



BSI Standards Publication

Systems and software engineering — Software life cycle processes

National foreword

This British Standard is the UK implementation of ISO/IEC/IEEE 12207:2017. It supersedes BS ISO/IEC 12207:2008, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/15, Software and systems engineering.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2018
Published by BSI Standards Limited 2018

ISBN 978 0 580 85508 5

ICS 35.080

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 March 2018.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

INTERNATIONAL
STANDARD

ISO/IEC/
IEEE
12207

First edition
2017-11

Systems and software engineering — Software life cycle processes

Ingénierie des systèmes et du logiciel — Processus du cycle de vie du logiciel



Reference number
ISO/IEC/IEEE 12207:2017(E)

Documents
© ISO/IEC 2017
© IEEE 2017



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2017, Published in Switzerland

© IEEE 2017

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO or IEEE at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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

stds.ipr@ieee.org
www.ieee.org

Contents

Page

Foreword.....	vi
Introduction	vii
1 Scope	1
1.1 Overview	1
1.2 Purpose	1
1.3 Field of application	1
1.4 Limitations	2
2 Normative references	2
3 Terms, definitions, and abbreviated terms	2
3.1 Terms and definitions	2
3.2 Abbreviated terms	11
4 Conformance.....	11
4.1 Intended usage.....	11
4.2 Full conformance.....	12
4.2.1 Full conformance to outcomes.....	12
4.2.2 Full conformance to tasks	12
4.3 Tailored conformance	12
5 Key concepts and application.....	13
5.1 Introduction.....	13
5.2 Software system concepts	13
5.2.1 Software systems.....	13
5.2.2 Software system structure	13
5.2.3 Enabling systems.....	15
5.2.4 Life cycle processes for the software system	16
5.3 Organization and project concepts.....	16
5.3.1 Organizations	16
5.3.2 Organization and project-level adoption	17
5.4 Life cycle concepts	17
5.4.1 Software life cycle stages	17
5.4.2 Life cycle model for the software system	17
5.5 Process concepts	19
5.5.1 Criteria for processes.....	19
5.5.2 Description of processes.....	19
5.5.3 General characteristics of processes	19
5.5.4 Tailoring.....	19
5.6 Process groups	19
5.6.1 Introduction.....	19
5.6.2 Agreement processes.....	21
5.6.3 Organizational project-enabling processes.....	22
5.6.4 Technical Management processes	22
5.6.5 Technical processes	22
5.7 Process application	22
5.8 Process reference model	23
6 Software life cycle processes	24
6.1 Agreement processes.....	24
6.1.1 Acquisition process	24
6.1.2 Supply process	27
6.2 Organizational Project-Enabling processes	28
6.2.1 Life cycle model management process.....	29
6.2.2 Infrastructure Management process	30
6.2.3 Portfolio Management process.....	31
6.2.4 Human Resource Management process.....	33

6.2.5	Quality Management process.....	34
6.2.6	Knowledge Management process.....	36
6.3	Technical Management processes.....	37
6.3.1	Project Planning process.....	38
6.3.2	Project assessment and control process.....	40
6.3.3	Decision Management process.....	43
6.3.4	Risk Management process.....	44
6.3.5	Configuration Management process.....	46
6.3.6	Information Management process.....	50
6.3.7	Measurement process.....	52
6.3.8	Quality Assurance process.....	53
6.4	Technical processes.....	55
6.4.1	Business or Mission Analysis process.....	56
6.4.2	Stakeholder Needs and Requirements Definition process.....	59
6.4.3	System/Software requirements definition process.....	63
6.4.4	Architecture Definition process.....	66
6.4.5	Design Definition process.....	71
6.4.6	System Analysis process.....	74
6.4.7	Implementation process.....	75
6.4.8	Integration process.....	79
6.4.9	Verification process.....	82
6.4.10	Transition process.....	85
6.4.11	Validation process.....	89
6.4.12	Operation process.....	92
6.4.13	Maintenance process.....	95
6.4.14	Disposal process.....	99
Annex A (normative)	Tailoring process.....	102
A.1	Introduction.....	102
A.2	Tailoring process.....	102
A.2.1	Purpose.....	102
A.2.2	Outcomes.....	102
A.2.3	Activities and tasks.....	102
Annex B (informative)	Examples of process information items.....	104
Annex C (informative)	Process Reference Model for assessment purposes.....	107
C.1	Introduction.....	107
C.2	Conformance with ISO/IEC 33004.....	107
C.2.1	General.....	107
C.2.2	Requirements for process reference models.....	107
C.2.3	Process descriptions.....	108
C.3	The process reference model.....	108
Annex D (informative)	Process integration and process constructs.....	109
D.1	Introduction.....	109
D.2	Process constructs and their usage.....	109
Annex E (informative)	Process views.....	111
E.1	Introduction.....	111
E.2	The process view concept.....	111
E.3	Process viewpoint.....	111
E.4	Process view for specialty engineering.....	112
E.5	Process view for interface management.....	114
E.6	Process view for software assurance (Information security).....	116
Annex F (informative)	Software system architecture modelling.....	120
F.1	Introduction.....	120
F.2	Views, models and model kinds used in software system architecture.....	120
F.2.1	Functional model.....	120
F.2.2	Static model.....	121
F.2.3	Data model.....	121
F.2.4	Behavioral model.....	121
F.2.5	Temporal model.....	121
F.2.6	Structural model.....	121

F.2.7	Network model	121
F.3	Other model considerations.....	121
Annex G	(informative) Application of software life cycle processes to a system of systems	123
G.1	Introduction.....	123
G.2	SoS characteristics and types	123
G.3	SE processes applied to systems of systems.....	124
G.3.1	General	124
G.3.2	Agreement processes.....	124
G.3.3	Organizational project enabling processes	124
G.3.4	Technical management processes	125
G.3.5	Technical processes	125
Annex H	(informative) Application of Agile.....	127
Annex I	(informative) Process Mapping from ISO/IEC/IEEE 12207:2008.....	129
Bibliography		143

List of Illustrations

Figure 1	—Software system and software system element relationship	14
Figure 2	—Example of software system-of-interest structure.....	14
Figure 3	—Software system-of-interest, its operational environment and enabling systems.....	15
Figure 4	—Software life cycle processes	21
Table B.1	— Sample information items by process.....	104
Figure D.1	— ISO/IEC/IEEE 12207:2017 and ISO/IEC/IEEE 15288:2015 process constructs.....	110
Table G.1	— System of Systems types	123
Table I.1	— Comparison of processes in ISO/IEC/IEEE 12207:2017 and the previous edition	129
Table I.2	— Comparison of process outcomes in ISO/IEC/IEEE 12207:2017 and software-related outcomes in the previous edition	131

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO, IEC, and IEEE shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Systems and software engineering*, in cooperation with the IEEE Computer Society Systems and Software Engineering Standards Committee, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

This first edition of ISO/IEC/IEEE 12207 cancels and replaces ISO/IEC 12207:2008 (second edition), which has been technically revised.

Changes in this revision of ISO/IEC/IEEE 12207 were developed in conjunction with a corresponding revision of ISO/IEC/IEEE 15288:2015, *Systems and software engineering – System life cycle processes*. The purpose of these revisions is to accomplish the harmonization of the structures and contents of the two documents, while supporting the requirements of the engineering and assessment communities.

This document was developed with the following goals:

- provide a common terminology between the revision of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207;
- where applicable, provide common process names and process structure between the revision of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207; and
- enable the user community to evolve towards fully harmonized standards, while allowing backward compatibility.

This revision is intended to achieve a fully harmonized view of the system and software life cycle processes.

Introduction

The complexity of software systems has increased to an unprecedented level. This has led to new opportunities, but also to increased challenges for the organizations that create and utilize systems. These challenges exist throughout the life cycle of a system and at all levels of architectural detail. This document provides a common process framework for describing the life cycle of systems created by humans, adopting a Software Engineering approach. Software Engineering is an interdisciplinary approach and means to enable the realization of successful software systems. It focuses on defining stakeholder needs and required functionality early in the development cycle, documenting requirements, and performing design synthesis and system validation while considering the complete problem. It integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation and maintenance. It considers both the business and the technical needs of all stakeholders with the goal of providing a quality product that meets the needs of users and other applicable stakeholders. This life cycle spans the conception of ideas through to the retirement of a system. It provides the processes for acquiring and supplying systems. It helps to improve communication and cooperation among the parties that create, utilize and manage modern software systems in order that they can work in an integrated, coherent fashion. In addition, this framework provides for the assessment and improvement of the life cycle processes.

The processes in this document form a comprehensive set from which an organization can construct software life cycle models appropriate to its products and services. An organization, depending on its purpose, can select and apply an appropriate subset to fulfill that purpose.

This document can be used in one or more of the following modes:

- a) By an organization — to help establish an environment of desired processes. These processes can be supported by an infrastructure of methods, procedures, techniques, tools and trained personnel. The organization may then employ this environment to perform and manage its projects and progress software systems through their life cycle stages. In this mode, this document is used to assess conformance of a declared, established environment to its provisions.
- b) By a project — to help select, structure and employ the elements of an established environment to provide products and services. In this mode, this document is used in the assessment of conformance of the project to the declared and established environment.
- c) By an acquirer and a supplier — to help develop an agreement concerning processes and activities. Via the agreement, the processes and activities in this document are selected, negotiated, agreed to and performed. In this mode, this document is used for guidance in developing the agreement.
- d) By process assessors — to serve as a process reference model for use in the performance of process assessments that may be used to support organizational process improvement.

Systems and software engineering — Software life cycle processes

1 Scope

1.1 Overview

This document establishes a common framework for software life cycle processes, with well-defined terminology, that can be referenced by the software industry. It contains processes, activities, and tasks that are applicable during the acquisition, supply, development, operation, maintenance or disposal of software systems, products, and services. These life cycle processes are accomplished through the involvement of stakeholders, with the ultimate goal of achieving customer satisfaction.

This document applies to the acquisition, supply, development, operation, maintenance, and disposal (whether performed internally or externally to an organization) of software systems, products and services, and the software portion of any system. Software includes the software portion of firmware. Those aspects of system definition needed to provide the context for software products and services are included.

This document also provides processes that can be employed for defining, controlling, and improving software life cycle processes within an organization or a project.

The processes, activities, and tasks of this document can also be applied during the acquisition of a system that contains software, either alone or in conjunction with ISO/IEC/IEEE 15288:2015, *Systems and software engineering—System life cycle processes*.

In the context of this document and ISO/IEC/IEEE 15288, there is a continuum of human-made systems from those that use little or no software to those in which software is the primary interest. It is rare to encounter a complex system without software, and all software systems require physical system components (hardware) to operate, either as part of the software system-of-interest or as an enabling system or infrastructure. Thus, the choice of whether to apply this document for the software life cycle processes, or ISO/IEC/IEEE 15288:2015, *Systems and software engineering—System life cycle processes*, depends on the system-of-interest. Processes in both documents have the same process purpose and process outcomes, but differ in activities and tasks to perform software engineering or systems engineering, respectively.

1.2 Purpose

The purpose of this document is to provide a defined set of processes to facilitate communication among acquirers, suppliers and other stakeholders in the life cycle of a software system.

This document is written for acquirers, suppliers, developers, integrators, operators, maintainers, managers, quality assurance managers, and users of software systems, products, and services. It can be used by a single organization in a self-imposed mode or in a multi-party situation. Parties can be from the same organization or from different organizations and the situation can range from an informal agreement to a formal contract.

The processes in this document can be used as a basis for establishing business environments, e.g., methods, procedures, techniques, tools and trained personnel. Annex A provides normative direction regarding the tailoring of these software life cycle processes.

1.3 Field of application

This document applies to the full life cycle of software systems, products, and services, including conception, development, production, utilization, support and retirement, and to their acquisition and supply, whether performed internally or externally to an organization. The life cycle processes of this document can be applied concurrently, iteratively and recursively to a software system and incrementally to its elements.

There is a wide variety of software systems in terms of their purpose, domain of application, complexity, size, novelty, adaptability, quantities, locations, life spans and evolution. This document describes the processes that comprise the life cycle of man-made software systems. It therefore applies to one-of-a-kind software systems, software systems for wide commercial or public distribution, and customized, adaptable software systems. It also applies to a complete stand-alone software system and to software systems that are embedded and integrated into larger, more complex and complete systems.

This document provides a process reference model characterized in terms of the process purpose and the process outcomes that result from the successful execution of the activity tasks. Annex B lists examples of artifacts and information items that may be associated with various processes. This document can therefore be used as a reference model to support process assessment as specified in ISO/IEC 33002:2015. Annex C provides information regarding the use of the software life cycle processes as a process reference model. Annex D describes the process constructs for use in the process reference model. Annex I provides the correspondence between this document and ISO/IEC/IEEE 12207:2008 at the level of process name and process outcome.

1.4 Limitations

This document does not prescribe a specific software life cycle model, development methodology, method, modelling approach, or technique. The users of this document are responsible for selecting a life cycle model for the project and mapping the processes, activities, and tasks in this document into that model. The parties are also responsible for selecting and applying appropriate methodologies, methods, models and techniques suitable for the project.

