information	Destination activity
Evaluation Reported Information	Creating activity
	Manage Risks (A.1.3.1)
	Manage the Project (A.1.3.2)
	Identify SPLCP Improvement Needs (A.1.3.3)
	Collect and Analyze Metric Data (A.1.3.5)
	Accept Software in Operational Environment (A.4.1.3)
	Identify Software Improvement Needs (A.4.3.1)
	Implement Problem Reporting Method (A.4.3.2)
	Confirm Security Accreditation (A.5.1.8)
Anomalies	Implement Problem Reporting Method (A.4.3.2)
	Confirm Security Accreditation (A.5.1.8)

A.5.1.8 Confirm Security Accreditation (Required)

A.5.1.8.1 Input information

Input information	Source activity
Security Objectives	Determine Security Objectives (A.1.1.5)
Evaluation Planned Information	Plan Evaluations (A.1.2.1)
Risk Management Reported Information	Manage Risks (A.1.3.1)
System Architecture	Develop System Architecture (A.2.2.2)
Imported Software Requirements	Identify Imported Software Requirements (A.2.3.1)
Imported Software Documentation	Import Software (A.2.3.4)
Software Requirements	Prioritize and Integrate Software Requirements (A.3.1.3)
Software Detailed Design	Perform Detailed Design (A.3.2.4)
Executable Code	Create Executable Code (A.3.3.1)
Evaluation Reported Information	Report Evaluation Results (A.5.1.7)
Anomalies	Report Evaluation Results (A.5.1.7)

A.5.1.8.2 Description

The Confirm Security Accreditation Activity shall provide the formal authority to operate. Project documentation shall be fully reviewed and approved in writing by an authorized decision authority accepting the risk for the operation. All project requirements shall be referenced within, and traceable to, the Security Accreditation documentation. Evidence that security objectives have been met and have been verified to conform to the applicable external protection profile (if applicable) or, alternatively, to the predefined level of acceptable risk shall be included.

Additional guidance on this topic is provided by ISO 15408 [B30].

Prior to the distribution of the Security Accreditation, the Conduct Audits Activity (A.5.1.3) may be invoked.

A.5.1.8.3 Output information

Output information	Destination activity
Security Accreditation	External
	Accept Software in Operational Environment (A.4.1.3)

A.5.2 Software Configuration Management Activity Group

The Software Configuration Management Activity Group identifies the items in a software development project and provides both for control of the identified items and for the generation of status information for management visibility and accountability throughout the SPLC. Items to be managed 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 (A.5.1). The software configuration management approach for a given project should be compatible with the configuration management approach that is being used on associated systems.

The activities of the Software Configuration Management Activity Group are

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

A.5.2.1 Develop Configuration Identification (Required)

A.5.2.1.1 Input information

Input information	Source activity
Deliverable List	External
SCMPI	Plan Configuration Management (A.1.2.2)
SPMPI	Plan Project Management (A.1.2.7)
SRMPI	Plan Release Management (A.1.2.9)
Traceability Matrix	Create Traceability Matrix (A.5.1.2)

A.5.2.1.2 Description

The Develop Configuration Identification 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 the configuration items that are to be addressed by the Configuration Identification. The identification shall support the software throughout the SPLC 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 as well as to provide a record of changes requested, approved, disapproved, and implemented.

All project products, including both internal (e.g., project planning information) and external (e.g., code) project deliverables, are subject to configuration management.

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. Traceability shall also connect requirements to verification methods, plans, and procedures.

Additional guidance on this topic is provided by IEEE Std 828 [B7] and IEEE Std 1028 [B14].

Prior to the distribution of the Configuration Identification, the following activities shall be invoked:

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

A.5.2.1.3 Output information

Output information	Destination activity
Configuration Identification	Plan Evaluations (A.1.2.1)
	Plan Release Management (A.1.2.9)
	Manage Software Releases (A.3.3.4)
	Perform Configuration Control (A.5.2.2)
	Perform Status Accounting (A.5.2.3)

A.5.2.2 Perform Configuration Control (Required)

A.5.2.2.1 Input information

Input information	Source activity
Items to be Controlled	Creating activity
Proposed Change	Creating activity
SCMPI	Plan Configuration Management (A.1.2.2)
SRMPI	Plan Release Management (A.1.2.9)
System Allocation Change Reported Information	Conduct Reviews (A.5.1.1)
Configuration Identification	Develop Configuration Identification (A.5.2.1)

A.5.2.2.2 Description

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

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

Additional guidance on this topic is provided by IEEE Std 828 [B7].

A.5.2.2.3 Output information

Output information	Destination activity
Controlled Item	Creating activity
	Implement Problem Reporting Method (A.4.3.2)
Change Status	Perform Status Accounting (A.5.2.3)

A.5.2.3 Perform Status Accounting (Required)

A.5.2.3.1 Input information

Input information	Source activity
SCMPI	Plan Configuration Management (A.1.2.2)
SRMPI	Plan Release Management (A.1.2.9)

Input information	Source activity
Configuration Identification	Develop Configuration Identification (A.5.2.1)
Change Status	Perform Configuration Control (A.5.2.2)

A.5.2.3.2 Description

The Perform Status Accounting Activity shall receive Configuration Identification and Change Status and shall create and update the Status Reported Information to reflect the status and history of each Controlled Item. The history of changes to each Controlled Item shall be maintained throughout the SPLC 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.

Additional guidance on this topic is provided by IEEE Std 828 [B7].

Prior to the distribution of the Status Reported Information, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

A.5.2.3.3 Output information

Output information	Destination activity
Controlled Item	Creating activity
Status Reported Information	External
	Manage the Project (A.1.3.2)

A.5.3 Documentation Development Activity Group

The Documentation Development Activity Group for software development and usage is the set of activities that plan, design, implement, edit, produce, distribute, and maintain the documents that are needed by developers and users. The purpose of the Documentation Development Activity Group is to provide timely software documentation to users 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 SPLCP, depending on the individual document and the timing of its development. Typically, there will be multiple documents, each at a different stage of development.

The activities of the Documentation Development Activity Group are

a) Implement Documentation (A.5.3.1)

b) Produce and Distribute Documentation (A.5.3.2)

A.5.3.1 Implement Documentation (Required)

A.5.3.1.1 Input information

Input information	Source activity
Input Information for Document	Creating activity
Documentation Planned Information	Plan Documentation (A.1.2.5)
SPMPI	Plan Project Management (A.1.2.7)
Imported Software Documentation	Import Software (A.2.3.4)

A.5.3.1.2 Description

The Implement Documentation Activity includes the design, preparation, and maintenance of documentation. 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.

The input information for this activity 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) Conduct Reviews (A.5.1.1)

b) Perform Configuration Control (A.5.2.2)

A.5.3.1.3 Output information

Output information	Destination activity
Document	Produce and Distribute Documentation (A.5.3.2)

A.5.3.2 Produce and Distribute Documentation (Required)

A.5.3.2.1 Input information

Input information	Source activity
Documentation Planned Information	Plan Documentation (A.1.2.5)
Document	Implement Documentation (A.5.3.1)

A.5.3.2.2 Description

The Produce and Distribute Documentation 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
Published Document	External
	Creating activity
	Retain Records (A.1.3.4)

A.5.4 Training Activity Group

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 information be completed early in the SPLC, prior to the time when personnel would be expected to apply the required expertise to the project. Plans for customer training should be prepared and reviewed with the customer.

