International Standard

# ISO/IEC/IEEE 24748-1

Second edition
2024-03

# Systems and software engineering — Life cycle management —

## Part 1: Guidelines for life cycle management

*Ingénierie des systèmes et du logiciel — Gestion du cycle de vie —*

*Partie 1: Lignes directrices pour la gestion du cycle de vie*



Reference number
ISO/IEC/IEEE 24748-1:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

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

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

# Contents

Page

# 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.

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 or www.iec.ch/members_experts/refdocs).

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.

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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

For an explanation of the voluntary nature of standards, 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 www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

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

This second edition cancels and replaces the first edition (ISO/IEC/IEEE 24748-1:2018), which has been technically revised.

The main changes are as follows:

— added system of systems topics based on ISO/IEC/IEEE 21839, ISO/IEC/IEEE 21840 and ISO/IEC/IEEE 21841;

— added references for interfacing and interoperating systems and general updates from ISO/IEC/IEEE 15288:2023;

— added more recent life cycle models such as DEVOPS.

A list of all parts in the ISO/IEC/IEEE 24748 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

# Introduction

The purpose of this document is to facilitate the use of the process content of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207, by providing unified and consolidated guidance on life cycle management of systems and software. This is to help ensure consistency in system concepts and life cycle concepts, models, stages, processes, process application, key points of view, adaptation and use in various domains as the two International Standards are used in combination. That in turn helps a project team design a life cycle model for the system-of-interest to facilitate managing the progress of their project. Hence, ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 are the documents that apply the concepts found in this document to specific processes.

NOTE ISO/IEC/IEEE 16326 and ISO/IEC/IEEE 24641 also apply the concepts found in this document, in the process context for project management and model-based approaches respectively.

This document also aids in identifying and planning use of life cycle processes described in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 that enable the project to be completed successfully, meeting its objectives/requirements for each stage and for the overall project.

There is also increasing recognition of the importance of helping to ensure that all life cycle stages and all aspects within each stage are supported with thorough guidance to enable alignment with any process documents that can be created later that focus on areas besides systems and software, including hardware, humans, data, processes (e.g. review process), procedures (e.g. operator instructions), facilities and naturally occurring entities (e.g. water, organisms, minerals).

By addressing these needs specifically in this document, the users of the process-focused ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 benefit not only from having one complementary document that addresses the management of life cycles of systems that provide products or services, but also from a framework that links life cycle management aspects to more than just the systems or software aspects of products or services. Additional discussion for system of systems can be found in ISO/IEC/IEEE 21839, ISO/IEC/IEEE 21840 and ISO/IEC/IEEE 21841.

In the context of this document, ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207, there is a continuum of human-made systems from those that use little to no software to those in which software is the primary interest. When software is the predominant system or element of interest, ISO/IEC/IEEE 12207 should be used. Both documents have the same process model, share most activities and tasks and differ primarily in descriptive notes. The determination of the applicability of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 should be decided by the nature of the system and its enabling systems. Often, a mixed tailoring of each standard can be appropriate.

ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 also have published guidance documents (ISO/IEC/IEEE 24748-2 and ISO/IEC/IEEE 24748-3), respectively, to support use of the two International Standards individually.

# Systems and software engineering — Life cycle management —

## Part 1: Guidelines for life cycle management

## 1 Scope

This document provides guidance for the life cycle management of systems and software, complementing the processes described in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207. This document:

— addresses systems concepts and life cycle concepts, models, stages, processes, process application, key points of view, adaptation and use in various domains and by various disciplines;

— establishes a common framework for describing life cycles, including their individual stages, for the management of projects that provide or acquire either products or services;

— defines the concept of a life cycle;

— supports the use of the life cycle processes within an organization or a project; organizations and projects can use these life cycle concepts when acquiring and supplying either products or services;

— provides guidance on adapting a life cycle model and the content associated with a life cycle or a part of a life cycle;

— describes the relationship between life cycles and their use in applying the processes in ISO/IEC/IEEE 15288 (systems aspects) and ISO/IEC/IEEE 12207 (software systems aspects);

— shows the relationships of life cycle concepts to the hardware, human, services, process, procedure, facility and naturally occurring entity aspects of projects;

— describes how its concepts relate to detailed process standards, for example, in the areas of measurement, project management, risk management and model-based systems and software engineering.

## 2 Normative references

There are no normative references in this document.

## 3 Terms and definitions

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

ISO, IEC and IEEE maintain terminology databases for use in standardization at the following addresses:

— IEC Electropedia: available at https://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEEE Standards Dictionary Online: available at https://dictionary.ieee.org/

NOTE Definitions for other system and software engineering terms can be found in ISO/IEC/IEEE 24765, available at www.computer.org/sevocab.

**3.1**
**acquirer**
*stakeholder* (3.49) that acquires or procures a *system* (3.51), *product* (3.33) or *service* (3.45) from a *supplier* (3.50)

Note 1 to entry: Other terms commonly used for an acquirer are buyer, *customer* (3.13), owner, purchaser, or internal/organizational sponsor.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.1]

**3.2**
**acquisition**
*process* (3.30) of obtaining a *system* (3.51), *product* (3.33) or *service* (3.45)

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.2]

**3.3**
**activity**
set of cohesive *tasks* (3.56) of a *process* (3.30)

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.3]

**3.4**
**agile**
development approach based on iterative development, frequent inspection and adaptation, and incremental deliveries in which *requirements* (3.39) and solutions evolve through collaboration in cross-functional teams and through continual *stakeholder* (3.49) feedback

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.1, modified — The defined term has been changed from "agile development" to "agile"; note 1 to entry has been removed.]

**3.5**
**agreement**
mutual acknowledgement of terms and conditions under which a working relationship is conducted

EXAMPLE Contract, memorandum of agreement.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.4]

**3.6**
**architecture**
fundamental concepts or properties of a *system* (3.51) in its *environment* (3.18) and governing principles for the realization and evolution of this system and its related *life cycle* (3.24) *processes* (3.30)

[SOURCE: ISO/IEC/IEEE 42020:2019, 3.3, modified — "entity" has been replaced with "system"; notes to entry have been removed.]

**3.7**
**artefact**
work *product* (3.33) that is produced and used during a project to capture and convey information

[SOURCE: ISO 19014-4:2020, 3.9, modified — The definition has been made singular.]

**3.8**
**audit**
independent examination of a work *product* (3.33) or set of work products to assess compliance with specifications, standards, contractual *agreements* (3.5), or other criteria

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.7]

**3.9**
**baseline**
formally approved version of a *configuration item* (3.12), regardless of media, formally designated and fixed at a specific time during the configuration item's *life cycle* (3.24)

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.8]

**3.10**
**concept of operations**
verbal and graphic statement, in broad outline, of an *organization's* (3.28) assumptions or intent in regard to an operation or series of operations of new, modified, or existing organizational *systems* (3.51)

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 to achieve an organizational performance objective. See also *operational concept* (3.26).

Note 2 to entry: The concept of operations provides the basis for bounding the operating space, system capabilities, *interfaces* (3.22) and operating *environment* (3.18).

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.9]

**3.11**
**concern**
matter of interest or importance to a *stakeholder* (3.49)

Note 1 to entry: A concern pertains to any influence on a *system* (3.51) in its *environment* (3.18), including developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ethical, ecological and social influences.

[SOURCE: ISO/IEC/IEEE 42020:2019, 3.8, modified — EXAMPLE has been removed; note 1 to entry has been added.]

**3.12**
**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* (3.30)

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.11]

**3.13**
**customer**
*organization* (3.28) or person that receives a *product* (3.33) or *service* (3.45)

EXAMPLE Consumer, client, *user* (3.60), *acquirer* (3.1), buyer, or purchaser.

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

[SOURCE: ISO /IEC/IEEE 15288:2023, 3.12]

**3.14**
**design**, noun
specification of *system elements* (3.52) and their relationships, that is sufficiently complete to support a compliant implementation of the *architecture* (3.6)

Note 1 to entry: Design provides the detailed implementation-level physical structure, behaviour, temporal relationships and other attributes of system elements.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.13]

**3.15**
**design characteristics**
design attributes or distinguishing features that pertain to a measurable description of a *product* (3.33) or *service* (3.45)

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.14]

**3.16**
**DevOps**
**development and operations**
set of principles and practices which enable better communication and collaboration between relevant *stakeholders* (3.49) for the purpose of specifying, developing, and operating software and *systems* (3.51), *products* (3.33) and *services* (3.45), and continuous improvements in all aspects of the *life cycle* (3.24)

Note 1 to entry: Extensions include DevSecOps which addresses *concerns* (3.11) related to *security* (3.44) throughout development and operations.

[SOURCE: ISO/IEC/IEEE 32675:2022, 3.1, modified —Note 1 to entry has been added.]

**3.17**
**enabling system**
*system* (3.51) that supports a *system-of-interest* (3.53) during its *life cycle* (3.24) *stages* (3.48) but does not necessarily contribute directly to its function during operation

EXAMPLE Production-enabling system, which is required when a system-of-interest enters the production stage.

Note 1 to entry: Each enabling system has a life cycle of its own.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.15, modified — The second sentence of note 1 to entry has been removed.]

**3.18**
**environment**
<system> context determining the setting and circumstances of all influences upon a *system* (3.51)

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.16]

**3.19**
**incident**
anomalous or unexpected event, set of events, condition, or situation at any time during the *life cycle* (3.24) of a *project* (3.34), *product* (3.33), *service* (3.45), or *system* (3.51)

Note 1 to entry: An incident is elevated and treated as a *problem* (3.29) when the cause of the incident needs to be analysed and corrected to prevent reoccurrence to avoid or minimise loss of life, or damage of property or natural resources.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.17]

**3.20**
**information item**
separately identifiable body of information that is produced, stored, and delivered for human use

[SOURCE: ISO/IEC/IEEE 15289:2019, 3.1.12, modified — The preferred term “information product” has been removed; notes to entry have been removed.]

**3.21**
**iteration**
<process> repeating the application of the same *process* (3.30) or set of processes on the same level of the *system* (3.51) structure

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.28]

**3.22**
**interface**
point at which two or more logical, physical, or both, *system elements* (3.52) or software system elements meet and act on or communicate with each other

[SOURCE: ISO/IEC/IEEE 24748-6:2023, 3.1.3]

**3.23**
**interoperating system**
*system* (3.51) that exchanges information with the *system-of-interest* (3.53) and uses the information that has been exchanged

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.20]

**3.24**
**life cycle**
evolution of a *system* (3.51), *product* (3.33), *service* (3.45), *project* (3.34) or other human-made entity from conception through *retirement* (3.41)

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.21]

**3.25**
**life cycle model**
framework of *processes* (3.30) and *activities* (3.3) concerned with the *life cycle* (3.24) which can be organized into *stages* (3.48), acting as a common reference for communication and understanding

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.22]

**3.26**
**operational concept**
verbal and graphic statement of an *organization's* (3.28) assumptions or intent in regard to an operation or series of operations of a specific *system* (3.51) or a related set of new, existing or modified 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* (3.18) from the *users'* (3.60) and *operators'* (3.27) perspective. See also *concept of operations* (3.10).

Note 2 to entry: The operational concept is about systems, while a concept of operations typically refers to organizations.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.23]

**3.27**
**operator**
individual or *organization* (3.28) that performs the operations of a *system* (3.51)

Note 1 to entry: The role of operator and the role of *user* (3.60) may 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 may perform operations on a system that is operated, or of a system that is operated, depending on whether or not operating instructions are placed within the system boundary.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.24]

**3.28**
**organization**
person or group of people that has its own functions with responsibilities, authorities and relationships to achieve its objectives

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

[SOURCE: ISO 9000:2015, 3.2.1, modified — Notes to entry have been removed; EXAMPLE has been added.]

**3.29**
**problem**
difficulty, uncertainty, or otherwise realized and undesirable event, set of events, condition, or situation that requires investigation and corrective action

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.26]

**3.30**
**process**
set of interrelated or interacting *activities* (3.3) that transforms inputs into outputs

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.27]

**3.31**
**process outcome**
observable result of the successful achievement of the *process purpose* (3.32)

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.30]

**3.32**
**process purpose**
high-level objective of performing the *process* (3.30) 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* (3.49).

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.29, modified — The term "high-level" has been hyphenated.]

**3.33**
**product**
output of an *organization* (3.28) that can be produced without any transaction taking place between the organization and the *customer* (3.13)

Note 1 to entry: The dominant element of a product is that it is generally tangible.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.32]

**3.34**
**project**
endeavour with defined start and finish criteria undertaken to create a *product* (3.33) or *service* (3.45) in accordance with specified *resources* (3.40) and *requirements* (3.39)

Note 1 to entry: A project is sometimes viewed as a unique *process* (3.30) comprising co-coordinated and controlled *activities* (3.3) and composed of activities from the *technical management* (3.58) and technical processes defined in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288.

Note 2 to entry: Continuous development approaches such as *agile* (3.4) and *DevOps* (3.16) can use different terminology for the creation of product and services.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.33, modified — Note 1 to entry has been updated to reference ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288.]

**3.35**
**qualification**
*process* (3.30) of demonstrating whether an entity is capable of fulfilling specified *requirements* (3.39)

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.39]

**3.36**
**quality characteristic**
inherent characteristic of a *product* (3.33), *service* (3.45), *process* (3.30), or *system* (3.51) related to a *requirement* (3.39)

[SOURCE: ISO 9000:2015, 3.10.2, modified — "an object" has been replaced with "a product, service, process, or system"; notes to entry have been removed.]

**3.37**
**quality management**
coordinated activities to direct and control an *organization* (3.28) with regard to quality

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.42]

**3.38**
**recursion**
<process> repeating the application of the same *process* (3.30)

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.31]

**3.39**
**requirement**
statement that translates or expresses a need and its associated constraints and conditions

[SOURCE: ISO/IEC/IEEE 29148:2018, 3.1.19, modified — Notes to entry have been removed.]

**3.40**
**resource**
asset that is utilised or consumed during the execution of a *process* (3.30)

Note 1 to entry: Resource includes diverse entities, such as funding, personnel, facilities, capital equipment, tools and utilities, such as power, water, fuel and communication infrastructures.

Note 2 to entry: Resources include those that are reusable, renewable or consumable.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.37]

**3.41**
**retirement**
<system> withdrawal of active support by the operation and maintenance *organization* (3.28), partial or total replacement by a new *system* (3.51), installation of an upgraded system, or final decommissioning and disposal

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.38]

**3.42**
**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* (3.43), 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, 1.1, modified — The last sentence in note 1 to entry has been added.]

**3.43**
**safety**
expectation that a *system* (3.51) does not, under defined conditions, lead to a state in which human life, health, property, or the *environment* (3.18) is endangered

Note 1 to entry: The term is alternately defined as freedom from *risks* (3.42) that are not tolerable.

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.48, modified — Note 1 to entry has been added.]

**3.44**
**security**
protection against intentional subversion or forced failure, containing 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

Note 1 to entry: Security includes authenticity, accountability, confidentiality, integrity, availability, non-repudiation, and reliability, all of which have the related issue of their assurance.

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.49, modified — Note 1 to entry has been added.]

**3.45**
**service**
output of an *organization* (3.28) with at least one *activity* (3.3) necessarily performed between the organization and the *customer* (3.13)

Note 1 to entry: The dominant elements of a service are generally intangible.

Note 2 to entry: A service is coherent, discrete and can be composed of other services.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.42]

**3.46**
**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* (3.52) of ISO/IEC/IEEE 12207 and of ISO/IEC/IEEE 15288. Software items are typically *configuration items* (3.12).

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.53, modified — Note 1 to entry has been updated to reference ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288.]

**3.47**
**software product**
set of computer programs, procedures, and possibly associated documentation and data

Note 1 to entry: A software product is a software *system* (3.51) viewed as the output [*product* (3.33)] resulting from a *process* (3.30).

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.54]

**3.48**
**stage**
period within the *life cycle* (3.24) 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.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.43]

**3.49**
**stakeholder**
individual or *organization* (3.28) having a right, share, claim, or interest in a *system* (3.51) or in its possession of characteristics that meet their needs and expectations

EXAMPLE End *users* (3.60), end user organizations, supporters, developers, *customers* (3.13), producers, trainers, maintainers, disposers, *acquirers* (3.1), *suppliers* (3.50), regulatory bodies, and people influenced positively or negatively by a system.

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

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.44]

**3.50**
**supplier**
*organization* (3.28) or an individual that enters into an *agreement* (3.5) with the *acquirer* (3.1) for the supply of a *product* (3.33) or *service* (3.45)

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.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.45]

**3.51**
**system**
arrangement of parts or elements that together exhibit a stated behaviour or meaning that the individual constituents do not

Note 1 to entry: A system is sometimes considered as a *product* (3.33) or as the *services* (3.45) 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. Alternatively, the word “system” is substituted simply by a context-dependent synonym, e.g. aircraft, though this potentially obscures a system principles perspective.

Note 3 to entry: A complete system includes all of the associated equipment, facilities, material, computer programs, firmware, technical documentation, services and personnel required for operations and support to the degree necessary for self-sufficient use in its intended *environment* (3.18).

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.46]

**3.52**
**system element**
discrete part of a *system* (3.51) that can be implemented to fulfil specified *requirements* (3.39)

EXAMPLE Hardware, software, data, humans, *processes* (3.30) [e.g. processes for providing *service* (3.45) to *users* (3.60)], procedures [e.g. *operator* (3.27) instructions], facilities, materials and naturally occurring entities, or any combination.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.47]

**3.53**
**system-of-interest**
**SoI**
*system* (3.51) whose *life cycle* (3.24) is under consideration

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.48]

**3.54**
**system of systems**
**SoS**
set of *systems* (3.51) or *system elements* (3.52) that interact to provide a unique capability that none of the constituent systems can accomplish on its own

Note 1 to entry: System elements can be necessary to facilitate the interaction of the constituent systems in the system of systems.

[SOURCE: ISO/IEC/IEEE 21839:2019, 3.1.4]

**3.55**
**systems engineering**
transdisciplinary and integrative approach to enable the successful realization, use, and *retirement* (3.41) of engineered *systems* (3.51) using systems principles and concepts and scientific, technological and management methods

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.50]

**3.56**
**task**
required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a *process* (3.30)

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.51]

**3.57**
**technical debt**
deferred cost of work not performed at an earlier point in the product *life cycle* (3.24)

[SOURCE: *Software Extension to the PMBOK® Guide Fifth Edition*]

**3.58**
**technical management**
application of technical and administrative *resources* (3.40) to plan, organize and control engineering functions

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.67]

**3.59**
**traceability**
discernible association among two or more logical entities, such as *requirements* (3.39), *system elements* (3.52), *verifications* (3.62), or *tasks* (3.56)

[SOURCE: ISO/IEC TR 29110-1:2016, 3.71, modified — "discernible" has been added; EXAMPLE has been removed.]

**3.60**
**user**
individual or group that interacts with a *system* (3.51) or benefits from a system during its utilization

Note 1 to entry: The role of user and the role of *operator* (3.27) are sometimes vested, simultaneously or sequentially, in the same individual or *organization* (3.28).

[SOURCE: ISO/IEC 25010:2011, 4.3.16, modified — The original note 1 to entry has been replaced by new one.]

**3.61**
**validation**
confirmation, through the provision of objective evidence, that the *requirements* (3.39) for a specific intended use or application have been fulfilled

Note 1 to entry: In a *life cycle* (3.24) context, validation involves the set of *activities* (3.3) for gaining confidence that a *system* (3.51) is able to accomplish its intended use, goals and objectives in an *environment* (3.18) like the operational environment. The right system was built.

[SOURCE: ISO 9000:2015, 3.8.13, modified — Notes 1 to 3 to entry have been removed; a new note 1 to entry has been added.]

**3.62**
**verification**
confirmation, through the provision of objective evidence, that specified *requirements* (3.39) have been fulfilled

Note 1 to entry: Verification is a set of *activities* (3.3) that compares a *system* (3.51) or *system element* (3.52) against the required characteristics. This includes, but is not limited to, specified requirements, design description and the system itself. The system was built right.

[SOURCE: ISO 9000:2015, 3.8.12, modified — Notes 1 to 3 to entry have been removed; a new note 1 to entry has been added.]

**3.63**
**view**
representation of a *system* (3.51) from the perspective of a related set of *concerns* (3.11)

Note 1 to entry: A view can be an operational, functional or architectural representation of a system.

[SOURCE: ISO/IEC/IEEE 24774:2021, 3.21, modified — Removed the word "whole" before "system" from the definition; the original note 1 to entry has been replaced by a new one.]

**3.64**
**viewpoint**
specification of the conventions for constructing and using a *view* (3.63)

[SOURCE: ISO/IEC/IEEE 24774:2021, 3.22, modified — Notes 1 to 3 to entry have been removed.]

# 4 Life cycle-related concepts

## 4.1 General

This clause addresses system and life cycle concepts. For completeness, process, organizational and project concepts are covered in Annexes A, B and C, respectively to provide guidance when they are considered further.

## 4.2 System concepts

### 4.2.1 General

This subclause is included to highlight and explain essential concepts on which this document is based. These concepts are directly applicable to software systems, as addressed in ISO/IEC/IEEE 12207, and systems, as addressed in ISO/IEC/IEEE 15288. See Annex A for additional information on process concepts. Additional discussion for essential concepts related to system of systems (SoS) can be found in ISO/IEC/IEEE 21839, ISO/IEC/IEEE 21840 and ISO/IEC/IEEE 21841.

### 4.2.2 Systems

A system is an arrangement of parts or elements that together exhibit behaviour or meaning that the individual constituents do not. The systems considered in this document are made by humans and utilised to provide products or services in defined environments for the benefit of users and other stakeholders. These systems can be configured with one or more of the following: hardware, software, services, humans, data, processes (e.g. processes for providing services to users), procedures (e.g. operator instructions), facilities and naturally occurring entities (e.g. water, organisms, minerals).

The perception and definition of a particular system, its architecture and its system elements depend on an observer's interests and responsibilities. One person's SoI can be viewed as a system element in another person's SoI. Conversely, it can be viewed as being part of the environment of operation for yet another person's SoI.

The following are key points regarding the characteristics of the SoI:

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 SoI 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 (e.g. operators) within a system;

f) a system can be viewed in isolation as an entity, (e.g. a product), or as a collection of functions capable of interacting with its surrounding environment, (e.g. a set of services).