This document does not establish a management system or require the use of any management system standard. However, it is intended to be compatible with the quality management system specified by ISO 9001, the service management system specified by ISO/IEC 20000-1 (IEEE Std 20000-1), and the information security management system specified by ISO/IEC 27000.

This document does not detail information items in terms of name, format, explicit content and recording media. ISO/IEC/IEEE 15289 addresses the content for life cycle process information items (documentation).

2 Normative references

There are no normative references in this document.

3 Terms, definitions, and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org>
- ISO Online browsing platform: available at <http://www.iso.org/obp>
- IEEE Standards Dictionary Online: available at <http://ieeexplore.ieee.org/xpls/dictionary.jsp>

Definitions for other terms typically can be found in ISO/IEC/IEEE 24765, *System and software engineering — Vocabulary*, available at <www.computer.org/sevocab>.

3.1.1

acquirer

stakeholder that acquires or procures a product or service from a supplier

Note 1 to entry: Other terms commonly used for an acquirer are buyer, customer, owner, purchaser or internal/organizational sponsor.

3.1.2**acquisition**

process of obtaining a system, product or service

3.1.3**activity**

set of cohesive tasks of a process

3.1.4**agile development**

software development approach based on iterative development, frequent inspection and adaptation, and incremental deliveries, in which requirements and solutions evolve through collaboration in cross-functional teams and through continual stakeholder feedback

[SOURCE: ISO/IEC/IEEE 26515: 2011]

3.1.5**agreement**

mutual acknowledgement of terms and conditions under which a working relationship is conducted

EXAMPLE

Contract, memorandum of agreement.

3.1.6**architecture**

<system> fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution

[SOURCE: ISO/IEC/IEEE 42010:2011]

3.1.7**architecture framework**

conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders

EXAMPLE 1
framework.

Generalised Enterprise Reference Architecture and Methodologies (GERAM) [ISO 15704] is an architecture

EXAMPLE 2

Reference Model of Open Distributed Processing (RM-ODP) [ISO/IEC 10746] is an architecture framework.

[SOURCE: ISO/IEC/IEEE 42010:2011]

3.1.8**architecture view**

work product expressing the architecture of a system from the perspective of specific system concerns

[SOURCE: ISO/IEC/IEEE 42010:2011]

3.1.9**architecture viewpoint**

work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific system concerns

[SOURCE: ISO/IEC/IEEE 42010:2011]

3.1.10**audit**

independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria

3.1.11

baseline

formally approved version of a configuration item, regardless of media, formally designated and fixed at a specific time during the configuration item's life cycle

[SOURCE: IEEE Std 828-2012]

3.1.12

business process

partially ordered set of enterprise activities that can be executed to achieve some desired end-result in pursuit of a given objective of an organization

3.1.13

concept of operations

verbal and/or graphic statement, in broad outline, of an organization's assumptions or intent in regard to an operation or series of operations

Note 1 to entry: The concept of operations frequently is embodied in long-range strategic plans and annual operational plans. In the latter case, the concept of operations in the plan covers a series of connected operations to be carried out simultaneously or in succession. The concept is designed to give an overall picture of the organization operations. See also operational concept (3.1.28).

Note 2 to entry: It provides the basis for bounding the operating space, system capabilities, interfaces and operating environment.

[SOURCE: ANSI/AIAA G-043A-2012e]

3.1.14

concern

<system> interest in a system relevant to one or more of its stakeholders

Note 1 to entry: A concern pertains to any influence on a system in its environment, including developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological and social influences.

[SOURCE: ISO/IEC/IEEE 42010:2011]

3.1.15

configuration item

item or aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process

EXAMPLE Software, firmware, data, hardware, humans, processes (e.g., processes for providing service to users), procedures (e.g., operator instructions and user manuals), facilities, services, materials, and naturally occurring entities

3.1.16

customer

organization or person that receives a product or service

EXAMPLE Consumer, client, user, acquirer, buyer, or purchaser.

Note 1 to entry: A customer can be internal or external to the organization.

3.1.17

design, verb

<process> to define the architecture, system elements, interfaces, and other characteristics of a system or system element

[SOURCE: ISO/IEC/IEEE 24765:2010, modified, changed 'components' to 'system element']

3.1.18

design, noun

result of the process in 3.1.17

Note 1 to entry: Information, including specification of system elements and their relationships, that is sufficiently complete to support a compliant implementation of the architecture

Note 2 to entry: Design provides the detailed implementation-level physical structure, behavior, temporal relationships, and other attributes of system elements.

3.1.19

design characteristic

design attributes or distinguishing features that pertain to a measurable description of a product or service

3.1.20

enabling system

system that supports a system-of-interest during its life cycle stages but does not necessarily contribute directly to its function during operation

EXAMPLE A configuration management system used to control software elements during software development.

Note 1 to entry: Each enabling system has a life cycle of its own. This document is applicable to each enabling system when, in its own right, it is treated as a system-of-interest.

3.1.21

environment

<system> context determining the setting and circumstances of all influences upon a system

[SOURCE: ISO/IEC/IEEE 42010:2011]

3.1.22

facility

physical means or equipment for facilitating the performance of an action, e.g., buildings, instruments, tools

3.1.23

incident

anomalous or unexpected event, set of events, condition, or situation at any time during the life cycle of a project, product, service, or system

3.1.24

information item

separately identifiable body of information that is produced, stored, and delivered for human use

[SOURCE: ISO/IEC/IEEE 15289:2015]

3.1.25

infrastructure

hardware and software environment to support computer system and software design, development, and modification

3.1.26

life cycle

evolution of a system, product, service, project or other human-made entity from conception through retirement

3.1.27

life cycle model

framework of processes and activities concerned with the life cycle, which can be organized into stages, acting as a common reference for communication and understanding

3.1.28

operational concept

verbal and graphic statement of an organization's assumptions or intent in regard to an operation or series of operations of a system or a related set of systems

Note 1 to entry: The operational concept is designed to give an overall picture of the operations using one or more specific systems, or set of related systems, in the organization's operational environment from the users' and operators' perspective. See also concept of operations (3.1.13).

3.1.29

operator

individual or organization that performs the operations of a system

Note 1 to entry: The role of operator and the role of user can be vested, simultaneously or sequentially, in the same individual or organization.

Note 2 to entry: An individual operator combined with knowledge, skills and procedures can be considered as an element of the system.

Note 3 to entry: An operator can perform operations on a system that is operated, or within a system that is operated, depending on whether or not operating instructions are placed within the system boundary.

3.1.30

organization

group of people and facilities with an arrangement of responsibilities, authorities and relationships

EXAMPLE company, corporation, firm, enterprise, institution, charity, sole trader, association, or parts or combination thereof.

Note 1 to entry: An identified part of an organization (even as small as a single individual) or an identified group of organizations can be regarded as an organization if it has responsibilities, authorities and relationships. A body of persons organized for some specific purpose, such as a club, union, corporation, or society, is an organization.

3.1.31

party

organization entering into an agreement

Note 1 to entry: In this document, the agreeing parties are called the acquirer and the supplier.

3.1.32

problem

difficulty, uncertainty, or otherwise realized and undesirable event, set of events, condition, or situation that requires investigation and corrective action

3.1.33

process

set of interrelated or interacting activities that transforms inputs into outputs

3.1.34

process outcome

observable result of the successful achievement of the process purpose

3.1.35

process purpose

high-level objective of performing the process and the likely outcomes of effective implementation of the process

Note 1 to entry: The purpose of implementing the process is to provide benefits to the stakeholders.

3.1.36

product

result of a process

Note 1 to entry: There are four agreed generic product categories: hardware (e.g., engine mechanical part); software (e.g., computer program procedures, and possibly associated documentation and data); services (e.g., transport); and processed materials (e.g., lubricant). Hardware and processed materials are generally tangible products, while software or services are generally intangible.

3.1.37

project

endeavour with defined start and finish criteria undertaken to create a product or service in accordance with specified resources and requirements

Note 1 to entry: A project is sometimes viewed as a unique process comprising coordinated and controlled activities and composed of activities from the Technical Management processes and Technical processes defined in this document.

3.1.38

<project> **portfolio**

collection of projects that addresses the strategic objectives of the organization

3.1.39

qualification

process of demonstrating whether an entity is capable of fulfilling specified requirements

3.1.40

quality assurance

part of quality management focused on providing confidence that quality requirements will be fulfilled

[SOURCE: ISO 9000:2015]

3.1.41

quality characteristic

inherent characteristic of a product, process or system related to a requirement

Note 1 to entry: Critical quality characteristics commonly include those related to health, safety, security assurance, reliability, availability and supportability.

3.1.42

quality management

coordinated activities to direct and control an organization with regard to quality

3.1.43

release

particular version of a configuration item that is made available for a specific purpose

EXAMPLE Test release.

3.1.44

requirement

statement that translates or expresses a need and its associated constraints and conditions

[SOURCE: ISO/IEC/IEEE 29148:2011, modified, NOTE has been removed.]

3.1.45

resource

asset that is utilized or consumed during the execution of a process

Note 1 to entry: Resources include those that are reusable, renewable or consumable.

EXAMPLE diverse entities such as funding, personnel, facilities, capital equipment, tools, and utilities such as power, water, fuel and communication infrastructures.

3.1.46

retirement

withdrawal of active support by the operation and maintenance organization, partial or total replacement by a new system, or installation of an upgraded system

3.1.47

risk

effect of uncertainty on objectives

Note 1 to entry: An effect is a deviation from the expected — positive or negative. A positive effect is also known as an opportunity.

Note 2 to entry: Objectives can have different aspects (such as financial, health and safety, and environmental goals) and can apply at different levels (such as strategic, organization-wide, project, product and process).

Note 3 to entry: Risk is often characterized by reference to potential events and consequences, or a combination of these.

Note 4 to entry: Risk is often expressed in terms of a combination of the consequences of an event (including changes in circumstances) and the associated likelihood of occurrence.

Note 5 to entry: Uncertainty is the state, even partial, of deficiency of information related to understanding or knowledge of an event, its consequence, or likelihood.

[SOURCE: ISO Guide 73:2009, definition 1.1]

3.1.48

safety

expectation that a system does not, under defined conditions, lead to a state in which human life, health, property, or the environment is endangered

3.1.49

security

protection against intentional subversion or forced failure; a composite of four attributes – confidentiality, integrity, availability, and accountability – plus aspects of a fifth, usability, all of which have the related issue of their assurance

[SOURCE: NATO AEP-67]

3.1.50

service

performance of activities, work, or duties

Note 1 to entry: A service is self-contained, coherent, discrete, and can be composed of other services.

Note 2 to entry: A service is generally an intangible product.

3.1.51

software element

system element that is software

3.1.52

software engineering

application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software

3.1.53

software item

source code, object code, control code, control data, or a collection of these items

Note 1 to entry: A software item can be viewed as a system element of this document and of ISO/IEC 15288:2015. Software items are typically configuration items.

3.1.54

software product

set of computer programs, procedures, and possibly associated documentation and data

Note 1 to entry: A software product is a software system viewed as the output (product) resulting from a process.

3.1.55

software system

system for which software is of primary importance to the stakeholders

Note 1 to entry: In the most general case, a software system is comprised of hardware, software, people, and manual procedures.

Note 2 to entry: In a software system, software is the leading driver in meeting system requirements.

3.1.56**software system element**

member of a set of elements that constitute a software system

Note 1 to entry: A software system element can include one or more software units, software elements, hardware units, hardware elements, services, and other system elements and systems.

Note 2 to entry: A software system element can be viewed as a system element.

3.1.57**software unit**

atomic-level software component of the software architecture that can be subjected to standalone testing

Note 1 to entry: Some software units are separately compilable pieces of code.

[SOURCE: ISO 26262-1:2011, modified, Note 1 to entry added.]

3.1.58**stage**

period within the life cycle of an entity that relates to the state of its description or realization

Note 1 to entry: As used in this document, stages relate to major progress and achievement milestones of the entity through its life cycle.

Note 2 to entry: Stages often overlap.

3.1.59**stakeholder**

individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations

EXAMPLE End users, end user organizations, supporters, developers, producers, trainers, maintainers, disposers, acquirers, supplier organizations and regulatory bodies.

Note 1 to entry: Some stakeholders can have interests that oppose each other or oppose the system.

3.1.60**supplier**

organization or an individual that enters into an agreement with the acquirer for the supply of a product or service

Note 1 to entry: Other terms commonly used for supplier are contractor, producer, seller, or vendor.

Note 2 to entry: The acquirer and the supplier sometimes are part of the same organization.

3.1.61**system**

combination of interacting elements organized to achieve one or more stated purposes

Note 1 to entry: A system is sometimes considered as a product or as the services it provides.

Note 2 to entry: In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g., aircraft system or database management system. Alternatively, the word “system” is substituted simply by a context-dependent synonym, e.g., aircraft or database, though this potentially obscures a system principles perspective.

