INTERNATIONAL STANDARD

**ISO/IEC/ IEEE**

**12207**

First edition

2017-11

**Systems and software engineering —**

**Software life cycle processes**

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

*logiciel*

Reference number

ISO/IEC/IEEE 12207:2017(E)

© ISO/IEC 2017

© IEEE 2017

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2017, Published in Switzerland

© IEEE 2017

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

ISO copyright office Institute of Electrical and Electronics Engineers, Inc Ch. de Blandonnet 8 • CP 401 3 Park Avenue, New York

CH-1214 Vernier, Geneva, Switzerland NY 10016-5997, USA

Tel. +41 22 749 01 11

Fax +41 22 749 09 47

copyright@iso.org stds.ipr@ieee.org

www.iso.org www.ieee.org

© ISO/IEC 2017 – All rights reserved

ii © IEEE 2017 – All rights reserved Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

**Contents** Page

**Foreword ......................................................................................................................................................................................... vi Introduction ................................................................................................................................................................................. vii**

**1 Scope .................................................................................................................................................................................. 1 1.1 Overview ........................................................................................................................................................................... 1 1.2 Purpose ............................................................................................................................................................................. 1 1.3 Field of application ........................................................................................................................................................ 1 1.4 Limitations ....................................................................................................................................................................... 2**

**2 Normative references ................................................................................................................................................... 2**

**3 Terms, definitions, and abbreviated terms ........................................................................................................... 2 3.1 Terms and definitions .................................................................................................................................................. 2 3.2 Abbreviated terms ...................................................................................................................................................... 11**

**4 Conformance ................................................................................................................................................................. 11 4.1 Intended usage ............................................................................................................................................................. 11 4.2 Full conformance ......................................................................................................................................................... 12 4.2.1 Full conformance to outcomes ................................................................................................................................ 12 4.2.2 Full conformance to tasks ........................................................................................................................................ 12 4.3 Tailored conformance ............................................................................................................................................... 12**

**5 Key concepts and application .................................................................................................................................. 13 5.1 Introduction .................................................................................................................................................................. 13 5.2 Software system concepts ........................................................................................................................................ 13 5.2.1 Software systems ......................................................................................................................................................... 13 5.2.2 Software system structure ....................................................................................................................................... 13 5.2.3 Enabling systems ......................................................................................................................................................... 15 5.2.4 Life cycle processes for the software system ..................................................................................................... 16 5.3 Organization and project concepts ........................................................................................................................ 16 5.3.1 Organizations ............................................................................................................................................................... 16 5.3.2 Organization and project-level adoption ............................................................................................................ 17 5.4 Life cycle concepts ...................................................................................................................................................... 17 5.4.1 Software life cycle stages .......................................................................................................................................... 17 5.4.2 Life cycle model for the software system ............................................................................................................ 17 5.5 Process concepts ......................................................................................................................................................... 19 5.5.1 Criteria for processes ................................................................................................................................................. 19 5.5.2 Description of processes ........................................................................................................................................... 19 5.5.3 General characteristics of processes .................................................................................................................... 19 5.5.4 Tailoring ......................................................................................................................................................................... 19 5.6 Process groups ............................................................................................................................................................. 19 5.6.1 Introduction .................................................................................................................................................................. 19 5.6.2 Agreement processes ................................................................................................................................................. 21 5.6.3 Organizational project-enabling processes ........................................................................................................ 22 5.6.4 Technical Management processes ......................................................................................................................... 22 5.6.5 Technical processes ................................................................................................................................................... 22 5.7 Process application .................................................................................................................................................... 22 5.8 Process reference model .......................................................................................................................................... 23**

**6 Software life cycle processes ................................................................................................................................... 24 6.1 Agreement processes ................................................................................................................................................. 24 6.1.1 Acquisition process .................................................................................................................................................... 24 6.1.2 Supply process ............................................................................................................................................................. 27 6.2 Organizational Project-Enabling processes ....................................................................................................... 28 6.2.1 Life cycle model management process ................................................................................................................. 29 6.2.2 Infrastructure Management process .................................................................................................................... 30 6.2.3 Portfolio Management process ............................................................................................................................... 31 6.2.4 Human Resource Management process ............................................................................................................... 33**