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

In this document, humans are considered both as users and as elements of a system. In the first case, the human user is a beneficiary of the operation of the system. In the second case, the human is an operator carrying out specified system functions, such as those involved in providing a service. An individual can be, simultaneously or sequentially, a user and an element of a system, for example, the pilot of a private aircraft in the civil aviation system performs as an operator of the aircraft while also benefitting from the transportation service provided by the aircraft.

Humans contribute to the performance and characteristics of many systems for numerous reasons, e.g. their special skills, the need for flexibility and for legal reasons. Whether they are users or operators, humans are highly complex, with behaviour that is frequently difficult to predict, and they need protection from harm. This requires that the system life cycle processes address human element factors in the areas of human factors engineering, system safety, health hazard assessment, workforce, personnel, training, ethical and cultural values. These issues are addressed by particular activities and iterations in the life cycle and are described in more detail in ISO 9241-210 and ISO 9241-220.

### 4.2.3 System structure

The system life cycle processes are described in relation to a system that is composed of a set of interacting physical, logical and other system elements, depicted in Figure 1, each of which can be implemented to fulfil its respective specified requirements. Responsibility for the implementation of any system element can therefore be delegated to another party through an agreement.

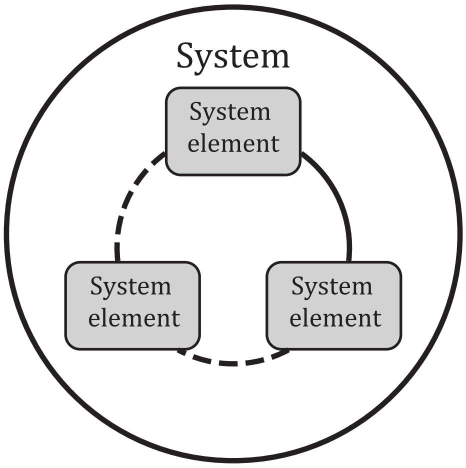



**Figure 1 — System and system element relationship**



The relationship between the system elements can be expressed in many forms, including hierarchies or networks. For more complex systems-of-interest, a prospective system element can itself need to be considered as a system (that in turn can be comprised of system elements) before a complete set of system elements can be defined with confidence, as indicated by Figure 2. In this manner, the system life cycle processes are applied recursively to a SoI to resolve its structure to the point where understandable and manageable system elements can be implemented or reused or acquired from another party. While Figure 2 implies 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. So, recursion is not necessarily linearly downward in all cases.

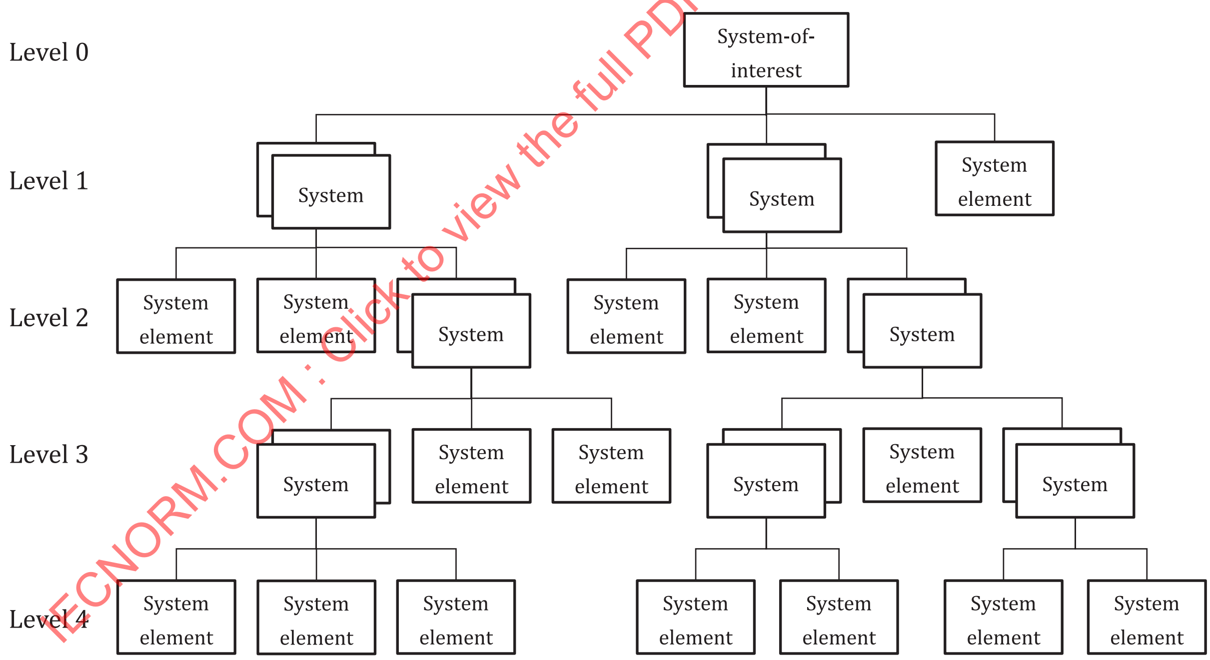



**Figure 2 — System-of-interest structure**

### 4.2.4 Enabling systems

Throughout the life cycle of a SoI, essential services are required from systems that are not directly a part of the operational environment of the SoI, e.g. new product development, mass-production system, training system, maintenance system. Each of these systems enables a part, for example, a stage, or stages, of the life cycle of the SoI to be conducted. Termed “enabling systems”, they facilitate progression of the SoI through its life cycle. The relationship between the services delivered to the operational environment by the SoI and the services delivered by the enabling systems to the SoI are shown in Figure 3. Enabling systems can be seen to contribute indirectly to the services provided by the SoI. Enabling systems can have their own customers, deployment approaches and use cases that can overlap other enabling systems. The interrelationships between the SoI and the enabling systems can be bi-directional or a one-way relationship at each stage in which the SoI and specific enabling systems interrelate. In addition to interacting with enabling systems, the SoI can also interact with other systems in the operating environment, shown as Other Interacting Systems A, B and C, and Enabling Systems 1, 2, 3, 4 and 5. Requirements for interfaces with enabling systems and other systems in the operational environment should be included in the requirements for the SoI.

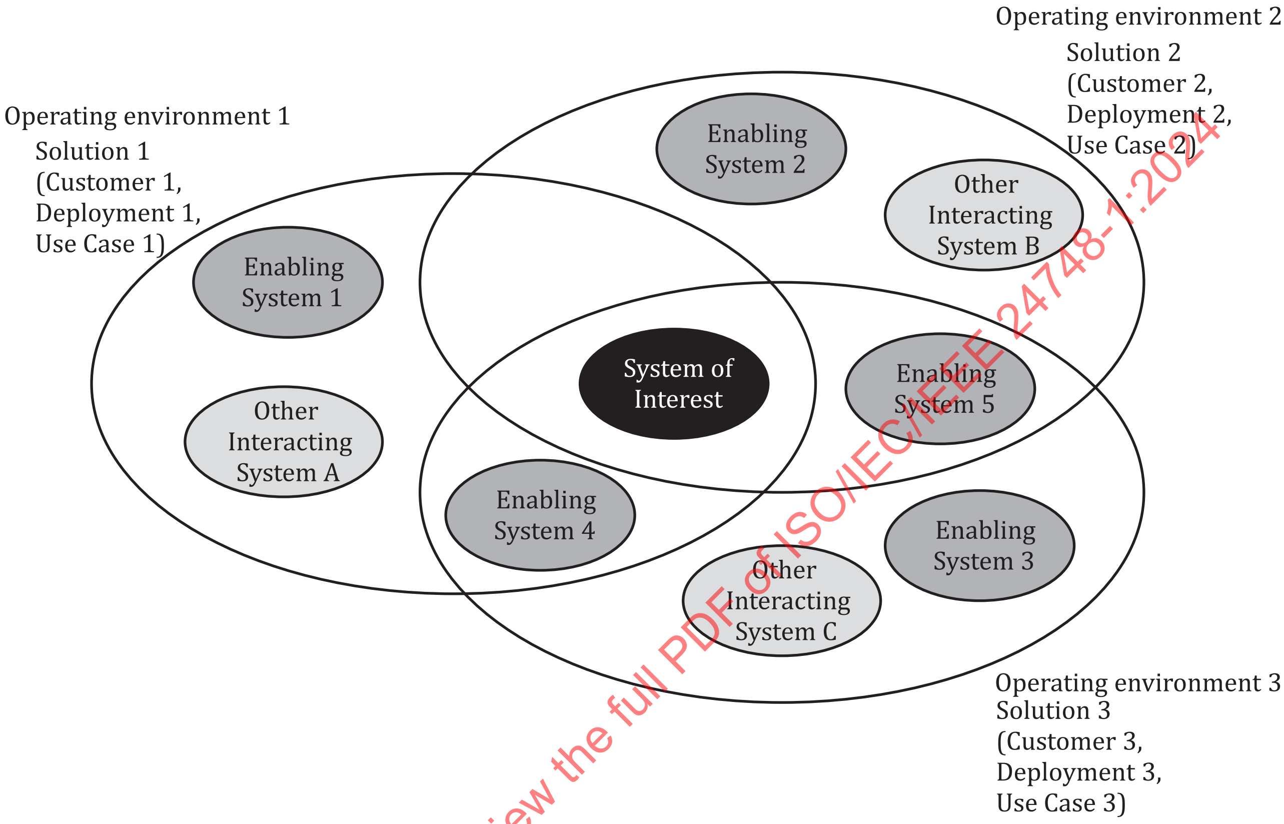



**Figure 3 — Services delivered by enabling systems to the system-of-interest**

During each stage in the system life cycle, the relevant enabling systems and the SoI 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 SoI can also be directly responsible for creating and using the enabling system. Creating the enabling systems can be viewed as a separate project and subsequently another SoI. A given enabling system can be used more than once (i.e. several times during a given stage, or in multiple stages, or both) during a system's life cycle, and that such use can have discontinuities. The enabling system can also be used concurrently in multiple life cycle stages. Thus, it is necessary to consider the full life cycle, maturity and concurrency requirements for the enabling systems. For example, a specific fixture can be used for retirement, for which there can be a significant time delay. Or the fixture can be needed for development, operations and maintenance, and have activities requiring the fixture concurrently. As another example, a training aid can be developed as part of the concept stage, then refined and reused in development, operation and maintenance stages.

See ISO/IEC/IEEE 24748-2:2024, Annex B for additional information about interfacing, enabling and interoperating systems.

## 4.3 Life cycle concepts

### 4.3.1 System life cycle model

Every system, whatever the kind or size, inherently evolves from its initial conceptualization through its eventual retirement. It is generally useful to build a life cycle model of this progression, showing each stage in the evolution, to help manage the evolution of the system. Models built for this purpose are termed life cycle models. Movement from one stage to another represents a decision point with specific criteria intended to be satisfied guiding stage entry and completion. These criteria usually are directly based either on the completion of specific tasks and demonstration of successful process outcomes, or readiness to start specific tasks in order to achieve required process outcomes. A life cycle model, then, is a decision-linked conceptual segmentation of the definition of the need for the system, its realization as a product or service or some mix of both, and its utilization, evolution and disposal. The actual progression of a system through the parts of the model, however done, is the system's life cycle. While the system has a life cycle, it is only by applying model(s) that subdivide the life cycle that the concept of stages becomes relevant. Models have stages, and any SoI can be regarded from the perspective of one or more models. Some models permit concurrency of stages while some do not. Which stage(s) a system is in depends on the models. A system can be in multiple life cycle stages at the same time.

A system life cycle model is segmented by stages because it facilitates planning, provisioning, operating and supporting the SoI. This segmentation by stages provides an orderly progression of a system through established decision-making gates to reduce risk and to enable satisfactory progress. This segmentation also allows the establishment of different development environments identifying processes, procedures, methods, techniques and tools appropriate to each stage. However, it does not negate using the same development environment for multiple stages. It can be useful to develop one or more specific process views from the viewpoint of stage transition decision making, following the information given in ISO/IEC/IEEE 24774:2021 Clause 6.

A secondary aspect of using a life cycle model is that it can help an organization think of its work and its processes within a larger framework, which can have useful business overtones. See, for example, the discussion of a services life cycle model in 6.3.4.1.

Several factors make system life cycle planning, provisioning and operation difficult to manage. Economics and market forces, as well as novelty, complexity and operational stability affect the length of a system's life cycle. Some systems have life cycles that are decades long (e.g. aircraft, satellites, ships) and some are very short (e.g. instruments and consumer electronics).

A typical system, however, progresses through stages where it is conceptualized, developed, produced as a product or service (or a mix of both), utilised, supported and retired. The life cycle model is the framework that helps ensure that the system can meet its required functionality throughout its life. Thus, to define system requirements and develop system solutions during the concept and development stages, competent people in the activities of other stages (e.g. production, utilization, support, retirement) are needed to perform trade-off analyses and to help make design decisions and arrive at a balanced solution. This helps ensure that a system has the necessary attributes designed in as early as possible. Also, it is essential to have the necessary enabling systems available to perform required stage functions.

A representative system life cycle model includes the stages: concept, development, production, utilization, support and retirement as shown in Figure 4. Although the figure shows stages as adjacent boxes, this should not be construed to mean a sequence or a linear progression between stages. An alternate system life cycle model is shown in Figure 5.

| | | | | | |
|---|---|---|---|---|---|
| Concept | Development | Production | Utilization | Support | Retirement |

**Figure 4 — Representative system life cycle model**

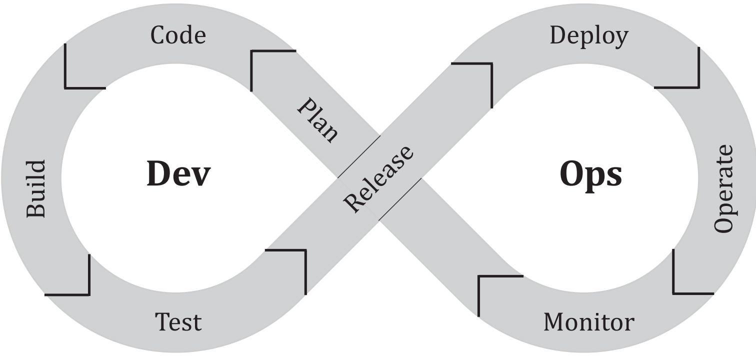
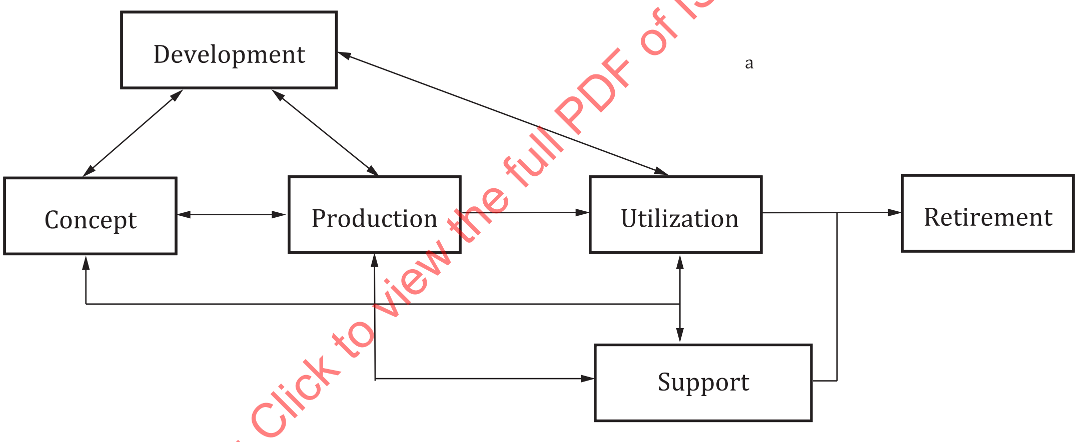


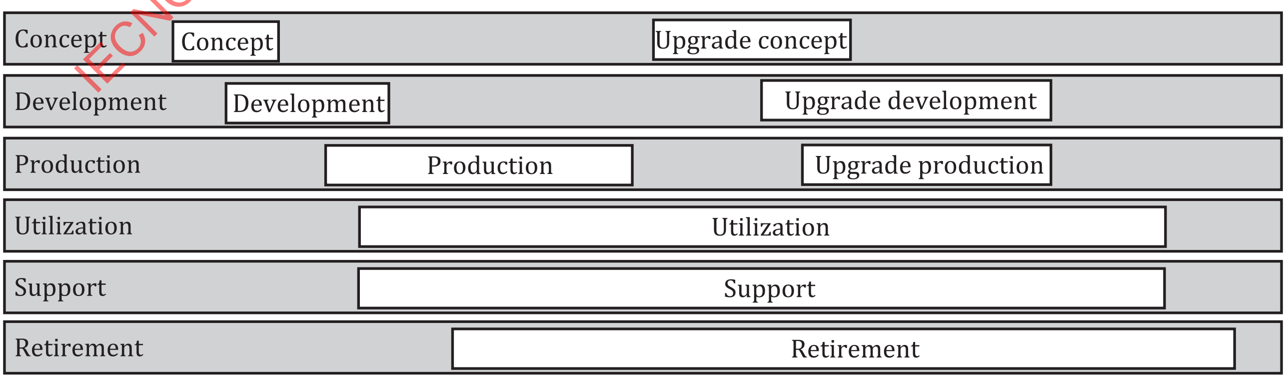

**Figure 5 — Development and operations (DevOps) example life cycle stages**

Stages, although drawn as discrete, are in practice interdependent, overlapping or concurrent. Further, figures sometimes implicitly convey a uniformity and single linearity of time progression that is not inherently part of a life cycle model: stages do not necessarily occur one after another in time sequence. Life cycle models can navigate non-sequentially between the various stages as shown in Figure 6. Or as shown in Figure 7, stages often occur in parallel and multiple times. So, one actuality of the "progression" of a system through its life cycle can be represented in many ways. Further, although each stage is sometimes represented as if it have approximately the same time duration as the other stages, the durations of different stages can be and usually are quite different. For example, for a given system, the concept stage can last only weeks, the development can take months, the utilization can continue for years and retirement can be accomplished in days.




a Iteration and recursion possible on all paths.

**Figure 6 — Life cycle model with some of the possible progressions**




NOTE 1 Utilization and support stages begin with first unit. For software, support can occur during development and before operational utilization.

NOTE 2 Retirement begins when first unit retires or is lost to damage. System elements can be retired prior to system retirement.

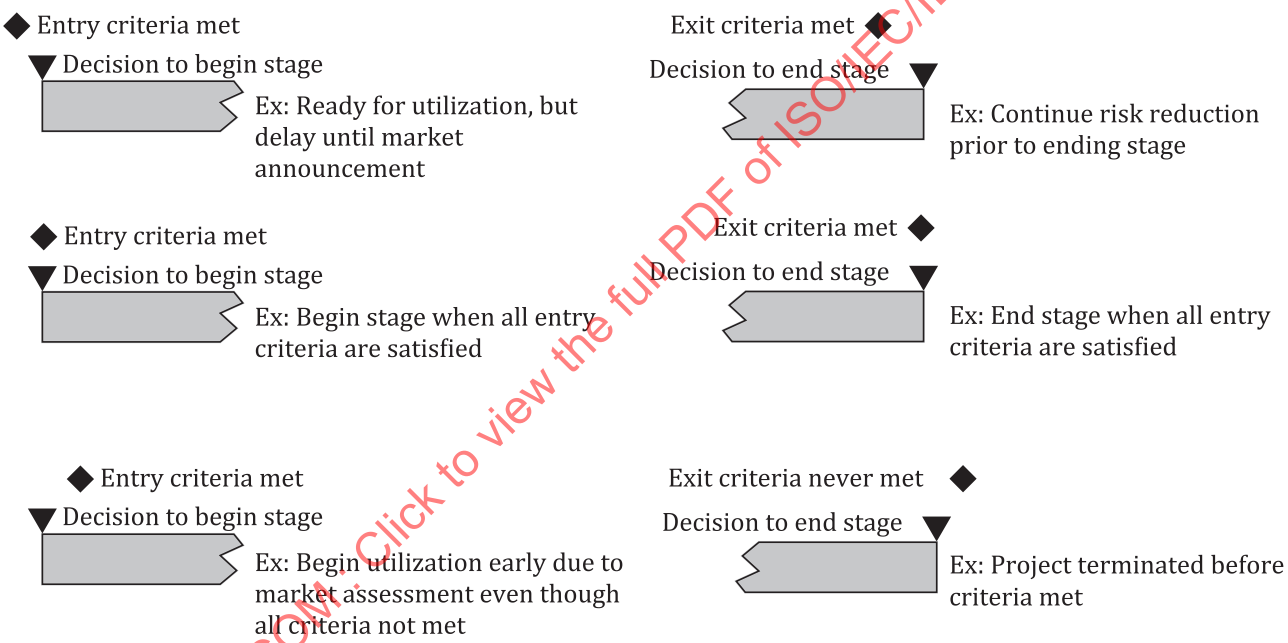
NOTE 3 Concept, development and production stages can be restarted for mid-life upgrade or service life extension.

NOTE 4 Support parallels utilization until last unit is removed from service. Support can end while units are still being utilised.

NOTE 5 Retirement stage continues until last unit is retired. It is not always possible to know if units are still being utilised.

**Figure 7 — Example of parallel life cycle stages**

A system progresses through its life cycle as the result of actions, performed and managed by people in organizations, using processes. The detail in the life cycle model is expressed in terms of these processes, their outcomes, relationships and occurrence. Since stages can have complex relationships, definition of entry and exit criteria for a specific stage are needed. Figure 8 shows some examples of stage entry and exit criteria and potential alignment of the decisions. Entry or exit criteria can be met (diamonds in the figure) before the decision, concurrent with the decision, after the decision, or not at all (triangles). Decision criteria can include an assessment of technical debt and its implications for other stages. Technical debt can accumulate in any life cycle stage.




NOTE Decision criteria can include stage entry/exit criteria, entry/exit criteria from other stages, risk assessment, external events.

**Figure 8 — Stages begin and end based on criteria or external events**

Four common principles associated with a life cycle model are the following:

— a system progresses through specific stages during its life;

— enabling systems should be available for each stage in order to achieve the outcomes of the stage;

— at specific life cycle stages, quality characteristics such as producibility, usability, supportability and disposability should be specified and designed or implemented into a system;

— stages begin and end based on criteria or external events.