Note 3 to entry: A system can include the associated equipment, facilities, material, software, firmware, technical documentation, services and personnel required for operations and support to the degree necessary for use in its intended environment.

Note 4 to entry: See for comparison: enabling system, system-of-interest, system of systems.

3.1.62**system element**

member of a set of elements that constitute a system

EXAMPLE Hardware, software, data, humans, processes (e.g., processes for providing service to users), procedures (e.g., operator instructions), facilities, materials, and naturally occurring entities or any combination.

Note 1 to entry: A system element is a discrete part of a system that can be implemented to fulfill specified requirements.

3.1.63

system-of-interest

SOI

system whose life cycle is under consideration

3.1.64

system of systems

SoS

set of systems that integrate or interoperate to provide a unique capability that none of the constituent systems can accomplish on its own

Note 1 to entry: Each constituent system is a useful system by itself, having its own management, goals, and resources, but coordinates within the SoS to provide the unique capability of the SoS.

3.1.65

systems engineering

interdisciplinary approach governing the total technical and managerial effort required to transform a set of stakeholder needs, expectations, and constraints into a solution and to support that solution throughout its life.

3.1.66

task

required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a process

3.1.67

technical management

application of technical and administrative resources to plan, organize and control engineering functions

3.1.68

trade-off

decision-making actions that select from various requirements and alternative solutions on the basis of net benefit to the stakeholders

3.1.69

traceability

degree to which a relationship can be established among two or more logical entities, especially entities having a predecessor-successor or master-subordinate relationship to one another, such as requirements, system elements, verifications, or tasks

EXAMPLE Software features and test cases are typically traced to software requirements.

3.1.70

user

individual or group that interacts with a system or benefits from a system during its utilization

Note 1 to entry: The role of user and the role of operator are sometimes vested, simultaneously or sequentially, in the same individual or organization.

[SOURCE: ISO/IEC 25010:2011, modified, Note 1 to entry added.]

3.1.71

validation

confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled

Note 1 to entry: A system is able to accomplish its intended use, goals and objectives (i.e., meet stakeholder requirements) in the intended operational environment. The right system was built.

Note 2 to entry: In a life cycle context, validation involves the set of activities for gaining confidence that a system is able to accomplish its intended use, goals and objectives in an environment like the operational environment.

3.1.72

verification

confirmation, through the provision of objective evidence, that specified requirements have been fulfilled

Note 1 to entry: Verification is a set of activities that compares a system or system element against the required characteristics. This includes, but is not limited to specified requirements, design, descriptions, and the system itself. The system was built right.

[SOURCE: ISO 9000:2015, modified, Note 1 to entry added.]

3.2 Abbreviated terms

CCB	Configuration Control Board
CM	Configuration Management
COTS	Commercial-Off-The-Shelf
FCA	Functional Configuration Audit
FOSS	Free and Open Source Software
GUI	Graphical User Interface
NDI	Non-Developmental Items
QA	Quality Assurance
PCA	Physical Configuration Audit
PESTEL	Political, Economic, Social, Technological, Environmental, and Legal
PMI	Project Management Institute
PMP	Project Management Plan
PRM	Process Reference Model
SCM	Software Configuration Management
SDP	Software Development Plan
SEMP	Systems Engineering Management Plan
SOI	System-of-Interest
SoS	System of Systems
SWOT	Strengths, Weaknesses, Opportunities, Threats
WBS	Work Breakdown Structure

4 Conformance

4.1 Intended usage

The requirements in this document are contained in Clause 6 and Annex A. This document provides requirements for a number of processes suitable for usage during the life cycle of a software system or product. It is recognized that particular projects or organizations may not need to use all of the processes provided by this document. Therefore, implementation of this document typically involves selecting and declaring a set of processes suitable to the organization or project. There are two ways that an implementation can be claimed to conform to the provisions of this document — full conformance and tailored conformance.

There are two criteria for claiming full conformance. Achieving either criterion suffices for conformance, although the chosen criterion (or criteria) shall be stated in the claim. Claiming “full conformance to tasks” asserts that all of the requirements of the activities and tasks of the declared set of processes are achieved. Alternatively, claiming “full conformance to outcomes” asserts that all of the required outcomes of the declared set of processes are

achieved. Full conformance to outcomes permits greater freedom in the implementation of conforming processes and can be useful for implementing processes to be used in the context of an innovative life cycle model.

NOTE 1 Options for conformance are provided for needed flexibility in the application of this document. Each process has a set of objectives (phrased as “outcomes”) and a set of activities and tasks that represent one way to achieve the objectives.

NOTE 2 Users who implement the activities and tasks of the declared set of processes can assert full conformance to tasks of the selected processes. Some users, however, might have innovative process variants that achieve the objectives (i.e., the outcomes) of the declared set of processes without implementing all of the activities and tasks. These users can assert full conformance to the outcomes of the declared set of processes. The two criteria — conformance to task and conformance to outcome — are necessarily not equivalent since specific performance of activities and tasks can require, in some cases, a higher level of capability than just the achievement of outcomes.

NOTE 3 When this document is used to help develop an agreement between an acquirer and a supplier, clauses of this document can be selected for incorporation in the agreement with or without modification. In this case, it is more appropriate for the acquirer and supplier to claim compliance with the agreement than conformance with this document.

NOTE 4 An organization (for example, national, industrial association, company) imposing this document, as a condition of trade, can specify and make public the minimum set of required processes, outcomes, activities, and tasks, which constitute suppliers’ compliance with the conditions of trade.

NOTE 5 Requirements of this document are marked by the use of the verb “shall”. Recommendations are marked by the use of the verb “should”. Permissions are marked by the use of the verb “may”. However, despite the verb that is used, the requirements for conformance are selected as described previously.

4.2 Full conformance

4.2.1 Full conformance to outcomes

A claim of full conformance declares the set of processes for which conformance is claimed. Full conformance to outcomes is achieved by demonstrating that all of the outcomes of the declared set of processes have been achieved. In this situation, the provisions for activities and tasks of the declared set of processes are guidance rather than requirements, regardless of the verb form that is used in the provision.

One intended use of this document is to facilitate process assessment and improvement. For this purpose, the objectives of each process are written in the form of ‘outcomes’ compatible with the provisions of ISO/IEC 33002. That standard provides for the assessment of the processes of this document, providing a basis for improvement. Users intending process assessment and improvement may use the process outcomes written in this document as the “process reference model” required by ISO/IEC 33002.

4.2.2 Full conformance to tasks

A claim of full conformance declares the set of processes for which conformance is claimed. Full conformance to tasks is achieved by demonstrating that all of the requirements of the activities and tasks of the declared set of processes have been achieved. In this situation, the provisions for the outcomes of the declared set of processes are guidance rather than requirements, regardless of the verb form that is used in the provision.

NOTE A claim of full conformance to tasks can be appropriate in contractual situations where an acquirer or a regulator requires detailed understanding of the suppliers’ processes.

4.3 Tailored conformance

When this document is used as a basis for establishing a set of processes that do not qualify for full conformance, the clauses of this document are selected or modified in accordance with the tailoring process prescribed in Annex A. The tailored text, for which tailored conformance is claimed, is declared. Tailored conformance is achieved by demonstrating that the outcomes, activities, and tasks, as tailored, have been achieved.

5 Key concepts and application

5.1 Introduction

This clause is included to highlight and to help explain essential concepts on which this document is based.

NOTE Further elaboration of these concepts can be found in ISO/IEC TS 24748-1, ISO/IEC TR 24748-2, and ISO/IEC TR 24748-3 on the application of life cycle management.

5.2 Software system concepts

5.2.1 Software systems

The software systems considered in this document are human-made, created and utilized to provide products or services in defined environments for the benefit of users and other stakeholders. These software systems can include the following system elements: hardware, software, data, humans, processes (e.g., processes for providing service to users), procedures (e.g., operator instructions), facilities, services, materials and naturally occurring entities. As viewed by the user, they are thought of as products or services.

This document applies to systems for which software is of primary importance to the stakeholders. It is based upon the general principles of systems engineering and software engineering. It is a fundamental premise of this document that software always exists in the context of a system. Since software does not operate without hardware, the processor upon which the software is executed can be considered as part of the system. Alternatively, hardware or services hosting the software system and handling communications with other systems can also be viewed as enabling systems or external systems in the operating environment.

The perception and definition of a particular software system, its architecture, and its elements depend on a stakeholder's interests and responsibilities. One stakeholder's system-of-interest can be viewed as a system element in another stakeholder's system-of-interest. Furthermore, a system-of-interest can be viewed as being part of the environment for another stakeholder's system-of-interest.

The following are key points regarding the characteristics of systems-of-interest:

- a) defined boundaries encapsulate meaningful needs and practical solutions;
- b) there is a hierarchical or other relationship between system elements;
- c) an entity at any level in the system-of-interest can be viewed as a system;
- d) a system comprises an integrated, defined set of subordinate system elements;
- e) humans can be viewed as both users external to a system and as system elements (i.e., operators) within a system; and
- f) a system can be viewed in isolation as an entity, i.e., a product; or as a collection of functions capable of interacting with its surrounding environment, i.e., a set of services.

Whatever the boundaries chosen to define the system, the concepts in this document are generic and permit a practitioner to correlate or adapt individual instances of life cycles to its system principles.

5.2.2 Software system structure

The life cycle processes in this document are described in relation to a software system that is composed of a set of interacting system elements (including software elements), each of which can be implemented to fulfill its respective specified requirements (Figure 1). Responsibility for the implementation of any system element may therefore be delegated to another party through an agreement.

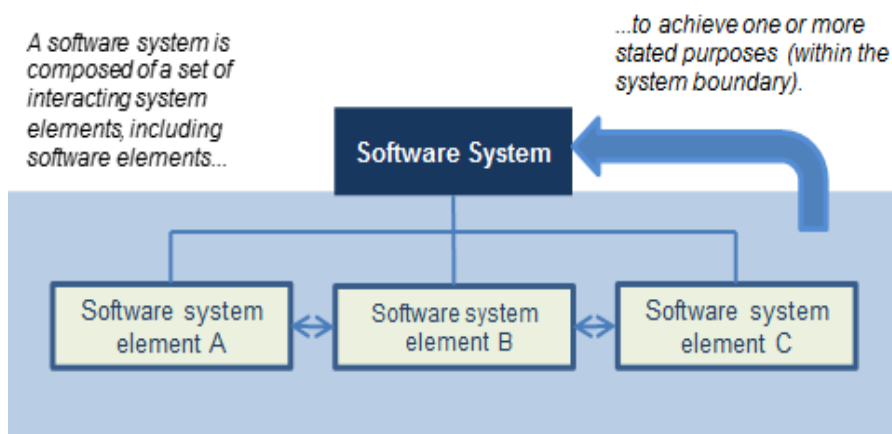


Figure 1 — Software system and software system element relationship

The relationship between the software system and its complete set of system elements can typically be represented showing relationships between the elements – often depicted as a hierarchy for the simplest of systems-of-interest. Decomposition is one approach to some software activities. Other approaches include the object-oriented approach, where the system elements are laid out in a flat (non-hierarchical) description such as in a network diagram. For more complex software systems-of-interest, a prospective system element may need to be considered as a system (that in turn is comprised of system elements) before a complete set of system elements can be defined with confidence (Figure 2). In this manner, the appropriate system life cycle processes are applied recursively to a system-of-interest to resolve its structure to the point where understandable and manageable software system elements can be implemented (created, adapted, acquired, or reused).