The activities of the Training Activity Group are

a) Develop Training Materials (A.5.4.1)

b) Validate the Training Program (A.5.4.2)

c) Implement the Training Program (A.5.4.3)

A.5.4.1 Develop Training Materials

A.5.4.1.1 Input information

Input information	Source activity
Applicable Information	External
	Creating activity
Training Planned Information	Plan Training (A.1.2.6)
SPMPI	Plan Project Management (A.1.2.7)
Imported Software Documentation	Import Software (A.2.3.4)
Software Detailed Design	Perform Detailed Design (A.3.2.4)

A.5.4.1.2 Description

The Develop Training Materials Activity shall consist of the identification and review of all available materials and input information that are 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) Conduct Reviews (A.5.1.1)

b) Implement Documentation (A.5.3.1)

The Perform Configuration Control Activity (A.5.2.2) should also be invoked.

A.5.4.1.3 Output information

Output information	Destination activity
Training Manual	Validate the Training Program (A.5.4.2)
Training Materials	Validate the Training Program (A.5.4.2)
Prepared Presentations	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
Training Planned Information	Plan Training (A.1.2.6)
Training Manual	Develop Training Materials (A.5.4.1)
Training Materials	Develop Training Materials (A.5.4.1)
Prepared Presentations	Develop Training Materials (A.5.4.1)

A.5.4.2.2 Description

The Validate the Training Program 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 of this activity 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 Manual and Updated Training Materials, the following activities shall be invoked:

a) Conduct Reviews (A.5.1.1)

b) Implement Documentation (A.5.3.1)

The Perform Configuration Control Activity (A.5.2.2) should be also invoked.

A.5.4.2.3 Output information

Output information	Destination activity
Training Feedback	Plan Training (A.1.2.6)
Updated Training Manual	Implement the Training Program (A.5.4.3)
Updated Training Materials	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
Staff Participants	External
Students	External
Training Planned Information	Plan Training (A.1.2.6)
Updated Training Manual	Validate the Training Program (A.5.4.2)
Updated Training Materials	Validate the Training Program (A.5.4.2)

A.5.4.3.2 Description

The Implement the Training Program Activity shall assure 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 all activities of the Training Activity Group.

A.5.4.3.3 Output information

Output information	Destination activity
Updated Skills Inventory	External
Trained Personnel	Creating activity
Training Feedback	Plan Training (A.1.2.6)

Annex B

(informative)

Mapping example

The purpose of this example is to show the mapping process, as presented in Clause 4, without constraining the reader to any specific methodologies or tools.

For purposes of this illustration, an adaptive software development process model has been selected. It is understood that any SPLCM could be interactive and iterative in the real world and would cause expansion of the mapped software project life cycle (SPLC) to reflect multiple instances of activities.

At each step, constraints should be identified that impact the development of the SPLCP. In the example in this annex, common constraints require consideration while verifying information flows, mapping information into deliverable documents, and adding actual dates and times to the SPLCP. Once the SPLCP is in place, the experience of the project might dictate necessary modifications to the SPLCP. In this case, some or all of Step A through Step K could have to be repeated.

Clause 4 describes a five-part approach to accomplishing this task. Table B.1 presents an overview of the steps (A through K) required to implement the five parts of Clause 4. Table B.2 through Table B.12 each presents one of the steps.

Table B.1—Developing an SPLCP

Step	Description	Input	Output	Notes
Implementing the standard (Clause 4) Establish the requirements for a software project life cycle process (SPLCP) (4.1)				
A	**Establish Requirements**	Needs and expectations of relevant stakeholders	Feasible set of requirements acceptable to stakeholders	
	The process architect, who is already familiar with a variety of software project life cycle models (SPLCMs), with the criteria for choosing among them, and with the criteria for determining the attributes and constraints of the desired end product and the project environment that affect this selection, assures that requirements for the SPLCP are established.			
Select software project life cycle model (SPLCM) (4.2)				
B	**Select the SPLCM**	Organizational capabilities, past experience, project attributes, constraints	Selected SPLCM	See Table B.3
	This step is to identify the SPLCM to which the activities will be mapped. This step could mean that the whole process of locating, evaluating, selecting, and acquiring an SPLCM shall be performed. For this example, a three-phase (speculate, collaborate, learn) adaptive software development model is selected. During this step, the process architect reviews the SPLCM to ensure it is appropriate for the specific project.			
Create software project life cycle (SPLC) (4.3)				
C	**Map required and applicable activities to SPLCM**	Selected SPLCM	Table of mapped activities	See Table B.4
	Having selected a SPLCM, perform a detailed mapping of this standard's activities against the SPLCM. This step involves the matching of the activities against the requirements of the SPLCM. This step provides a checklist to assure that all activities are mapped and that all SPLCM requirements are covered by an activity(ies). Activity(ies) not mapped will be noted in Step F.			

Step	Description	Input	Output	Notes
Place the activities in executable sequence (4.3.1)				
D	**List activities and invocations**	List of Activities Not Used	List of activities used	See Table B.5
	List each activity and invocation for all the pieces of the SPLCM. 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 Subphase II.3 are listed in numerical order.			
E	**Place activities in executable sequence**	List of activities used	Sequenced list of activities (SPLC)	See Table B.6
	This step further organizes the activities within the SPLCM subphases to refine executable order relationships.			
Develop and justify List of Activities Not Used (4.3.2)				
F	**Develop and justify List of Activities Not Used**	Table of mapped activities	List of Activities Not Used	See Table B.7
	Activities not used for this example are listed below: A.1.2.3 A.2.3.2, A.2.3.3, and A.2.3.4 A.3.2.2 A.4.2.1, A.4.2.2, and A.4.2.3 A.4.4.1, A.4.4.2, and A.4.4.3 A.5.1.8 A.5.4.1, A.5.4.2, and A.5.4.3			
Verify the map (4.3.3)				
G	**Verify information flow**	Sequenced list of activities (SPLC)	Adjusted list of activities	See Table B.8
	The input information 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 E, it is a necessary check to be sure that all information will be available to the activities that need it, when they need it.			
Establish software project life cycle process (SPLCP) (4.4)				
H	**Map information into deliverable documents**	Adjusted list of activities; deliverable requirements	List of activities with mapped deliverables (SPLC)	See Table B.9
	Each SPLCM requires and defines formatted content of its own set of output products. These products are, for the most part, the specific documents that the SPLCM delivers. Note that the term document does not imply any particular medium. This step compares the output information generated by each activity with the SPLCM-required document(s) into which it must go. Once again, the order of the mapping, this time from Step F, might have to be modified. If a particular document, specified by the selected SPLCM, is to be created at a particular point in the development schedule, all the activities that contribute information to be recorded in that document must have had an opportunity to generate such information.			
I	**Add OPAs**	OPAs	Initial SPLCP	See Table B.10
	OPAs are now added to the fully mapped activities and deliverable documents at the appropriate points in the SPLCP. Adding the OPAs expands the SPLCP beyond the minimum set of activities specified in this standard and produces a fully robust SPLCP for the development project.			
J	**Add project planning information**	Project-specific activities	Final SPLCP	See Table B.11
	Throughout these steps, the project manager could be adding planning specifics to the evolving SPLCP. Additions are normally identified during the continuing audit and review process.			

Step	Description	Input	Output	Notes
Validate software project life cycle process (SPLCP) (4.5)				
K	**Validate SPLCP**	Requirements established in Step A	Validated SPLCP	See Table B.12

Table B.2—Establish requirements (for SPLCP) (Step A)

No additional detail offered here.