### 4.3.2 System life cycle stages

Life cycles vary according to the nature, purpose, use and prevailing circumstances of the system. Nevertheless, despite an apparently limitless variety in system life cycles, there is an underlying, essential set of stages that exists in the complete life cycle of any system. Each stage has a distinct purpose and contribution to the whole life cycle and should be considered when planning and executing the system life cycle.

Stages can represent divisions of the life cycle associated with a system and they relate to the state of the system description or realization of the system's set of products or services. The stages describe the major progress and achievement milestones of the system throughout its life cycle. Because the life cycle model makes a distinction between various stages, some criteria, decisions or gates can be used to guide the transition between stages. These criteria, decisions or gates can facilitate consistent approaches and outcomes as the system transitions between stages of the model. The stages thus provide organizations with a framework within which management has high-level visibility and control of project and technical processes. Stages can include sub-stages to further divide the life cycle.

Table 1 shows a commonly encountered example of life cycle stages. Also shown are the principal purposes of each of these stages and the possible decision options used to manage the achievement and risk associated with progression through the life cycle.

**Table 1 — Example of stages, their purposes and major decision gates**

<table>
<tr><th>Life cycle stages</th><th>Purpose</th><th>Decision options</th></tr>
<tr><td>Concept</td><td>Identify stakeholders' needs<br>Explore concepts<br>Propose viable solutions</td><td rowspan="6">— Begin subsequent stage or stages<br>— Continue this stage<br>— Go to or restart another stage<br>— Hold project activity<br>— Terminate project</td></tr>
<tr><td>Development</td><td>Refine system requirements<br>Create solution description<br>Build system<br>Verify and validate system</td></tr>
<tr><td>Production</td><td>Produce systems<br>Inspect and test</td></tr>
<tr><td>Utilization</td><td>Operate system to satisfy users' needs</td></tr>
<tr><td>Support</td><td>Provide sustained system capability</td></tr>
<tr><td>Retirement</td><td>Store, archive or dispose of system</td></tr>
</table>

Organizations employ stages differently to satisfy contrasting business and risk mitigation strategies. Using stages concurrently and, in a few cases, even in different orders, can lead to life cycle forms with distinctly different characteristics. Sequential, incremental, concurrent or evolutionary life cycle forms are frequently used. Alternatively, a suitable hybrid of these can be developed. The selection and development of such life cycle forms by an organization depend on several factors, including the business context, the nature and complexity of the system, the stability of requirements, the technology opportunities, the need for different system capabilities at different times and the availability of budget and resources. In addition, major decision gates, often called milestones and reviews to focus on making the decisions can be incorporated by an organization on an incremental basis within a stage, as well as at the end of a stage, to further manage risks.

Just as all the system elements of a system contribute to the system as a whole, each stage of the life cycle should be considered during any other stage of the life cycle. As a consequence, the contributing parties need to coordinate and cooperate with each other throughout the life cycle. This synergism of the life cycle stages and the functional contributors is necessary for successful project actions. Close communication with project team members from the different functions and organizations responsible for other life cycle stages leads to consistency in the life cycle.

### 4.3.3 Stages in a SoI and its enabling systems

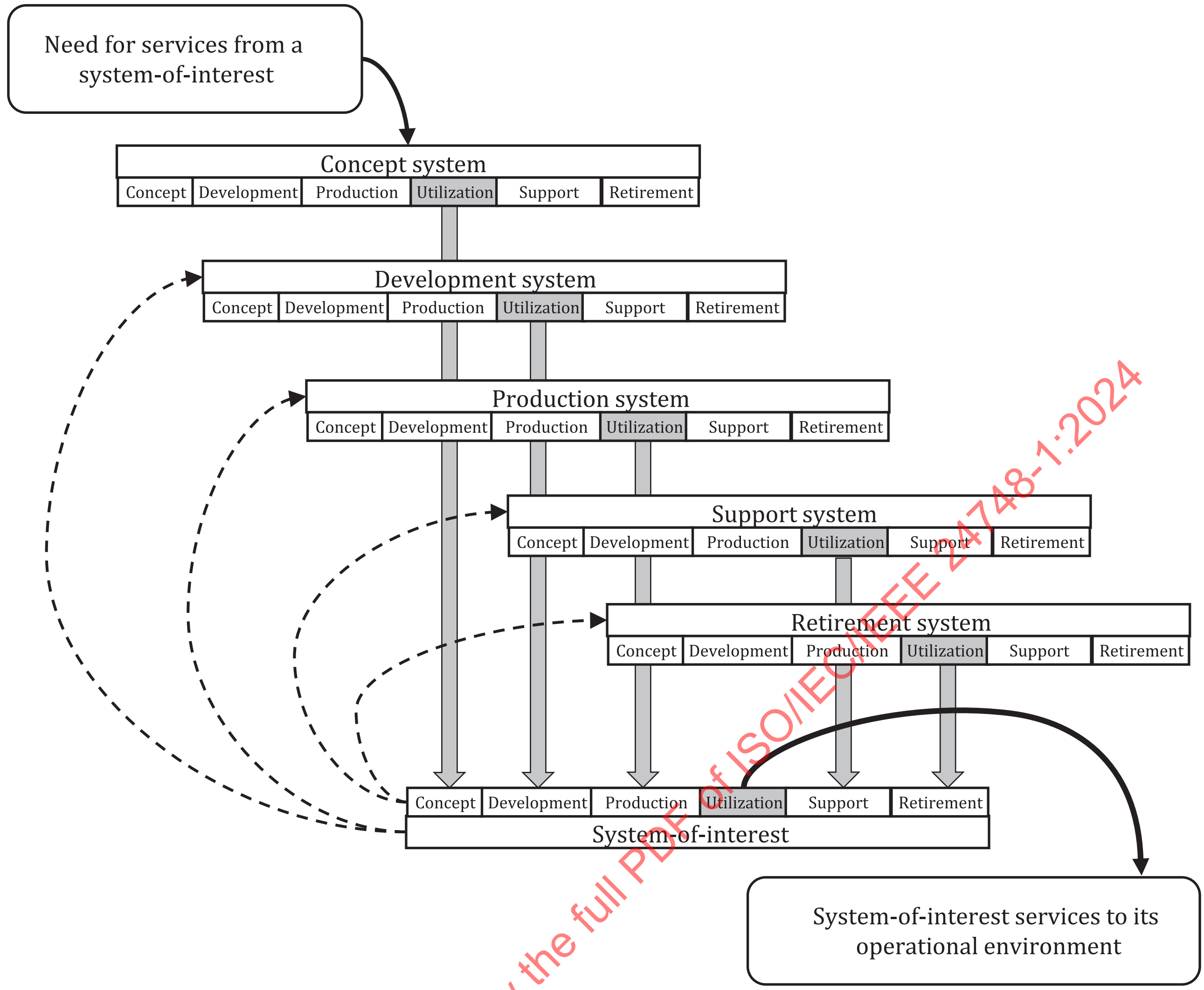
As with any system, each enabling system also has its own life cycle. Each life cycle is linked and synchronized to that of the SoI. For example, if an enabling system does not already exist, its requirement is defined during the concept stage of the SoI (or later if lead times permit), before the enabling system is utilised as shown in Figure 9 to provide its particular service to the SoI.

An enabling system can pre-exist the SoI. Enabling systems can, for example, be an existing part of the infrastructure of the organization responsible for the SoI or be in a service supplier's organization. Pre-existing enabling systems can introduce additional constraints on the SoI.

Each enabling system can itself be considered as a SoI, having in turn its own enabling systems. Therefore, the concepts in this document can also be applied to enabling systems.

Combinations of multiple life cycle models can be applied, such that iteration (e.g. typical agile development) and serial (e.g. waterfall) life cycle models are applied to different system components of SoI, while DevOps can be introduced during the initial development or after an initial system is operational, to closely involve development and operations in a continuous integration/continuous delivery model. System elements within the SoI can use life cycle models that differ from that used to model the SoI.

Likewise, constituent systems (CS) participating in an SoS can use different life cycle models and be at various stages within them. The SoS can be requested to interact with CS that are in different stages. That is because CS are managed and operated independently from SoS. For example, some CS can be in their utilization stage while others can still be in their concept or development stages, while the SoS is in its development stage.




**Key**

- - - -▸ the system-of-interest requirements for enabling services

⇨ the enabling systems services delivered to the system-of-interest

**Figure 9 — System interaction with typical enabling systems**



# 5 Life cycle stages

## 5.1 General

Life cycle stages are the specific framework within which system life cycle processes are applied throughout the system life cycle. The value of stages is that entry into and exit from each stage represents a transition in the primary activities applied to the SoI and a decision point. Progression from one stage to another for a SOI or some increment of the system can be explicitly gated with decision points, exit criteria and entry criteria. At a decision gate, progress is evaluated against criteria for each stage. These criteria determine when a stage can be exited and when other appropriate stages can be commenced.

NOTE ISO/IEC/IEEE 24748-8 provides more rigorous requirements for technical reviews and audits in support of decision gates.

Each life cycle process of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 can be invoked at any point throughout the life cycle (see 4.2.1). The order in which processes are invoked, and when they are invoked, is driven by the project requirements and context: there is no unique definitive order or time sequence for their use within a stage or across several stages. For a given process, the outcome or outcomes for one stage can differ from those in other stages. Outcome or outcomes for each process invoked during a given stage to support the exit criteria for that stage and the entrance criteria for any ensuing stages.

System, software, service (or other) life cycle processes can be invoked concurrently, iteratively, recursively or time dependently in any applicable stages of an individual system, software, service or other life cycle model. applicable to a given project. The scale and rigour of process application in the listed stages and the duration of these stages are guided by the varying business and technical needs of the projects defining and using the life cycle model.

The processes invoked are generally performed by systems other than the SoI, e.g. by enabling systems.

This document describes the following six stages as examples:

a) concept stage;

b) development stage;

c) production stage;

d) utilization stage;

e) support stage;

f) retirement stage.

Other life cycle models and development approaches exist. For example, agile is an iterative and incremental approach to product delivery. DevOps (development and operations) is the set of principles and practices which enable better communication and collaboration between relevant stakeholders for the purpose of specifying, developing and operating software, and systems products and services, and continuous improvements in all aspects of the life cycle. Having similar objectives, agile and DevOps approaches are frequently combined.

For each example stage, an overview description is given, followed by a detailed description with purpose and outcomes.

Outcomes common to all stages include the following.

— Plans and exit criteria for the stage are provided for common use by stakeholders, including records of their preliminary and revised versions.

— Risks and mitigating actions are identified for common use by stakeholders, including records of their preliminary and revised versions.

— Technical debt and its implications for other stages are assessed.

— Enabling systems are implemented and available.

— Stage exit criteria are achieved.

— Authorization to proceed to the appropriate stage or stages, based on the specific life cycle model in use by the project, is achieved.

— Information items are generated or updated.

## 5.2 Concept stage

### 5.2.1 Overview

The concept stage begins with initial recognition of a need or a requirement for a new SoI or for the modification to an existing SoI. This is an initial exploration, fact finding and planning period when economic technical, strategic and market bases are assessed through acquirer/market survey, business or mission analysis, solution space identification and feasibility analysis and trade-off studies. Acquirer/user feedback to the concept is obtained.

One or more alternative concepts to meet the identified needs or requirements are developed through analysis, feasibility evaluations, estimations (such as cost, schedule, market intelligence and logistics), trade-off studies, risk assessment and experimental or prototype development and demonstration. Concepts can be generated from existing systems. The need for one or more enabling systems for adaptation, development, production, utilization, support and retirement of the SoI is identified and candidate solutions are included in the evaluation of alternatives in order to arrive at a balanced life cycle solution. Typical outputs of this stage are stakeholder requirements, concepts of operation, assessment of feasibility, preliminary system requirements, outline architecture and design solutions in the form of drawings, models, prototypes, etc., and concept plans for enabling systems, including whole life cycle cost and human resource requirements estimates and preliminary project schedules. Decisions are made whether to continue with the implementation of a solution in the development stage or to cancel further work.

It is presumed that the organization has available enabling systems for the concept stage that consist of the methods, techniques, tools and competent human resources to undertake market/economic analysis and forecasting or mission analysis, feasibility analysis, trade-off analysis, technical analysis, whole life cost estimation, modelling, simulation and prototyping.



### 5.2.2 Purpose

The concept stage is executed to assess new business opportunities or mission assignments for feasibility and to discern stakeholder needs and requirements. The concept stage can be extended into evaluation of conceptual architecture and design solution.

### 5.2.3 Outcomes

In addition to the outcomes common to all stages, outcomes of the concept stage are listed below.

a) New concepts are identified with stakeholders' consensus that offer such things as new capabilities, enhanced overall performance or reduced stakeholders' total ownership costs over the system life cycle.

b) An assessment is performed on feasible SoI concepts, with initial architectural and other solutions, including enabling systems throughout the life cycle, for closure against technical and business or mission stakeholder objectives.

c) Preliminary stakeholder, system and usability requirements (preliminary technical specifications for the selected SoI and usability specifications for the envisaged human-machine interaction) are prepared and baselined. See ISO 9241-210 for additional guidance on human-machine interactions.

d) Refined outcomes and cost estimates for stages of the system life cycle model are provided for common use by stakeholders.

e) Risk identification, assessment and mitigation plans are prepared for common use by stakeholders for this and subsequent stages. The system life cycle model can also be the subject of risk management.

f) Identification and initial specification of the services that are needed from enabling systems throughout the life of the SoI are defined.

g) Concepts for execution of all succeeding stages are provided for common use by stakeholders, including preliminary plans and preliminary entry and exit criteria.

h) Definition of the enabling system services required in subsequent life cycle stages is provided for relevant stakeholders.

i) Project budget and schedule baselines and life cycle ownership cost estimates are provided for common use by stakeholders.

j) Review is performed for any existing systems or system elements that can be leveraged to accelerate meeting the customer's need.

k) Review is performed for system of systems (SoS) considerations if the system will operate in an existing or developing ecosystem of other systems.

## 5.3 Development stage

### 5.3.1 Overview

The development stage includes sufficiently detailed technical refinement of the system requirements, system architecture and the design solution and transforms these into one or more feasible products that enable one or more services during the utilization stage. The SoI can be a prototype in this stage or an operational system being considered for adaptation to meet a new need or market. The hardware, software, operator, process and facility interfaces are specified, analysed, designed, fabricated, integrated, tested and evaluated, as applicable and the requirements for production, training and support facilities, and transitions are defined. This stage also involves considering and incorporating into the design the applicable constraints of other stages (production, utilization, support and retirement) and their enabling systems' requirements and capabilities. Feedback is obtained from stakeholders, including those who produce, operate, use, support and retire the SoI through such means as a series of technical or other reviews. Outputs are a SoI or a prototype of the final SoI, refined requirements for enabling systems or the enabling systems themselves and all documentation and cost estimates of other stages.

Planning for this stage includes preparing to establish an infrastructure of development enabling systems consisting of facilities, processes, procedures, methods, techniques, tools and competent human resources to undertake analysis, modelling and simulation, prototyping, design, integration, test, transition and documentation. These items are modified, developed or acquired to be available when needed to support development.



### 5.3.2 Purpose

The development stage is executed to develop a SoI that meets stakeholder requirements and can be produced, tested, evaluated, operated, supported and retired.

### 5.3.3 Outcomes

In addition to the outcomes common to all stages, outcomes of the development stage are listed below.

a) Architecture models and system designs are created for the SoI.

b) A structure for the SoI, is comprised of engineered system elements, for example, hardware elements, software elements, human elements, process elements, facility elements and the interfaces (internal and external) of all such elements.

c) Verification and validation results are available for assurances of SoI capabilities with their recorded evidence.

d) Transition planning is conducted to help stakeholders who perform system transition to their own environments of use, including establishment of hardware, software and facility elements, implementation of operating procedures and training in support of the utilization stage.

e) Evidence supporting a decision is available for assurances of SoI capabilities, with all risks and benefits considered, that the SoI meets all specified requirements and is producible, operable, supportable and capable of retirement and is cost-effective for stakeholders.

f) Refined and baselined requirements for the enabling systems are provided for common use by stakeholders, along with methods and tools for establishing and maintaining traceability between requirements and the developed system.

g) A prototype or operable SoI is available.

h) Refined outcomes and cost estimates for the production, utilization, support and retirement stages are provided for common use by stakeholders.

## 5.4 Production stage

### 5.4.1 Overview

The production stage begins with the approval to produce the SoI. The SoI can be individually produced, assembled, integrated and tested, as appropriate, or can be mass-produced. Initial planning for this stage can begin in another stage. During production, the system can undergo enhancements or redesigns, the enabling systems can be reconfigured, and production staff can be retrained in order to continue evolving a cost-effective product or service from the stakeholder view. That is, the production stage and the development stage can overlap or execute concurrently.

NOTE For software systems, production can involve enabling systems, physical media and packaging.

It is presumed that the organization has budget and enabling systems that consist of production equipment, facilities, tools, processes, procedures and competent human resources. These items are developed or acquired in order to be available when needed to enable production. See ISO/IEC 26550 for additional guidance and considerations related to product line engineering and management.

### 5.4.2 Purpose

The production stage is executed to produce or manufacture the SoI, test it and produce related enabling systems as needed.

### 5.4.3 Outcomes

In addition to the outcomes common to all stages, outcomes of the production stage are listed below.

a) Qualification of the production capability is determined enough for assurances of SoI capabilities.

b) Resources, material, services and system elements are acquired to support the target production quantity goals.

c) An operable system is produced according to approved and qualified production information.

d) The packaged product is transferred to distribution channels or acquirer.

e) Updated concepts are provided for execution of all other stages.

f) A verified and quality-assured SoI is accepted by the acquirer.



## 5.5 Utilization stage

### 5.5.1 Overview

The utilization stage begins with installation and transition to use of the system. The utilization stage is executed to operate the product at the intended operational sites to deliver the required services, material and data with continuing operational and cost-effectiveness. This stage ends when the SoI is taken out of service.

Planning for this stage begins in other stages. This stage includes those processes related to use of the system to provide services, as well as monitoring performance and identifying, classifying and reporting of anomalies, deficiencies and failures. The response to identified problems includes taking no action, maintenance and minor (low cost/temporary) modification, major (permanent) modification and SoI life extensions, and end-of-life retirement. Some decisions can require a change to other life cycle stages, such as development (implying a major modification of the SoI).

During this stage, the product or services can evolve, giving rise to different configurations. Enabling systems can likewise evolve. The operator operates the different configurations and the responsible product supplier manages the status and descriptions of the various versions and configurations of the product or services in use.

It is presumed that the organization has available at the utilization stage the enabling system which can include elements such as facilities, hardware and software, processes, procedures, trained personnel, documentation and data. These items are developed or acquired in order to be available when needed to support utilization.

### 5.5.2 Purpose

The utilization stage is executed to operate the system, to deliver services within intended environments and to help achieve continuing operational effectiveness.

### 5.5.3 Outcomes

In addition to the outcomes common to all stages, outcomes of the utilization stage are listed below.

a) Trained personnel are assigned who have the competence to operate the SoI and provide operational services.

b) An installed SoI is deployed such that it is capable of being operated and of providing sustainable operational services.

c) Confirmed conformance to required service objectives.

d) New opportunities for SoI enhancement are identified through stakeholder feedback.

NOTE It is not always possible to know when the last unit has been removed from operation.

## 5.6 Support stage

### 5.6.1 Overview

The support stage begins with the provision of maintenance, logistics and other support for the SoI's operation and use. Planning for this stage begins in other stages. The support stage is completed with the retirement of the SoI and termination of support services. However, support can be revoked/ended and the support stage completed, even if units are still in use.

This stage includes those processes related to providing services that support utilization of the SoI. This stage also includes processes to use and monitor the support system itself and its services, including the identification, classification and reporting of anomalies, deficiencies and failures of the support system and services. Actions to be taken as a result of identified problems with the support system include maintenance and minor modification of the support system and services, major modification of the support system or services, as well as end-of-life retirement of the support system and services.

During this stage, the support system and services can evolve under different versions or configurations. The support organization operates the different versions or configurations and the responsible product organization manages the status and descriptions of the various versions and configurations of the support system and services in use.

It is presumed that the supporting organization has available the enabling systems, which consist of facilities, equipment, tools, processes, procedures, trained support personnel and maintenance manuals. The items making up the support enabling system are developed and acquired in order to be ready when needed to support the SoI.

### 5.6.2 Purpose

The support stage is executed to provide logistics, maintenance and support services that enable continuing SoI operation and a sustainable service.

#### 5.6.3 Outcomes

In addition to the outcomes common to all stages, outcomes of the support stage are listed below.

a) Trained and competent personnel are assigned to maintain the SoI and provide the support services.

b) Organizational and enabling system interfaces are provided for problem resolution and corrective actions.

c) Product and services and the provision of all related support services, including logistics and security are deployed to the operational sites.

d) Problems or deficiencies are identified; appropriate parties (user, development, production or support) are informed of the need for corrective action, including sharing of historical resolution records.

e) Product and service are maintained and design deficiencies are corrected.

f) All required logistics support is operationally provided, including a spare parts inventory sufficient to satisfy operational availability goals.

NOTE Transition from the support stage can be to the retirement stage or the concept stage for an enhanced SoI.

### 5.7 Retirement stage

#### 5.7.1 Overview

The retirement stage provides for the removal of a SoI and related operational and support services, including appropriate disposal of specified system elements. Planning for the retirement stage begins in other stages. This stage begins when a SoI is taken out of service.

This stage includes those processes related to operating the system that enables retirement of the SoI (the retirement enabling system), including appropriate disposal of specified enabling system elements, and also includes monitoring performance of that enabling system and the identification, classification and reporting of anomalies, deficiencies, and failures of the retirement enabling system. Actions to be taken as a result of identified problems include maintenance and minor modification of the retirement enabling system, major modification of the retirement enabling system and end-of-life retirement of the retirement enabling system itself.

It is presumed that the organization has access to enabling systems, which consists of facilities, tools, processes, procedures, equipment, trained personnel and, as appropriate, access to recycling, disposal or containment facilities. The items making up the retirement enabling system are developed and acquired in order to be ready when needed to perform retirement functions.

This stage is applicable whenever a SoI reaches its end-of-service life. Such end-of-service life can be the result of replacement by a new system, irreparable wear, catastrophic failure, no further use to the user (e.g. through change in mission or business direction), or when it is no longer cost-effective to continue operating and supporting the SoI.