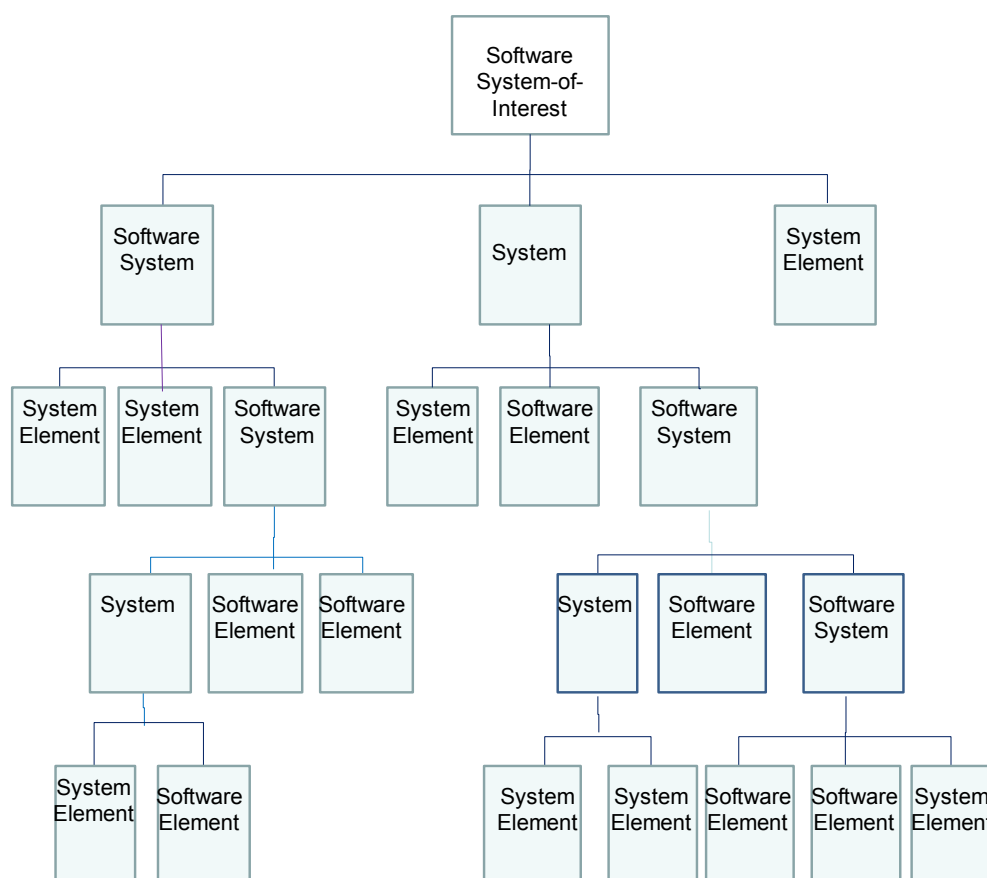


Figure 2 —Example of software system-of-interest structure

While Figures 1 and 2 imply a hierarchical relationship, in reality there are an increasing number of systems that, from one or more aspects, are not hierarchical, such as networks and other distributed systems. Annex G discusses the concept of a system of systems (SoS).

NOTE Decomposition is an activity fundamental to many software activities. Not all decompositions imply the designation of new software system elements and the corresponding recursive application of the activity. Designation of a decomposed construct as an element is necessary only when it is appropriate to apply distinct requirements, design, or implementation activities to its development. One example of an appropriate situation is when the element is to be developed by a distinct organization. Another example is when management determines that it is appropriate to distinctly monitor the status of the development or customization of the element.

5.2.3 Enabling systems

Throughout the life cycle of a system-of-interest, essential services are required from systems that are not directly a part of the operational environment of the system-of-interest, e.g., modelling system, training system, maintenance system. Each of these systems enables a part, e.g., a stage of the life cycle of the system-of-interest to be conducted. Termed “enabling systems”, they facilitate progression of the system-of-interest through its life cycle.

The relationship between the services delivered to the operational environment by the system-of-interest and the services delivered by the enabling systems to the system-of-interest is shown in Figure 3. Enabling systems can be seen to contribute indirectly to the services provided by the system-of-interest. The interrelationships between the system-of-interest and the enabling systems can be bidirectional or a one-way relationship. In addition to interacting with enabling systems, the system-of-interest can also interact with other systems in the operating environment, shown as Systems A, B, and C. Requirements for interfaces with enabling systems and other systems in the operational environment are included in the requirements for the system-of-interest.

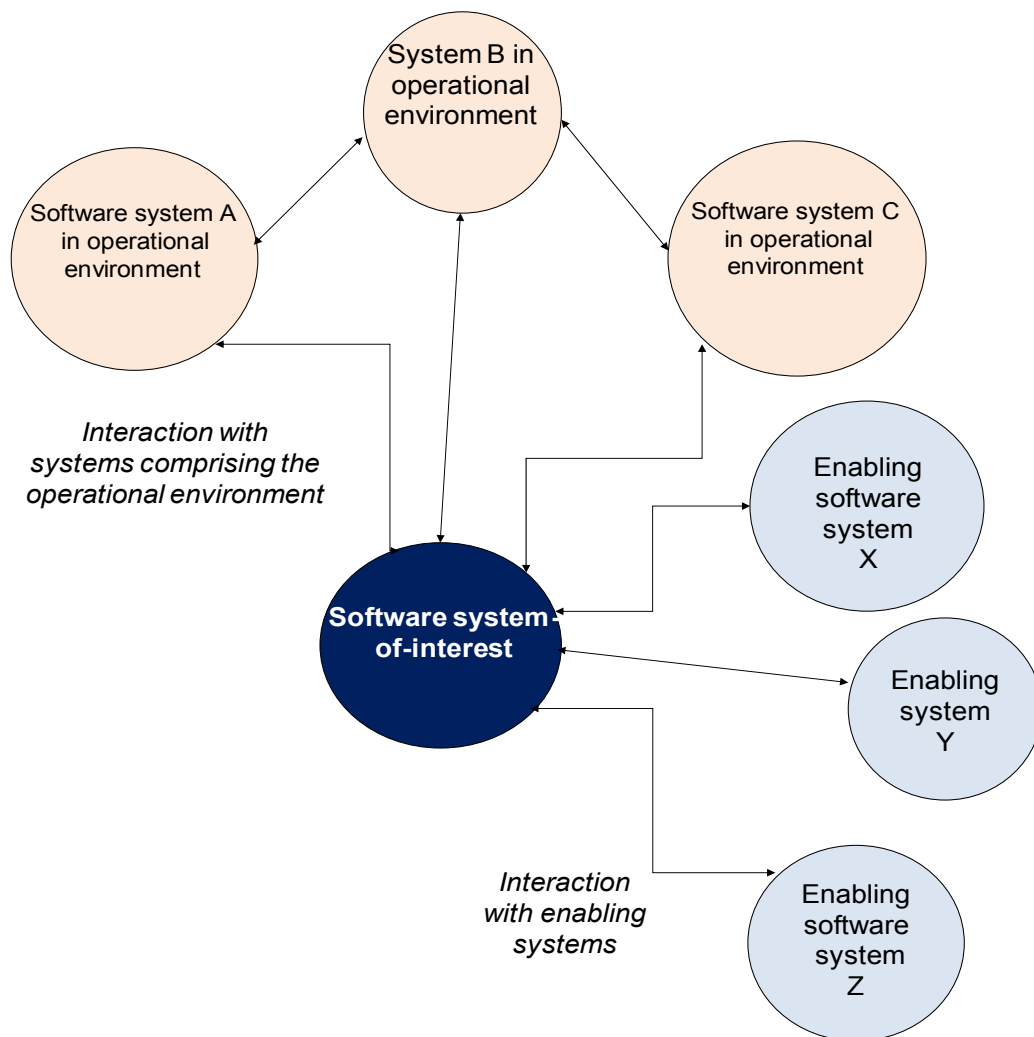


Figure 3 —Software system-of-interest, its operational environment and enabling systems

During a stage in the software life cycle, the relevant enabling systems and the system-of-interest are considered together. Since they are interdependent, they can also be viewed as a system. When a suitable enabling system does not already exist, the project that is responsible for the system-of-interest can be directly responsible for

creating and using the enabling system. Creating the enabling system can be viewed as a separate project and subsequently as another system-of-interest.

Further elaboration of these concepts can be found in ISO/IEC/IEEE 24748 (all parts) on the application of life cycle processes.

NOTE Enabling systems in software development include software development and test environments for target platforms.

5.2.4 Life cycle processes for the software system

In the software system, the requirements, architecture, and design processes at the system level result in an allocation of the system requirements to various elements. The software system-of-interest is implemented primarily by analyzing the software system requirements, architecture, and design and determining which functions will be implemented in software or by other elements, implementing the software and other elements, and integrating the elements as a software system. Therefore, a software product or service can be treated as an element of a software system.

In some cases, the architectural definition of a software system can indicate that it is appropriate to consider it as comprising a set of distinct subordinate elements. In turn, each of the software elements can be treated as a distinct software system as described previously. In these cases, this document may be applied recursively to procure or develop the subordinate elements.

This document has a strong relationship with ISO/IEC/IEEE 15288:2015, *Systems and Software Engineering--System Life Cycle Processes*, and is more applicable to software systems. To account for situations in which both ISO/IEC/IEEE 15288:2015 and ISO/IEC/IEEE 12207:2017 are applied (e.g., a development of a system containing software, or the development of a software system containing hardware), their process structures are harmonized to be identical. The processes of this document directly correspond to processes of ISO/IEC/IEEE 15288 with specialization for software products and services.

In the case where the system non-software elements have primary importance, an organization may decide to apply ISO/IEC/IEEE 15288 to perform the appropriate life cycle processes, activities and tasks. For each software element of the system, the organization may apply this document to create, adapt, acquire, or reuse the software elements.

5.3 Organization and project concepts

5.3.1 Organizations

When an organization, as a whole or a part, enters into an agreement, it is sometimes called a “party” to the agreement. Parties can be from the same organization or from separate organizations. An organization can be as small as a single individual, if the individual is assigned responsibilities and authorities.

In informal terms, the organization that is responsible for executing a process is sometimes referred to by the name of that process. For example, the organization executing the Acquisition process is sometimes called the “acquirer”. Other examples include supplier, implementer, maintainer, and operator.

A few other terms are applied to organizations in this document: “user” can be the organization or individuals that directly engage with or benefit from the utilization of the product or service; “customer” refers to the user and acquirer collectively; and “stakeholder” refers to an individual or organization with an interest in the system.

The processes and organizations are only related functionally. This document does not dictate or imply a structure for an organization, nor does it specify that particular processes are to be executed by particular parts of the organization. It is the responsibility of the organization that implements this document to define a suitable structure for the organization and assign appropriate roles for the execution of processes.

The processes in this document form a comprehensive set to serve various organizations. An organization, small or large, depending on its business purpose or its acquisition strategy, can select an appropriate set of the processes (and associated activities and tasks) to fulfill that purpose. An organization can perform one process or more than one process.

This document is intended to be applied by an organization internally or externally by two or more organizations. When applied internally, the two agreeing parties typically act under the terms of an agreement that may vary in formality under different circumstances. When applied externally, the two agreeing parties typically act under the terms of a contract. This document uses the term “agreement” to apply to either situation.

For the purpose of this document, any project is assumed to be conducted within the context of an organization. This is important, because a project is dependent upon various outcomes produced by the business processes of the organization, e.g., employees to staff the project and facilities to house the project. For this purpose, this document provides a set of “Organizational Project-Enabling” processes. These processes are not assumed to be adequate to operate a business; instead the processes, considered as a collection, are intended to state the minimum set of dependencies that the project places upon the organization.

5.3.2 Organization and project-level adoption

Modern businesses strive to develop a robust set of life cycle processes that are applied repeatedly to the projects and services of the business. Therefore, this document is intended to be useful for adoption at either the organization level or at the project level. An organization can adopt the document and supplement it with appropriate procedures, practices, tools and policies. In turn, a project of the organization typically conforms to the organization’s processes rather than conforming directly to this document.

In some cases, projects may be executed by an organization that does not have an appropriate set of processes adopted at the organizational level. Such a project may apply the provisions of this document directly to the project.

5.4 Life cycle concepts

5.4.1 Software life cycle stages

Life cycles vary according to the nature, purpose, use and prevailing circumstances of the software system. Using stages concurrently and in different orders can lead to life cycle forms with distinctly different characteristics. Each stage has a distinct purpose and contribution to planning and executing the whole life cycle of the software system. Per ISO/IEC TS 24748-1, the typical system life cycle stages include concept, development, production, utilization, support, and retirement. Use of these terms to define stages is not normative. A common set of stages for a software system is concept exploration, development, sustainment, and retirement, with transitions between stages for the system as a whole and for its elements.

The stages represent the major life cycle periods associated with a software system and they relate to the state of the software system description or the software system itself. The stages describe the major progress and achievement milestones of the software system through its life cycle. They give rise to the primary decision gates of the life cycle. These decision gates are used by organizations to understand and manage the inherent uncertainties and risks associated with costs, schedule and functionality when creating or utilizing a software system. Using stages thus provides organizations with a framework within which organization management has high-level visibility and control of project and technical processes. Organizations define and employ stages differently to satisfy contrasting business and risk mitigation strategies.

The life cycle processes defined in this document are not aligned to any specific stage in a software life cycle. All of the life cycle processes involve planning, performance, and evaluation activities that should be considered for use at every stage.

Further elaboration of these concepts can be found in ISO/IEC/IEEE 24748 (all parts), on the application of life cycle management.

5.4.2 Life cycle model for the software system

Every software system has a life cycle. A life cycle can be described using an abstract functional model that represents the conceptualization of a need for the system, its realization, utilization, evolution and disposal.

A software system progresses through its life cycle as the result of actions, performed and managed by people in organizations, using processes for execution of these actions. The detail in the life cycle model is expressed in terms of these processes, their outcomes, relationships and sequence.

This document does not prescribe any particular life cycle model. Instead it defines a set of processes, termed life cycle processes, which can be used in the definition of the system's life cycle. Also, this document does not prescribe any particular sequence of processes within the life cycle model. The sequence of the processes is determined by project objectives and by selection of the life cycle model. Often, the development stage is subdivided more finely and in different ways.