Table B.3—Select the SPLCM (Step B)

ID	Phase segments	Primary output
I.	Project initiation	
I.1	Identify the mission	Mission values
I.2	Identify project team	Resources
I.3	Create mission artifacts	Project vision, project data sheet product mission profile product specification (outline)
I.4	Obtain approval	Approval to proceed
I.5	Share mission values	Quality measures
II.	Adaptive cycle planning	
II.1	Determine time boxes	Cycle plan
II.2	Write objective statements	Product specification (draft)
II.3	Define product components	List of product components
II.4	Assign components to cycles	Cycle plan
II.5	Plan project	Project schedule
II.6	Develop project task list	Project work breakdown structure
III	Concurrent component engineering	
III.1	Develop components	Something useful delivered to a customer
III.2	Manage project	Status reports
III.3	Prepare for final quality assurance	Test plan, cases, scripts
III.4	Prepare for quality review	Agenda
IV	Quality review	
IV.1	Conduct cycle review	Focus group results, change requests
IV.2	Determine next steps	Updated cycle plan
IV.3	Conduct cycle post mortem	Corrective actions
V	Final quality assurance and release	
V.1	Perform tests	
V.2	Evaluate test results	
V.3	Fix problems	
V.4	Make decision	
V.5	Transition to production	
V.6	Close project	

Table B.4—Map activities to SPCLM (Step C)

Activities of IEEE Std 1074 (preceded by corresponding subclause numbers) Legend: NU = not used; X = mapped activity	SPLCM phases				
	I. Project initiation	II. Adaptive cycle planning	III. Concurrent component engineering	IV. Quality review	V. Final quality assurance & release
A.1.1 Project Initiation Activity Group					
A.1.1.1 Develop SPLCP (Required)	X				
A.1.1.2 Perform Estimations (Required)	X				
A.1.1.3 Allocate Project Resources (Required)	X				
A.1.1.4 Define Metrics (Required)	X				
A.1.1.5 Determine Security Objectives (Required)	X				
A.1.2 Project Planning Activity Group					
A.1.2.1 Plan Evaluations (Required)		X	X		
A.1.2.2 Plan Configuration Management (Required)		X			
A.1.2.3 Plan System Transition	NU	NU	NU	NU	NU
A.1.2.4 Plan Installation		X			
A.1.2.5 Plan Documentation (Required)	X	X			
A.1.2.6 Plan Training		X			
A.1.2.7 Plan Project Management (Required)		X			
A.1.2.8 Plan Integration		X			
A.1.2.9 Plan Release Management		X			
A.1.3 Project Monitoring and Control Activity Group					
A.1.3.1 Manage Risks (Required)	X	X	X		
A.1.3.2 Manage the Project (Required)			X		
A.1.3.3 Identify SPLCP Improvement Needs (Required)				X	X
A.1.3.4 Retain Records (Required)			X		
A.1.3.5 Collect and Analyze Metric Data (Required)			X		
A.1.3.6 Close Project (Required)					X
A.2.1 Concept Exploration Activity Group					
A.2.1.1 Identify Ideas or Needs (Required)	X	X			
A.2.1.2 Formulate Potential Approaches (Required)		X			
A.2.1.3 Conduct Feasibility Studies (Required)		X			
A.2.1.4 Refine and Finalize the Idea or Need (Required)		X			
A.2.2 System Allocation Activity Group					
A.2.2.1 Analyze System Functions		X			
A.2.2.2 Develop System Architecture		X			
A.2.2.3 Allocate System Requirements		X			

Activities of IEEE Std 1074 (preceded by corresponding subclause numbers) Legend: NU = not used; X = mapped activity	SPLCM phases				
	I. Project initiation	II. Adaptive cycle planning	III. Concurrent component engineering	IV. Quality review	V. Final quality assurance & release
A.2.3 Software Importation Activity Group					
A.2.3.1 Identify Imported Software Requirements	NU	NU	NU	NU	NU
A.2.3.2 Evaluate Software Import Sources	NU	NU	NU	NU	NU
A.2.3.3 Define Software Import Method	NU	NU	NU	NU	NU
A.2.3.4 Import Software	NU	NU	NU	NU	NU
A.3.1 Software Requirements Activity Group					
A.3.1.1 Define and Develop Software Requirements		X			
A.3.1.2 Define Interface Requirements		X			
A.3.1.3 Prioritize and Integrate Software Requirements		X			
A.3.2 Design Activity Group					
A.3.2.1 Perform Architectural Design		X			
A.3.2.2 Design Database	NU	NU	NU	NU	NU
A.3.2.3 Design Interfaces		X			
A.3.2.4 Perform Detailed Design		X			
A.3.3 Implementation Activity Group					
A.3.3.1 Create Executable Code			X		
A.3.3.2 Create Operating Documentation			X		
A.3.3.3 Perform Integration			X		
A.3.3.4 Manage Software Releases			X		
A.4.1 Installation Activity Group					
A.4.1.1 Distribute Software			X	X	X
A.4.1.2 Install Software			X	X	
A.4.1.3 Accept Software in Operational Environment				X	
A.4.2 Operation and Support Activity Group					
A.4.2.1 Operate the System			X		
A.4.2.2 Provide Technical Assistance and Consulting			X		
A.4.2.3 Maintain Support Request Log	NU	NU	NU	NU	NU
A.4.3 Maintenance Activity Group					
A.4.3.1 Identify Software Improvement Needs					X
A.4.3.2 Implement Problem Reporting Method		X			
A.4.3.3 Reapply SPLCP					X
A.4.4 Retirement Activity Group					
A.4.4.1 Notify User	NU	NU	NU	NU	NU
A.4.4.2 Conduct Parallel Operations	NU	NU	NU	NU	NU
A.4.4.3 Retire System	NU	NU	NU	NU	NU

Activities of IEEE Std 1074 (preceded by corresponding subclause numbers) Legend: NU = not used; X = mapped activity	SPLCM phases				
	I. Project initiation	II. Adaptive cycle planning	III. Concurrent component engineering	IV. Quality review	V. Final quality assurance & release
A.5.1 Evaluation Activity Group					
A.5.1.1 Conduct Reviews (Required)	X	X	X	X	X
A.5.1.2 Create Traceability Matrix		X			
A.5.1.3 Conduct Audits				X	
A.5.1.4 Develop Test Procedures			X		
A.5.1.5 Create Test Data			X		
A.5.1.6 Execute Tests			X		X
A.5.1.7 Report Evaluation Results (Required)	X		X		X
A.5.1.8 Confirm Security Accreditation	NU	NU	NU	NU	NU
A.5.2 Software Configuration Management Activity Group					
A.5.2.1 Develop Configuration Identification (Required)		X			
A.5.2.2 Perform Configuration Control (Required)		X			
A.5.2.3 Perform Status Accounting (Required)		X			
A.5.3 Documentation Development Activity Group					
A.5.3.1 Implement Documentation (Required)		X			
A.5.3.2 Produce and Distribute Documentation (Required)		X			
A.5.4 Training Activity Group					
A.5.4.1 Develop Training Materials	NU	NU	NU	NU	NU
A.5.4.2 Validate the Training Program	NU	NU	NU	NU	NU
A.5.4.3 Implement the Training Program	NU	NU	NU	NU	NU

Table B.5—List of activities and invocations (Step D)