#### 5.7.2 Purpose

The retirement stage is executed to provide for the removal of a SoI and related operational and support services and to operate and support the retirement system itself.

#### 5.7.3 Outcomes

In addition to the outcomes common to all stages, outcomes of the retirement stage are listed below.

a) Disposal constraints are provided as inputs to requirements, architecture, design and implementation.

b) Any enabling systems or services needed for disposal are available.

c) Related operational and support services are terminated.

d) The system elements or waste products are destroyed, stored, reclaimed or recycled in accordance with safety and security requirements.

e) The environment is returned to its original or an agreed state.

f) Records of disposal actions and analysis are available.

# 6 Life cycle model adaptation

## 6.1 General

No two projects are the same. Each organization is driven by the nature of its mission or business, its social responsibilities and its forward strategy. These provide constraints on available opportunities that the organization and its projects can exploit. To help exploit opportunities, the organization establishes policies and procedures to guide the performance of projects. Variations in organizational policies and procedures, acquisition methods and strategies, project size and complexity, system requirements and development methods, among other things, influence how a system is acquired, developed, operated or maintained. See Annex C for additional information on project concepts.

## 6.2 Adaptation sequence

### 6.2.1 General

Figure 10 gives an illustrative sequence of steps that can be followed to adapt the life cycle model to a particular need. First, the adaptations reflecting the project environment are addressed, and then inputs on possible changes are solicited from potentially affected parties. After that, the specific life cycle model is selected.

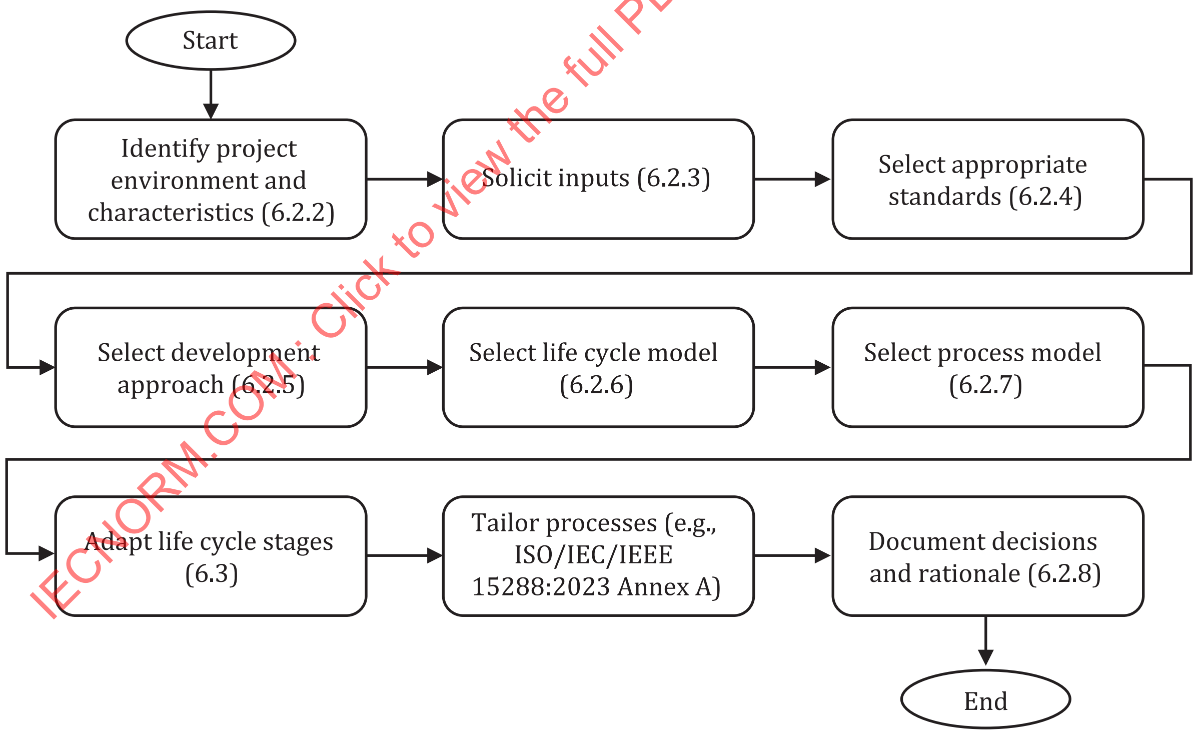



**Figure 10 — Adaptation sequence**

### 6.2.2 Identification of the project environment and characteristics

The project's organizational and enterprise characteristics can be determined by considering such issues as enterprise strategy, organizational policies and procedures. Identify the relevant policies, procedures and regulatory constraints of the organizations involved, particularly of the acquirer and supplier, with which the project should comply. Examples include policies and procedures related to ethics, security, safety, privacy, risk management, use of an independent verification and validation agent, use of a specific computer programming language and hardware resourcing. Pertinent laws and regulations that can impact the project, including those related to environment, public safety and privacy, should be identified and subsequently monitored for compliance.

### 6.2.3 Solicitation of inputs

The requirements derived from the relevant business or mission and contractual needs are major drivers in adapting the life cycle model to support using the processes from an International Standard. Affected parties should be involved in the adaptation decisions. These parties can help ensure that the resulting adapted life cycle model is feasible and useful. Where possible, include feedback from previous projects. For example, the life cycle model for a project using ISO/IEC/IEEE 12207 can be adapted according to the contract between a supplier and the purchaser of a software product. A customer can require the design of software to be carried out and not full development of a software system, which can be reflected in one or more stages and their relationships in the life cycle model. In another instance, if the customer requirements are for safety critical software, it can be appropriate for the acquirer to require the execution of activities and tasks beyond those provided in ISO/IEC/IEEE 12207, which can, for example, affect the exit criteria of a specific life cycle stage or stages.

### 6.2.4 Selection of appropriate standards

ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 provide an interoperable suite of processes suitable for the life cycles of systems, with primary interest of either software or hardware. Projects can vary in their relative emphasis upon system and software aspects. This variation should inform the choice of standards appropriate to the project.

— In the context of this document, ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207, there is a continuum of human-made systems from those that use little to no software to those in which software is the primary interest. When software is the predominant system or element of interest, ISO/IEC/IEEE 12207 should be used. Both documents have the same process model, share most activities and tasks and differ primarily in descriptive notes. The determination of the applicability of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 should be decided by the nature of the system and its enabling systems. Often, a mixed tailoring of each standard can be appropriate.

— For projects for which the SoI is an SoS or a constituent of a broader SoS, ISO/IEC/IEEE 21841, ISO/IEC/IEEE 21839 and ISO/IEC/IEEE 21840 can be used to augment the processes in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288.

### 6.2.5 Selection of development approach

Determine which development approach is most relevant and applicable for the system, such as sequential, evolutionary, incremental agile or pre-planned product improvement. Each such development strategy prescribes certain processes and activities that can be performed sequentially, repeated and combined, or performed in parallel. In these development strategies, the life cycle processes in the relevant International Standards should be mapped to the selected development strategy and, in turn, mapped to the specifics of the life cycle model used for the project. For evolutionary, incremental and pre-planned product improvement approaches, the outputs of one project activity can feed into the next or be performed in parallel. In these cases, the documentation should be complete at the end of an activity or a task and the exit criteria of the life cycle model should reflect that fact. Annex E provides guidance on development approaches and build planning.

Determination of the development approach should be performed for the system and for each system element that is itself developmental and similarly for all enabling systems.

### 6.2.6 Selection of life cycle model

Identify the relevant life cycle model stages needed for the project, as well as their entry and exit criteria and their relationship (serial, parallel, wholly or partly combined). Define the outcomes that determine successful completion of each of the stages and the milestones or decision gates that distinguish them. See 6.3 for life cycle model adaptation guidance. Also, select and prioritize the appropriate processes in ISO/IEC/IEEE15288 or ISO/IEC/IEEE12207 that can be implemented to achieve the outcomes of the stages.

### 6.2.7 Selection of process model

To help establish these policies and procedures and to determine the resources needed by the organization, International Standards, such as ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288, can be used to provide specific standardized processes for use within one or more life cycle models. However, in the interest of cost reduction and quality improvement, the processes from the International Standards can be tailored and the life cycle models can be adapted for an individual project to reflect the variations appropriate to the organization, project and system. The framework for tailoring processes is given ISO/IEC/IEEE 15288:2023, Clause 4 and Annex A and ISO/IEC/IEEE 12207:2017, Clause 4, 5.5.4 and Annex A. See ISO/IEC/IEEE 24748-2 and ISO/IEC/IEEE 24748-3 for practical advice for implementing the life cycle model management process.

NOTE In implementing a suite of processes intended to be applied uniformly throughout an organization, it is usually preferable to start with those processes that will achieve the most significant returns, rather than attempting to implement all of the International Standard at once.

International Standards, such as ISO/IEC/IEEE 15288 or ISO/IEC/IEEE 12207 do not define the sequencing of activities of processes and they do not prescribe any particular life cycle model. Mapping the organization's current processes, practices and methods to the relevant processes, activities and tasks is useful at this stage. The mapping can be used to verify the completeness of the approach; that is, to identify where gaps exist between the current situation and the target situation where processes from the International Standard are used.

### 6.2.8 Documentation of decisions and rationale

When applying International Standards, such as ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288, a mapping of the defined processes and activities onto the selected life cycle model should be documented, together with the determined relationships and the reasons for adopting this approach. The verification of this work is to demonstrate that the outcomes of each stage to be achieved by the processes selected to implement the stage. This information should be incorporated into the project management planning as it provides a reference framework for evaluating the success or otherwise of the approach taken.

Specific guidance and examples of adaptation can be found in ISO/IEC/IEEE 24748-2 and ISO/IEC/IEEE 24748-3, which provide the application guidance for ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207, respectively.

## 6.3 Life cycle model adaptation guidance

### 6.3.1 General

ISO/IEC/IEEE 24748-3:2020, Clause 5 and ISO/IEC/IEEE 24748-2:2024, Clause 6 provide general guidance on adapting applications of ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288, respectively. This subclause provides guidance on life cycle model scope and stage adaptation, as well as adaptations for specific domains, disciplines and specialities.

Whatever life cycle model is used in dealing with problems arising during the execution of life cycle processes, it is useful to maintain a unified problem reporting capability. By its nature, a unified capability can deal with large numbers of problems. It is therefore useful to categorize the problems in various ways. Annex G describes two classifications: by category and by priority. Others can be applicable in particular situations.

Additional adaptation considerations can be found in the conformance requirements of ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288.

### 6.3.2 Scope adaptation

As an example, if an organization does development only and is not involved in the utilization, support or retirement life cycle stages, that organization can adapt the scope of ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 accordingly by selecting only the appropriate processes. The policies and procedures called for in the non-applicable parts of the International Standard can be omitted from in the organization's policies and procedures. Additionally, inputs that can help shape the policies and procedures of an organization are the following:

a) life cycle model and related entry or exit criteria used by the organization for decision making, as well as for establishing milestone reviews of a project; select one or more appropriate life cycle models for the project, since hardware, software, humans and other aspects of the system can have their own life cycles; determine whether the life cycle model is a sub-part of the life cycle model of the SoI, or enabling system or is the complete life cycle model;

b) resource availability and the resources the organization is willing to commit;

c) expertise and skills available to the organization to provide the organization's products and services;

d) technology available for the organization's products and services.

### 6.3.3 Stage adaptation

Depending on the specific SoI and the environment in which a project is established to realize the SoI, stages can be combined, eliminated or added to the generic model. For example, an organization can buy a concept, establish a project to design and produce a system, then turn the result over to another organization for marketing to consumers or operation and maintenance. In another case, there can be repeated iterations between concept and development stages or lengthy evolution after fielding of a system. Each stage in these cases can have different criteria for entry and exit, as well as possibly drawing on different processes. Accordingly, each stage in the life cycle, as well as the juxtaposition of each stage with those before and after it, requires specific consideration for the project and system to be realized.

### 6.3.4 Life cycle model adaptation for domains, disciplines and specialties

#### 6.3.4.1 Adaptations for domains

The life cycle model that is appropriate at the system level can need adaptation if the SoI falls largely or entirely within one domain, such as software. Further, the life cycle model that best reflects events in one domain (such as software) are not equally suitable for others (such as hardware, humans, processes, procedures, facilities and naturally occurring entities), as is illustrated by the examples shown in Table 2.

The life cycle stages shown are illustrative only; this is not an exhaustive list of possible life cycle models. For example, using life cycle models is just as relevant to services as it is to products. A life cycle model for service management can include stages of: service strategy, service design, service transition, service operation and continual service improvement, as shown in Table 2. The advantage to a service organization of using a life cycle model approach is that it gives them a unifying framework for examining what processes they need to have to do their work. If a service management organization viewed itself as doing nothing but service operations, not thinking in terms of a life cycle model, it can, for example, fail to consider continual service improvement or not occasionally re-examine its service strategy. That can make the organization less desirable to a customer compared to a service management organization that uses a life cycle model framework to keep its competitive view as wide as possible.

For all models, the length of any one stage or of the entire model does not represent any uniform time scale or coincident start or stop timing.

The basic point is that each domain's perspective on the system element's life cycle should be considered. As a result, when a system has elements that span multiple domains;

a) every domain's life cycle model should be thought through; and

b) the life cycle models and their stages should be considered as a whole, and care is taken that they work in concert.

**Table 2 — Example of life cycle models for different system types**

| System type | Stages |
|---|---|
| System | Concept, development, production, utilization, support, retirement |
| Software | Concept, development, operations and maintenance, retirement |
| Hardware | Concept, design, fabrication, operations and maintenance, retirement |
| Services | Service strategy, service design, service transition, service operation, continual service improvement |
| Human | Skill-needs definition, acquisition, training, skills use and maturation, retirement |
| Facility | Rendering, structure and site design, permitting, construction, operations and maintenance, retirement |
| Process | Output definition, process mapping, writeup, pilot use, use and improvement, retirement |
| Natural entity | Acquisition, development, exploitation, retirement |

#### 6.3.4.2 Adaptations for disciplines

In general, life cycle models do not need to be developed or adapted for a specific discipline (such as mechanical engineering, electrical engineering, civil engineering, quality, system administration). Instead, the processes associated with the life cycle model(s) in which that discipline is used are adapted to reflect the overall considerations discussed in 6.1, possibly with additional adaptations for the discipline itself. For example, the agreement processes associated with a facility can be adapted to the specific facility project. In addition, some details can be adapted to reflect civil engineering concepts, practices or terminology.

#### 6.3.4.3 Adaptations for specialties

##### 6.3.4.3.1 General

The focus of International Standards for systems and software is on the engineering, operation, maintenance and disposal of complex man-made products. A cluster of interests exists, generally called specialty engineering, which includes but is not limited to such areas as availability, maintainability, reliability, safety, human factors and usability. Within ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288, these "ilities" requirements are referred to as "critical quality characteristics." Each product can have critical quality characteristics that should be considered, with possible adaptations of specific processes, so that the product can be successful. These critical qualities, although requiring special knowledge and expertise in one area, generally cannot be evaluated in isolation from each other. For example, assessments of safety, human factors and environmental compatibility can need to be done as one integrated effort and that assessment can link to yet others. For specialty areas that are critical to the success of the product, each stage should have additional exit criteria tailored specifically for the level of planning, analysis, or verification that should be achieved in that specialty area by the end of each stage. Particularly for highly cost- or schedule-constrained efforts, if not closely monitored, these specialty areas are at risk of becoming sources of hidden technical debt. Some, though obviously not all, of the critical qualities are discussed below to illustrate adaptations for specialities. See D.4 and D.6 for additional information on process views for specialty engineering and security.

##### 6.3.4.3.2 Human

Human interaction with products and services associated with a system should be looked at from the perspective of impacts on operators, users, maintainers, support personnel and the general public, including ethics and cultural values. Impacts should be analysed to determine adverse impacts that should be avoided or mitigated through product related design requirements that can mitigate the adverse impacts identified.

It is important to get feedback from related stakeholders and end-users frequently to confirm that the system meets their needs and to improve the system.

NOTE ISO/IEC/IEEE 42010 provides additional information on how user/human views affect requirements and design. ISO/IEC/IEEE 24748-7000 addresses processes to realize human values in system requirements and design.

#### 6.3.4.3.3 Health

Planned usage rates and environments, operational concepts and other requirements can present health risks with respect to potential damage to human life including operators and others (people and animals) that come in contact with the product or exist within its operational environment. Use cases, operating environments, electromagnetic radiation, heat and noise emissions, waste materials and potential failure modes should be analysed to determine such risks. Outcomes from such analyses should include specific health concerns and recommendations related to design requirements that can prevent the health hazards identified.

NOTE Health issues can persist after the product is retired.

#### 6.3.4.3.4 Safety

Operational concepts and other product requirements can present safety risks with respect to potential damage to human life, property and the environment. Use cases, human-machine interfaces, operating environments, electromagnetic radiation, heat and noise emissions and potential failure modes should be analysed to determine such risks. Outcomes from such analyses should include specific safety concerns and recommendations related to design requirements that can prevent the safety hazards identified.

#### 6.3.4.3.5 Security and privacy

Operational concepts, usage environments and other product requirements can present security risks with respect to the product and its users. Risks include:

a) access and damage to personnel, properties and information;

b) corruption, theft or compromise of sensitive information;

c) denial of approved access to property and information;

d) unauthorized system access;

e) loss of life or property.

Applicable areas of security should be analysed to include physical security, communications security, computer security and electronic emissions security. Outcomes from such analyses should include specific security concerns and recommendations related to design requirements that can mitigate the security risks identified.

#### 6.3.4.3.6 Interoperability

Data flows are common in many types of systems that have controls as well as in most software systems. The potential failure causes of data (or information) to not flow properly should be analysed to include use of appropriate communication connectivity protocols, standardized interfaces and frequent regression testing upon software and hardware integration. Outcomes from such analyses should include specific interoperability concerns and recommendations related to design requirements that can improve interoperability.

#### 6.3.4.3.7 Usability

Systems should be designed to accommodate users with the widest range of capabilities while delivering effective, efficient, trustworthy, and satisfying results. Systems that include human elements depend on operators of the product performing tasks within specified times and required accuracies and with efficient and effective resource utilization. Use cases, human-machine interfaces, operating environments and training and operating procedures should be defined based on targets for usability such as understandability, learnability, operability and attractiveness, and evaluated against quality of use criteria, such as effectiveness, productivity, safety and satisfaction as proposed by ISO 9241-210. Outcomes from such analyses should include specific usability concerns and recommendations related to design requirements that can improve usability.

#### 6.3.4.3.8 Dependability

Dependability is the characteristic of systems that operate when needed, without failure or unscheduled down time. Measures of dependability include mean-time-between-failures (MTBF) and mean-time-to-repair (MTTR). Such factors should be analysed to determine their impact over the product life. Outcomes from such analyses should include specific dependability concerns and recommendations related to design requirements that can help make the product more dependable.

NOTE When dependability of the SoI is addressed in association with open systems dependability, application of the following four process views from IEC 62853 can be considered: change accommodation, accountability, failure response and consensus building.

#### 6.3.4.3.9 Environmental impacts

The impacts on the environment from short and long-term use of a product and disposing of hazardous materials related to its use or retirement, can present risks to all life forms. Risks to the general public include loss of life, illness and lowering of the standard of living. Environmental impacts as a result of use or disposal of waste products (e.g. by-products, materials) from product use or from disposal of the product or one of its elements that have reached end-of-life, should be analysed. Environmental assessments should be made as early as possible. Outcomes from such analyses can be included in an environmental assessment and should include specific environmental impact concerns and recommendations related to design requirements that can reduce risks related to product use and eventual disposal.

## 6.4 Evaluation-related activities

Persons who are involved in any activity of the life cycle of a project or a process can conduct evaluations either on their own or on other's products and activities. This document groups these evaluations into five categories, which are listed below. The first four evaluation categories are at the project level; the last one is at the organizational level. These evaluations should be selected and adapted proportional to the scope, magnitude, complexity and criticality of the project or of the organization. The problem, non-conformance and improvement reports from these evaluations feed back into one or more of the life cycle processes.

a) Process-internal evaluations are conducted by personnel performing the assigned tasks within the process during their day-to-day activities.

b) Verification and validation are conducted by some combination of the acquirer, the supplier, or an independent party, to verify and validate the products in varying depth depending on the project. These evaluations do not replace other evaluations but supplement them.

c) Joint reviews and audits are conducted in a joint forum by the reviewing and reviewed parties to evaluate status and compliance of products and activities on a pre-agreed-to schedule. Additional guidance on joint reviews is given in Annex F.

d) Quality assurance can be conducted by personnel independent of project management. The goal is to assure conformance of the products and processes with the contract requirements and adherence to the established plans. This process can use the results from a), b) and c) as inputs. This process can coordinate its activities with those of a), b) and c).

e) Improvement of processes and of activities like monitoring and reviews can be a follow-on to evaluations. Process improvement should be conducted to achieve the acceptable or desired level of quality in processes and products for an organization or its projects and contracts This is conducted regardless of project or contract requirements.

## 7 Relationship with detailed process standards

This document facilitates the use of the process content of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 by providing unified and consolidated guidance on life cycle management of systems and software. Adaptations include incorporating lessons learned during the life cycle management process through a feedback mechanism into one or more other processes within or across other system and/or specialty disciplines to improve system design and use. Further guidance on the application of the processes of each applicable International Standard comes from other International Standards and documents.

ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 define a generic, top-level framework based upon a set of processes that can be combined into various suitable life cycle models. These documents do not, and are not intended to, define in detail the systems engineering, software engineering or the engineering of systems. However, the documents are expected to strengthen the relationships among systems engineering, software engineering and other affected engineering disciplines. They are intended to do this through provision of a consistent life cycle process model, process purposes and outcomes, and terminology among the various domains. They are also intended to establish interactions and improve communication between the various engineering disciplines needed to create systems.

Other International Standards can address either domains (e.g. aircraft, cloud services) in detail, or treat specific disciplines or specialties (e.g. ISO 9241-220). They can be used as the basis for building applicable sets of life cycle processes that provide activities to achieve a stated goal. The processes defined by such documents are likely to be invoked during the whole life cycle of the system.