One oft-cited set of software development stages are elicitation, requirements, design, construction, and testing—the predictive or “waterfall” model. If the stages are considered as sequential, then each stage is required to produce correct results before proceeding to the next stage. In practice, this is extremely difficult to achieve unless the requirements are known well and the initial cost estimates are accurate. In performing a waterfall, one risks performing extensive rework that does not properly fall within any of the planned stages, hence probably does not fall within any budget.

NOTE 1 Winston Royce, commonly recognized as an early analyst of life cycle process models, described the need for rework stages rather than the “waterfall” (a term that he did not use). Unfortunately, the rework stages were dropped from the “waterfall” model as it was popularly understood.

To deal with the issues of incompletely known requirements and inaccurate estimates, a number of other types of models have been proposed: incremental, spiral, iterative, and evolutionary (adaptive). These life cycle models can incorporate agile techniques and methods. These models can typically involve repeated performance of the life cycle processes and stages during the life cycle, e.g., for different increments of the software product, for more precise handling of exceptions to common functions, or for requirements that were not fully defined at the outset. These models can be applied across stages, such as development and utilization or deployment. Use of these models can affect software release strategies and acquisition strategies for software services.

EXAMPLE Software elements can be developed incrementally, and then held for block operational release at a convenient time in the organization's business cycle.

The “incremental development” model includes initial planning, initial requirements analysis, initial architectural definition, and initial validation, but allocates design, implementation, verification (and sometimes delivery) activities to a series of stages, each of which provides a portion of the intended functionality. The approach provides for some flexibility to respond to inaccurate cost or schedule estimates by moving functionality to later increments.

The “spiral” variation on incremental developmental proposes ordering the development of functionality based on risk, with the riskiest problems considered in the early increments. This provides some protection against cost surprises occurring late in the development cycle.

The “iterative development” model performs initial planning and then consists of a cyclic process of prototyping, testing, analyzing and refining the requirements and the solution. “Iterative” models repeatedly perform the life cycle processes to deliver prioritized system functions sooner, with refined or more complex elements of the system coming in later iterations.

The “evolutionary model” is intended to deal with incomplete knowledge of requirements. It provides for initial planning and initial architecture definition, but allocates requirements analysis, design, construction, verification, validation and delivery to a series of stages. Delivered capabilities that do not meet user needs can be reworked in subsequent stages of the evolution.

“Agile” methods actually can be applied within a variety of models. While Agile methods are common in executing an evolutionary lifecycle model, they can be used in other lifecycle models at various stages. What the methods have in common is an emphasis on continuous inspection and collaboration in the rapid production of working software in an environment where changes, including changes to requirements, are expected. Annex H provides information on the application of this document in an agile context.

NOTE 2 Selecting the name of a type of model does not satisfy the requirement to define a model comprised of stages, with defined purpose and outcomes accomplished via the processes of this document.

NOTE 3 ISO/IEC TS 24748-1, ISO/IEC TR 24748-2, ISO/IEC TR 24748-3, and ISO/IEC/IEEE 24748-4 provide additional detail regarding life cycle models and stages. The models described in this clause apply not only to software systems but also to other systems as described in ISO/IEC/IEEE 15288:2015.

5.5 Process concepts

5.5.1 Criteria for processes

The determination of the life cycle processes in this document is based upon three basic principles:

- 1) Each life cycle process has strong relationships among its outcomes, activities and tasks.
- 2) The dependencies among the processes are reduced to the greatest feasible extent.
- 3) A process is capable of execution by a single organization in the life cycle.

5.5.2 Description of processes

Each process of this document is described in terms of the following attributes:

- a) The title conveys the scope of the process as a whole.
- b) The purpose describes the goals of performing the process.
- c) The outcomes express the observable results expected from the successful performance of the process.
- d) The activities are sets of cohesive tasks of a process.
- e) The tasks are requirements, recommendations, or permissible actions intended to support the achievement of the outcomes.

The processes and process groups in this document are identical in their purpose and outcomes with those in ISO/IEC/IEEE 15288:2015, *System and software engineering – System life cycle processes*, with one exception: the System/Software Requirements Definition process of this document is renamed from the System Requirements Definition process of ISO/IEC/IEEE 15288:2015. To emphasize this harmonization of systems and software system processes, the process purposes and outcomes are presented in boxes in Clause 6.

Software-specific activities, tasks, and work products are applied to achieve the outcomes of the processes in this document. Annex E provides additional process views.

Additional detail regarding this form of process description can be found in ISO/IEC TR 24774.

5.5.3 General characteristics of processes

In addition to the basic attributes described in the previous subclause, processes may be characterized by other attributes common to all processes. ISO/IEC 33020:2015 identifies common process attributes that characterize six levels of achievement within a measurement framework for process capability. Annex C includes the list of process attributes that contribute to the achievement of higher levels of process capability as defined in ISO/IEC 33020:2015.

5.5.4 Tailoring

Annex A, which is normative, defines the basic activities needed to perform tailoring. Note that tailoring may diminish the perceived value of a claim of conformance to this document. This is because it is difficult for other organizations to understand the extent to which tailoring may have deleted desirable provisions. An organization asserting a single-party claim of conformance to this document may find it advantageous to claim full conformance to a smaller list of processes rather than tailored conformance to a larger list of processes.

5.6 Process groups

5.6.1 Introduction

This document groups the activities that can be performed during the life cycle of a software system into four process groups. Each of the life cycle processes within those groups is described in terms of its purpose and

desired outcomes with a set of related activities and tasks that can be performed to achieve those outcomes. The four process groups and the processes included in each group are depicted in Figure 4 as follows:

- a) Agreement processes;
- b) Organizational Project-Enabling Processes;
- c) Technical Management Processes; and
- d) Technical Processes.

The processes described in this document are not intended to preclude or discourage the use of additional processes that organizations find useful. The order of the subclauses in which the processes are defined in this document does not determine the order in which the processes are performed during the system life cycle or any of its stages. A description of each process group is provided in the four subclauses that follow.

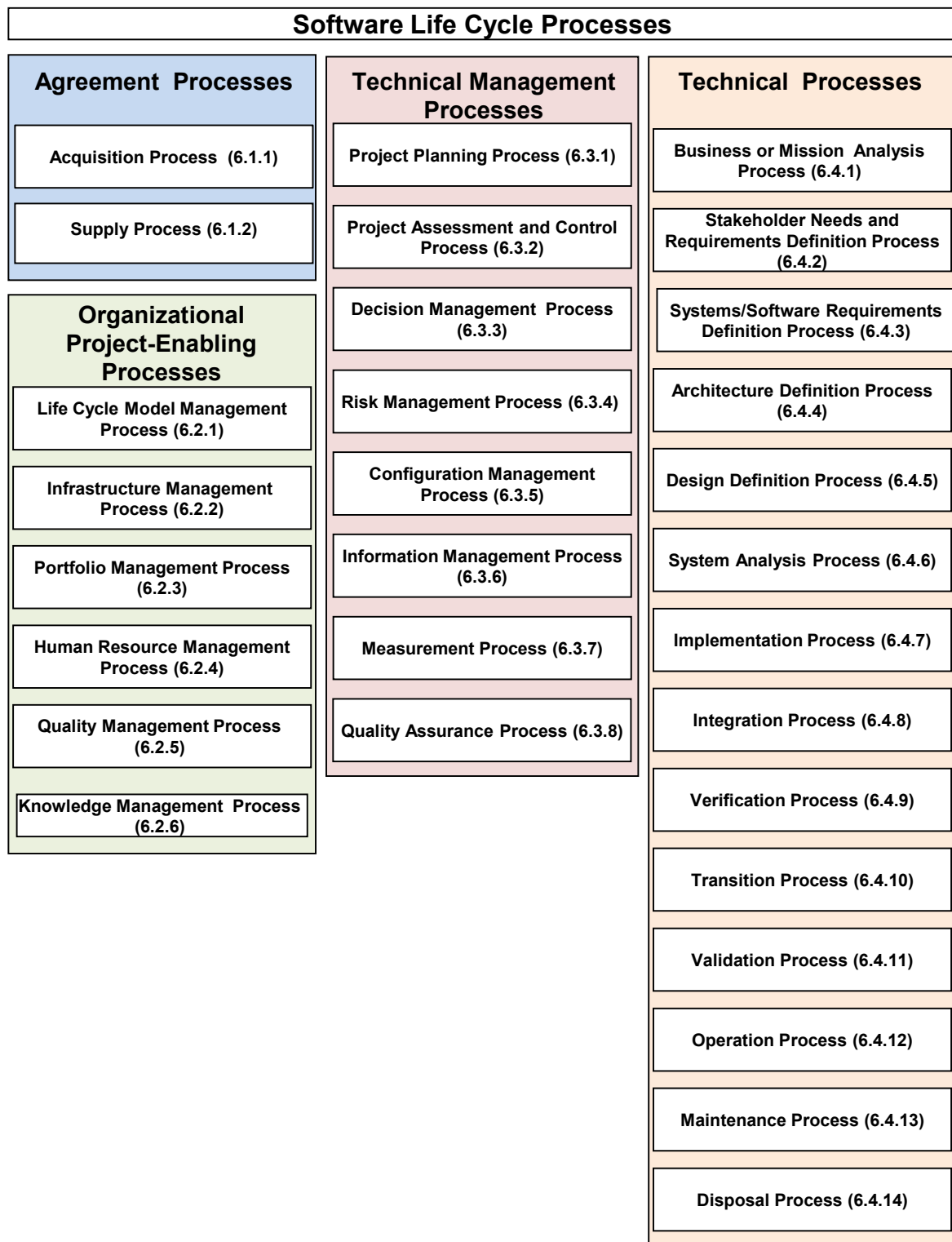


Figure 4 — Software life cycle processes

5.6.2 Agreement processes

Organizations are producers and users of software systems. One organization (acting as an acquirer) can task another (acting as a supplier) for products or services. This is achieved using agreements. Agreements allow both acquirers and suppliers to realize value and support business strategies for their organizations.

The Agreement processes are organizational processes that apply outside of the span of a project's life, as well as for a project's lifespan. Generally, organizations act simultaneously or successively as both acquirers and suppliers of software systems. The Agreement processes can be used with less formality when the acquirer and the supplier are in the same organization. Similarly, they can be used within the organization to agree on the respective

responsibilities of organization, project and technical functions. Figure 4 lists the processes contained in this process group.

5.6.3 Organizational project-enabling processes

The Organizational Project-Enabling processes are concerned with providing the resources to enable the project to meet the needs and expectations of the organization's stakeholders. The Organizational Project-Enabling processes are typically concerned at a strategic level with the management and improvement of the organization's business or undertaking, with the provision and deployment of resources and assets, and with its management of risks in competitive or uncertain situations. The Organizational Project-Enabling processes apply outside the span of a project's life, as well as during a project's lifespan.

The Organizational Project-Enabling processes establish the environment in which projects are conducted. The organization establishes the processes and life cycle models to be used by projects; establishes, redirects, or cancels projects; provides resources required, including human and financial; and sets and monitors the quality measures for software systems and other deliverables that are developed by projects for internal and external customers.

The Organizational Project-Enabling processes create a strong business image for many organizations and imply commercial and profit-making motives. Nevertheless, the Organizational Project-Enabling processes are equally relevant to non-profit organizations, since they are also accountable to stakeholders, are responsible for resources, and encounter risk in their undertakings. This document can be applied to non-profit organizations as well as to profit-making organizations. Figure 4 lists the processes contained in this process group.

5.6.4 Technical Management processes

The Technical Management processes are concerned with managing the resources and assets allocated by organization management and with applying them to fulfill the agreements into which the organization or organizations enter. The Technical Management processes relate to the technical effort of projects, in particular to planning in terms of cost, timescales and achievements, to the checking of actions to help ensure that they comply with plans and performance criteria and to the identification and selection of corrective actions that recover shortfalls in progress and achievement. These processes are used to establish and perform technical plans for the project, manage information across the technical team, assess technical progress against the plans for the software system, products, or services, control technical tasks through to completion, and aid in decision-making.

NOTE 1 Technical management is 'the application of technical and administrative resources to plan, organize and control engineering functions'. (ISO/IEC/IEEE 24765:2010)

Typically, several projects will co-exist in any one organization. The Technical Management processes can be employed at a corporate level to meet internal needs. Figure 4 lists the processes contained in this process group.

NOTE 2 Technical Management processes are applied during the performance of each Technical process.

5.6.5 Technical processes

The Technical processes are concerned with technical actions throughout the life cycle. Technical processes transform the needs of stakeholders into a product or service. By applying that product or operating that service, technical processes, provide sustainable performance, when and where needed in order to meet the stakeholder requirements and achieve customer satisfaction. The Technical processes are applied in order to create and use a software system, whether it is in the form of a model or is an operational product. The Technical processes apply at any level in a hierarchy of software system structure and at any stage in the life cycle. Figure 4 lists the processes contained in this process group.

5.7 Process application

The life cycle processes defined in this document can be used by any organization when acquiring, using, creating, or supplying a software system. They can be applied at any level in a system's hierarchy and at any stage in the life cycle.

The functions these processes perform are defined in terms of specific purposes, outcomes and the set of activities and tasks that constitute the process.

