

AS/NZS ISO/IEC/IEEE 24748.1:2024
ISO/IEC/IEEE 24748-1:2022



Australian/New Zealand Standard™

Systems and software engineering — Life cycle management

Part 1: Guidelines for life cycle management



AS/NZS ISO/IEC/IEEE 24748.1:2024

This Joint Australian/New Zealand Standard™ was prepared by Joint Technical Committee IT-015, Software and Systems Engineering. It was approved on behalf of Standards Australia's Standards Development and Accreditation Committee on 24 June 2024 and by the New Zealand Standards Approval Board on 03 July 2024.

This Standard was published on 16 August 2024.

The following are represented on Committee IT-015:

- Australian Computer Society
- Australian Digital Health Agency
- Australian Society for Technical Communication (NSW)
- Business NSW
- Department of Defence (Australian Government)
- Engineers Australia
- Griffith University
- Institute of IT Professionals New Zealand
- International Council on Systems Engineering (INCOSE) New Zealand
- IT Service Management Forum Australia
- Systems Engineering Society of Australia
- Universities New Zealand
- University of New South Wales
- University of Southern Queensland
- University of Technology Sydney

This Standard was issued in draft form for comment as DR AS/NZS ISO/IEC/IEEE 24748.1:2024.

Keeping Standards up-to-date

Ensure you have the latest versions of our publications and keep up-to-date about Amendments, Rulings, Withdrawals, and new projects by visiting:

www.standards.org.au

www.standards.govt.nz

ISBN 978 1 76139 760 8

Australian/New Zealand Standard™

Systems and software engineering — Life cycle management

Part 1: Guidelines for life cycle management

Originated as SA/SNZ TR ISO/IEC 24748.1:2014.
Revised and redesignated as AS/NZS ISO/IEC/IEEE 24748.1:2022.
Second edition 2024.

COPYRIGHT

© ISO/IEC/IEEE 2024 — All rights reserved

© Standards Australia Limited/the Crown in right of New Zealand, administered by the New Zealand Standards Executive 2024

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher, unless otherwise permitted under the Copyright Act 1968 (Cth) or the Copyright Act 1994 (New Zealand).

Preface

This Standard was prepared by the Joint Standards Australia/Standards New Zealand Committee, IT-015 Software and Systems Engineering, to supersede AS/NZS ISO/IEC/IEEE 24748.1:2022.

The objective of this document is to provide guidance for the life cycle management of systems and software, complementing the processes described in AS/NZS ISO/IEC/IEEE 15288 and AS ISO/IEC/IEEE 12207.

This document —

- (a) 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;
- (b) 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;
- (c) defines the concept of a life cycle;
- (d) 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;
- (e) provides guidance on adapting a life cycle model and the content associated with a life cycle or a part of a life cycle;
- (f) describes the relationship between life cycles and their use in applying the processes in AS/NZS ISO/IEC/IEEE 15288 (systems aspects) and AS ISO/IEC/IEEE 12207 (software systems aspects);
- (g) shows the relationships of life cycle concepts to the hardware, human, services, process, procedure, facility and naturally occurring entity aspects of projects; and
- (h) 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.

This document is identical with, and has been reproduced from, ISO/IEC/IEEE 24748-1:2024 *Systems and software engineering — Life cycle management — Part 1: Guidelines for life cycle management*.

As this document has been reproduced from an international document, a full point substitutes for a comma when referring to a decimal marker.

Australian or Australian/New Zealand Standards that are identical adoptions of international normative references may be used interchangeably. Refer to the online catalogue for information on specific Standards.

The terms “normative” and “informative” are used in Standards to define the application of the appendices or annexes to which they apply. A “normative” appendix or annex is an integral part of a Standard, whereas an “informative” appendix or annex is only for information and guidance.