International Standards are, in turn, supported by application guidance, such as ISO/IEC/IEEE 24748-2 for systems and ISO/IEC/IEEE 24748-3 for software. This document thus provides spanning guidance in two key areas of interest across domains and disciplines, complemented by the specific conformance and guidance documents for each area. A more detailed view of the emphasis given each area (e.g. process definitions) by ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 and their related documents is shown in Table 3.

**Table 3 — Overview of coverage and emphasis among International Standards and Guidance Documents**

| Area | ISO/IEC/IEEE 15288 | ISO/IEC/IEEE 12207 | ISO/IEC/IEEE 24748-1 | ISO/IEC/IEEE 24748-2 | ISO/IEC/IEEE 24748-3 |
|---|---|---|---|---|---|
| Process definitions | Systems engineering and common | Software engineering | General overview: what a process is and pointer to standards | n/a | n/a |
| Life cycle concepts | Summary | | Detail | n/a | n/a |
| Life cycle stages | Summary | | Detail | n/a | n/a |
| Life cycle tailoring | Process requirements | | General guidance | Specific detail for systems engineering | Specific detail for software engineering |
| Life cycle application/usage | n/a | | General guidance | Domain-specific guidance | Domain-specific guidance |
| Life cycle model examples/illustrations | n/a | | General guidance | Domain-specific examples | Domain-specific examples |
| Terminology | Systems engineering | Software engineering | Life cycle and pointer to standards | As needed | As needed |
| System process key concepts | Summary | n/a | n/a | Detail in systems context | n/a |
| Software process key concepts | n/a | Summary | n/a | n/a | Detail in software context |

**Table 3** *(continued)*

| Area | ISO/IEC/IEEE 15288 | ISO/IEC/IEEE 12207 | ISO/IEC/IEEE 24748-1 | ISO/IEC/IEEE 24748-2 | ISO/IEC/IEEE 24748-3 |
|---|---|---|---|---|---|
| Organization/ project application | Summary | | Summary in life cycle context | Detail in systems context | Detail in software context |
| Process application | Summary | | Summary in life cycle context | Detail in systems context | Detail in software context |
| Process tailoring | Normative requirements | | Summary in life cycle context | Example for systems | Example for software |
| Process reference model | Detail | | General description and pointer to standards | n/a | n/a |
| Specialty applications | Summary | | Summary in life cycle context | Detail in systems context | Detail in software context |
| Conformance | Included | | n/a | n/a | n/a |

# Annex A
(informative)

# Process concepts

## A.1 General

### A.1.1 Overview

Application of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 presupposes an understanding of process concepts.

NOTE 1 Process concepts are introduced in ISO/IEC/IEEE 12207:2017, 5.5, and ISO/IEC/IEEE 15288:2023, 5.6.

NOTE 2 ISO/IEC/IEEE 24774 provides guidance on process descriptions, which have been applied to the process descriptions used ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207.

The focus of ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 is on the processes that are applied within a life cycle. The processes can be used by organizations (e.g. functional organizations and projects) that play the role of acquirer, supplier (e.g. main contractor, subcontractor or service provider) or management to fulfil responsibilities pertaining to the SoI. Additionally, the processes in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 can be used as a reference model for assessments under the ISO/IEC 33000 family of standards.

A process is an integrated set of activities that transform inputs (e.g. a set of data such as requirements) into desired outputs (e.g. a set of data describing a desired solution). Controls and enablers are associated with processes. These relationships are illustrated in Figure A.1 and described in A.1.2 through A.1.5.

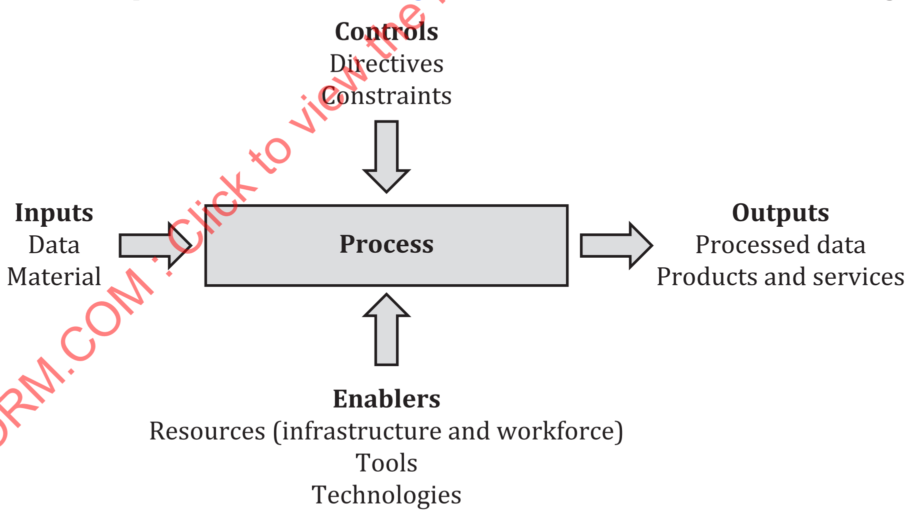



**Figure A.1 — Example process inputs and outputs**

An activity is a set of cohesive tasks. A task is a requirement, recommendation or permissible action, intended to contribute to the achievement of one or more process outcomes.

A task is expressed in the form of a requirement, self-declaration, recommendation or permissible action. See ISO/IEC/IEEE 12207:2017, 4.1, NOTE 5 and ISO/IEC/IEEE 15288:2023, 4.1, NOTE 4 for the verbal forms used to differentiate between for forms of tasks.

Within a life cycle stage, processes are performed as required to achieve stated objectives. The progression of a system through its life is the result of actions managed and performed by people in one or more organizations using the processes selected for a life cycle stage.

### A.1.2 Inputs

Inputs can come from outside an organization or project, or from other processes that precede or accompany the process being examined. Examples of inputs to a process include:

a) information, such as requirements, interface or architecture definitions;

b) data, such as measurements and test reports;

c) material that either ends up in the output or is consumed in producing the output;

d) services that are part of a chain of services, such as setting up a computer prior to, or coincident with establishing an account.

### A.1.3 Outputs

Outputs can go to other processes or back to the same process (recursive processing) inside the organization, project (or both), or they can go outside the project or organization, or both. Examples of outputs parallels the examples given for inputs in A.1.2. However, the outputs are often (but not necessarily) transformed in some way by the process being examined.

### A.1.4 Controls

Processes can be controlled by organizational or organization management directives and constraints and by governmental regulations and laws. Examples of such controls on a process include:

a) the project agreement;

b) the interfaces with processes used on other systems for which the project is responsible (see 5.6.3);

c) the applicable system life cycle stage or stages;

d) internal standard practices of the organization, or the part of the organization, that has project responsibility.

### A.1.5 Enablers

Each process can have a set of process-enabling mechanisms including:

a) the workforce that performs the tasks related to the process;

b) other resources required by the process such as facilities, equipment and funds;

c) tools (e.g. software and hardware, automated, manual) required for performing the process activities;

d) technologies required by persons performing the activities including methods, procedures and techniques.



## A.2 Process application

The processes defined in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 can be used by any organization when acquiring and using, as well as when creating and supplying, software or a system. They can be applied at any level in a software's or a system's structure and at any stage in the life cycle.

The life cycle processes are based on principles of ownership (a process is associated with a responsibility, discussed further in B.2) and modularity. That is, the processes are the following:

— strongly cohesive, meaning that all the parts of a process are strongly related;

— loosely coupled, meaning that the number of interfaces among the processes is kept to a minimum.

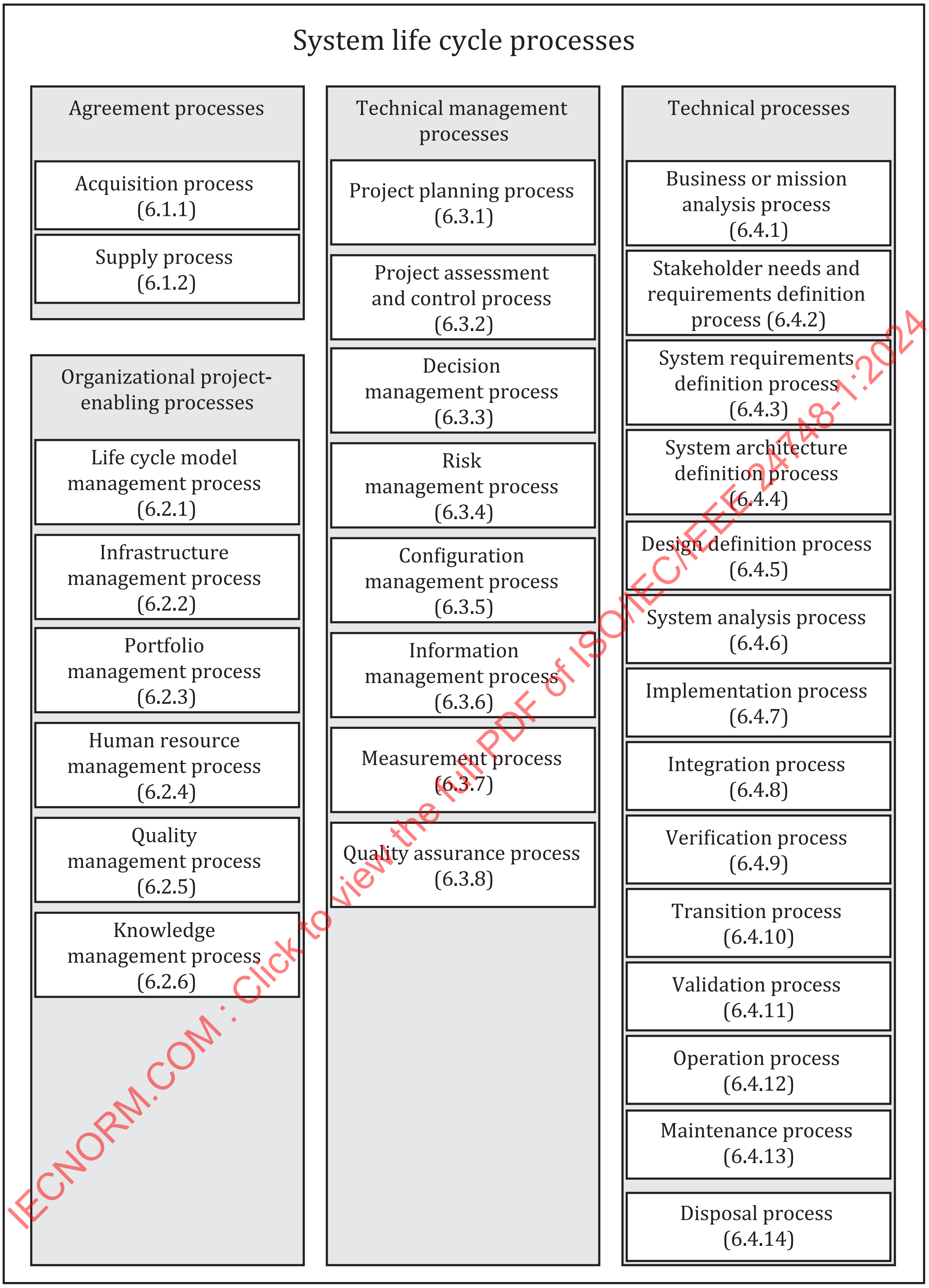
The processes described in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 are not intended to preclude or discourage the use of additional processes that organizations find useful.

Organizations, when considering a new project, should select a life cycle model and the necessary life cycle processes to satisfy applicable life cycle stage entry or exit criteria. Decisions as to which processes to select should be based on cost-benefit or risk reduction. Within a life cycle stage, processes are performed as required to achieve stated objectives. The progression of a system through its life is the result of actions managed and performed by people in one or more organizations using the processes selected for a life cycle stage.

ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 provide a specific example of four groups of system life cycle processes: agreement, organizational project-enabling, technical management and technical. Each process has a specific purpose, a set of expected outcomes and a set of activities and tasks. Each group of processes is described in ISO/IEC/IEEE 15288:2023, Clause 6 or ISO/IEC/IEEE 12207:2017, Clause 6 and summarized in A.3.

NOTE The process groups in ISO/IEC/IEEE 15288:2023, and ISO/IEC/IEEE 12207:2017 are identical. There are differences between the two standards at lower levels within a given group (i.e. purpose, outcomes, activities and tasks). Those differences do not affect the discussion of project groups in this subclause.

Each system life cycle process in Figure A.2 can be invoked, as required, at any point, and at multiple points, throughout the life cycle and there is no definitive order or time sequence in their use. 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 system. An individual system life cycle is thus a complex system of processes with concurrent, iterative, recursive and time dependent characteristics. These statements are true for a SoI and any enabling systems.






NOTE 1 In Figure A.2, the subclause numbers refer to the subclauses in ISO/IEC/IEEE 15288:2023, and ISO/IEC/IEEE 12207:2017 where the processes are described, not to the subclauses in this document.

NOTE 2 The title of ISO/IEC/IEEE 12207:2017, Figure 4 is "Software life cycle processes" and the process in ISO/IEC/IEEE 12207:2017, 6.4.3 is titled "System/Software requirements definition process".

**Figure A.2 — System life cycle processes in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207**

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

The changing nature and complexity of the influences on the 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 is thus dynamic, responding to the many external influences on the system.

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. Precedence, particularly in similar market and product sectors can assist the selection of stages and the application of life cycle processes to build an appropriate and effective life cycle model for any system.

## A.3 Process groups of ISO/IEC/IEEE 15288:2023 and ISO/IEC/IEEE 12207:2017

The four process groups of ISO/IEC/IEEE 12207:2017 and ISO/IEC/IEEE 15288:2023, as well as the primary relationships between the groups, are portrayed in Figure A.3. The role of the organizational project-enabling and technical management groups of processes is to achieve the project goals within applicable life cycle stages to satisfy an agreement. Organizational project-enabling processes provide enabling resources and infrastructure that are used to create, support and monitor projects and to assess project effectiveness. The technical management processes help ensure adequate planning, assessment and control activities are performed to manage processes and life cycle stages.

Appropriate processes are selected from the technical processes and used to populate projects in order for the project to perform life cycle related work.

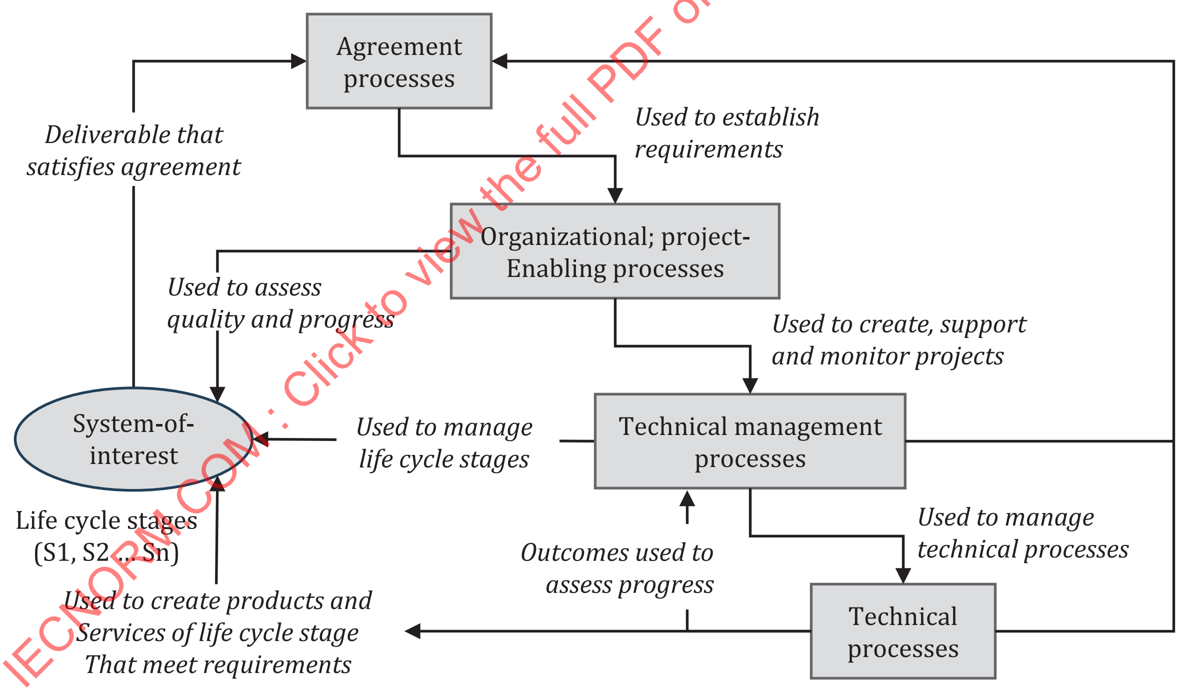



**Figure A.3 — Role of the ISO/IEC/IEEE 15288:2023 and ISO/IEC/IEEE 12207:2017 processes**

Projects can need to establish relationships with other projects within the organization, as well as those in other organizations. Such relationships are established through the agreement processes of acquisition and supply as shown in Figure A.4. The degree of formality of the agreement is adapted to the internal or external business relationships between projects.

NOTE An example and discussion of the use of the agreement processes is provided in ISO/IEC/IEEE 24748-2:2024, 6.7.2.

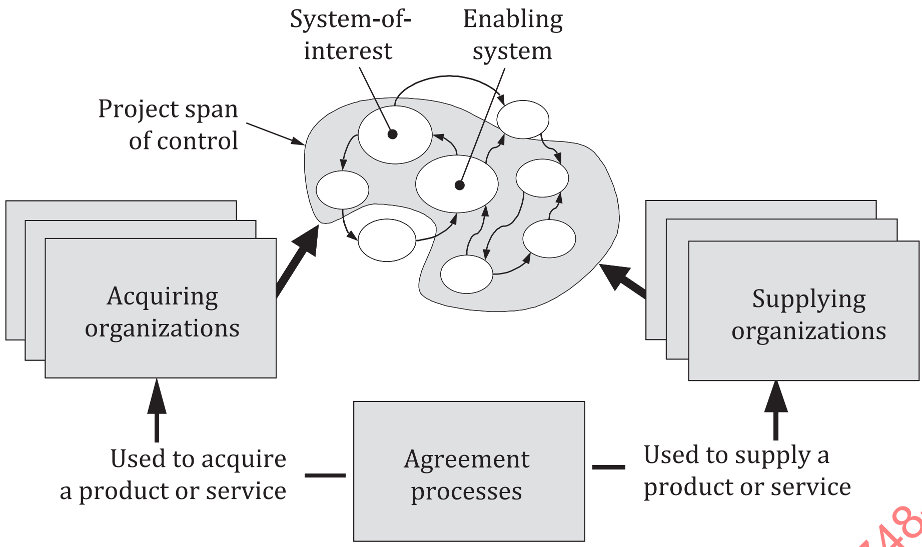



**Figure A.4 — Use of agreement processes**



## A.4 Agreement processes

The agreement processes are applicable for establishing the relationship and requirements between an acquirer and supplier. The agreement processes provide the basis for initiation of other project processes to enable arriving at an agreement to conceive, develop, produce, utilise, support or retire a system and to acquire or supply related services.

The agreement processes can be used for several purposes, such as:

a) to form and specify completion of an agreement between an acquirer and a supplier for work on a system at any level of the system structure;

b) to establish and carry out agreements to acquire a system or related enabling system services;

c) to obtain work efforts by consultants, subcontractors, organizations, projects or individuals or teams within a project;

d) to provide the basis for closing an agreement after the system has been delivered or work has been completed and payment made.

## A.5 Organizational project-enabling processes

Organizational project-enabling processes are for that part of the general management that is responsible for establishing and implementing projects related to the products and services of an organization. Thus, the organization through these distinct processes provides the services that constrain and enable the projects, directly or indirectly, to meet their requirements.

The organizational project-enabling processes included in International Standards, such as ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 are not necessarily the only processes used by an organization for governance of its business. For example, organizations also have processes for managing accounts receivable, accounts payable, payroll processing and marketing. These business-related processes are not directly within the scope of the mentioned standards and thus are not discussed further in this document. The organizational project-enabling processes of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 are constrained in their scope to the aspects of those processes that are required to bound and guide the project, even though there are implicit relationships that the organization should address elsewhere, for example, in the human resource management process.

For multiple projects involved in or interfacing with an organization or for a teaming arrangement among external organizations, other organizational project-enabling processes can be appropriately tailored.

To perform these processes, it is not intended that a new organizational unit or discipline within an organization be created. Identified and defined roles, responsibilities and authorities can be assigned to individuals or established organizational units. When necessary, a new organizational unit can be formed.

The organizational project-enabling processes have specific objectives to fulfil:

a) provide the proper environment so that projects within the organization can accomplish their purpose and objectives;

b) establish an orderly approach to starting, stopping and redirecting projects;

c) define organizational policies and procedures that set forth the relevant life-cycle processes of the International Standards that are applicable to projects within the organization and its constituent parts;

d) select and provide appropriate models, methods and tools to projects so that they can complete process activities efficiently and effectively;

e) provide adequate resources for the project to meet budget, schedule and performance requirements within acceptable risks and train human resources for completing their responsibilities;

f) deliver project work products of a suitable quality to customers;

g) retain knowledge acquired during the execution of a project in a form that is accessible for future needs.

## A.6 Technical management processes

The technical management processes are used to manage technical process activities and to assure satisfaction of an agreement. Technical management processes are performed to establish and update plans, to assess progress against plans and system requirements, to measure and control work efforts, to make required decisions, to manage risks and configurations and to capture, store and disseminate information. Outcomes from performing the technical management processes help in the accomplishment of the technical processes.

The technical management processes apply to technology exploration projects that are most often part of larger projects. When that is the case, the appropriate technical management processes are performed at each level of the system structure. These processes also apply when performing organizational project-enabling processes or carrying out the activities related to a life cycle stage, including utilization, support and retirement.

When several projects co-exist within one organization, technical management processes should be defined to allow for the management of the resources and performance of the multiple projects.

## A.7 Technical processes

The technical processes are applicable across all life cycle stages. For example, the following technical processes from ISO/IEC/IEEE 12207:2017 and ISO/IEC/IEEE 15288:2023 should be performed to engineer a system.

a) Business or mission analysis process

b) Stakeholder needs and requirements definition process

c) System requirements definition process

d) System architecture definition process

e) Design definition process

f) System analysis process

g) Implementation process

h) Integration process

i) Verification process

j) Transition process

k) Validation process

l) Operation process

m) Maintenance process

n) Disposal process

NOTE The titles of these processes are identical to those in ISO/IEC/IEEE 12207:2017 except for c), which is system/software requirements definition process, and d) which is architecture definition process in that document.