Each life cycle process in Figure 4 can be invoked, as required, at any time throughout the life cycle. The order that the processes are presented in this document does not imply any prescriptive order in their use. However, sequential relationships are introduced by the definition of a life cycle model. The detailed purpose and timing of use of these processes throughout the life cycle are influenced by multiple factors, including social, trading, organizational and technical considerations, each of which can vary during the life of a software system. An individual software life cycle is thus created through a selection and application of processes that will normally possess concurrent, iterative, recursive and time-dependent characteristics.

Concurrent use of processes can exist within a project (e.g., when design actions and preparatory actions for building a software system are performed at the same time), and between projects (e.g., when system elements are designed at the same time under different project responsibilities).

When the application of the same process or set of processes is repeated on the same system, the application is referred to as iterative. The iterative use of processes is important for the progressive refinement of process outputs, e.g., the interaction between successive verification actions and integration actions can incrementally build confidence in the conformance of the product. Iteration is not only appropriate but also expected. New information is created by the application of a process or set of processes. Typically, this information takes the form of questions with respect to requirements, analyzed risks or opportunities. Such questions should be resolved before completing the activities of a process or set of processes.

The recursive use of processes, i.e., the repeated application of the same process or set of processes applied to successive levels of system elements in a system's structure, is a key aspect of the application of this document. The outputs of processes at any level, whether information, artifacts or services, are inputs to the processes used at the level below (e.g., during top down design) or at the level above (e.g., during software system realization). The outcomes from one application are used as inputs to the next lower (or higher) system in the system structure to arrive at a more detailed or mature set of outcomes. Such an approach adds value to successive systems in the system structure.

The changing nature of the influences on the software system (e.g., operational environment changes, new opportunities for system element implementation, modified structure and responsibilities in organizations) requires continual review of the selection and timing of process use. Process use in the life cycle can be dynamic, responding to the many external influences on the software system. The life cycle approach also allows for incorporating the changes in the next stage. The life cycle stages assist the planning, execution and management of life cycle processes in the face of this complexity in life cycles by providing comprehensible and recognizable high-level purpose and structure. The set of processes within a life cycle stage are applied with the common goal of satisfying the exit criteria for that stage or the entry criteria of the formal progress reviews within that stage.

The discussion in this section on iterative and recursive use of software life cycle processes is not meant to imply any specific hierarchical, vertical, or horizontal structure for the system-of-interest, enabling system, organization, or project.

Where justified by product quality risks, detailed descriptions of process instances in the context of the specific product may also be created. Instantiation of processes involves identifying specific success criteria for a process instance, derived from the product requirements, and identifying the specific activities and tasks needed to achieve the success criteria, derived from the activities and tasks identified in this document. Creating detailed descriptions of process instances enables better management of product quality risks by establishing the link between the process and the specific product requirements.

Further elaboration of these concepts can be found in ISO/IEC/IEEE 24748 (all parts) on the application of life cycle processes.

5.8 Process reference model

Annex C defines a process reference model (PRM) at a level of abstraction higher than that of the detailed requirements contained in Clause 6. The PRM is applicable to an organization that is assessing its processes in order to determine the capability of these processes. The purpose and outcomes are a statement of the goals of the performance of each process. This statement of goals permits assessment of the effectiveness of the processes in ways other than simple conformity assessment.

NOTE In this document, the term “process reference model” is used with the same meaning as ISO/IEC 33001:2015: “model comprising definitions of processes in a domain of application described in terms of process purpose and outcomes, together with an architecture describing the relationships between the processes”.

6 Software life cycle processes

6.1 Agreement processes

This subclause specifies the requirements for the establishment of agreements with organizational entities external and internal to the organization.

The Agreement processes consist of the following:

- a) Acquisition process – used by organizations for acquiring products or services; and
- b) Supply process – used by organizations for supplying products or services.

These processes define the activities necessary to establish an agreement between two organizations. If the Acquisition process is invoked, it provides the means for conducting business with a supplier. This may include products that are supplied for use as an operational software system, services in support of operational activities, software elements of a system, or elements of a software system being provided by a supplier. If the Supply process is invoked, it provides the means for an agreement in which the result is a product or service that is provided to the acquirer.

NOTE Security is an increasing concern in systems and software engineering. See ISO/IEC 27036, *Security techniques - Information security for supplier relationships*, for requirements and guidance for suppliers and acquirers on how to secure information in supplier relationships. Specific aspects of information security supplier relationships are addressed in ISO/IEC 27036-3:2013 and ISO/IEC 27036-4 (in development).

6.1.1 Acquisition process

6.1.1.1 Purpose

The purpose of the Acquisition process is to obtain a product or service in accordance with the acquirer's requirements.

NOTE As part of this process, the agreement is modified when a change request affecting the terms of the agreement is agreed to by both the acquirer and supplier.

6.1.1.2 Outcomes

As a result of the successful implementation of the Acquisition process:

- a) A request for supply is prepared.
- b) One or more suppliers are selected.
- c) An agreement is established between the acquirer and supplier.
- d) A product or service complying with the agreement is accepted.
- e) Acquirer obligations defined in the agreement are satisfied.

6.1.1.3 Activities and tasks

The acquirer shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Acquisition process.

NOTE 1 The activities and resulting agreement from this process often apply to suppliers in the supply chain, including subcontracted suppliers.

NOTE 2 IEEE Std 1062-2015, *IEEE Recommended Practice for Software Acquisition*, contains detailed activities for software acquisition alternatives, including custom-developed, off-the-shelf, and software as a service. IEEE Std 1062-2015 also provides software acquisition guidance for technical data rights and intellectual property considerations, for consideration

when safety assurance or information security requirements are of concern, and checklists of relevant issues for organizational consideration when establishing software acquisition processes.

a) **Prepare for the acquisition.** This activity consists of the following tasks:

1) Define a strategy for how the acquisition will be conducted.

NOTE 1 This strategy describes or references the life cycle model, risks and issues mitigation, a schedule of milestones, and selection criteria if the supplier is external to the acquiring organization. It also includes key drivers and characteristics of the acquisition, such as responsibilities and liabilities; specific models, methods, or processes; level of criticality; formality; and priority of relevant trade factors.

NOTE 2 The Decision Management process and System Analysis process are often used to support trade-offs for the acquisition strategy. Examples include: determining a make or buy decision, as well as the suitability of specific COTS or modified OTS solutions and supplier evaluation.

NOTE 3 If the strategy calls for acquisition of specific commercial-off-the-shelf software or open source software, acquisition can be limited to identifying the supplier, accepting or negotiating the conditions in a pre-defined license or lease or maintenance agreement, determining rights to intellectual property and data rights in the software system, and agreeing on the price.

NOTE 4 A factor to consider in agreements between suppliers is data rights and lateral access to constituent data and intellectual property. As an example, suppliers for one system component may need to collaborate with suppliers for another component and share source code. Agreements can enable this collaboration.

2) Prepare a request for the supply of a product or service that includes the requirements.

NOTE 1 If a supplier is external to the organization, then the request includes the business practices with which a supplier is expected to comply and the criteria for selecting a supplier.

NOTE 2 A definition of requirements is provided to one or more suppliers. The requirements are the stakeholder or the system/software requirements, depending on the type of acquisition approach, and using the associated requirements definition process.

NOTE 3 The acquirer develops the requirements by itself or retains a supplier to develop them. If the acquirer retains a supplier to develop requirements, the acquirer retains approval authority for the requirements developed by the supplier.

b) **Advertise the acquisition and select the supplier.** This activity consists of the following tasks:

1) Communicate the request for the supply of a product or service to potential suppliers; and

2) Select one or more suppliers.

NOTE To obtain competitive solicitations, proposals to supply are evaluated and compared against the selection criteria and ranked. The justification for rating each proposal is declared and suppliers commonly are informed why they were or were not selected.

c) **Establish and maintain an agreement.** This activity consists of the following tasks:

NOTE Project cost, schedule, and performance are monitored through the Project Assessment and Control process. Any identified issues that require agreement modifications are referred to this activity. Any proposals for changes to system elements or information are controlled through the Change Management activity of the Configuration Management process.

1) Develop an agreement with the supplier that includes acceptance criteria.

NOTE 1 This agreement ranges in formality from a written contract to a verbal agreement. Appropriate to the level of formality, the agreement establishes requirements, development and delivery milestones, verification, validation and acceptance conditions, exception-handling procedures, agreement change management procedures, and payment schedules, so that both parties of the agreement understand the basis for executing the agreement. Other provisions for agreements include rights and restrictions associated with technical data and intellectual property, acceptance test preparation and test environment details; and the extent of supplier involvement. The agreement identifies process requirements that need to be imposed on participating subcontractors, such as configuration management requirements, reporting of risks, and reporting of measures and measurement analysis.

NOTE 2 Acceptance criteria, such as acceptance testing, relate to how the product or service will satisfy its intended use in its operational environment. Acceptance testing can be performed using the Validation process. Exceptions that arise during the conduct of the agreement or with the delivered product or service are resolved according to the procedures established in the agreement.

2) Identify necessary changes to the agreement.

NOTE In requesting a change to the agreement, the acquirer or the supplier details its specifications, rationale, and background.

3) Evaluate impact of changes on the agreement.

NOTE Any change is investigated for impacts to project plans, schedule, cost, technical capability, and quality. A change can be handled within the existing agreement, can require a modification to the agreement, or can require a new agreement.

4) Negotiate the agreement with the supplier.

NOTE Agreement terms are negotiated between the acquirer and supplier. Negotiation occurs for the initial agreement, and as required for any changes. Changed agreements are based on the required change and identified impacts. Details are discussed and changed during negotiation, after which the acquirer and supplier accept the terms of an agreement and the agreement commences. For a written contract, this occurs when the contract is signed or as specified in the agreement.

5) Update the agreement with the supplier, as necessary.

NOTE 1 The result of the agreement modification is incorporated into the project plans and communicated to all affected parties.

NOTE 2 Agreements can specify the conditions under which the agreement will be terminated by either party, e.g., unexpected changes in strategy or available funding, or lack of satisfactory progress.

d) **Monitor the agreement.** This activity consists of the following tasks:

1) Assess the execution of the agreement.

NOTE 1 This includes confirmation that all parties are meeting their responsibilities according to the agreement. The Project Assessment and Control process is used to evaluate projected cost, schedule, performance, and the impact of undesirable outcomes on the organization. This information is combined with other assessments of the execution of the terms of the agreement. If execution of the agreement does not result in an acceptable product or service, the acquirer or supplier can terminate the agreement as allowed in its terms.

NOTE 2 Acceptance testing can be performed using the Validation process. Exceptions that arise during the conduct of the agreement or with the delivered product or service are resolved according to the procedures established in the agreement.

2) Provide data needed by the supplier and resolve issues in a timely manner.

e) **Accept the product or service.** This activity consists of the following tasks:

1) Confirm that the delivered product or service complies with the agreement.

NOTE If the agreed requirements are satisfied and the acceptance criteria are met, the supplier has fulfilled its obligation. Unaddressed exceptions, e.g., disputes over conduct of acceptance testing or product suitability for intended use, are a matter for agreement provisions relating to disputes, arbitration or applicable law and regulation.

2) Provide payment or other agreed consideration.

3) Accept the product or service from the supplier, or other party, as directed by the agreement.

4) Close the agreement.

NOTE The project is closed by the Portfolio Management process.

6.1.2 Supply process

6.1.2.1 Purpose

The purpose of the Supply process is to provide an acquirer with a product or service that meets agreed requirements.

NOTE As part of this process, the agreement is modified when a change request affecting the terms of the agreement is agreed to by both the acquirer and supplier.

6.1.2.2 Outcomes

As a result of the successful implementation of the Supply process:

- a) An acquirer for a product or service is identified.
- b) A response to the acquirer's request is produced.
- c) An agreement is established between the acquirer and supplier.
- d) A product or service is provided.
- e) Supplier obligations defined in the agreement are satisfied.
- f) Responsibility for the acquired product or service, as directed by the agreement, is transferred.

6.1.2.3 Activities and tasks

The supplier shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Supply process.

a) **Prepare for the supply.** This activity consists of the following tasks:

- 1) Determine the existence and identity of an acquirer who has a need for a product or service.

NOTE This is often generated through the Business or Mission Analysis process. For a product or service developed for consumers, an agent, e.g., a marketing function within the supplier organization, often represents the acquirer.

- 2) Define a supply strategy.

NOTE This strategy describes or references the life cycle model, risks and issues mitigation, and a schedule of milestones. It also includes key drivers and characteristics of the acquisition, such as responsibilities and liabilities; specific models, methods, or processes; level of criticality; formality; and priority of relevant trade factors.

b) **Respond to a request for supply of products or services.** This activity consists of the following tasks:

- 1) Evaluate a request for the supply of a product or service) to determine feasibility and how to respond.
- 2) Prepare a response that satisfies the solicitation.

c) **Establish and maintain an agreement.** This activity consists of the following tasks:

- 1) Negotiate an agreement with the acquirer that includes acceptance criteria.