Life cycle segment or phase	IEEE 1074 activities	Comments
I. Project initiation		
I.0	A.1.1.1 Develop SPLCP (Required)	
I.1 Identify the mission	A.1.1.2 Perform Estimations (Required)	
	A.1.1.5 Determine Security Objectives (Required)	
	A.1.3.1 Manage Risks (Required)	
I.2 Identify project team	A.1.1.3 Allocate Project Resources (Required)	
I.3 Create mission artifacts	A.1.2.5 Plan Documentation (Required)	
	A.1.1.4 Define Metrics (Required)	
I.4 Share mission values	A.1.1.4 Define Metrics (Required)	
	A.2.1.1 Identify Ideas or Needs (Required)	Results needed
	A.5.1.1 Conduct Reviews (Required)	Quality criteria
I.5 Obtain approval	A.5.1.1 Conduct Reviews (Required)	

Life cycle segment or phase	IEEE 1074 activities	Comments
	A.5.1.7 Report Evaluation Results (Required)	
II. Adaptive cycle planning		
II.1 Determine time boxes	A.1.2.7 Plan Project Management (Required)	First cycle
II.2 Write objective statements (for each cycle)	A.1.2.7 Plan Project Management (Required)	First cycle
II.3 Define product components	A.1.3.1 Manage Risks	
	A.2.1.1 Identify Ideas or Needs (Required)	First cycle
	A.2.1.2 Formulate Potential Approaches (Required)	
	A.2.1.3 Conduct Feasibility Studies (Required)	
	A.2.1.4 Refine and Finalize the Idea or Need (Required)	
	A.2.2.1 Analyze System Functions	
	A.2.2.2 Develop System Architecture	
	A.2.2.3 Allocate System Requirements	
	A.3.1.1 Define and Develop Software Requirements	
	A.3.1.2 Define Interface Requirements	
	A.3.1.3 Prioritize and Integrate Software Requirements	
	A.3.2.1 Perform Architectural Design	
	A.3.2.3 Design Interfaces	
	A.3.2.4 Perform Detailed Design	
	A.5.1.1 Conduct Reviews (Required)	
	A.5.1.2 Create Traceability Matrix	
	A.5.3.1 Implement Documentation (Required)	
	A.5.3.2 Produce and Distribute Documentation (Required)	
II.4 Assign components to cycle	A.5.2.1 Develop Configuration Identification (Required)	First cycle
	A.5.2.2 Perform Configuration Control (Required)	
	A.5.2.3 Perform Status Accounting (Required)	
II.5 Plan project	A.1.2.1 Plan Evaluations (Required)	
	A.1.2.2 Plan Configuration Management (Required)	
	A.1.2.4 Plan Installation	
	A.1.2.5 Plan Documentation (Required)	
	A.1.2.6 Plan Training	
	A.1.2.7 Plan Project Management (Required)	
	A.1.2.8 Plan Integration	
	A.1.2.9 Plan Release Management	
	A.1.3.1 Manage Risks (Required)	
	A.4.3.2 Implement Problem Reporting Method	
II.6 Develop project task list	A.1.2.7 Plan Project Management (Required)	
III Concurrent component engineering		
III.1 Develop components	A.3.3.1 Create Executable Code	Cycle 1 components
	A.3.3.2 Create Operating Documentation	
	A.3.3.3 Perform Integration	
	A.3.3.4 Manage Software Releases	
	A.5.1.1 Conduct Reviews (Required)	
	A.5.1.4 Develop Test Procedures	Unit tests; integration tests

Life cycle segment or phase	IEEE 1074 activities	Comments
	A.5.1.5 Create Test Data	
	A.5.1.6 Execute Tests	
	A.5.1.7 Report Evaluation Results (Required)	
III.2 Manage project	A.1.3.1 Manage Risks (Required)	
	A.1.3.2 Manage the Project (Required)	
	A.1.3.4 Retain Records (Required)	
	A.1.3.5 Collect and Analyze Metric Data (Required)	
III.3 Prepare for final quality assurance	A.5.1.4 Develop Test Procedures	Combined cycle 1 through cycle n test procedures
	A.5.1.5 Create Test Data	
III.4 Prepare for quality review	A.1.2.1 Plan Evaluations (Required)	First cycle
	A.5.1.4 Develop Test Procedures	Cycle 1 tests
	A.5.1.5 Create Test Data	
	A.5.1.6 Execute Tests	Cycle 1 tests
	A.4.1.1 Distribute Software	Set of cycle 1 components
	A.4.1.2 Install Software	
	A.4.2.1 Operate the System	
	A.4.2.2 Provide Technical Assistance and Consulting	
IV. Quality review		
IV.1 Conduct cycle review	A.5.1.1 Conduct Reviews (Required)	First cycle
	A.5.1.3 Conduct Audits	
IV.2 Determine next steps	A.1.3.3 Identify SPLCP Improvement Needs (Required)	
	A.4.1.1 Distribute Software	
	A.4.1.2 Install Software	
	A.4.1.3 Accept Software in Operational Environment	
IV.3 Conduct cycle post mortem	A.5.1.1 Conduct Reviews (Required)	
V. Final quality assurance and release		
V.1 Perform tests	A.5.1.6 Execute Tests	
V.2 Evaluate test results	A.5.1.7 Report Evaluation Results (Required)	
V.3 Fix problems	A.4.3.1 Identify Software Improvement Needs	
V.4 Make decision	A.5.1.1 Conduct Reviews (Required)	
V.5 Transition to production	A.4.1.1 Distribute Software	
V.6 Close project	A.1.3.3 Identify SPLCP Improvement Needs (Required)	When all cycles are complete
	A.4.3.3 Reapply SPLCP	
	A.1.3.6 Close Project (Required)	

Table B.6—Place activities in executable sequence (Partial list of sequenced activities) (Step E)

This step further organizes the activities within the SPLCM subphases to refine executable order relationships. The result for Subphase II.3 starting with A.3.1.1 would look like this.		
II. Adaptive cycle planning		
II.3 Define product components	…	
	…	
	A.3.1.1 Define and Develop Software Requirements	
	A.5.3.1 Implement Documentation	Draft product specification
	A.5.2.2 Perform Configuration Control	
	A.3.1.2 Define Interface Requirements	
	A.5.3.1 Implement Documentation	Draft product specification
	A.5.2.2 Perform Configuration Control	
	A.3.1.3 Prioritize and Integrate Software Requirements	
	A.5.3.1 Implement Documentation	Draft product specification
	A.5.2.2 Perform Configuration Control	
	A.5.1.1 Conduct Reviews	
	A.1.3.1 Manage Risks	
	A.5.3.1 Implement Documentation	
	A.5.2.2 Perform Configuration Control	

Table B.7—Develop and justify List of Activities Not Used (in the first cycle) (Step F)

Activity not used	Justification for not using
A.1.2.3 Plan System Transition	No system transition is needed or planned for
A.2.3.1 Identify Imported Software Requirements	No imported software will be used.
A.2.3.2 Evaluate Software Import Sources	No imported software will be used.
A.2.3.3 Define Software Import Method	No imported software will be used.
A.2.3.4 Import Software	No imported software will be used.
A.3.2.2 Design Database	No database design will be needed. Existing structure will be utilized.
A.4.2.3 Maintain Support Request Log	Cycle 1 corrective actions will be managed using the internal problem reporting method
A.4.4 Retirement Activity Group	First cycle in the project; no retirement considerations
A.4.4.1 Notify User	First cycle in the project
A.4.4.2 Conduct Parallel Operations	First cycle in the project
A.4.4.3 Retire System	First cycle in the project
A.5.4.1 Develop Training Materials	No training necessary for initial cycle
A.5.4.2 Validate the Training Program	No training necessary for initial cycle
A.5.4.3 Implement the Training Program	No training necessary for initial cycle
A.5.1.8 Confirm Security Accreditation	Security needs will be covered in last cycle

Table B.8—Verify information flow (Step G)

A review of the input/output tables of this standard is performed in light of the sequencing of activities to ensure that information needed by a particular activity has been produced by an earlier activity and to minimize rework caused by returning to an earlier process to redo or generate additional information.