Contents

Preface	ii
Foreword	v
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Life cycle-related concepts	11
4.1 General	11
4.2 System concepts	11
4.2.1 General	11
4.2.2 Systems	12
4.2.3 System structure	12
4.2.4 Enabling systems	13
4.3 Life cycle concepts	15
4.3.1 System life cycle model	15
4.3.2 System life cycle stages	18
4.3.3 Stages in a Sol and its enabling systems	19
5 Life cycle stages	20
5.1 General	20
5.2 Concept stage	22
5.2.1 Overview	22
5.2.2 Purpose	22
5.2.3 Outcomes	22
5.3 Development stage	23
5.3.1 Overview	23
5.3.2 Purpose	23
5.3.3 Outcomes	23
5.4 Production stage	24
5.4.1 Overview	24
5.4.2 Purpose	24
5.4.3 Outcomes	24
5.5 Utilization stage	25
5.5.1 Overview	25
5.5.2 Purpose	25
5.5.3 Outcomes	25
5.6 Support stage	25
5.6.1 Overview	25
5.6.2 Purpose	26
5.6.3 Outcomes	26
5.7 Retirement stage	26
5.7.1 Overview	26
5.7.2 Purpose	27
5.7.3 Outcomes	27
6 Life cycle model adaptation	27
6.1 General	27
6.2 Adaptation sequence	27
6.2.1 General	27
6.2.2 Identification of the project environment and characteristics	28
6.2.3 Solicitation of inputs	28
6.2.4 Selection of appropriate standards	28
6.2.5 Selection of development approach	29
6.2.6 Selection of life cycle model	29

6.2.7	Selection of process model.....	29
6.2.8	Documentation of decisions and rationale.....	30
6.3	Life cycle model adaptation guidance.....	30
6.3.1	General.....	30
6.3.2	Scope adaptation.....	30
6.3.3	Stage adaptation.....	31
6.3.4	Life cycle model adaptation for domains, disciplines and specialties.....	31
6.4	Evaluation-related activities.....	34
7	Relationship with detailed process standards.....	34
Annex A	(informative) Process concepts.....	37
Annex B	(informative) Organizational concepts.....	51
Annex C	(informative) Project concepts.....	53
Annex D	(informative) Process views.....	58
Annex E	(informative) Guidance on development approaches and build planning.....	69
Annex F	(informative) Candidate joint stakeholder reviews.....	73
Annex G	(informative) Problem reporting capability.....	76
	Bibliography.....	78
	IEEE notices and abstract.....	81

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.

NOTES

Australian/New Zealand Standard

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 Sol can be viewed as a system element in another person's Sol. Conversely, it can be viewed as being part of the environment of operation for yet another person's Sol.

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

- 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 Sol 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.

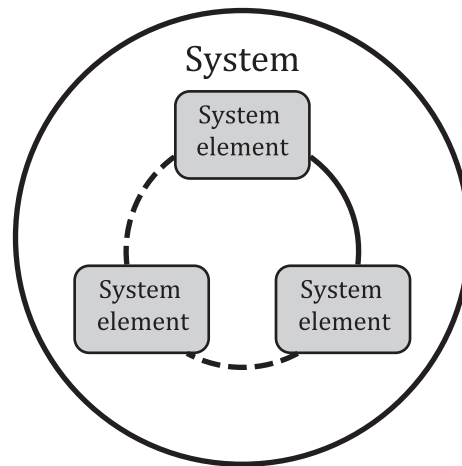


Figure 1 — System and system element relationship

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 Sol 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.