These processes should be performed to satisfy the entry or exit criteria of a system life cycle stage or set of stages. For example, they can be used during early system life cycle stages to create a feasible system concept, determine technology needs and establish future developmental costs, schedules and risks. During ensuing life cycle stages the technical processes can be used to define, realize and make use of a new system. During later system life cycle stages, they can be used on legacy systems to make technology refreshments or technology insertions, as well as to correct variations from expected performance during production, utilization, support or retirement. These processes apply to a SoI and its enabling systems.

The last three technical processes listed (operations process, maintenance process and disposal process) can be used during any system life cycle to accomplish the objectives of a life cycle stage and support the technical processes used for engineering a system. The operations process and the maintenance process can be performed, as applicable, to support a particular version of a system. The disposal process can be performed to deactivate legacy systems, to dispose of legacy systems and to safely dispose of by-products from system use.

There can be additional technical processes used for a specific technical domain within a system, such as software, hardware, humans or facilities.

## A.8 Processes under key views

ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 contain processes that are applicable throughout a life cycle. However, these processes can be used in different ways by different organizations and parties with different views and objectives. This subclause presents processes and their relationships under key views.

Figure A.5 depicts illustrative examples of the life cycle processes and their relationships under different views of the usage of either document. The basic views shown are contract, management, technical management, engineering, operating, support and project. Further discussion of views, their creation and use, is given in Annex D.

NOTE ISO/IEC/IEEE 24641:2023, Figure A.3 provides a high-level process view from the viewpoint of the model-based system and software engineering reference framework to show the interrelationships among the life cycle model defined in this document and the process groups described in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207.

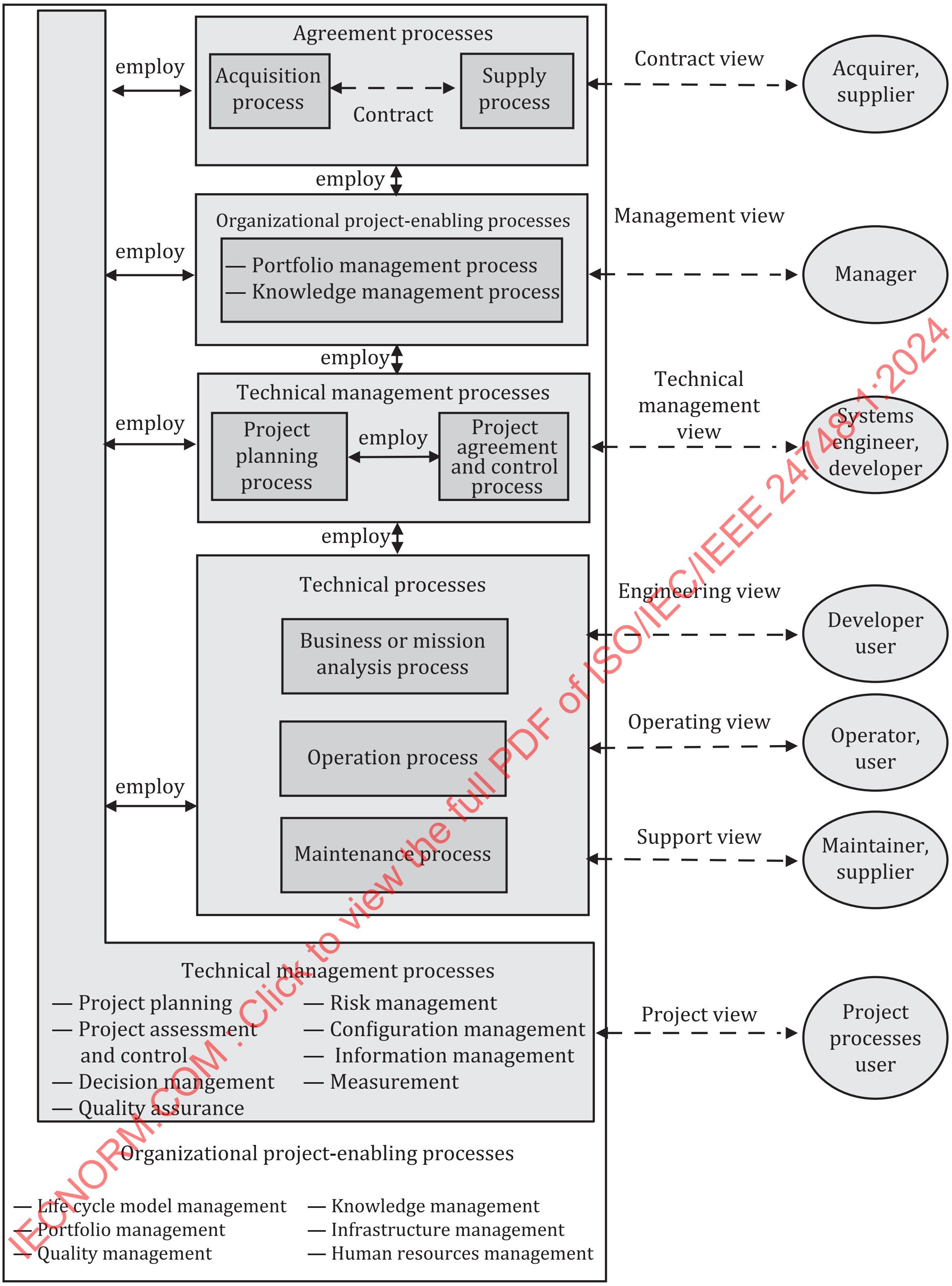



**Figure A.5 — Life cycle processes roles and relationships**

Under the contract view, acquirer and supplier parties negotiate and enter into a contract or other agreement and employ the acquisition process and supply process respectively. Under the management view, the organization establishes, uses and maintains a process framework that services multiple projects and complements their technical effort. Under the technical management view, the organization initiates, supports and controls projects. Under the engineering view, the developer or maintainer conducts its respective engineering tasks to produce or modify products or services. Under the operating view, the operator provides operation service for the users. Under the support view, the maintainers and suppliers apply parts and services to sustain and improve the operation of the system. Under the project view, parties (such as configuration management or information management) provide supporting services to others in fulfilling specific, unique tasks. Also shown (see the bottom box) are some of the organizational project-enabling processes provided by the organization to support multiple projects; these are employed by an organization at a higher, e.g. corporate, level to establish and implement an underlying structure made up of associated life cycle processes and personnel and their continuous improvement.

The processes and parties (or stakeholders) are only related functionally. They do not dictate a structure for an organization.

An organization gets its name from the process it performs, for example, it is called an acquirer when it performs the acquisition process.

An organization can perform one process or more than one process, a process can be performed by one organization or more than one organization. For example, under one contract or application of this document, a given organization can perform the acquisition process and the supply process.

In the documents themselves, the relationships between the processes are only static. The more important dynamic, real-life relationships between the processes, between the parties and between the processes and the parties are established when the appropriate document is applied on projects in a manner specific to that project. Each process (and the organization performing it) contributes to the project in its own unique way. The acquisition process (and the acquirer) contributes by defining the product or service (or both) to be obtained. The supply process (and the supplier) contributes by providing the product or service. The integration process (and the integrator) contributes by “looking” to the system for correct derivation and definition of software, hardware and other products, as well as services, by supporting proper integration of products and services back into an overall system. The operation process (and the operator) contributes by operating the products or providing the services in the system's environment for the benefit of the users, the business and the mission. The maintenance process (and the maintainer) contributes by maintaining and sustaining the products and services for operational fitness and by providing support and advice to the user community. Each process contributes by providing unique, specialized functions to other processes as needed.

Where processes are applied to system elements of a specific nature, the relevant processes from the standard that applies to that kind of element are invoked, as well as appropriate system processes. This holds true for elements of all types. Thus, there is a complementary duality of process interplay as the particulars of a system are instantiated. As a specific example, the validation process for software is invoked as part of the overall process of validating the system, to help ensure that the software is validated as part of the system.

All of these statements apply to enabling systems, as well as the SoI.

NOTE Process concepts are introduced in ISO/IEC/IEEE 15288:2023, 5.6.

The focus of ISO/IEC/IEEE 15288:2023 is on the processes that are applied within a life cycle. The processes can be used by organizations (e.g. functional organizations and projects) that play the role of acquirer, supplier (e.g. main contractor, subcontractor or service provider) or management to fulfil responsibilities pertaining to the SoI. Additionally, the processes in ISO/IEC/IEEE 15288:2023 can be used as a reference model for assessments under the ISO/IEC 33000 family of standards.



## A.9 Iterative, recursive, and concurrent application of processes

### A.9.1 General

Three forms of process application – iterative, recursive and concurrent – are essential and useful for executing the requirements of ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288. The iterative use of processes, i.e. the repeated application of a process or set of processes at the same hierarchical level of structural detail, 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. The recursive use of processes, i.e. the repeated application of the same process or set of processes applied to successive levels of detail in a system's hierarchical structure is a key aspect of the application of ISO/IEC/IEEE 15288 or ISO/IEC/IEEE 12207. Concurrent application of processes is when processes are applied in parallel or at the same time. From a high-level perspective, the three application approaches are shown in Figure A.6. The outputs of processes at any level, whether information, artefacts or services are inputs to the same processes used at the level above or level below. This results in a response, information, artefact or service, which can then modify the original output. In this way, the outputs across all levels of the system can be resolved and consistency achieved, e.g. system element descriptions that conform to an architecture.

NOTE The names of the process groups in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 as shown in Figure A.6 do not match the names of stages in most life cycle models.

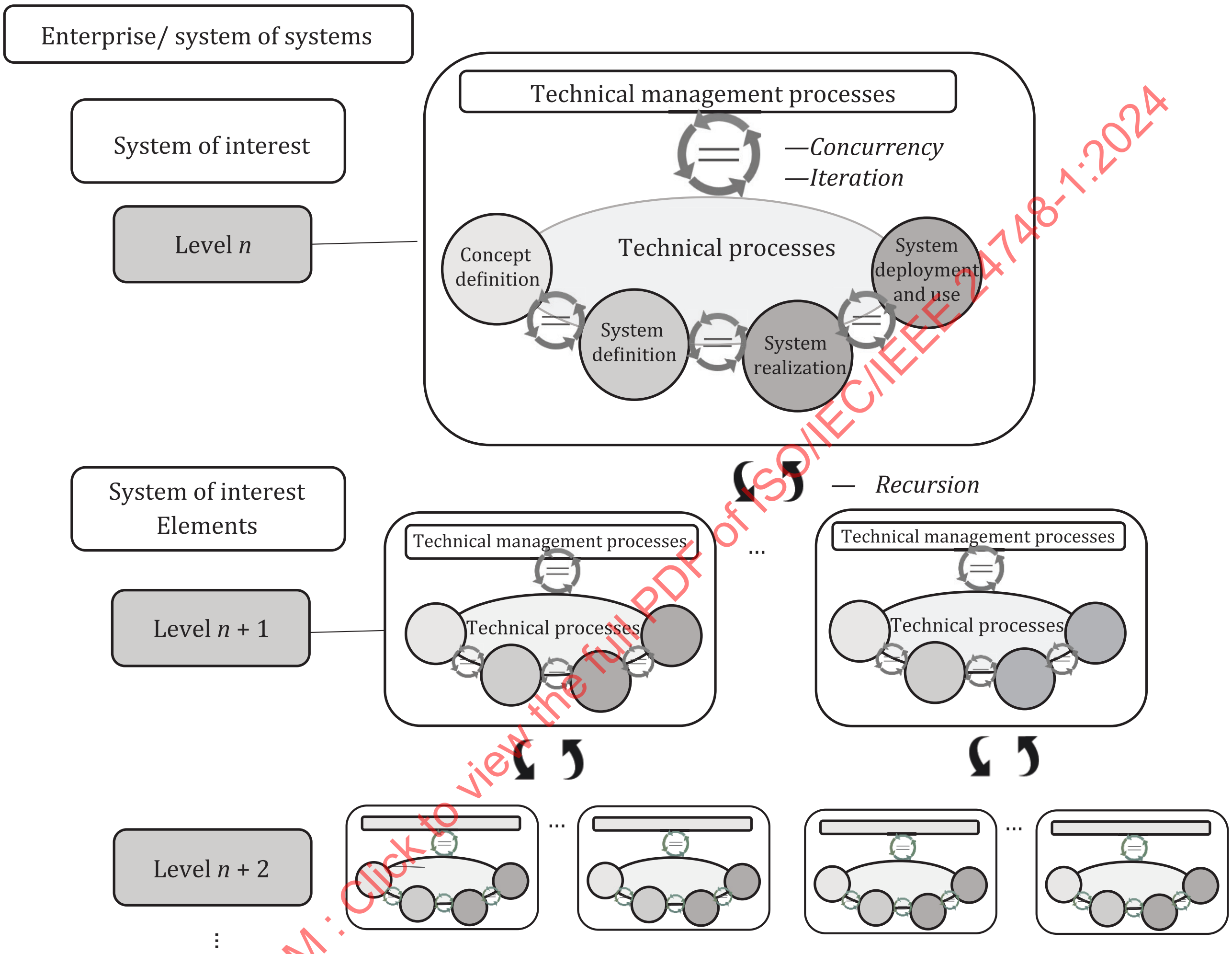



**Figure A.6 — Iteration, recursion, and concurrent process application**



### A.9.2 Iterative application of processes

When the application of the same process or set of processes is repeated on the same system, at the same level, the application is referred to as iterative. The iterative application of processes is illustrated in Figure A.7.

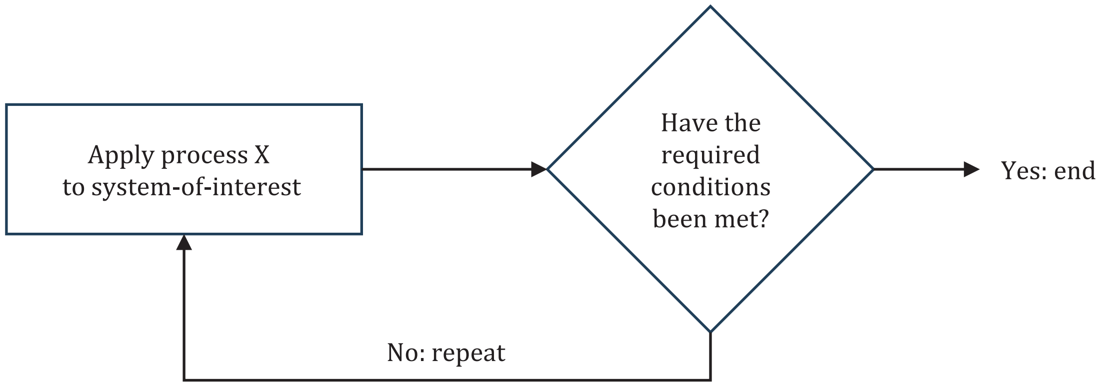



**Figure A.7 — Iterative application of process(es)**

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, analysed risks or opportunities. Such questions should be resolved before completing the activities of a process or set of processes. When re-application of activities or processes can resolve the questions, then it is useful to do so. Processes should be repeated until an acceptable quality of results is obtained prior to applying the next process or set of activities to a SoI. In this case iteration adds value to the system to which the processes are being used.

### A.9.3 Recursive application of processes

When the same set of processes or the same set of process activities are applied to successive levels of system elements within the system structure until some condition is satisfied, the application form is referred to as recursive. Figure A.8 illustrates the recursive application of processes to systems. The most common applications of recursion are in computer programs, mathematics and language.

The following describes how X is applied recursively to a system element, where:

- S is the system-of-interest, which is also considered as a system element of itself, and
- E and E′ vary in system elements

```
Apply X to S considered as a system element of itself.
   where
      Apply X to E is defined by {
        … Apply some activities of X to E; …
        if (E is not small enough [or low level enough] {
            for each sub-element E′ of E
            Apply X to E′;
     }
        …Apply the rest of activities of X to E; …
     }
```

Pseudocode example for a factorial function:

```
function factorial is:
input: integer n such that n >= 0
output: [n × (n-1) × (n-2) × … × 1]
1.  if n is 0, return 1
2.  otherwise, return [ n × factorial(n-1) ]
end factorial
```

**Figure A.8 — Recursive application of processes**

In systems development, the system life cycle technical processes are applied recursively at successive levels of the system structural hierarchy.

### A.9.4 Concurrent application of processes

When the same set of processes are conducted in parallel or at the same time, the application form is considered concurrent. It is not necessary for processes to be performed serially, especially when one process is not dependent on another for information or results. For example, the risk management process and measurement process usually are performed in a continual and concurrent manner.

## A.10 Methods and tools

In practice, there are many situations where system size and complexity, project duration and the number of contributing organizations require process execution to be supported by methods and tools.

NOTE See ISO/IEC/IEEE 24641 for additional information on model-based approaches to life cycle management.

The selection of methods and tools depend on many factors including stage in the life cycle, level in the system's hierarchy and application domain. As a result, neither ISO/IEC/IEEE 12207, ISO/IEC/IEEE 15288, nor this document includes discussions of specific methods and tools. Nevertheless, there are some issues that the user of ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288 should bear in mind when selecting and using methods and tools to accomplish life cycle process activities or related tasks. Five such issues are listed below.

a) A method or tool should not dictate the process to be followed but should support the set of activities of a selected process. Methods should be selected to fit the system life cycle stage.

b) Selection of tools should be based on multiple factors, including cost, capabilities, as well as on connectivity to other tools that provide inputs or use outputs of the tool being considered for use. The engineering data produced should be in an appropriate form to enable the data to be captured, stored and available as long as it is needed. Those members of organizations, projects and other stakeholders who have the need should be given access authority to the data.

c) The training requirements for application of the method or tool should be considered. The initial, as well as subsequent training time after a user has not used the tool for a period of time, should be included in the consideration.

d) Enabling systems as well as tool administration should be considered.

e) Methods and tools have their own life cycles. Alignment between the life cycle stages of the methods, tools and the SoI should be considered.

# Annex B
(informative)

# Organizational concepts

## B.1 General

An organization is a body of entities organized for some specific purpose and can be as diverse as a corporation, agency, society, union or club. An organization can be part of a parent organization (e.g. a society being parent to clubs that are a part of it). When an organization enters into an agreement, it is a party. Parties can have the same parent organization, although it is equally possible that they have different parent organizations (e.g. two clubs from the same society making an agreement or two clubs from two different societies making an agreement). The following explanation on the relationship between organizations and projects should be understood together with the structure of projects detailed in Annex C.

For the purpose of ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288, any project is assumed to be conducted within the context of one or more organizations. This is important because a system 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, ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 provide a set of organizational project-enabling processes. The organizational project-enabling processes are not assumed to be adequate to operate a business, nor are the technical management processes assumed to be adequate to operate a project. Instead, the processes, considered as a collection, are intended to state the minimum set of dependencies that the project places upon the organization.

ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 describe the set of processes that comprise the life cycle of any human-made system. Therefore, ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 are designed so that they can be tailored for a project of any type, size and complexity, whether focused on tangible products, services or a mix of both.

The sequence of the processes, activities and tasks in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 does not dictate the sequence of their application in a life cycle model. It is intended that the project select, order, tailor and iterate the processes, activities and tasks as applicable or appropriate.

On any given project, ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 can be separately applied more than once. For example, in a given system implementation project, an acquirer can request a supplier to perform system implementation, with the acquirer and the supplier executing one application of ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288. The supplier can then request its sub-contractor to perform all or part of the system development, e.g. develop the software. In both situations, it is necessary to adapt ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288 to reflect the arrangements.

NOTE 1 4.2.3 provides more detail on structure in systems and projects.

NOTE 2 4.2.4 provides more detail on enabling systems.

NOTE 3 ISO/IEC/IEEE 16326 provides more information on projects and project management.

NOTE 4 In the context of SoS, see ISO/IEC/IEEE 21840:2019, Clause 5, for key concepts and application to SoS, including differences between systems and SoS, and managerial and operational independence.

## B.2 Process responsibility

Typically, organizations distinguish different areas of managerial responsibility and action through agreements, as indicated in Figure B.1. Different organizations (or parts thereof) as parties, and different areas of responsibility within an organization, mutually establish their working relationships and acknowledge their respective responsibilities through such agreements. These agreements unify and coordinate the contributions made by different parties in order that they can meet a common business purpose.

Together, these agreements contribute to the organization's overall capability to trade. This document employs a process model based on three primary organizational areas (or levels) of responsibility: organization, project and technical. Within each organization, a coordinated set of organizational project-enabling, technical management and technical processes contribute to the effective creation and use of systems, and therefore to achieving the organization's goals.

An organization can perform one or more processes. A process can be performed by one organization or more than one organization. The execution of individual processes, activities, and tasks can be by different parties with responsibilities and accountability, as stated in agreements. The responsibility aspect of the life cycle architecture facilitates adaptation and application of International Standards, such as ISO/IEC/IEEE 15288 or ISO/IEC/IEEE 12207 on a project, in which many parties can be legitimately involved.