Table B.9—Map information into deliverable documents (Step H)

All output information determined by the selected set of activities is mapped into this project's particular document/information set. This example shows a selected set of mapped information. See Annex D for a sample information mapping template.

Information mapping template			
Activity	**Clause**	**Output information**	**Mapped deliverables**
Develop SPLCP (Required)	A.1.1.1	SPLCP	Adaptive Software Development Process
		List of Activities Not Used	Project plan
Perform Estimations (Required)	A.1.1.2	Project Estimates	Project plan
		Estimation Assumptions	Project plan
Allocate Project Resources (Required)	A.1.1.3	Resource Allocations	Project plan
Define Metrics (Required)	A.1.1.4	Defined Metrics	Project plan
Plan Installation	A.1.2.4	Software Installation Planned Information	Cycle plan
Plan Integration	A.1.2.8	Integration Planned Information	Cycle plan
Plan Release Management	A.1.2.9	SRMPI	Cycle plan
Define and Develop Software Requirements	A.3.1.1	Preliminary Software Requirements	Product specification
		Installation Requirements	Product specification
Define Interface Requirements	A.3.1.2	Software Interface Requirements	Product specification
Prioritize and Integrate Software Requirements	A.3.1.3	Software Requirements	Product specification
Create Executable Code	A.3.3.1	Executable Code	Component code
Conduct Reviews (Required)	A.5.1.1	In-Process Review Results	Cycle post mortem report
		Process Improvement Recommendations	Cycle post mortem report
		Management Status Reported Information	Management review results
Conduct Audits	A.5.1.3	Audit Results Information	Customer focus group results
Develop Test Procedures	A.5.1.4	Test Procedures	Unit, integration, cycle, system
Implement Documentation (Required)	A.5.3.1	Document	Product specification, project data sheet, cycle post mortem reports, peer review results, management review results

Table B.10—Add OPAs (Step I)

For this example, the project will use the following process assets:
Collaboration methods and tools
Post mortem procedure
Peer review procedure

Table B.11—Add project planning information (Step J)

Results of initial and subsequent life cycle reviews and life cycle post mortems are used to help determine the next steps and serve as the basis for updates to the document set.

Table B.12—Validate SPLCP (Step K)

No additional detail offered here.

Annex C

(informative)

Information mapping template

This information mapping template is designed to assist project managers in identifying project-critical deliverables and assuring 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	Clause	Output information	Project deliverables
Project Management Section of activity groups	**A.1**		
Project Initiation Activity Group	A.1.1		
Develop SPLCP (Required)	A.1.1.1	SPLCP	
		List of Activities Not Used	
Perform Estimations (Required)	A.1.1.2	Project Estimates	
		Estimation Assumptions	
Allocate Project Resources (Required)	A.1.1.3	Resource Allocations	
Define Metrics (Required)	A.1.1.4	Defined Metrics	
		Collection and Analysis Methods	
Determine Security Objectives (Required)	A.1.1.5	Security Objectives	
Project Planning Activity Group	A.1.2		
Plan Evaluations (Required)	A.1.2.1	Evaluation Planned Information	
Plan Configuration Management (Required)	A.1.2.2	SCMPI	
Plan System Transition	A.1.2.3	Transition Planned Information	
		Transition Impact Statement	
Plan Installation	A.1.2.4	Software Installation Planned Information	
Plan Documentation (Required)	A.1.2.5	Documentation Planned Information	
Plan Training	A.1.2.6	Training Planned Information	
Plan Project Management (Required)	A.1.2.7	SPMPI	
		PR&RPI	
		Retirement Planned Information	
		Security Planned Information	
		Support Planned Information	
Plan Integration	A.1.2.8	Integration Planned Information	
Plan Release Management	A.1.2.9	SRMPI	

Activity	Clause	Output information	Project deliverables
Project Monitoring and Control Activity Group	A.1.3		
Manage Risks (Required)	A.1.3.1	Risk Management Reported Information	
Manage the Project (Required)	A.1.3.2	Project Management Reported Information	
		Anomalies	
Identify SPLCP Improvement Needs (Required)	A.1.3.3	Environment Improvement Needs	
Retain Records (Required)	A.1.3.4	Historical Project Records	
Collect and Analyze Metric Data (Required)	A.1.3.5	Analysis Reported Information	
Close Project (Required)	A.1.3.6	Project Archival Information	
Pre-Development Section of activity groups	**A.2**		
Concept Exploration Activity Group	A.2.1		
Identify Ideas or Needs (Required)	A.2.1.1	Preliminary Statement of Need	
Formulate Potential Approaches (Required)	A.2.1.2	Constraints and Benefits	
		Potential Approaches	
Conduct Feasibility Studies (Required)	A.2.1.3	Recommendations	
Refine and Finalize the Idea or Need (Required)	A.2.1.4	Statement of Need	
System Allocation Activity Group	A.2.2		
Analyze System Functions	A.2.2.1	Functional Description of the System	
Develop System Architecture	A.2.2.2	System Architecture	
		Security Requirements	
Allocate System Requirements	A.2.2.3	System Human and Hardware Requirements	
		System Software Functional Requirements	
		System Interface Requirements	
Software Importation Activity Group	A.2.3		
Identify Imported Software Requirements	A.2.3.1	Imported Software Requirements	
Evaluate Software Import Sources	A.2.3.2	Selected Software Import Source	
		Candidate Software Import Methods	
Define Software Import Method	A.2.3.3	Selected Software Import Method	
Import Software	A.2.3.4	Imported Software	
		Imported Software Documentation	
Development Section of activity groups	**A.3**		
Software Requirements Activity Group	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	

Activity	Clause	Output information	Project deliverables
Design Activity Group	A.3.2		
Perform Architectural Design	A.3.2.1	Software Architectural Design	
Design Database	A.3.2.2	Database Design	
Design Interfaces	A.3.2.3	Interface Design	
Perform Detailed Design	A.3.2.4	Software Detailed Design	
Implementation Activity Group	A.3.3		
Create Executable Code	A.3.3.1	Source Code (when required)	
		Executable Code	
		Database	
Create Operating Documentation	A.3.3.2	Operating Documentation	
Perform Integration	A.3.3.3	Integrated Software	
Manage Software Releases	A.3.3.4	Released Product Package	
Post-Development Section of activity groups	**A.4**		
Installation Activity Group	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	
		Historical Project Records	
		Installed Software System	
Operation and Support Activity Group	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 Activity Group	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 SPLCP	A.4.3.3	Maintenance Recommendations	
		Resolved Problem Reported Information	
		Updated Report Log	
Retirement Activity Group	A.4.4		
Notify User	A.4.4.1	Official Notification	

