International Standard

ISO/IEC/IEEE 12207

Second edition
2026-04

Systems and software engineering — Software life cycle processes

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

Reference number
ISO/IEC/IEEE 12207:2026(en)

COPYRIGHT PROTECTED DOCUMENT

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 documents 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 IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments 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/JTC 1, *Information technology,* Subcommittee 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 12207:2017), which has been technically revised.

The main changes are as follows:

— clarifications and updates to reflect current practices in selected technical processes, including business or mission analysis, system architecture definition, implementation, integration, operations, and maintenance;

— improvements to selected technical management processes, including risk management and configuration management;

— updates to Clause 5, key concepts, including a more precise description of iteration, recursion, system-of-systems, and quality characteristics;

— new content in Clause 5 on concept and system definition, and expanded content on agile methods, process application, and system concepts.

— revised Annex D on model-based systems and software engineering (MBSSE)

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 complexity of software systems continues to increase to unprecedented levels. Technology change has led to new opportunities, but also to increased challenges for the organizations that create and utilise systems. There is a continuum of human-made systems from those that use little or no software to those in which software is the primary interest. It is rare to encounter a complex system without software, and all software systems require physical system elements (hardware) to operate, either as part of the software system-of-interest (SoI) or as an enabling system or infrastructure. These challenges exist throughout the life cycle of a system and at all levels of architectural detail.

The purpose of this document is to provide a defined set of software life cycle processes. This document provides a common process framework for engineering the life cycle of systems created by humans, adopting a software engineering approach. Software engineering is an interdisciplinary approach and means to enable software. This document concerns software systems that are configured with software elements and with one or more of the following: hardware elements, data, humans, processes, services, procedures, and facilities.

This document provides processes for use by acquirers, suppliers, and other stakeholders in the life cycle of a software system, such as developers, integrators, operators, maintainers, managers, quality managers, and users of software services and products. It covers defining stakeholder needs, concerns, priorities, and constraints for the required functionality and non-functional characteristics early in the life cycle, establishing requirements, and concurrent design synthesis and system validation while considering the complete problem. It integrates disciplines and specialties into a team effort, forming a structured set of process that proceeds from concept through production operation and maintenance and sustainment to disposal. It considers both the business and the technical needs of stakeholders with the goal of providing a quality product that meets the needs of users and other applicable stakeholders. It provides the processes for acquiring and supplying systems. It helps to improve communication and cooperation among the parties that create, utilise, and manage modern software systems so they can work in an integrated, coherent fashion. Finally, this document provides the framework for assessment and improvement of the life cycle processes.

There is a wide variety of software systems in terms of their purpose, domain, complexity, size, novelty, adaptability, quantity, location, life span, and evolution. The processes in this document form a comprehensive set from which an organization can construct software life cycle models appropriate to its products and services. An organization can select and apply an appropriate subset of these processes to fulfil its specific objectives.

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

— By an organization — to help establish an environment of desired processes. These processes can be supported by an infrastructure of methods, procedures, techniques, tools, and trained personnel. The organization can then employ this environment to perform and manage its projects and progress systems through their life cycle stages. In this mode, this document can be used to assess conformance of a declared, established environment to its provisions. It can be used by a single organization in a self-imposed mode or in a multi-party situation. Parties can be from the same organization or from different organizations, and the situation can range from an informal agreement to a formal contract.

— By a project: to help select, structure, and employ the elements of an established environment to provide products and services. In this mode this document is used in the assessment of conformance of the project to the declared and established environment.

— By an acquirer and a supplier: to help develop an agreement concerning processes and activities. Via the agreement, the processes and activities in this document are selected, negotiated, agreed to, and performed. In this mode this document is used for guidance in developing the agreement.

— By process assessors: to serve as a process reference model for use in the performance of process assessments that can be used to support organizational process improvement.

This document is related to ISO/IEC/IEEE 15288, which covers system engineering processes. The choice of whether to apply this document for the software life cycle processes, or ISO/IEC/IEEE 15288:2023, depends

on the system-of-interest (SoI). Processes in both documents have the same process purpose and process outcomes, but differ in activities and tasks to perform software engineering or systems engineering, respectively.

The requirements in this document are intended to be compatible with the requirements of the quality management system provided by ISO 9001, the service management system provided by ISO/IEC 20000-1, the IT asset management system provided by ISO/IEC 19770-1, and the information security management system provided by ISO/IEC 27001.