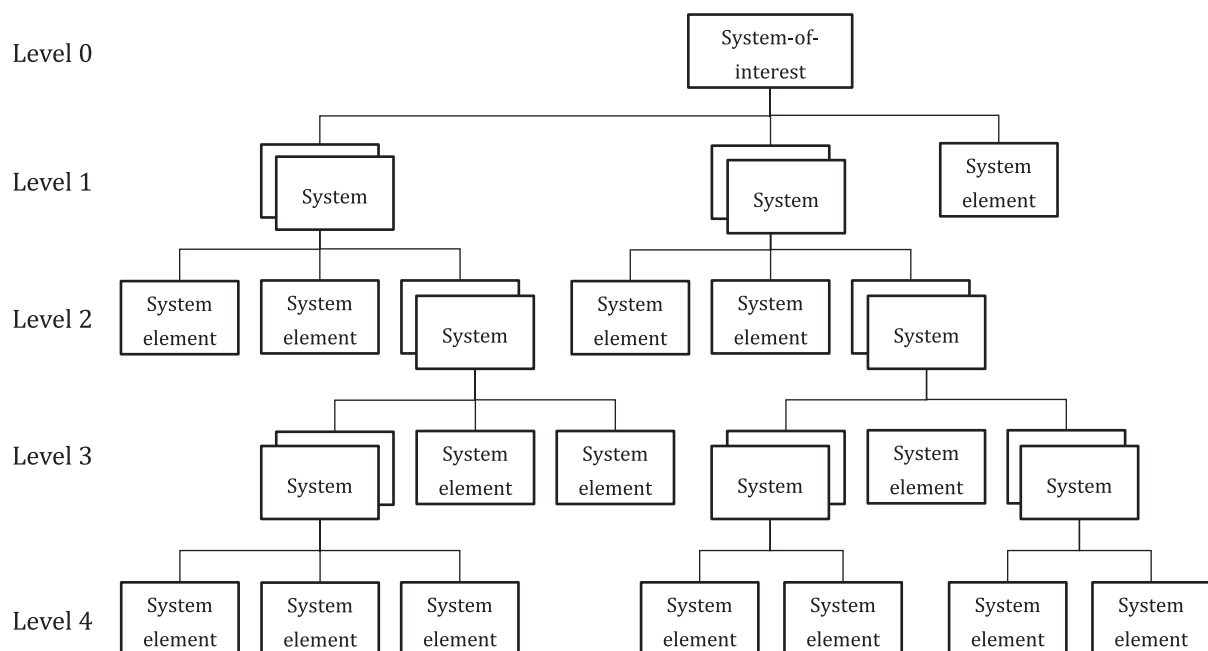


Figure 2 — System-of-interest structure

4.2.4 Enabling systems

Throughout the life cycle of a Sol, essential services are required from systems that are not directly a part of the operational environment of the Sol, 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 Sol 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.