Activity	Clause	Output information	Project deliverables
Conduct Parallel Operations	A.4.4.2	Parallel Operations Log	
Retire System	A.4.4.3	Archive Reported Information	
		Post Operation Review Reported Information	
Support Section of activity groups	**A.5**		
Evaluation Activity Group	A.5.1		
Conduct Reviews (Required)	A.5.1.1	In-Process Review Results	
		Post-Implementation Review Reported Information	
		Process Improvement Recommendations	
		Management Status Reported Information	
		Traceability Analysis 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	
		Test Data	
Execute Tests	A.5.1.6	Test Summary Reported Information	
		Tested Software	
		Anomalies	
Report Evaluation Results (Required)	A.5.1.7	Evaluation Reported Information	
		Anomalies	
Confirm Security Accreditation	A.5.1.8	Security Accreditation	
Software Configuration Management Activity Group	A.5.2		
Develop Configuration Identification (Required)	A.5.2.1	Configuration Identification	
Perform Configuration Control (Required)	A.5.2.2	Change Status	
		Controlled Item	
Perform Status Accounting (Required)	A.5.2.3	Controlled Item	
		Status Reported Information	
Documentation Development Activity Group	A.5.3		
Implement Documentation (Required)	A.5.3.1	Document	
Produce and Distribute Documentation (Required)	A.5.3.2	Published Document	
Training Activity Group	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	

Activity	Clause	Output information	Project deliverables
Implement the Training Program	A.5.4.3	Updated Skills Inventory	
		Trained Personnel	
		Training Feedback	

Annex D

(informative)

Sample software project life cycle models (SPLCMs)

SPLCMs vary with the expected deliverables of the project. For example, a life cycle model for a project to deliver only requirements, not a finished software product, would not include any activities beyond those necessary to deliver the requirements.

D.1 Requirements-defining software project life cycle model (SPLCM)

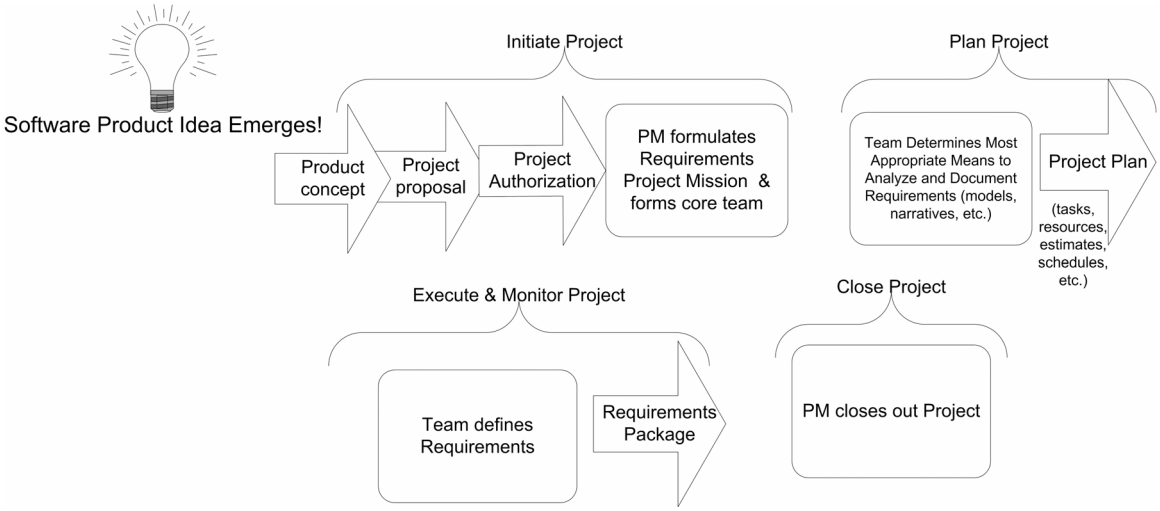

Figure D.1—Requirements-defining SPLCM

In this model, the project's mission is to define the functional and nonfunctional requirements for a software system. The software itself will not be built as part of this project. (It may later be built by another group and another project.)

The following activities can be applied to this project life cycle model:

- Project Initiation Activity Group (**A.1.1)**
  - Develop SPLCP (Required) (A.1.1.1)
  - Perform Estimations (Required) (A.1.1.2)
  - Allocate Project Resources (Required) (A.1.1.3)
  - Define Metrics (Required) (A.1.1.4)
  - Determine Security Objectives (Required) (A.1.1.5)
- Project Planning Activity Group (A.1.2)
  - Plan Evaluations (Required) (A.1.2.1)
  - Plan Configuration Management (Required) (A.1.2.2)

  - Plan Documentation (Required) (A.1.2.5)
  - Plan Project Management (Required) (A.1.2.7)
- Project Monitoring and Control Activity Group (A.1.3)
  - Manage Risks (Required) (A.1.3.1)
  - Manage the Project (Required) (A.1.3.2)
  - Identify SPLCP Improvement Needs (Required) (A.1.3.3)
  - Retain Records (Required) (A.1.3.4)
  - Collect and Analyze Metric Data (Required) (A.1.3.5)
  - Close Project (Required) (A.1.3.6)
- Concept Exploration Activity Group (A.2.1)
  - Identify Ideas or Needs (Required) (A.2.1.1)
  - Formulate Potential Approaches (Required) (A.2.1.2)
  - Conduct Feasibility Studies (Required) (A.2.1.3)
  - Refine and Finalize the Idea or Need (Required) (A.2.1.4)
- System Allocation Activity Group (A.2.2)
  - Analyze System Functions (A.2.2.1)
  - Develop System Architecture (A.2.2.2)
  - Allocate System Requirements (A.2.2.3)
- Software Importation Activity Group (A.2.3)
  - Identify Imported Software Requirements (A.2.3.1)
- Software Requirements Activity Group (A.3.1)
  - Define and Develop Software Requirements (A.3.1.1)
  - Define Interface Requirements (A.3.1.2)
  - Prioritize and Integrate Software Requirements (A.3.1.3)
- Evaluation Activity Group (A.5.1)
  - Conduct Reviews (Required) (A.5.1.1)
  - Report Evaluation Results (Required) (A.5.1.7)
- Software Configuration Management Activity Group (A.5.2)
  - Develop Configuration Identification (Required) (A.5.2.1)
  - Perform Configuration Control (Required) (A.5.2.2)
  - Perform Status Accounting (Required) (A.5.2.3)
- Documentation Development Activity Group (A.5.3)
  - Implement Documentation (Required) (A.5.3.1)
  - Produce and Distribute Documentation (Required) (A.5.3.2)
- Training Activity Group (A.5.4) (not applicable)

D.2 System retirement software project life cycle model (SPLCM)