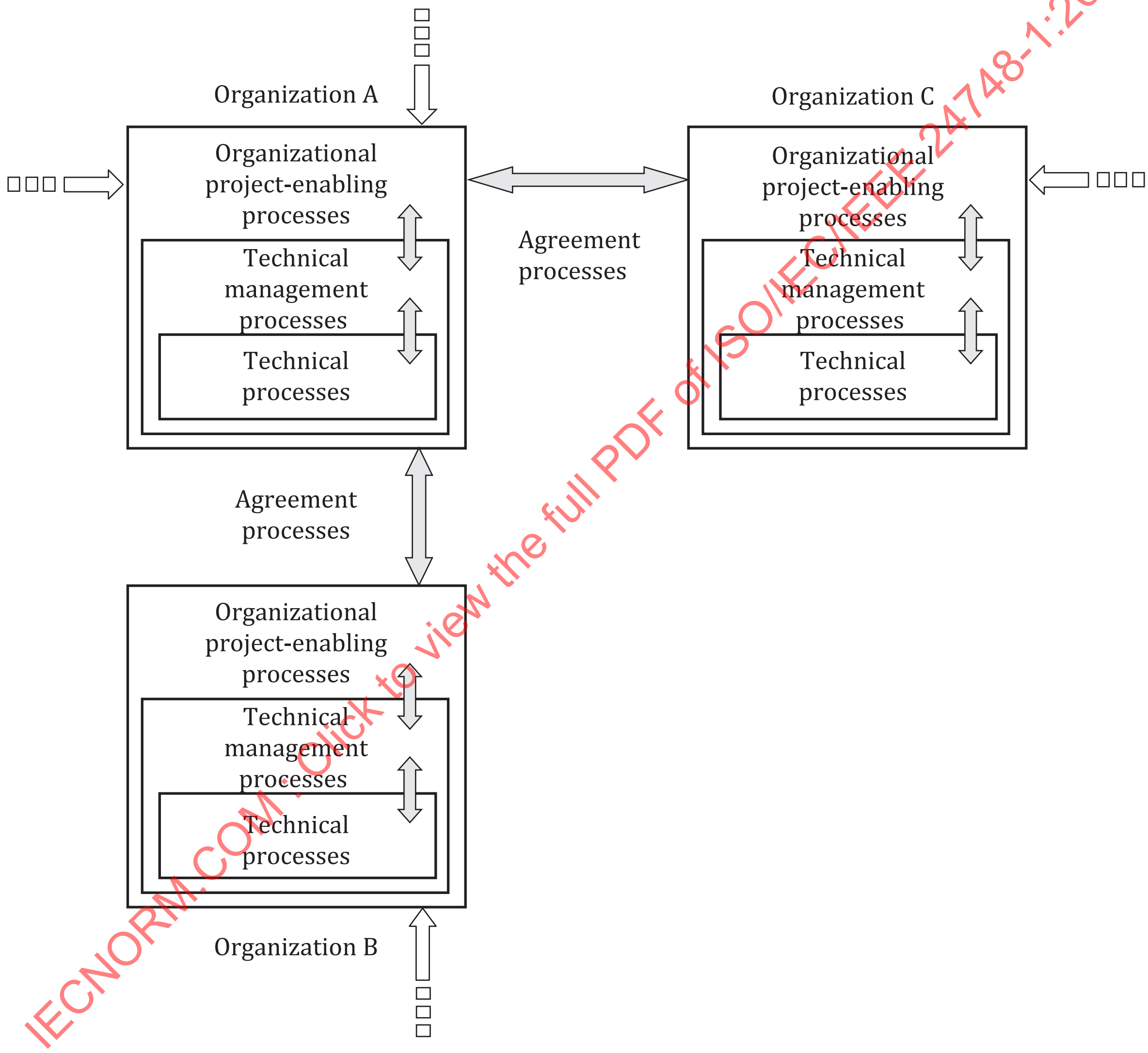



**Figure B.1 — Agreements on responsibilities for organizational project-enabling, technical management and technical processes among cooperating organizations**

# Annex C
(informative)

# Project concepts

## C.1 Structure in systems and projects

The scope of responsibility that an organization assigns to its projects is related to a number of factors:

a) complexity of the system or software engineering effort;

b) typical span of control and staffing levels compared to the norm for the organization;

c) expected duration of the effort;

d) participation and direct support from organizational process-enabling groups.

In short, the complexity of the system design is one factor among many determining the number and types of projects associated with the system engineering effort. Accordingly, each system in the structure illustrated in Figure 2 (or Figure 3) can be the responsibility of a separate project. This can be true whether the system is hierarchical, as used for the illustration, or other system structure, such as a network, or a mix of structures. The point is that there can be (and typically is) a strong correlation between levels of detail in the system structure and levels of responsibility in a set of projects. Each project characteristically has responsibility for acquiring and using system elements subordinate to it and creating and supplying to the level superior to it, as can be the case for hierarchical, non-hierarchical, or mixed, system structures.

Any particular project normally views its system as the SoI and whilst it can influence higher system levels, it does not have responsibility for them. However, even though it does not necessarily have responsibility for each system element considered by itself, it does have responsibility for all system elements that constitute its SoI and consequently for the output of projects at all levels subordinate to it, as shown in Figure C.1.

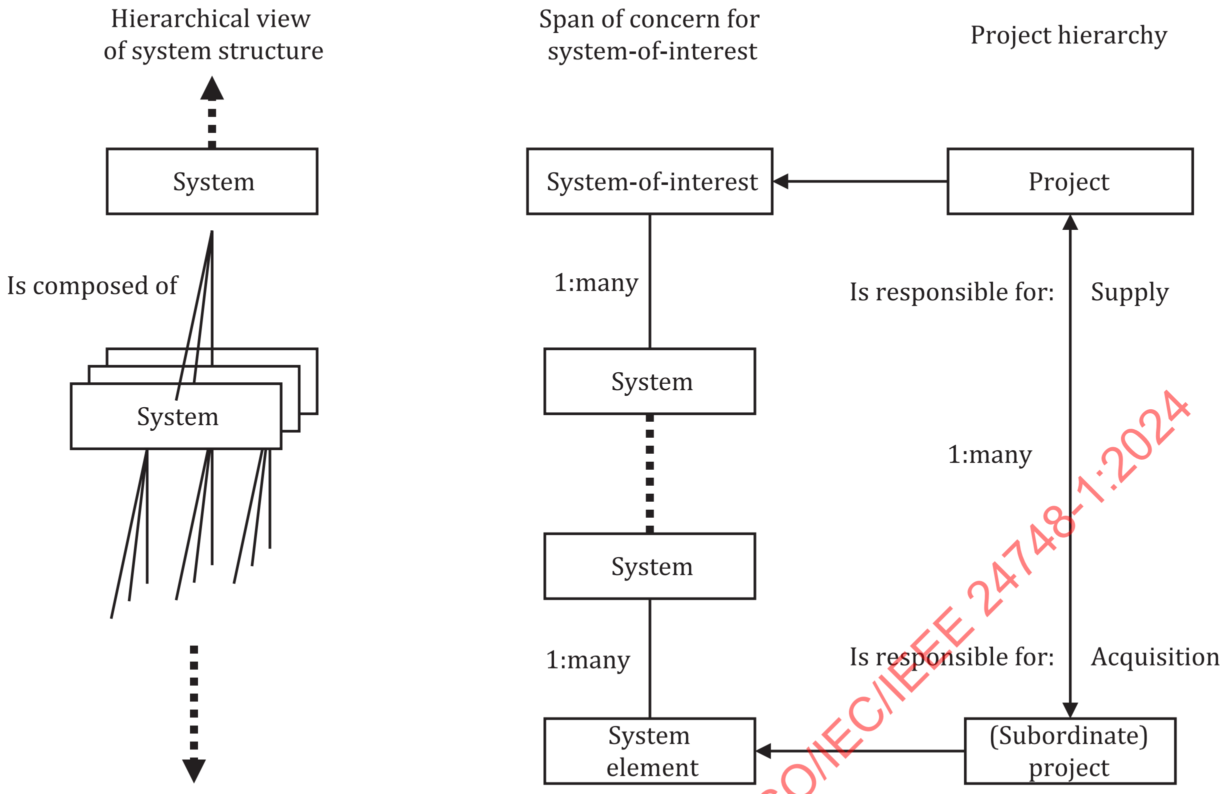



**Figure C.1 — System and project hierarchies**

In practice, the risks associated with implementing systems that fulfil specified requirements typically diminish with descending level of detail in the SoI's structure and eventually are no longer of direct attention or concern to the particular project. At this level (not necessarily the same level down different paths of SoI decomposition), a system element can be acquired with acceptable risk and the detail of its composition can remain hidden below some level. For example, if the SoI for the project were a radar, in normal practice, the formulation and cure of the slurry mix for the resistive part of a composition resistor on a printed circuit board in a subassembly in a system element of the SoI are not addressed directly. Rather, the requirements at the SoI and possibly the system element level can drive choices that can force a particular selection of this material in order to meet all the requirements for the higher-level elements. From the SoI view, the system elements can appear to be where specialist disciplines or particular implementation technology practices are present.

## C.2 Project relationships

A relationship can exist between a project and other projects, and subprojects. A subproject as used in Figure C.2 is a set of resources and tasks organized to undertake a portion of a project. A subproject can be considered a project by those assigned the work. Figure C.2 illustrates typical relationships between projects, which can be within a single organizational entity, or across entities. In this example, Organization B has internal agreements to partition 2 projects into subprojects. Additionally, it has agreements with Organizations A and C for subprojects to support Organization B's projects.

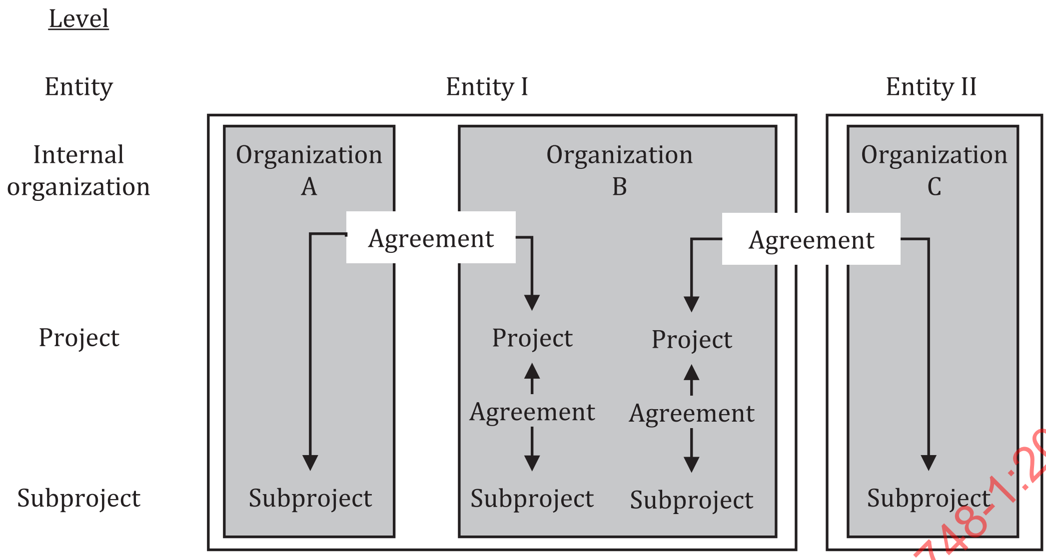



**Figure C.2 — Relationships between projects**



Project relationships are established, maintained and changed through formal or informal agreements in accordance with organizational policies and procedures, as appropriate. Depending on the type of project relationship involved, agreements can exist within a single organization, or can span organizational boundaries. Relationships, and therefore the establishing agreements, can be between a project and a specific organizational element or elements, among multiple projects or among a project and its subprojects. Agreements provide a mutual understanding of the problem to be solved, work to be done, established constraints, deliverables and clearly defined responsibilities and accountability.

Another kind of relationship, and agreement, not shown in Figure C.2, can be applicable when two or more organizations cooperate on a single project. In this case, it is important to define each organization's authorities, responsibilities and rights, including the sharing of proprietary information applicable to the project in the agreement.

Regardless of the kind of relationship, with its corresponding agreement, there is some basic information needed to do the work required in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288. Each agreement, whether formal or informal, should carefully consider, and address as appropriate, the activities and tasks of the agreement processes. The following additional aspects should be considered.

a) Responsibilities for the work expected to be done can be in the form of work statements or internal charters, but in any event, should be defined for all entities involved.

b) Products, services and data, created within the project can be formalized as deliverables either to another part of the organization or to external entities. In such cases, the means of delivering and accepting these deliverables should be made explicit.

c) technical or management reviews can be required within the project, or other parts of the organization, to support the reviews formalized in the agreement. Such reviews need to be driven by specific criteria for their initiation, completion and documentation.

NOTE A model is provided in ISO/IEC/IEEE 24748-2:2024, 6.7.2 for the application of ISO/IEC/IEEE 15288:2023 processes to reach an agreement.

## C.3 Enabling system relationships

Another relationship among projects is one that involves enabling systems. The project is responsible for ensuring that required enabling systems are available when needed to fulfil the functions of the SoI or enable the SoI to be realized. Some or possibly all enabling systems can be outside the direct responsibility (boundary) of the project. Some or all of the enabling systems can be needed for subsequent projects and stages and possibly throughout the system life cycle. Thus, the project should take appropriate actions to ensure their availability for use. Other enabling systems can be easily made available, for example by rental or purchase. If any enabling systems do not exist, they should be created and be made available in time to provide required services.

It is within the project's span of interest that not only the SoI should appropriately be made available but also all enabling systems that are needed for the project. Potentially, some or all of these enabling systems can be needed for subsequent projects and stages and possibly throughout the system life cycle. The project should identify any enabling systems that are needed and take appropriate actions to ensure their availability for use at the appropriate times.

Agreements should be established between the project and the internal or external organization or organizations, as applicable, to ensure that specified enabling system services are provided when needed. A project span of interest is illustrated in Figure C.3.

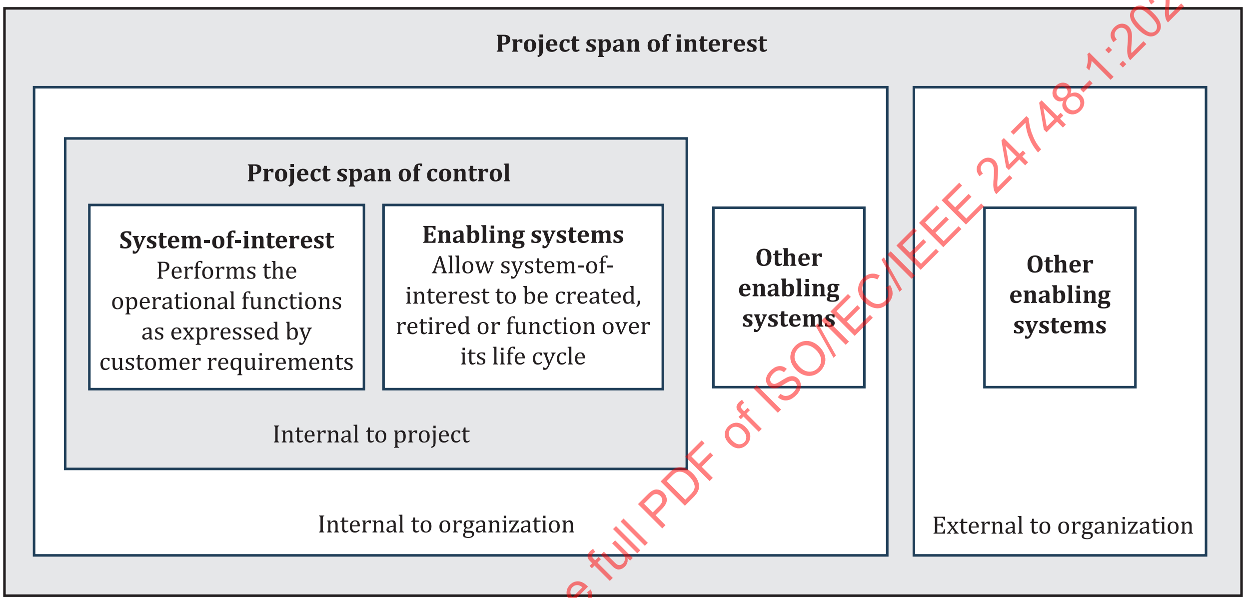



**Figure C.3 — Project span of interest**

## C.4 Hierarchy of projects

The system that the project is responsible for is considered a SoI. Each subordinate or sub-project is considered as a project itself. A resultant structure of projects can then be formed. The basic relationship of Figure C.3, which illustrates the project span of interest, can be combined with the hierarchical view of a system structure as is portrayed in more detail in Figure C.4.

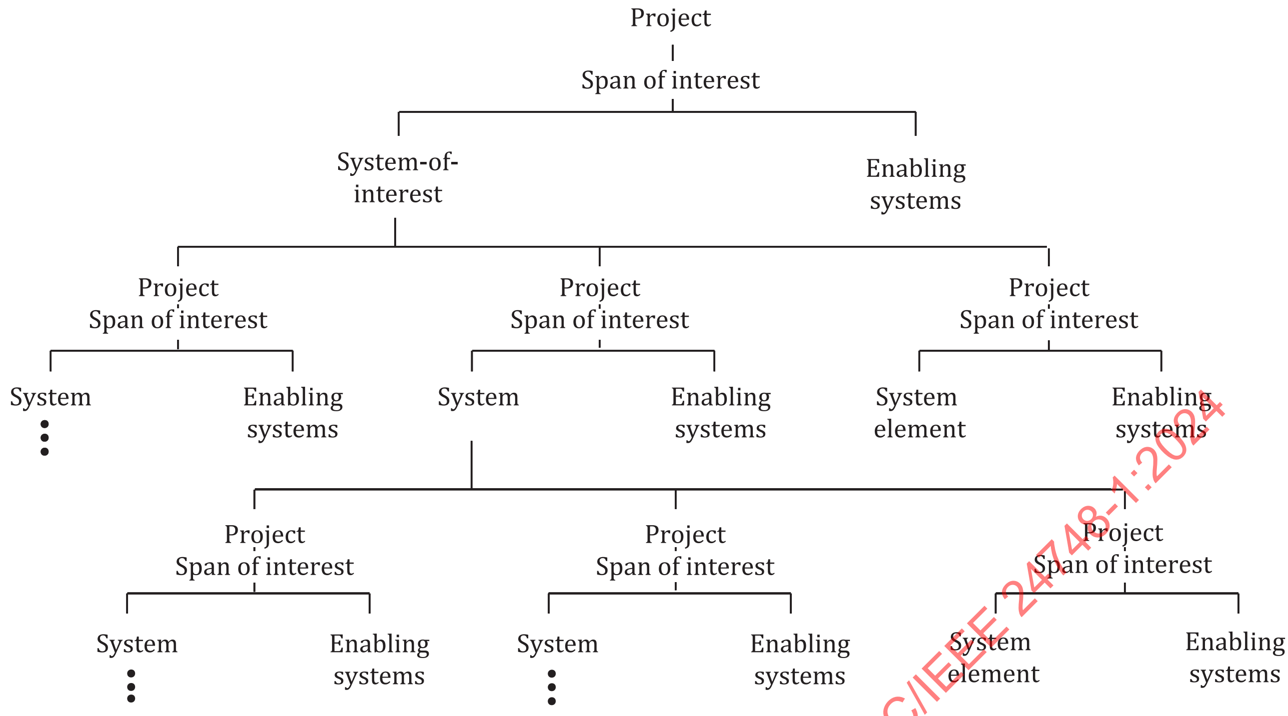



**Figure C.4 — Hierarchy of projects**

Figure C.4 shows only the lower level of projects of one system. Each system, however, should be decomposed into lower-level projects until each consists of only a system element and its enabling systems. Two such projects in Figure C.4 end with a system element. Each project should be carried out using applicable system life cycle processes to the extent required by requirements and to satisfy applicable life cycle stage entry or exit criteria. The hierarchical organizational model for projects aligned with systems is less applicable to projects with a specialty engineering function (e.g. security, safety, testing, infrastructure services), or across multiple subsystems; for software; or for continuing integration, operations and maintenance work.

As explained in Figure C.3, the enabling systems of Figure C.4 can be under project control or, if external to the project, under the control of other organizations. However, the project should work with these other organizations through agreements to ensure that the required enabling systems are available when needed to support the SoI during its life cycle.

# Annex D
(informative)

# Process views

## D.1 General

There are instances where those representing a particular engineering interest can benefit from a selected set of process activities that directly and succinctly address their concern. For such interests, a process view can be developed to organize processes, activities and tasks selected from ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 to provide a focus to their particular concern in a manner that cuts across all or parts of the life cycle. This Annex provides a process viewpoint that can be used to define process views in these instances and gives examples from the viewpoints of the following:

a) specialty engineering;

b) interface management;

c) security.

## D.2 The process view concept

There can be cases where a unified focus is needed for activities and tasks that are selected from disparate processes to provide visibility to a significant concept or thread that cuts across the processes employed across the life cycle. It is useful to advise users of the standards on how to identify and define these activities for their use, even though they cannot locate a single process that addresses their specific concern.

For this purpose, the concept of a process view has been formulated. Like a process, the description of a process view includes a statement of purpose and outcomes. Unlike a process, the description of a process view does not include activities and tasks. Instead, the description includes guidance explaining how the outcomes can be achieved by employing the activities and tasks of the various processes in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288. Process views can be constructed using the process viewpoint template found in D.3.

## D.3 Process viewpoint

A process view conforms to a process viewpoint. The process viewpoint provided here can be used to create process views.

The process viewpoint is defined by:

— its stakeholders: users of the standard;

— the concerns it frames: the processes needed to reflect a particular engineering interest.

The contents of resulting process views should include the following:

— process view name;

— process view purpose;

— process view outcomes;

— identification and description of the processes, activities and tasks that implement the process view, and references to the sources for these processes, activities and tasks in other standards.

NOTE The requirements for documenting viewpoints can be found in ISO/IEC/IEEE 24774:2021, 6.2 and 6.3.

## D.4 Process view for specialty engineering

### D.4.1 Specialty engineering process view overview and purpose

This subclause provides an example of applying the process viewpoint to yield a process view for specialty engineering to illustrate how a project can assemble processes, activities and tasks of ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 to provide focused attention to the achievement of product characteristics that have been selected as being of special interest.

This example treats the cluster of interests, generally called specialty engineering, which includes but is not limited to such areas as availability, maintainability, reliability, safety, human factors and usability. Within ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288, these "ilities" requirements are referred to as "critical quality characteristics". These characteristics determine how well the product meets its specified requirements in a specific area selected for focus.

NOTE This is a generalized instance of a process view that covers a broad set of functional and non-functional characteristics related to specialty engineering. It provides a broad view across the processes. If a specific critical quality characteristic has a high priority relative to other characteristics, a specific process view can be created for that characteristic, including more detailed information and requirements.

Name: Specialty engineering process view

Purpose: The purpose of the specialty engineering process view is to provide objective evidence that the system achieves satisfactory levels of certain critical quality characteristics selected for special attention.

### D.4.2 Specialty engineering process view outcomes

The outcomes for the specialty engineering process view are:

a) Product critical quality characteristics are selected for special attention.

b) Requirements for the achievement of the critical quality characteristics are defined.

c) Measures for the requirements are selected and related to the desired critical quality characteristics.

d) Approaches for achieving the desired critical quality characteristics are defined and implemented.

e) The extent of achievement of the requirements is continually monitored.

f) The extent of achievement of the critical quality characteristics are specified and developed.