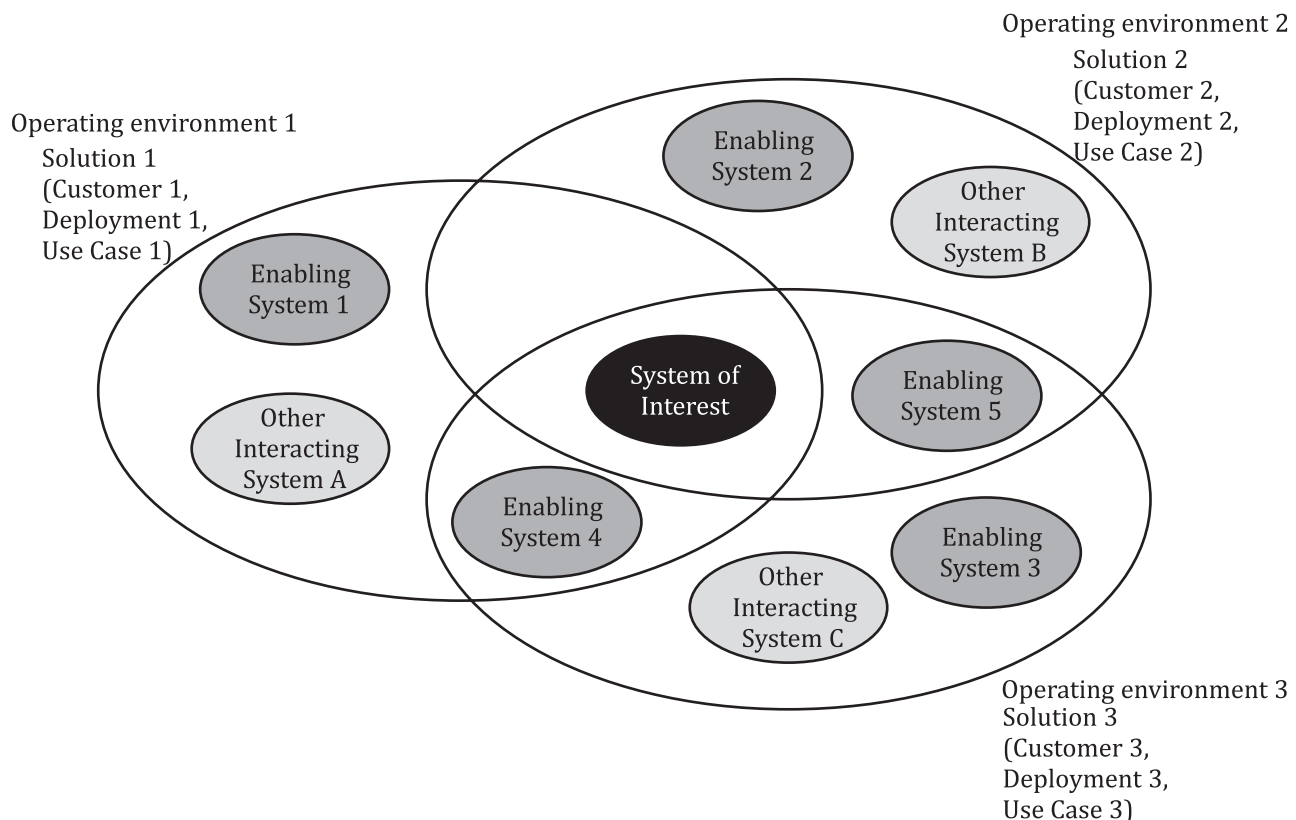


Figure 3 — Services delivered by enabling systems to the system-of-interest

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](#).

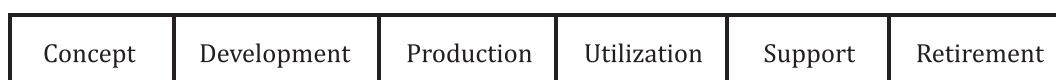


Figure 4 — Representative system life cycle model

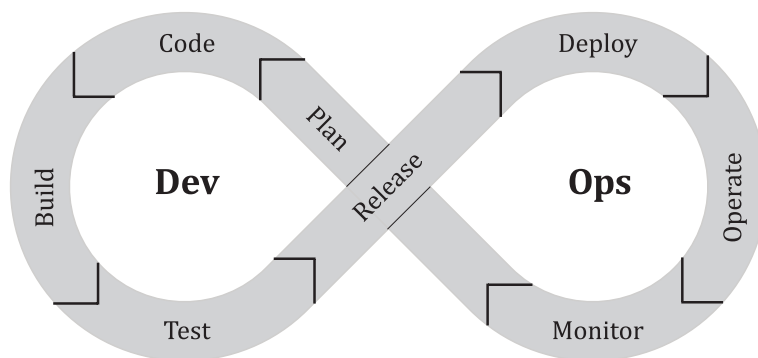
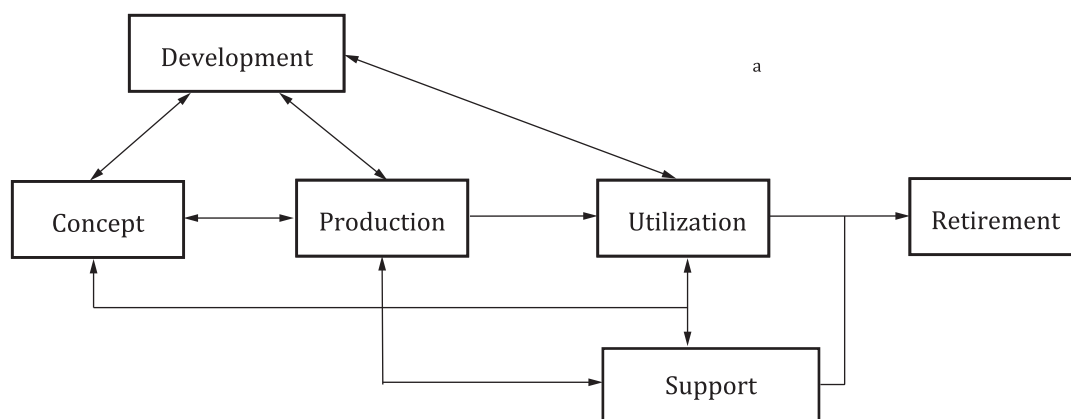