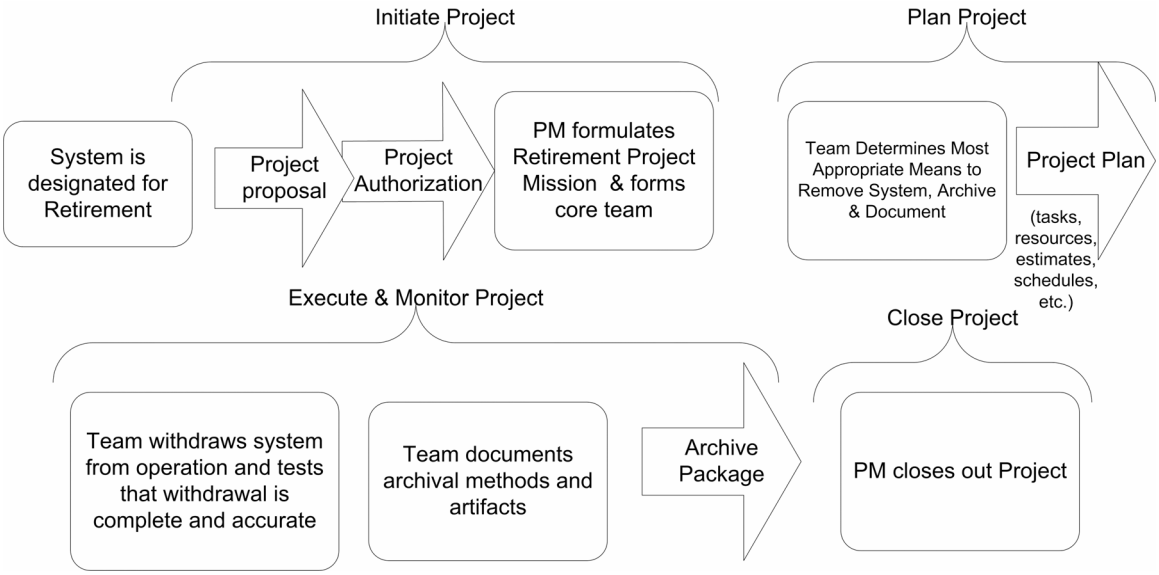

Figure D.2—System retirement SPLCM

In this model, the project's mission is to retire the software system. No software will be built as part of this project. Another system may have already been installed to replace this system, or this system may not be replaced.

The following activities can be applied to this project life cycle model:

- Project Initiation Activity Group (A.1.1)
  - Develop SPLCP (Required) (A.1.1.1)
  - Perform Estimations (Required) (A.1.1.2)
  - Allocate Project Resources (Required) (A.1.1.3)
  - Define Metrics (Required) (A.1.1.4)
  - Determine Security Objectives (Required) (A.1.1.5)
- Project Planning Activity Group (A.1.2)
  - Plan Evaluations (Required) (A.1.2.1)
  - Plan Configuration Management (Required) (A.1.2.2)
  - Plan Documentation (Required) (A.1.2.5)
  - Plan Project Management (Required) (A.1.2.7)
- Project Monitoring and Control Activity Group (A.1.3)
  - Manage Risks (Required) (A.1.3.1)
  - Manage the Project (Required) (A.1.3.2)
  - Identify SPLCP Improvement Needs (Required) (A.1.3.3)

  - Retain Records (Required) (A.1.3.4)
  - Collect and Analyze Metric Data (Required) (A.1.3.5)
  - Close Project (Required) (A.1.3.6)
- Concept Exploration Activity Group (A.2.1) (not applicable)
- System Allocation Activity Group (A.2.2) (not applicable)
- Software Importation Activity Group (A.2.3) (not applicable)
- Software Requirements Activity Group (not applicable) (A.3.1)
- Design Activity Group (A.3.2) (not applicable)
- Implementation Activity Group (A.3.3) (not applicable)
- Installation Activity Group (A.4.1) (not applicable)
- Operation and Support Activity Group (not applicable) (A.4.2)
- Maintenance Activity Group (A.4.3) (not applicable)
- Retirement Activity Group (A.4.4)
  - Notify User (A.4.4.1)
  - Retire System (A.4.4.3)
- Evaluation Activity Group (A.5.1)

NOTE—The retirement activities and the success of the clean and complete retirement of the system shall be evaluated.

  - Conduct Reviews (Required) (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)
  - Report Evaluation Results (Required) (A.5.1.7)
  - Confirm Security Accreditation (A.5.1.8)
- Software Configuration Management Activity Group (A.5.2)

NOTE—The system being retired shall be archived appropriately.

  - Develop Configuration Identification (Required) (A.5.2.1)
  - Perform Configuration Control (Required) (A.5.2.2)
  - Perform Status Accounting (Required) (A.5.2.3)
- Documentation Development Activity Group (A.5.3)

NOTE—The final disposition of the system being retired shall be documented appropriately.

  - Implement Documentation (Required) (A.5.3.1)
  - Produce and Distribute Documentation (A.5.3.2)
- Training Activity Group (A.5.4) (not applicable)

D.3 Development and delivery software project life cycle model (SPLCM)

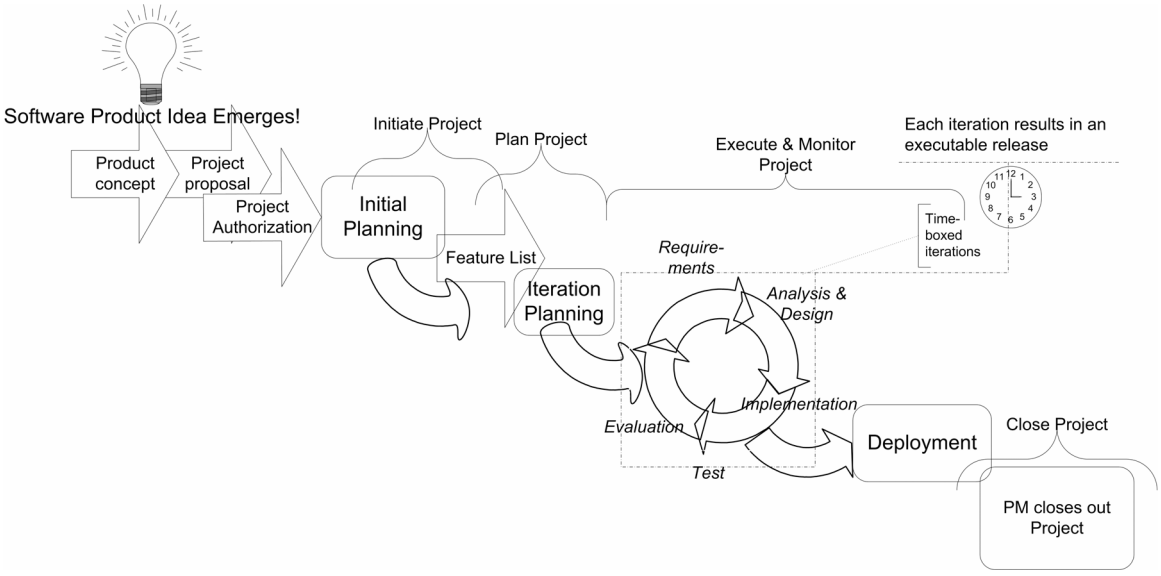

Figure D.3—Development and delivery SPLCM

In this model, the project's mission is to plan, develop, and deliver a software system. The software will be built in successive development iterations. Each iteration is time-boxed, and each iteration delivers a particular set of features as an executable system.