NOTE The outcomes permit the possibility that the desired critical quality characteristics cannot be directly measured but instead can be argued and inferred based on other product or process characteristics that can be measured.

### D.4.3 Specialty engineering process view processes, activities and tasks

This process view can be implemented using the following processes, activities and tasks from ISO/IEC/IEEE 12207:2017 and ISO/IEC/IEEE 15288:2023.

NOTE 1 ISO/IEC 25030 can be useful in specifying software product quality requirements.

NOTE 2 INCOSE Systems Engineering Handbook contains descriptions and elaboration about many of the specialty engineering areas and the associated critical quality characteristics.

NOTE3 Cross-references in this list are to clauses/subclauses in ISO/IEC/IEEE 15288:2023 and ISO/IEC/IEEE 12207:2017, not in this document.

a) The agreement processes (6.1) provide for the establishment of expectations and responsibilities related to each specialty engineering area, including legal agreements and licensing requirements. Generally, this is done in the context of establishing and maintaining agreement processes for all aspects of the system. In this case, or even when separate agreement process application is required for a specialty engineering area, all parts of the acquisition process (6.1.1) and supply process (6.1.2) can be applicable.

b) The organizational project-enabling processes (6.2) 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. Generally, this is done in the context of applicability of the processes from all viewpoints to one or more systems across multiple projects. However, specialty engineering stakeholders should review all parts of the six organizational project-enabling processes to assure that their interests are adequately addressed.

c) The project assessment and control process (6.3.2) provides for monitoring the extent of achievement of the requirements and critical quality characteristics and communicating the results to stakeholders and managers. Relevant activities and tasks include (b) (6), (b) (7), (b) (9) and (b) (10).

d) The decision management process (6.3.3) provides assessment of alternative requirements, architecture characteristics and design characteristics against the decision criteria, including the critical quality characteristics. Results of these comparisons are ranked via a suitable selection model and are then used to decide on an optimal solution. Relevant activities and tasks include (b) all tasks; and (c) (1).

e) The risk management process (6.3.4), in its entirety, provides for identifying, evaluating and handling risks of the system, including those related to meeting the critical quality characteristics.

f) The configuration management process (6.3.5) manages and controls system elements and configurations over the life cycle. Relevant activities and tasks include (a) (1) and (d) (1) for each discipline to be addressed under specialty engineering.

g) The information management process (6.3.6), in its entirety, provides for the specification, development and maintenance of information items for documenting and communicating the extent of achievement. Information items used for the purpose of critical quality characteristics are sometimes specialized in nature. Sources for the description of these information items include industry associations, regulators and specific standards.

h) The measurement process (6.3.7), in its entirety, provides for defining an approach that relates measures to the required critical quality characteristics.

i) The quality assurance process (6.3.8) addresses identified anomalies (incident and problems) that relate to the achievement of critical quality characteristics.

j) The business or mission analysis process (6.4.1) provides for the definition of the problem space and characterization of the solution space, including the relevant trade-space factors and preliminary life cycle concepts. This includes developing an understanding of the context and any key parameters, such as the critical quality characteristics (e.g. safety hazards, human interfaces, operational characteristics and system assurance context). Relevant activities and tasks include (b) (1) and (b) (2); (c) (1); (d) (1); and (e) (2).

k) The stakeholder needs and requirements definition process (6.4.2) provides for the selection and definition of characteristics, including critical quality characteristics, and associated information items. The activities and the documentation are useful in identifying, prioritizing, defining and recording requirements for the critical quality characteristics. Relevant activities and tasks include (a) (1); (b) (2), (b) (4) and (b) (6); (c) all tasks; (d) all tasks; (e) (2); and (f) (3).

l) The system requirements definition process (6.4.3) provides for the specification of parameters for the critical quality characteristics and the selection of measures for tracking the achievement of these requirements with respect to the specific system to be developed. Relevant activities and tasks include (a) (1); (b) all tasks; (c) (2); and (d) (3).

NOTE 4 The title of process 6.4.3 is software requirements definition process in ISO/IEC/IEEE 12207:2017 and system requirements definition process in ISO/IEC/IEEE 15288:2023,

m) The system architecture definition process (6.4.4) provides for the identification of stakeholder concerns from an architecture perspective. These concerns often translate into expectations or constraints across the life cycle stages that relate to the critical quality characteristics, such as utilization (e.g. availability, security, effectiveness, usability), support (e.g. reparability, obsolescence management), evolution of the system and of the environment (e.g. adaptability, scalability, survivability), production (e.g. manufacturability, testability), retirement (e.g. environmental impact, transportability), etc. This process further addresses those critical quality characteristic requirements that drive the architecture decisions, including the assessment of the architecture with respect to the concerns and associated characteristics. Relevant activities and tasks include (b) (1); (c) (1); (c) (2), (c) (3), (c) (4) and (c) (5); (d) (2); and (e) (2).

n) The design definition process (6.4.5) provides for the determination of necessary design characteristics, which includes the critical quality characteristics, such as security of design criteria for the specialty characteristics and the evaluation of alternative designs with respect to those criteria. Relevant activities and tasks include (a) (3); (b) (1), (b) (3), (b) (4), (b) (5) and (b) (7); (c) (2); and (d) (4).

o) The system analysis process (6.4.6) provides for the level of analysis needed to understand the trade space with respect to the critical quality characteristics through the conduct of mathematical analysis, modelling, simulation, experimentation and other techniques. The analysis results are input to trades made through the decision management process in support of other technical processes. Relevant activities and tasks include (a) all tasks; and (b) all tasks.

p) The implementation process (6.4.7) provides for recording the evidence that critical quality requirements have been met. Relevant activities and tasks include (b) (3).

q) The integration process (6.4.8) provides for planning the integration, including the considerations for critical quality characteristics, and the assurance that the achievement of the characteristics is determined and recorded. Relevant activities and tasks include (a) (1); (b) (5); and (c) (1).

r) The verification process (6.4.9), provides for the planning and execution of a strategy to perform verification, including the critical quality characteristics. The selected verification strategy can introduce design constraints that can affect the achievement of the characteristics. Relevant activities and tasks include (a) (1) and (3); (b) (1), (b) (2); and (c) (1) and (c) (2).

s) The transition process (6.4.10) provides for installing the system in its operational environment. Because some specialty properties involve a trade-off between design constraints and operational constraints, attention to installation is often important. Relevant activities and tasks include (a) (4); and (b) (4), (b) (6) and (b) (7).

t) The validation process (6.4.11) provides evidence that the services provided by the system meet the stakeholders' needs, including the critical quality characteristics. Relevant activities and tasks include (a) (1) and (a) (3); (b) (1) and (b) (2); (c) (1) and (c) 2).

u) The operation process (6.4.12) provides for usage of the system. Assuring that critical quality characteristics are appropriately achieved involves monitoring the operation of the system. Relevant activities and task include (b) (3), (b) (4) and b (5); (c) (1) and (c) (2); and (d) (1) and (d) (2).

v) The maintenance process (6.4.13) sustains the capabilities of the system, helping to ensure its ongoing availability to provide its functions, including its critical quality characteristics. This includes failure analysis, maintenance tasks and logistics tasks needed to assure continued operation of the system. Relevant activities and tasks include (b) all tasks; (c) all tasks; and (d) (1) and (d) (2).

w) The disposal process (6.4.14) ends the existence of a system. The inherent need to anticipate disposal can place constraints on development. In fact, these constraints can themselves be critical quality characteristics. Relevant activities and tasks include (a) (2); (b) (1) and (b) (2); and (c) (3).

## D.5 Process view for interface management

### D.5.1 Interface management process view overview and purpose

This subclause provides an example of applying the process viewpoint to yield a process view for interface management, intended to illustrate how a project can assemble processes, activities and tasks of ISO/IEC/IEEE 15288:2023, to provide focused attention to the achievement of product characteristics that have been selected as being of special interest.

This example treats a specific instance of a process view, called interface management, which includes but is not limited to interface definition, design and change management. Within ISO/IEC/IEEE 15288:2023, the tasks that comprise interface management are fully contained within the existing processes.

Name: Interface management process view

Purpose: The purpose of the interface management process view is to facilitate of the identification, definition, design and management of interfaces of the system.

### D.5.2 Interface management process view outcomes

The outcomes for the interface management process view are:

a) Business or mission needs related to interfaces are identified.

b) Stakeholder needs related to interfaces are identified.

c) Requirements for the interfaces are defined.

d) Interfaces between system elements, as well as interfaces between the system and external systems are identified and defined.

e) Approaches for achieving the desired interface characteristics are defined and implemented.

f) The extent of realization of the interface requirements is continually monitored.

g) The extent of achievement of the interface requirements are specified and developed.

### D.5.3 Interface management process view processes, activities and tasks:

This process view can be implemented using the following processes, activities and tasks from ISO/IEC/IEEE 15288:2023:

NOTE 1 INCOSE Systems Engineering Handbook contains descriptions and elaboration about interface management.

NOTE 2 Cross-references in this list are to clauses/subclauses in ISO/IEC/IEEE 15288:2023 and ISO/IEC/IEEE 12207:2017, not in this document.

a) The agreement processes (6.1) provide for the establishment of expectations and responsibilities related to interface management, including legal agreements and licensing requirements. Generally, this is done in the context of establishing and maintaining agreement processes for all aspects of the system. In this case, or even when separate agreement process application is required, such as for a specific interface, all parts of the acquisition process (6.1.1) and supply process (6.1.2) can be applicable.

b) The organizational project-enabling processes (6.2) 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. Generally, this is done in the context of applicability of the processes from all viewpoints to one or more systems across multiple projects. However, interface management stakeholders should review all parts of the six organizational project-enabling processes to assure that their interests are adequately addressed.

c) The project assessment and control process (6.3.2) provides for monitoring the extent of achievement of the requirements, including interfaces, and communicating the results to stakeholders and decision makers. Relevant activities and tasks include (b) (6), (b) (7), (b) (9) and (b) (10).

d) The decision management process (6.3.3) provides assessing alternative requirements, architecture characteristics and design characteristics against the decision criteria, including the interfaces. Results of these comparisons are ranked, via a suitable selection model and are then used to decide on an optimal solution. Relevant activities and tasks include (b) all tasks; and (c) (1).

e) The risk management process (6.3.4), in its entirety, provides for identifying, evaluating and handling risks of the system, including those related to interfaces.

f) The configuration management process (6.3.5) manages and controls system elements and configurations over the life cycle. All activities and tasks in this process are particularly relevant to interface management.

g) The information management process (6.3.6), in its entirety, provides for the specification, development and maintenance of information items for documenting and communicating the extent of achievement.

h) The measurement process (6.3.7), in its entirety, provides for defining an approach that relates measures to the required interface information needs, and then generating and using those measures to address the identified interface information needs.

i) The quality assurance process (6.3.8) addresses identified anomalies (incident and problems) that relate to the achievement of interface requirements. The business or mission analysis process (6.4.1) provides for the definition of the problem space and characterization of the solution space, including the description of the environment and context, as well as preliminary operational concepts. It often identifies external systems that must interface with the SoI. Relevant activities and tasks include (b) (1) and (b) (2); and (c) (1).

j) The business or mission analysis process (6.4.1) provides for the definition of the problem space and characterization of the solution space, including the relevant trade-space factors and preliminary life cycle concepts. This includes developing an understanding of the context and any key parameters, such as the critical interfaces within the SoI, as well as those to external systems, and similarly for all enabling systems. Relevant activities and tasks include (b) (1) and (b) (2); (c) (1); and (d) (1).

k) The stakeholder needs and requirements definition process (6.4.2) provides for the definition of operational concepts and the interactions of the system with users and the intended environment (including other systems). It often identifies external systems that must interface with the SoI. Relevant activities and tasks include (b) all tasks; (d) (1) and (d) (3).

l) The system requirements definition process (6.4.3) provides for the definition of the interface requirements. Relevant activities and tasks include (a) (1); (b) all tasks; (c) all tasks; and (d) all tasks.

m) The system architecture definition process (6.4.4) provides for the identification of interfaces from an architecture perspective as the architecture models evolve. This process further describes and defines the interfaces to the extent needed for the architecture description. Relevant activities and tasks include (a) (2) and (a) (4); (b) (1) (d) all tasks; (e) (2) and (e) (5) and (e) (6).

n) The design definition process (6.4.5) provides for the refinement and full definition of the interfaces and the creation of the necessary information items. Relevant activities and tasks include (b) 6) and (b) (7); and (d) (2) through (d) (4).

o) The system analysis process (6.4.6) provides for the level of analysis needed to understand the trade space with respect to the interface requirements and definition through the conduct of mathematical analysis, modelling, simulation, experimentation and other techniques. The analysis results are input to trades made through the decision management process in support of other technical processes. Relevant activities and tasks include (a) all activities; and (b) all tasks.

p) The implementation process (6.4.7) provides for development of the interfaces and recording the evidence that interface requirements for an implemented system element have been met. Relevant activities and tasks include (b) (3).

q) The integration process (6.4.8) provides for planning the integration, including the considerations for interfaces between system elements. It also includes the integration of systems or system elements and interfaces. Relevant activities and tasks include (a) (1); (b) all tasks; and (c) (1).

r) The verification process (6.4.9), provides evidence that the services provided by the system meet the system requirements, including interface requirements. The process provides for the planning and execution of a strategy to perform verification, including the interface requirements. The selected verification strategy can introduce interface constraints that can affect their achievement. Relevant activities and tasks include (a) (1) and (a) (3); (b) (1), (2); and (c) (1).

s) The transition process (6.4.10) provides for installing the system in its operational environment. This includes identifying constraints and checking the installation and operational state of the interfaces. Relevant activities and tasks include (a) (4); and (b) (3), (b) (4), (b) (6) and (b) (7).

t) The validation process (6.4.11) provides evidence that the services provided by the system meet the stakeholders' needs, including the interface requirements and the internal and external interfaces. The selected validation strategy can introduce interface constraints that can affect their achievement. Relevant activities and tasks include (a) (1), (a) (2) and (a) (3); (b) (1) and (b) (2); (c) (1) and (c) (2).

u) The operation process (6.4.12) provides for usage of the system. There also can be constraints to the interfaces for operations. Assuring that the interface requirements are appropriately achieved involves monitoring the operation of the system. Relevant activities and task include (a) (2), (b) (3) and (b) (4); and (c) (1) and (c) (2).

v) The maintenance process (6.4.13) sustains the capabilities of the system, helping to ensure its ongoing availability to provide its functions, including its interfaces. This includes failure analysis, maintenance tasks and logistics tasks needed to assure continued operation of the system. There also can be constraints to the interfaces for maintenance. Relevant activities and tasks include (a) (3); (b) all tasks; and (d) (1) and (d) (2).

w) The disposal process (6.4.14) ends the existence of a system. It can require activities to disengage interfaces, including the termination of cooperative agreements with external systems involved with external interfaces to the SoI. The inherent need to anticipate disposal can place constraints on the interfaces. Relevant activities and tasks include (a) (2) and (b) (1) and (b) (2).

## D.6 Process view for security

### D.6.1 Security process view overview and purpose

This subclause provides an example of applying the process viewpoint to yield a process view for security, intended to illustrate how a project can assemble processes, activities and tasks of ISO/IEC/IEEE 15288:2023 to provide focused attention to the achievement of system characteristics that have been selected as being of special interest.

This example is focused on protection against intentional subversion or forced failure due to the architecture, design, implementation, operation, support, or disposition of any of the types of elements (hardware, software, data, facilities, etc.) that can comprise a system, including enabling systems, delivering any mix of services and products.

Name: Security process view

Purpose: The purpose of the security process view is to provide objective evidence that the system is capable of achieving satisfactory levels of certainty, that sufficient protection is achieved against intentional subversion or forced failure due to the architecture, design, implementation, operation, support, or disposition of any of the types of elements that can comprise a system, including enabling systems, delivering any mix of services and products.

### D.6.2 Security process view outcomes

The outcomes for the security process view are:

a) Product and service security characteristics are selected for special attention.

b) Requirements for the achievement of the security characteristics are defined.

c) Measures for the requirements are selected and related to the desired security characteristics.

d) Approaches for achieving the desired security characteristics are defined and implemented.

e) The extent of realization of the security requirements is continuously monitored.

f) The extent of achievement of the security characteristics is specified and developed.

NOTE The outcomes permit the possibility that the desired security characteristics cannot be directly measured, but instead can be inferred based on other product or process characteristics that can be measured.

### D.6.3 Security process view processes, activities and tasks

This process view can be implemented using the following processes, activities and tasks from ISO/IEC/IEEE 15288:2023:

NOTE Cross-references in this list are to clauses/subclauses in ISO/IEC/IEEE 15288:2023 and ISO/IEC/IEEE 12207:2017, not in this document.



a) The agreement processes (6.1) provide for the establishment of expectations and responsibilities related to each aspect of security, including legal agreements and licensing requirements. Generally, this is done in the context of establishing and maintaining agreement processes for all aspects of the system. In this case, or even when separate agreement process application is required for a specific security area, all parts of the acquisition process (6.1.1) and supply process (6.1.2) can be applicable.

b) The organizational project-enabling processes (6.2) 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. Generally, this is done in the context of applicability of the processes from all viewpoints to one or more systems across multiple projects. However, security stakeholders should review all parts of the six organizational project-enabling processes to assure that their interests are adequately addressed.

c) The project assessment and control process (6.3.2) provides for monitoring the extent of achievement of the requirements and critical quality characteristics and communicating the results to stakeholders and managers. Relevant activities and tasks include (b) (6), (b) (7), (b) (9) and (b) (10).

d) The decision management process (6.3.3), in its entirety, assesses alternative requirements, architecture characteristics and design characteristics against the decision criteria, including the security characteristics. Results of these comparisons are ranked, via a suitable selection model, and are then used to decide on an optimal solution. Stakeholders can make decisions based on the justification provided.

e) The risk management process (6.3.4), in its entirety, provides for the evaluation of the potential for not being able to achieve the necessary security for each system element type as well as the overall system, resulting in a risk to one or more stakeholders, or resulting in the system not being used as intended. Security is a risk category in each risk analysis.

f) The configuration management process (6.3.5) manages and controls system elements and configurations over the life cycles. Relevant activities and tasks include (a) (1), (d) (1), (e) in its entirety.

g) The information management process (6.3.6), in its entirety, provides the information about the achievement of security to the relevant stakeholders, including regulatory or approval authorities. The information about security is collected to support a set of arguments that justify a claim about the security of a system. Due to the sensitivity of the data, additional care is given to identify the appropriate audiences for the various security measures. Information management includes the protection of sensitive information items regarding security.

h) The measurement process (6.3.7), in its entirety, provides a common platform for collecting information about the security of each element of the system and each aspect of security.

i) The quality assurance process (6.3.8) evaluates project and supplier processes for conformance to security requirements and procedures. It addresses identified anomalies (incident and problems) that relate to the achievement of security characteristics. Relevant activities and tasks include (c) and (e) (all tasks).

j) The business or mission analysis process (6.4.1) provides for understanding the security environment for the SoI, considering relevant laws, policies, risks and constraints related to security. Relevant activities and tasks include (b) (1), (c) (1) and (d) (1).

k) The stakeholder needs and requirements definition process (6.4.2) provides for the selection and definition of risks and threats to missions or information and incorporates this knowledge in defining security related requirements, including confidentiality, availability and integrity in the context of how the system is intended to function, including consideration of misuse and abuse scenarios. Stakeholders need to agree upon what aspects of security are sufficient. Relevant activities and tasks include (a) (2), (b) all tasks, (c) (2) and (c) (4).

l) The system requirements definition process (6.4.3) provides for the selection and definition of security related requirements, including confidentiality, availability and integrity in the context of how the software is intended to function, including requirements for system security in the event of misuse and abuse. Relevant activities and tasks include (b) (3) and (b) (4) and (c) (3).

m) The system architecture definition process (6.4.4) provides for the identification of stakeholder concerns from an architecture viewpoint through threat modelling and assessing the vulnerability of the product architecture and design to potential attacks to gain an understanding of the threat landscape and the relevant architecture elements. Relevant activities and tasks include (a) (2) and (a) (4); (b) (1); (c) (all tasks); (d) (5) and (f) (1), (f) (2) and (f) (5).

n) The design definition process (6.4.5) provides sufficient detailed data and information about the system and its elements to enable the security implementation consistent with architectural entities as defined in models and views of the system architecture. Relevant activities and tasks include (a) (3) and (a) (4); (b) (3), (b) (4) and (b) (7); (c) (2) and (d) (2).

o) The system analysis process (6.4.6) provides for the level of analysis needed to understand the trade space with respect to the critical quality characteristics through the conduct of mathematical analysis, modelling, simulation, experimentation and other techniques. The analysis results are input to trades made through the decision management process in support of other technical processes. Relevant activities and tasks include (a) all tasks; and (b) all tasks.

p) The implementation process (6.4.7) provides for the use of security practices to avoid common errors that lead to exploitable system vulnerabilities, and the use of a variety of testing techniques including inspection for competent authenticity, penetration testing, fuzz testing, static analysis testing and dynamic testing to identify and address system weaknesses and vulnerabilities. Relevant activities and tasks include (a) (1); and (b) (1), (b) (2), (b) (3) and (b) (4).

q) The integration process (6.4.8) provides for planning the integration, including the considerations for security characteristics, and the assurance that the achievement of the characteristics is determined and recorded. Implementation of interface standards wherever practical promotes system and element sustainability and element reusability. Relevant activities and tasks include (a) (2) and (a) (3); (b) (5); and (c) (1) and (c) (2).

r) The verification process (6.4.9) provides for the planning and execution of a strategy to verify that system requirements, including the security characteristics, have been achieved. The selected verification strategy can introduce testing for security weaknesses in the development process or during sustainment. Threat analysis provides input into the creation of test plans and cases. The results include the information required to effect the remedial actions that correct nonconformances in the system and account for uncertainty in verification activities, such as test tool reliability and level of the uncertainty in results (i.e. rates of false positives and false negatives). Additional considerations include the resiliency of the system to function. Relevant activities and tasks include (a) (1), (a) (4), (a) (5) and (a) (6): (b) (all tasks); (c) (1) and (c) (2).

s) The transition process (6.4.10) establishes a capability for a system to provide services specified by stakeholder requirements in the operational environment. Relevant activities and tasks include (a) (2); (b) (1), (b) (4), (b) (5), (b) (6); and (c) (1) and (c) (2).