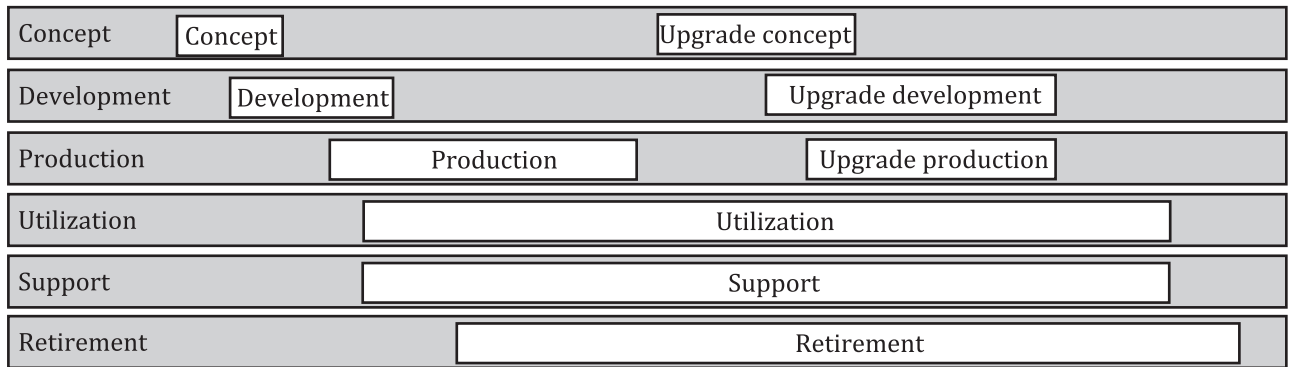
Figure 5 — Development and operations (DevOps) example life cycle stages

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.

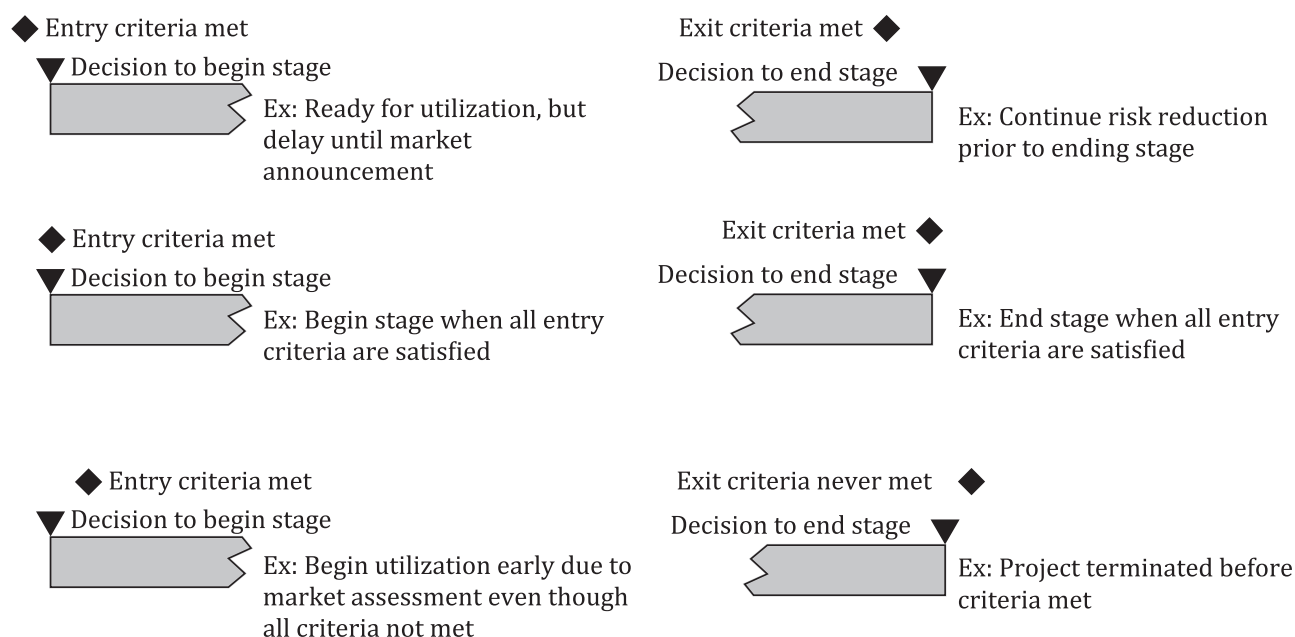
NOTE 3 Concept, development and production stages can be restarted for mid-life upgrade or service life extension.

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

Life cycle stages	Purpose	Decision options
Concept	Identify stakeholders' needs Explore concepts Propose viable solutions	<ul style="list-style-type: none"> — Begin subsequent stage or stages — Continue this stage — Go to or restart another stage — Hold project activity — Terminate project
Development	Refine system requirements Create solution description Build system Verify and validate system	
Production	Produce systems Inspect and test	
Utilization	Operate system to satisfy users' needs	
Support	Provide sustained system capability	
Retirement	Store, archive or dispose of system	

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