The following activities can be applied to this project life cycle model:

- Project Initiation Activity Group (A.1.1)
  - Develop SPLCP (Required) (A.1.1.1)
  - Perform Estimations (Required) (A.1.1.2)
  - Allocate Project Resources (Required) (A.1.1.3)
  - Define Metrics (Required) (A.1.1.4)
  - Determine Security Objectives (Required) (A.1.1.5)
- Project Planning Activity Group (A.1.2)
  - Plan Evaluations (Required) (A.1.2.1)
  - Plan Configuration Management (Required) (A.1.2.2)
  - Plan System Transition (A.1.2.3)
  - Plan Installation (A.1.2.4)
  - Plan Documentation (Required) (A.1.2.5)
  - Plan Training (A.1.2.6)
  - Plan Project Management (Required) (A.1.2.7)
  - Plan Integration (A.1.2.8)

- Plan Release Management (A.1.2.9)
- Project Monitoring and Control Activity Group (A.1.3)
  - Manage Risks (Required) (A.1.3.1)
  - Manage the Project (Required) (A.1.3.2)
  - Identify SPLCP Improvement Needs (Required) (A.1.3.3)
  - Retain Records (Required) (A.1.3.4)
  - Collect and Analyze Metric Data (Required) (A.1.3.5)
  - Close Project (Required) (A.1.3.6)
- Concept Exploration Activity Group (A.2.1)
  - Identify Ideas or Needs (Required) (A.2.1.1)
  - Formulate Potential Approaches (Required) (A.2.1.2)
  - Conduct Feasibility Studies (Required) (A.2.1.3)
  - Refine and Finalize the Idea or Need (Required) (A.2.1.4)
- System Allocation Activity Group (A.2.2)
  - Analyze System Functions (A.2.2.1)
  - Develop System Architecture (A.2.2.2)
  - Allocate System Requirements (A.2.2.3)
- Software Importation Activity Group (A.2.3)
  - Identify Imported Software Requirements (A.2.3.1)
  - Evaluate Software Import Sources (A.2.3.2)
  - Define Software Import Method (A.2.3.3)
  - Import Software (A.2.3.4)
- Software Requirements Activity Group (A.3.1)
  - Define and Develop Software Requirements (A.3.1.1)
  - Define Interface Requirements (A.3.1.2)
  - Prioritize and Integrate Software Requirements (A.3.1.3)
- Design Activity Group (A.3.2)
  - Perform Architectural Design (A.3.2.1)
  - Design Database (A.3.2.2)
  - Design Interfaces (A.3.2.3)
  - Perform Detailed Design (A.3.2.4)
- Implementation Activity Group (A.3.3)
  - Create Executable Code (A.3.3.1)
  - Create Operating Documentation (A.3.3.2)
  - Perform Integration (A.3.3.3)

- Manage Software Releases (A.3.3.4)
- Installation Activity Group (A.4.1)
  - Distribute Software (A.4.1.1)
- Operation and Support Activity Group (not applicable) (A.4.2)
- Maintenance Activity Group (A.4.3) (not applicable)
- Retirement Activity Group (not applicable) (A.4.4)
- Evaluation Activity Group (A.5.1)
  - Conduct Reviews (Required) (A.5.1.1)
  - Create Traceability Matrix (A.5.1.2)
  - 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)
  - Report Evaluation Results (Required) (A.5.1.7)
  - Confirm Security Accreditation (A.5.1.8)
- Software Configuration Management Activity Group (A.5.2)
  - Develop Configuration Identification (Required) (A.5.2.1)
  - Perform Configuration Control (Required) (A.5.2.2)
  - Perform Status Accounting (Required) (A.5.2.3)
- Documentation Development Activity Group (A.5.3)
  - Implement Documentation (Required) (A.5.3.1)
  - Produce and Distribute Documentation (A.5.3.2)
- Training Activity Group (A.5.4)
  - Develop Training Materials (A.5.4.1)
  - Validate the Training Program (A.5.4.2)
  - Implement the Training Program (A.5.4.3)

Annex E

(informative)

Glossary

For the purposes of this document, the following terms and definitions apply. These and other terms within IEEE standards are found in *The Authoritative Dictionary of IEEE Standards Terms* [B5].

activity: A defined body of work to be performed, including its required input information and output information.

activity group: A set of related activities.

constraint: A restriction on software project life cycle process (SPLCP) development.

external: An input information source or output information destination that is outside the purview of this standard and, therefore, may or may not exist.

mapping: Establishing a sequence of the activities in IEEE Std 1074 according to a selected software project life cycle model (SPLCM).

organizational process asset (OPA): An artifact that defines some portion of an organization's software project environment.

process architect: The person or group that has primary responsibility for creating and maintaining the software project life cycle process(s) (SPLCPs).

product: Any output of the software development activities, e.g., document, code, model.

Annex F

(informative)

Bibliography

[B1] AS 3563.1, Software Quality Management System, Part 1: Requirements.[3]

[B2] Capability Maturity Model Integrated, SEI 2002.[4]

[B3] IEEE/EIA 12207.0, Standard for Information Technology—Software Life Cycle Processes.[5]

[B4] IEEE Software Engineering Standards Collection.

[B5] IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition.

[B6] IEEE Std 730, IEEE Standard for Software Quality Assurance Plans.

[B7] IEEE Std 828, IEEE Standard for Software Configuration Management Plans.

[B8] IEEE Std 829, IEEE Standard for Software Test Documentation.

[B9] IEEE Std 830, IEEE Recommended Practice for Software Requirements Specifications.

[B10] IEEE Std 982.1, IEEE Standard Dictionary of Measures of the Software Aspects of Dependability.

[B11] IEEE Std 1008, IEEE Standard for Software Unit Testing.

[B12] IEEE Std 1012, IEEE Standard for Software Verification and Validation Plans.

[B13] IEEE Std 1016, IEEE Recommended Practice for Software Design Descriptions.

[B14] IEEE Std 1028, IEEE Standard for Software Reviews and Audits.

[B15] IEEE Std 1044, IEEE Standard for Classification of Software Anomalies.

[B16] IEEE Std 1045, IEEE Standard for Software Productivity Metrics.

[B17] IEEE Std 1058, IEEE Standard for Software Project Management Plans.

[B18] IEEE Std 1061, IEEE Standard for a Software Quality Metrics Methodology.

[B19] IEEE Std 1062, IEEE Recommended Practice for Software Acquisition.

[B20] IEEE Std 1063, IEEE Standard for Software User Documentation.

[B21] IEEE Std 1175, IEEE Trial-Use Standard Reference Model for Computing System Tool Interconnections.

[3] AS publications are available from Standards Australia (http://www.standards.org.au/).
[4] SEI publications are available from the Software Engineering Institute of Carnegie Mellon University (http://www.sei.cmu.edu/).
[5] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/). EIA publications are available from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (http://global.ihs.com/).

[B22] IEEE Std 1219, IEEE Standard for Software Maintenance.

[B23] IEEE Std 1220, IEEE Standard for Application and Management of the Systems Engineering Process.

[B24] IEEE Std 1228, IEEE Standard for Software Safety Plans.

[B25] IEEE Std 1233, IEEE Guide to Developing System Requirements Specifications.

[B26] IEEE Std 1471, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.

[B27] IEEE Std 1490, IEEE Guide—Adoption of PMI Standard—Guide to the Project Management Body of Knowledge.

[B28] IEEE Std 1540, IEEE Standard For Software Life Cycle Processes—Risk Management.

[B29] ISO 9001:2000, Quality management systems — Requirements.[6]

[B30] ISO 15408, Information Technology — Security techniques.

[B31] ISO/IEC 14598, Software Products Evaluation.[7]

[B32] ISO/IEC 15939, Software Engineering — Software Measurement Process.

[B33] ISO/IEC 90003, Software Engineering — Guidelines for the Application of ISO 9001:2000 to Computer Software.

[6] ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[7] ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO/IEC publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (http://global.ihs.com/). Electronic copies are available in the United States from the American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).