Systems and software engineering — Software life cycle processes

1 Scope

This document establishes a common framework for software life cycle processes. Its terminology can be referenced and applied across the software industry. It contains processes, activities and tasks that can be applied during the acquisition of a software system, product, or service and during the supply, development, operation, maintenance, and disposal of software products and services. This is accomplished through the involvement of stakeholders, with the goal of achieving customer satisfaction. This document includes those aspects of system definition needed to provide the context for software systems and services. This document also provides processes that can be employed for defining, controlling, and improving software life cycle processes within an organization or a project.

This document is applicable to one-of-a-kind software systems, software systems for wide commercial or public distribution, and customised, adaptable software systems. Software includes the software portion of firmware. It applies to a complete stand-alone software system and to software systems that are embedded and integrated into larger more complex and complete systems of systems (SoS). The processes, activities, and tasks of this document can also be applied during the acquisition of a system that contains software.

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

This document can be applied in organizations and software projects using a variety of formal engineering approaches. It is applicable for agile approaches and methods, which are most widely used for software development, sustainment, and maintenance, and which are believed to be more affordable and to deliver usable products more quickly.

This document does not identify or require any specific software life cycle model, development methodology, method, modelling approach, or techniques for selecting a life cycle model for the organization or project and mapping the processes, activities, and tasks in this document into that model. Using engineering judgment to help achieve the desired level of quality is also outside the scope of this document.

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

2 Normative references

There are no normative references in this document.

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

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

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

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

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

— 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.

3.1.1
acquirer
stakeholder (3.1.76) that acquires or procures a *system* (3.1.78), *product* (3.1.49), or *service* (3.1.65) from a *supplier* (3.1.77)

Note 1 to entry: Other terms commonly used for an acquirer are buyer, *customer* (3.1.19), owner, purchaser, or internal organizational sponsor.

3.1.2
acquisition
process (3.1.45) of obtaining a *system* (3.1.78), *product* (3.1.49) or *service* (3.1.65)

3.1.3
activity
set of cohesive *tasks* (3.1.83) of a *process* (3.1.45)

3.1.4
agile development
development approach based on iterative development, frequent inspection and adaptation, and incremental deliveries, in which *requirements* (3.1.56) and solutions evolve through collaboration in cross-functional teams and through continual *stakeholder* (3.1.76) feedback

3.1.5
agreement
mutual acknowledgement of terms and conditions under which a working relationship is conducted

EXAMPLE Contract, memorandum of agreement.

3.1.6
architecture
fundamental concepts or properties of an entity in its *environment* (3.1.25) and governing principles for the realization and evolution of this entity and its related *life cycle* (3.1.34)*processes* (3.1.45)

[SOURCE: ISO/IEC/IEEE 42010:2022, 3.2]

3.1.7
architecture description framework
conventions, principles and practices for the description of *architectures* (3.1.6) established within a specific domain of application or community of *stakeholders* (3.1.76)

EXAMPLE Generalized Enterprise Reference Architecture and Methodologies (GERAM) (ISO 15704:2019, Annex B), Reference Model of Open Distributed Processing (RM-ODP) (ISO/IEC 10746-1), Reference Model of Open Distributed Processing (RM-ODP), Unified Architecture Framework (UAF), and NATO Architecture Framework (NAF).

[SOURCE: ISO/IEC/IEEE 42010:2022, 3.5, modified — The abbreviated term and note 1 to entry have been removed.]

3.1.8
architecture view
information part comprising portion of an *architecture* (3.1.6) description

[SOURCE: ISO/IEC/IEEE 42010:2022, 3.7, modified — EXAMPLE has been removed.]

3.1.9
architecture viewpoint
set of conventions for the creation, interpretation and use of an *architecture view* (3.1.8) to frame one or more *concerns* (3.1.16)

[SOURCE: ISO/IEC/IEEE 42010:2022, 3.8, modified — Notes to entry have been removed.]

3.1.10
artefact
work *product* (3.1.49) that is produced and used during a *project* (3.1.50) to capture and convey information

EXAMPLE Models, *stakeholder* (3.1.76)*requirements* (3.1.56), *system* (3.1.78) requirements, *architecture* (3.1.6) descriptions, *design* (3.1.20) descriptions, source code, implemented *system elements* (3.1.79), verified or validated system.

3.1.11
audit
independent examination of a work *product* (3.1.49) or set of work products to assess compliance with contractual *agreements* (3.1.5) or conformance to specifications, standards, or other criteria

3.1.12
baseline
formally approved version of a *configuration item* (3.1.17), regardless of media, formally designated and fixed at a specific time during the configuration item's *life cycle* (3.1.34)

[SOURCE: IEEE Std 828-2012]

3.1.13
business process
partially ordered set of enterprise *activities* (3.1.3) that can be executed to achieve some desired end-result in pursuit of a given objective of an *organization* (3.1.41)

3.1.14
capability
ability to do something useful under a particular set of conditions

[SOURCE: ISO/IEC/IEEE 24641:2023, 3.1.3, modified — Note 1 to entry has been removed.]