NOTE This agreement ranges in formality from a written contract to a verbal agreement. The Supplier confirms that the requirements, delivery milestones, and acceptance conditions are achievable, that exception handling and agreement change management procedures and payment schedules are acceptable, and that they establish a basis for executing the agreement without unnecessary risks. Issues are discussed and resolved during negotiation, after which the acquirer and supplier accept the terms of an agreement and the agreement commences. For a contract, this occurs when the contract is signed.

- 2) Identify necessary changes to the agreement.

NOTE In requesting a change to the agreement, the acquirer or the supplier details its specifications, rationale, and background.

- 3) Evaluate impact of changes on the agreement.

NOTE Any change is investigated for impacts to project plans, schedule, cost, technical capability, or quality. A change can be handled within the existing agreement, can require a modification to the agreement, or can require a new agreement.

- 4) Negotiate the agreement with the acquirer, as necessary.

NOTE Changes to agreement terms are negotiated between the supplier and acquirer. This includes changes due to changing market context. Negotiation occurs for the initial agreement, and as required for any changes. Changed agreements are based on the required change and identified impacts.

- 5) Update the agreement with the acquirer, as necessary.

NOTE The result of the agreement modification is incorporated into the project plans and communicated to all affected parties.

d) **Execute the agreement.** This activity consists of the following tasks:

- 1) Execute the agreement according to the established project plans.

NOTE A supplier sometimes adopts, or agrees to use, acquirer processes.

- 2) Assess the execution of the agreement.

NOTE This includes confirmation that all parties are meeting their responsibilities according to the agreement. The Project Assessment and Control process is used to evaluate projected cost, schedule, performance, and the impact of undesirable outcomes on the organization. The change management activity of the Configuration Management process is used to control changes to the system elements. This information is combined with other assessments of the execution of the terms of the agreement. If execution of the agreement does not result in an acceptable product or service, the acquirer or supplier can terminate the agreement as allowed in its terms.

e) **Deliver and support the product or service.** This activity consists of the following tasks:

- 1) Deliver the product or service in accordance with the agreement criteria.

NOTE As stated in the agreement, acceptance criteria, such as acceptance testing, relate to how the product or service will satisfy its intended use in its operational environment. Unaddressed exceptions, e.g., disputes over conduct of acceptance testing or product suitability for intended use, are a matter for agreement provisions relating to disputes, arbitration or applicable law and regulation.

- 2) Provide assistance to the acquirer in support of the delivered product or service, per the agreement.
- 3) Accept and acknowledge payment or other agreed consideration.
- 4) Transfer the product or service to the acquirer, or other party, as directed by the agreement.
- 5) Close the agreement.

NOTE 1 The project is closed by the Portfolio Management process.

NOTE 2 Agreements can specify the conditions under which the agreement will be terminated by either party, e.g., unexpected changes in strategy or available funding, or lack of satisfactory progress.

6.2 Organizational Project-Enabling processes

The Organizational Project-Enabling processes help ensure the organization's capability to acquire and supply products or services through the initiation, support and control of projects. These processes provide resources and infrastructure necessary to support projects and help ensure the satisfaction of organizational objectives and

established agreements. They are not intended to be a comprehensive set of business processes that enable strategic management of the organization's business.

The Organizational Project-Enabling Processes consist of the following:

- a) Life Cycle Model Management process;
- b) Infrastructure Management process;
- c) Portfolio Management process;
- d) Human Resource Management process;
- e) Quality Management process; and
- f) Knowledge Management process.

6.2.1 Life cycle model management process

6.2.1.1 Purpose

The purpose of the Life Cycle Model Management process is to define, maintain, and assure availability of policies, life cycle processes, life cycle models, and procedures for use by the organization with respect to the scope of this document.

This process provides life cycle policies, processes, models, and procedures that are consistent with the organization's objectives, that are defined, adapted, improved and maintained to support individual project needs within the context of the organization, and that are capable of being applied using effective, proven methods and tools.

6.2.1.2 Outcomes

As a result of the successful implementation of the Life Cycle Model Management process:

- a) Organizational policies and procedures for the management and deployment of life cycle models and processes are established.
- b) Responsibility, accountability, and authority within life cycle policies, processes, models, and procedures are defined.
- c) Life cycle models and processes for use by the organization are assessed.
- d) Prioritized process, model, and procedure improvements are implemented.

6.2.1.3 Activities and tasks

The organization shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Life Cycle Model Management process.

- a) **Establish the process.** This activity consists of the following tasks:

NOTE The detail of the life cycle implementation within a project is dependent upon the complexity of the work, the methods used, and the skills and training of personnel involved in performing the work. A project tailors policies, processes, models, and procedures according to its requirements and needs, while maintaining alignment with regulations and organizational policies. Annex A contains information on tailoring.

- 1) Establish policies and procedures for process management and deployment that are consistent with organizational strategies.

- 2) Establish the processes that implement the requirements of this document and that are consistent with organizational strategies.
- 3) Define the roles, responsibilities, accountabilities, and authorities to facilitate implementation of processes and the strategic management of life cycles.
- 4) Define business criteria that control progression through the life cycle.

NOTE The decision-making criteria regarding entering and exiting each life cycle stage and key milestones are established. These are sometimes expressed in terms of business achievement.

- 5) Establish standard life cycle models for the organization that are comprised of stages and define the purpose and outcomes for each stage.

NOTE The life cycle model comprises one or more stage models, as needed. It is assembled as a sequence of stages that can overlap or iterate, as appropriate for the system-of-interest's scope, magnitude, complexity, changing needs and opportunities. Stages are illustrated in ISO/IEC TS 24748-1 using a commonly encountered example of life cycle stages. Specific examples for systems and software are provided in ISO/IEC TR 24748-2 and ISO/IEC TR 24748-3. The life cycle processes and activities are selected, tailored as appropriate and employed in a stage to fulfill the purpose and outcomes of that stage.

b) Assess the process. This activity consists of the following tasks:

NOTE ISO/IEC 33002:2015 provides a more detailed set of process assessment activities and tasks that can be aligned with the tasks shown below.

- 1) Monitor process execution across the organization.

NOTE This includes monitoring performance, analyzing process measures, and reviewing trends with respect to compliance with regulations, organizational policies, business criteria, and feedback from the projects regarding the effectiveness and efficiency of the processes.

- 2) Conduct periodic reviews of the life cycle models used by the projects.

NOTE This includes confirming the continuing suitability, adequacy and effectiveness of the life cycle models used by the projects and making improvements as appropriate. This includes the stages, processes and achievement criteria that control progression through the life cycle.

- 3) Identify improvement opportunities from assessment results.

NOTE Improvements can affect the stages, processes, and achievement criteria that control progression through the life cycle.

c) Improve the process. This activity consists of the following tasks:

- 1) Prioritize and plan improvement opportunities.
- 2) Implement improvement opportunities and inform relevant stakeholders.

NOTE Process Improvement includes improvements to any of the processes in the organization. Lessons learned are captured and available.

6.2.2 Infrastructure Management process

6.2.2.1 Purpose

The purpose of the Infrastructure Management process is to provide the infrastructure and services to projects to support organization and project objectives throughout the life cycle.

This process defines, provides and maintains the facilities, tools, and communications and information technology assets needed for the organization's business with respect to the scope of this document.

6.2.2.2 Outcomes

As a result of the successful implementation of the Infrastructure Management process:

- a) The requirements for infrastructure are defined.
- b) The infrastructure elements are identified and specified.
- c) Infrastructure elements are developed or acquired.
- d) The infrastructure is available.

6.2.2.3 Activities and tasks

The organization shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Infrastructure Management process.

a) **Establish the infrastructure.** This activity consists of the following tasks:

- 1) Define project infrastructure requirements.

NOTE 1 Infrastructure element examples include facilities, tools, hardware, software, services, and standards. In addition to the general infrastructure resources common to an organization to support its business processes, an organization can also provide projects with unique or shared enabling systems to support the project's technical processes.

NOTE 2 The infrastructure resource needs for the project are considered in context with other projects and resources within the organization, as well as within the policies and strategic plans of the organization. Business constraints and timelines that influence and control provision of infrastructure resources and services for the project are also evaluated. Project plans and future business needs contribute to the understanding of the resource infrastructure that is required. Physical factors (e.g., facilities), logistics needs, and human factors (including health and safety aspects) are also considered.

NOTE 3 ISO/IEC 27036, *Information security for supplier relationships*, provides guidance for addressing security of outsourced infrastructure.

- 2) Identify, obtain and provide infrastructure resources and services that are needed to implement and support projects.

NOTE An inventory asset registry is often established to track infrastructure elements and support reuse of infrastructure assets.

b) **Maintain the infrastructure.** This activity consists of the following tasks:

- 1) Evaluate the degree to which delivered infrastructure resources satisfy project needs.
- 2) Identify and provide improvements or changes to the infrastructure resources as the project requirements change.

6.2.3 Portfolio Management process

6.2.3.1 Purpose

The purpose of the Portfolio Management process is to initiate and sustain necessary, sufficient and suitable projects in order to meet the strategic objectives of the organization.

This process commits the investment of adequate organization funding and resources, and sanctions the authorities needed to establish selected projects. It performs continued assessment of projects to confirm they justify, or can be redirected to justify, continued investment.

For software systems, portfolio management also commonly refers to the management of a product line (portfolio of assets, products, and enabling systems, or service catalogue) to meet organizational or customer needs and objectives and support changes in technology. Management of assets is achieved through management of projects.

6.2.3.2 Outcomes

As a result of the successful implementation of the Portfolio Management process:

- a) Business venture opportunities, investments or necessities are qualified and prioritized.
- b) Projects are identified.
- c) Resources and budgets for each project are allocated.
- d) Project management responsibilities, accountability, and authorities are defined.
- e) Projects meeting agreement and stakeholder requirements are sustained.
- f) Projects not meeting agreement or satisfying stakeholder requirements are redirected or terminated.
- g) Projects that have completed agreements and satisfied stakeholder requirements are closed.

6.2.3.3 Activities and tasks

The organization shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Portfolio Management process.

a) **Define and authorize projects.** This activity consists of the following tasks:

- 1) Identify potential new or modified capabilities or missions.

NOTE The organization business strategy, concept of operations, or gap analysis or opportunity analysis is reviewed for current gaps, problems, or opportunities. A new capability or enterprise need is usually determined in the Business or Mission Analysis process, further defined in the Stakeholder Needs and Requirements Definition process, and managed through this process.

- 2) Prioritize, select and establish new business opportunities, ventures or undertakings.

NOTE These are usually consistent with the business strategy and action plans of the organization. The potential projects are prioritized, and thresholds established, to determine which projects will be executed. The characteristics of identified projects are often determined, including stakeholder value, risks and barriers to success, dependencies and inter-relationships, constraints, resource needs and mutual contention for resources. Each potential project is then assessed with respect to likelihood of success and cost-benefit. The Decision Management and System Analysis processes provide details on performing an analysis of alternatives.

- 3) Define projects, accountabilities and authorities.
- 4) Identify the expected goals, objectives, and outcomes of each project.
- 5) Identify and allocate resources for the achievement of project goals and objectives.
- 6) Identify multi-project interfaces and dependencies to be managed or supported by each project.

NOTE 1 This includes the use or reuse of enabling systems used by more than one project and the use or reuse of common system elements, including software elements, by more than one project.

NOTE 2 Understanding each project in the context of the enterprise architecture helps to ensure interfaces and constraints are identified.

- 7) Specify the project reporting requirements and review milestones that govern the execution of each project.

- 8) Authorize each project to commence execution of project plans.

NOTE Refer to the Project Planning process for additional information on developing project plans. Project plans are most useful when developed and approved early in the project life cycle.

b) **Evaluate the portfolio of projects.** This activity consists of the following tasks:

- 1) Evaluate projects to confirm ongoing viability.

NOTE Viability includes the following criteria:

- i) The project is making progress towards achieving established goals and objectives.
 - ii) The project is complying with project directives.
 - iii) The project is being conducted according to approved project life cycle policies, processes, and procedures.
 - iv) The project remains viable, as indicated by, for example, continuing need for the service, practicable product implementation, and acceptable investment benefits.
- 2) Act to continue or redirect projects that are satisfactorily progressing or can be expected to progress satisfactorily by appropriate redirection.

c) **Terminate projects.** This activity consists of the following tasks:

- 1) Where agreements permit, act to cancel or suspend projects whose disadvantages or risks to the organization outweigh the benefits of continued investments.

NOTE Capture of lessons learned from canceled or failing projects can be especially useful for organizational improvement or use on other projects.

- 2) After completion of the agreement for products and services, act to close the projects.

NOTE Closure is accomplished in accordance with organizational policies and procedures, and the agreement.

6.2.4 Human Resource Management process

6.2.4.1 Purpose

The purpose of the Human Resource Management process is to provide the organization with necessary human resources and to maintain their competencies, consistent with business needs.

This process provides a supply of skilled and experienced personnel qualified to perform life cycle processes to achieve organization, project, and stakeholder objectives.

6.2.4.2 Outcomes

As a result of the successful implementation of the Human Resource Management process:

- a) Skills required by projects are identified.
- b) Necessary human resources are provided to projects.
- c) Skills of personnel are developed, maintained or enhanced.
- d) Conflicts in multi-project resource demands are resolved.

6.2.4.3 Activities and tasks

The organization shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Human Resource Management process:

a) **Identify skills.** This activity consists of the following tasks:

- 1) Identify skill needs based on current and expected projects.
- 2) Identify and record skills of personnel.

b) **Develop skills.** This activity consists of the following tasks:

- 1) Establish skills development strategy.

NOTE This plan includes types and levels of training, categories of personnel, schedules, personnel resource requirements, and training needs.

- 2) Obtain or develop training, education or mentoring resources.

NOTE These resources include training materials that are developed by the organization or external parties, training courses that are available from external suppliers, computer based instruction.

- 3) Provide planned skill development.
- 4) Maintain records of skill development.

c) **Acquire and provide skills.** This activity consists of the following tasks:

NOTE This includes the recruitment and retention of personnel with experience levels and skills necessary to properly staff projects; staff assessment and review, e.g., their proficiency, motivation, ability to work in a team environment, as well as the need to be retrained, reassigned or reallocated.

- 1) Obtain qualified personnel when skill deficits are identified.

NOTE This includes using outsourced resources.

- 2) Maintain and manage the pool of skilled personnel necessary to staff ongoing projects.
- 3) Make project assignments based on project and staff-development needs.
- 4) Motivate personnel, e.g., through career development and reward mechanisms.
- 5) Control multi-project management interfaces to resolve personnel conflicts.

NOTE This includes conflicts of capacity in organizational infrastructure and supporting services and personnel resources among ongoing projects; or from project personnel being over-committed.

6.2.5 Quality Management process

6.2.5.1 Purpose

The purpose of the Quality Management process is to assure that products, services and implementations of the quality management process meet organizational and project quality objectives and achieve customer satisfaction.

6.2.5.2 Outcomes

As a result of the successful implementation of the Quality Management process:

- a) Organizational quality management policies, objectives, and procedures are defined and implemented.

- b) Quality evaluation criteria and methods are established.
- c) Resources and information are provided to projects to support the operation and monitoring of project quality assurance activities.
- d) Quality assurance evaluation results are gathered and analyzed.
- e) Quality management policies and procedures are improved based upon project and organizational results.

6.2.5.3 Activities and tasks

The organization shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Quality Management process.

NOTE Refer to ISO 9001:2015 for information and requirements to establish a Quality Management System.

a) **Plan quality management.** This activity consists of the following tasks:

- 1) Establish quality management policies, objectives, and procedures.

NOTE 1 ISO 9004:2009 contains guidelines for performance improvements.

NOTE 2 The policies, objectives, and procedures are based on the business strategy for customer satisfaction and risk management considerations.

- 2) Define responsibilities and authority for implementation of quality management.

NOTE Resources for quality management are often assigned from distinct organizations for independence from project management.

- 3) Define quality evaluation criteria and methods.
- 4) Provide resources and information for quality management.

b) **Evaluate quality management.** This activity consists of the following tasks:

- 1) Gather and analyze quality assurance evaluation results, in accordance with the defined criteria.
- 2) Assess customer satisfaction.

NOTE ISO 10004:2012 contains guidelines for monitoring and measuring customer satisfaction. The quality of the software system is also demonstrated by user satisfaction.

- 3) Conduct periodic reviews of project Quality Assurance activities for compliance with the Quality Management policies, objectives, and procedures.

NOTE Quality evaluation criteria and methods are established. Quality assessments address compliance with the project procedures and product conformance with quality characteristics.

- 4) Monitor the status of quality improvements on processes, products, and services.

c) **Perform corrective and preventive action.** This activity consists of the following tasks:

- 1) Plan corrective actions when quality management objectives are not achieved.
- 2) Plan preventive actions when there is a sufficient risk that quality management objectives will not be achieved.
- 3) Monitor corrective and preventive actions to completion and inform relevant stakeholders.

NOTE 1 Implementation of corrective and preventive action is performed in other relevant processes, such as Life Cycle Model Management or Project Assessment and Control.

NOTE 2 ISO 9001:2015, 0.3.3 and Annex A.4 describe preventive action to eliminate potential nonconformities as part of risk-based thinking.

6.2.6 Knowledge Management process

6.2.6.1 Purpose

The purpose of the Knowledge Management process is to create the capability and assets that enable the organization to exploit opportunities to re-apply existing knowledge.

This encompasses knowledge, skills, and knowledge assets, including system elements.

NOTE The re-application of existing knowledge is known as knowledge reuse and includes the reuse of knowledge about or from software elements.

6.2.6.2 Outcomes

As a result of the successful implementation of the Knowledge Management process:

- a) A taxonomy for the application of knowledge assets is identified.
- b) The organizational knowledge, skills, and knowledge assets are developed or acquired.
- c) The organizational knowledge, skills, and knowledge assets are available.
- d) Knowledge management usage data is gathered and analyzed.

6.2.6.3 Activities and tasks

The organization shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Knowledge Management process:

a) **Plan knowledge management.** This activity consists of the following tasks:

- 1) Define the knowledge management strategy.

NOTE 1 This knowledge management strategy generally includes:

- i) Identifying domains and their potential for the reapplication of knowledge.
- ii) Plans for obtaining and maintaining knowledge, skills, and knowledge assets for their useful life.
- iii) Characterization of the types of knowledge, skills, and knowledge assets to be collected and maintained.
- iv) Criteria for accepting, qualifying, and retiring knowledge, skills, and knowledge assets.
- v) Procedures for controlling changes to the knowledge, skills, and knowledge assets.
- vi) Plans, mechanisms, and procedures for protection, control, and access to classified or sensitive data and information.
- vii) Mechanisms for storage and retrieval.

NOTE 2 Knowledge management includes knowledge shared both internally within the organization and knowledge that is shared outside the organization with designated stakeholders, acquirers, and business partners, subject to intellectual property and non-disclosure agreements.

- 2) Identify the knowledge, skills, and knowledge assets to be managed.
- 3) Identify projects that can benefit from the application of the knowledge, skills, and knowledge assets.

b) **Share knowledge and skills throughout the organization.** This activity consists of the following tasks:

- 1) Establish and maintain a classification for capturing and sharing knowledge and skills across the organization.

NOTE This classification can include expert, common, and domain knowledge and skills, as well as lessons learned from other tasks.

- 2) Capture or acquire knowledge and skills.
- 3) Share knowledge and skills across the organization.

c) **Share knowledge assets throughout the organization.** This activity consists of the following tasks:

- 1) Establish a taxonomy to organize knowledge assets.

NOTE 1 The taxonomy includes the following:

- i) Definition of the boundaries of domains and their relationships to others.
- ii) Domain models capturing essential common and different features, capabilities, concepts, and functions.
- iii) An architecture for a family of systems within the domain, including their common and different features.

NOTE 2 See ISO/IEC 26550 for more information on product line models. Refer to ISO/IEC/IEEE 42010:2011 for requirements on architecture frameworks, viewpoints, model kinds, views and models.

- 2) Develop or acquire knowledge assets.

NOTE Knowledge assets include system elements or their representations (e.g., reusable code libraries, reference architectures) architecture or design elements (e.g., architecture or design patterns), processes, criteria, or other technical information (e.g., training materials) related to domain knowledge, and lessons learned.

- 3) Share knowledge assets across the organization.

NOTE Automated search capabilities improve access to knowledge assets.

d) **Manage knowledge, skills, and knowledge assets.** This activity consists of the following tasks:

- 1) Maintain knowledge, skills, and knowledge assets.
- 2) Monitor and record the reuse of knowledge, skills, and knowledge assets.
- 3) Periodically reassess the currency of technology and market needs for the knowledge assets.

NOTE Assess the business benefits which the organization gained through the use of knowledge management practices.

6.3 Technical Management processes

The Technical Management processes are used to establish and evolve plans, to execute the plans, to assess actual achievement and progress against the plans, and to control execution through to fulfillment. Individual Technical Management processes may be invoked at any time in the life cycle and at any level in a hierarchy of projects, as required by plans or unforeseen events. The Technical Management processes are applied with a level of rigor and formality that depends on the risk and complexity of the project.

The scope of a technical management process is the technical management of a project or its products, to include the software product or system-of-interest.

NOTE This set of Technical Management processes is performed so that software system-specific technical processes can be conducted effectively. They do not comprise a management system or a comprehensive set of processes for project management, as that is not the scope of this document.

The Technical Management processes consist of the following:

- a) Project Planning process;
- b) Project Assessment and Control process;
- c) Decision Management process;
- d) Risk Management process;
- e) Configuration Management process;
- f) Information Management process;
- g) Measurement process; and
- h) Quality Assurance process.

Project Planning and Project Assessment and Control processes are key to all management practices. These processes establish the general approach for managing a project or a process. The other processes in this group provide a specific focused set of tasks for achieving a specialized management objective. They are all evident in the management of any undertaking, ranging from a complete organization down to a single life cycle process and its tasks. In this document, the project has been chosen as the context for describing processes. The same processes can also be applied in the performance of services.

6.3.1 Project Planning process

6.3.1.1 Purpose

The purpose of the Project Planning process is to produce and coordinate effective and workable plans.

This process determines the scope of the project management and technical activities, identifies process outputs, tasks and deliverables, establishes schedules for task conduct, including achievement criteria, and required resources to accomplish tasks. This is an ongoing process that continues throughout a project, with regular revisions to plans.

NOTE The strategies defined in each of the other processes provide inputs and are integrated in the Project Planning process. The Project Assessment and Control process is used to assess whether the plans are integrated, aligned, and feasible.

6.3.1.2 Outcomes

As a result of the successful implementation of the Project Planning process:

- a) Objectives and plans are defined.
- b) Roles, responsibilities, accountabilities, and authorities are defined.
- c) Resources and services necessary to achieve the objectives are formally requested and committed.
- d) Plans for the execution of the project are activated.

6.3.1.3 Activities and tasks

The project shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Project Planning process.

a) **Define the project.** This activity consists of the following tasks:

1) Identify the project objectives and constraints.

NOTE 1 Objectives and constraints include performance and other quality aspects, cost, time and customer and user satisfaction. Each objective is identified with a level of detail that permits selection, tailoring and implementation of the appropriate processes and activities.

NOTE 2 ISO/IEC 15026 *Systems and software assurance*, ISO/IEC 27001 *Information Security Management System* and ISO/IEC 27036, *Information Security for Supplier Relationships*, provide additional guidance on objectives and constraints related to assurance and security.

2) Define the project scope as established in the agreement.

NOTE This includes the relevant activities required to satisfy business decision criteria and complete the project successfully. A project can have responsibility for one or more stages in the complete software system life cycle. Project Planning includes defining appropriate actions for maintaining project plans, performing assessments and controlling the project.

3) Define and maintain a life cycle model that is comprised of stages using the defined life cycle models of the organization.

NOTE ISO/IEC TS 24748-1 provides detailed information regarding life cycle stages and the definition of an appropriate life cycle model. It defines a general set of exemplar system life cycle stages, including Concept, Development, Production, Utilization, Support and Retirement. It also identifies a generic exemplar set of software life cycle stages, including Needs determination, Concept exploration and definition, Demonstration and evaluation, Engineering/development, Production/manufacturing, Deployment/sales, Operations, Maintenance and support, and Retirement.

4) Establish a work breakdown structure (WBS) based on the deliverable products or the evolving architecture of the software system.

NOTE 1 Each element of the software system architecture, and appropriate processes and activities, are described with a level of detail that is consistent with identified risks. Related tasks in the work breakdown structure are grouped for performance. Project tasks identify work items being developed or produced. The Practice Management Standard for Work Breakdown Structures of the Project Management Institute (PMI) contains additional details on WBSs.

NOTE 2 For projects with agile or iterative methods, a WBS element can correspond to the primary features, from a user perspective, to be produced during iterations.

5) Define and maintain the processes that will be applied on the project.

NOTE 1 These processes are based on the defined processes of the organization (see Life Cycle Model Management process). Annex A contains information on tailoring that can be used to address project-specific needs. The definition of the processes includes the entry and exit criteria, inputs, process sequence constraints (predecessor/successor relationships), process concurrency requirements (what processes and tasks are to be worked concurrently with other process area tasks or activities), Measures of Effectiveness/Measures of Performance attributes, and scope and cost parameters (for critically important cost estimation).

NOTE 2 Identifying interfaces with other projects or organizational units is addressed through the Portfolio Management process.

b) **Plan project and technical management.** This activity consists of the following tasks:

1) Define and maintain a project schedule based on management and technical objectives and work estimates.

NOTE This includes definition of the duration, relationship, dependencies and sequence of activities, achievement milestones, resources employed and the reviews and schedule reserves for risk management necessary to achieve timely completion of the project.