3.1.15
concept of operations
verbal and graphic statement, in broad outline, of an *organization's* (3.1.41) assumptions or intent regarding an operation or series of operations of new, modified, or existing organizational *systems* (3.1.78)

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* (3.1.39).

Note 2 to entry: The concept of operations provides the basis for bounding the operating space, system *capabilities* (3.1.14), *interfaces* (3.1.33) and operating *environment* (3.1.25).

[SOURCE: ANSI/AIAA G-043B-2018, 5.2, modified — The second definition has been used; the last two sentences of Note 1 to entry have been removed; Note 2 to entry has been added.]

3.1.16
concern
matter of interest or importance to a *stakeholder* (3.1.76)

Note 1 to entry: A concern pertains to any influence on a *system* (3.1.78) in its *environment* (3.1.25), 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.]

3.1.17
configuration item
item or aggregation of *hardware* (3.1.29) or *software* (3.1.66) with associated information and data

Note 1 to entry: *Information items* (3.1.31) can be considered as part of the software and managed as part of the configuration item, or can be managed using the information management *process* (3.1.45).

3.1.18
constituent system
independent *system* (3.1.78) that forms part of a *system of systems (SoS)* (3.1.81)

Note 1 to entry: Constituent systems can be part of one or more SoS. Each constituent system is a useful system by itself, having its own development, *management* (3.1.38), utilization, goals, and *resources* (3.1.60), but interacts within the SoS to provide the unique *capability* (3.1.14) of the SoS.

[SOURCE: ISO/IEC/IEEE 21839:2019, 3.1.1]

3.1.19
customer
organization (3.1.41) or person that receives a *product* (3.1.49) or *service* (3.1.65)

EXAMPLE Consumer, client, *user* (3.1.88), *acquirer* (3.1.1), buyer or purchaser.

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

3.1.20
design
specification of *system elements* (3.1.79) and their relationships, that is sufficiently complete to support a compliant implementation of the *architecture* (3.1.6)

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

3.1.21
design characteristic
design (3.1.20) attribute or distinguishing feature that pertains to a measurable description of a *product* (3.1.49), or *service* (3.1.65)

3.1.22
digital engineering
software engineering (3.1.69) discipline that uses the underlying data, modelling, simulation, and analytical functions for physical or logical *systems* (3.1.78) conceptualization, development, test and evaluation, and integrated control of operational performance

3.1.23
enabling system
system (3.1.78) that supports a *system-of-interest* (3.1.80) during its *life cycle* (3.1.36)*stages* (3.1.75) but does not necessarily contribute directly to its function during operation

EXAMPLE A configuration management system used to control *software elements* (3.1.68) during *software* (3.1.66) development.

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

3.1.24
engineering
application of a systematic, disciplined, quantifiable approach to structures, machines, *products* (3.1.49), systems, or *processes* (3.1.45)

3.1.25
environment
context determining the setting and circumstances of all influences upon a *system* (3.1.78)

3.1.26
facility
physical means or equipment for facilitating the performance of an action, e.g. buildings, instruments, tools

3.1.27
firmware
ordered set of instructions and associated data stored in a way that is functionally independent of main storage, usually in read-only memory

3.1.28
governance
practice of establishing and enforcing strategic goals and objectives, organizational policies and performance parameters

3.1.29
hardware
physical equipment used to *process* (3.1.45), store, or transmit computer programs or data

[SOURCE: ISO/IEC 19770-1:2017, 3.21]

3.1.30
incident
anomalous or unexpected event or set of events at any time during the *life cycle* (3.1.36) of a *project* (3.1.50), *product* (3.1.49), *service* (3.1.65), or *system* (3.1.78)

Note 1 to entry: An incident is elevated and treated as a *problem* (3.1.44) when the cause of the incident is analysed and corrected to prevent recurrence, or to avoid or minimise loss of life, or damage to property or natural *resources* (3.1.60). Some incidents do not involve further analysis, as they result from known errors where workarounds or solutions are already known.

3.1.31
information item
information product
separately identifiable body of information that is produced, stored, and delivered for human use

Note 1 to entry: A document produced to meet information *requirements* (3.1.59) can be an information item, part of an information item, or a combination of several information items.

Note 2 to entry: An information item can be produced in several versions during a *project* (3.1.50) or *system* (3.1.78) *life cycle* (3.1.36).

[SOURCE: ISO/IEC/IEEE 15289:2019, 3.1.12, modified — The preferred term "information product" has been changed to an admitted term.]

3.1.32
infrastructure
hardware (3.1.29) and *software* (3.1.66)*environment* (3.1.25) to support *system* (3.1.78) and software *design* (3.1.20), development, operation and modification

3.1.33
interface
point at which two or more logical or physical *system elements* (3.1.79) meet to act on or communicate with each other

3.1.34
interoperating system
system (3.1.78) that exchanges information with the *system-of-interest* (3.1.80) and uses the information that has been exchanged

3.1.35
iteration
repeating the application of the same *process* (3.1.45) or set of processes at the same level of abstraction on the same *system* (3.1.78) or *system element* (3.1.79)

3.1.36
life cycle
evolution of a *system* (3.1.78), *product* (3.1.49), *service* (3.1.65), *project* (3.1.50) or other human-made entity from conception through *retirement* (3.1.61)

3.1.37
life cycle model
framework of *processes* (3.1.45) and *activities* (3.1.3) concerned with the *life cycle* (3.1.36) which can be organized into *stages* (3.1.75), acting as a common reference for communication and understanding

3.1.38
management
oversight and direction of *activities* (3.1.3) and controls to achieve the strategic objectives set by the *organization's* (3.1.41) governing body

3.1.39
operational concept
verbal and graphic statement of an *organization's* (3.1.41) assumptions or intent in regard to an operation or series of operations of a specific *system* (3.1.78) or a related set of specific 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* (3.1.25) from the *users'* (3.1.88) and *operators'* (3.1.40) perspectives.

Note 2 to entry: The operational concept is about systems, while a *concept of operations* (3.1.15) typically refers to organizations.

[SOURCE: ANSI/AIAA G-043B-2018, 5.2, modified — The third definition has been used; the first sentence in Note 1 to entry has been removed; Note 2 to entry has been added.]

3.1.40
operator
individual or *organization* (3.1.41) that performs the operations of a *system* (3.1.78)

Note 1 to entry: The role of operator and the role of *user* (3.1.88) can be vested, simultaneously or sequentially, in the same individual or organization.

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

Note 3 to entry: An operator may perform operations on a system that is operated, or of a system that is operated, depending on whether operating instructions are placed within the system boundary.

3.1.41
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.

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

3.1.42
party
organization (3.1.41) entering into an *agreement* (3.1.5)

Note 1 to entry: The agreeing parties are called the *acquirer* (3.1.1) and the *supplier* (3.1.77).

3.1.43
portfolio
collection of *projects* (3.1.50) or *systems* (3.1.78) that address the strategic or business objectives of the *organization* (3.1.41)

3.1.44
problem
difficulty, uncertainty, or otherwise realised and undesirable condition or situation that is investigated and can receive corrective action

3.1.45
process
set of interrelated or interacting *activities* (3.1.3) that transform inputs into outputs

3.1.46
process outcome
observable result of the successful achievement of the *process purpose* (3.1.47)

3.1.47
process purpose
high level objective of performing the *process* (3.1.45) and the likely outcomes of effective implementation of the process

Note 1 to entry: The purpose of implementing the process is to provide benefits to the *stakeholders* (3.1.76)

3.1.48
process reference model
PRM
model comprising definitions of *processes* (3.1.45) in a domain of application described in terms of *process purpose* (3.1.47) and outcomes, together with an *architecture* (3.1.6) describing the relationships between the processes

[SOURCE: ISO/IEC 33001:2015, 3.3.16, modified — The abbreviated term "PRM" has been added.]

3.1.49
product
result of a *process* (3.1.45)

Note 1 to entry: ISO/IEC/IEEE 15288 defines a product as "output of an *organization* (3.1.41) that can be produced without any transaction taking place between the organization and the *customer* (3.1.19)".

Note 2 to entry: There are four generic product categories: *hardware* (3.1.29), e.g. engine mechanical part; *software* (3.1.66), e.g. computer program procedures, and possibly associated documentation and data; *services* (3.1.65), e.g. transport; and processed materials, e.g. lubricant. Hardware and processed materials are generally tangible products, while software or services are generally intangible.

3.1.50
project
endeavour with defined start and finish criteria undertaken to create a *product* (3.1.49) or *service* (3.1.65) in accordance with specified *resources* (3.1.60) and *requirements* (3.1.59)

Note 1 to entry: A project is sometimes viewed as a unique *process* (3.1.45) comprising coordinated and controlled *technical management* (3.1.85) and technical *activities* (3.1.3).

Note 2 to entry: Continuous development approaches such as agile and DevOps can create products and services without the defined start and finish criteria of projects.