iii

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

**6.2.5 Quality Management process ................................................................................................................................... 34 6.2.6 Knowledge Management process ........................................................................................................................... 36 6.3 Technical Management processes .......................................................................................................................... 37 6.3.1 Project Planning process ........................................................................................................................................... 38 6.3.2 Project assessment and control process .............................................................................................................. 40 6.3.3 Decision Management process ................................................................................................................................ 43 6.3.4 Risk Management process ........................................................................................................................................ 44 6.3.5 Configuration Management process ...................................................................................................................... 46 6.3.6 Information Management process ......................................................................................................................... 50 6.3.7 Measurement process ................................................................................................................................................ 52 6.3.8 Quality Assurance process ........................................................................................................................................ 53 6.4 Technical processes .................................................................................................................................................... 55 6.4.1 Business or Mission Analysis process ................................................................................................................... 56 6.4.2 Stakeholder Needs and Requirements Definition process ............................................................................ 59 6.4.3 System/Software requirements definition process ......................................................................................... 63 6.4.4 Architecture Definition process .............................................................................................................................. 66 6.4.5 Design Definition process ......................................................................................................................................... 71 6.4.6 System Analysis process ............................................................................................................................................ 74 6.4.7 Implementation process ........................................................................................................................................... 75 6.4.8 Integration process ..................................................................................................................................................... 79 6.4.9 Verification process .................................................................................................................................................... 82 6.4.10 Transition process....................................................................................................................................................... 85 6.4.11 Validation process ....................................................................................................................................................... 89 6.4.12 Operation process ....................................................................................................................................................... 92 6.4.13 Maintenance process .................................................................................................................................................. 95 6.4.14 Disposal process .......................................................................................................................................................... 99**

**Annex A** (normative) **Tailoring process ........................................................................................................................... 102 A.1 Introduction ............................................................................................................................................................... 102 A.2 Tailoring process ...................................................................................................................................................... 102 A.2.1 Purpose ........................................................................................................................................................................ 102 A.2.2 Outcomes ..................................................................................................................................................................... 102 A.2.3 Activities and tasks .................................................................................................................................................. 102**

**Annex B** (informative) **Examples of process information items ............................................................................... 104**

**Annex C** (informative) **Process Reference Model for assessment purposes ......................................................... 107 C.1 Introduction ............................................................................................................................................................... 107 C.2 Conformance with ISO/IEC 33004 ....................................................................................................................... 107 C.2.1 General ......................................................................................................................................................................... 107 C.2.2 Requirements for process reference models .................................................................................................. 107 C.2.3 Process descriptions ................................................................................................................................................ 108 C.3 The process reference model ............................................................................................................................... 108**

**Annex D** (informative) **Process integration and process constructs ........................................................................ 109 D.1 Introduction ............................................................................................................................................................... 109 D.2 Process constructs and their usage .................................................................................................................... 109**

**Annex E** (informative) **Process views ................................................................................................................................ 111 E.1 Introduction ............................................................................................................................................................... 111 E.2 The process view concept ...................................................................................................................................... 111 E.3 Process viewpoint .................................................................................................................................................... 111 E.4 Process view for specialty engineering ............................................................................................................. 112 E.5 Process view for interface management ........................................................................................................... 114 E.6 Process view for software assurance (Information security) .................................................................... 116**

**Annex F** (informative) **Software system architecture modelling .............................................................................. 120 F.1 Introduction ............................................................................................................................................................... 120 F.2 Views, models and model kinds used in software system architecture ................................................. 120 F.2.1 Functional model ...................................................................................................................................................... 120 F.2.2 Static model ................................................................................................................................................................ 121 F.2.3 Data model .................................................................................................................................................................. 121 F.2.4 Behavioral model ...................................................................................................................................................... 121 F.2.5 Temporal model ........................................................................................................................................................ 121 F.2.6 Structural model ....................................................................................................................................................... 121**

iv

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

**F.2.7 Network model .......................................................................................................................................................... 121 F.3 Other model considerations .................................................................................................................................. 121**

**Annex G** (informative) **Application of software life cycle processes to a system of systems ........................... 123 G.1 Introduction ................................................................................................................................................................ 123 G.2 SoS characteristics and types ................................................................................................................................ 123 G.3 SE processes applied to systems of systems ..................................................................................................... 124 G.3.1 General ......................................................................................................................................................................... 124 G.3.2 Agreement processes ............................................................................................................................................... 124 G.3.3 Organizational project enabling processes ...................................................................................................... 124 G.3.4 Technical management processes ....................................................................................................................... 125 G.3.5 Technical processes ................................................................................................................................................. 125**

**Annex H** (informative) **Application of Agile ..................................................................................................................... 127 Annex I** (informative) **Process Mapping from ISO/IEC/IEEE 12207:2008 ............................................................. 129 Bibliography ............................................................................................................................................................................... 143**

**List of Illustrations**

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

**Table I.2 — Comparison of process outcomes in ISO/IEC/IEEE 12207:2017 and software-related outcomes in the previous edition ........................................................................................................................ 131**

v

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

**Foreword**

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

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

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

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

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

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

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

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

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

This document was developed with the following goals:

— provide a common terminology between the revision of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207;

— where applicable, provide common process names and process structure between the revision of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207; and

— enable the user community to evolve towards fully harmonized standards, while allowing backward compatibility.

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

vi

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

**Introduction**

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

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

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

a) By an organization — to help establish an environment of desired processes. These processes can be supported by an infrastructure of methods, procedures, techniques, tools and trained personnel. The organization may then employ this environment to perform and manage its projects and progress software systems through their life cycle stages. In this mode, this document is used to assess conformance of a declared, established environment to its provisions.

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

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

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

vii

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**INTERNATIONAL STANDARD ISO/IEC/IEEE 12207:2017(E) Systems and software engineering — Software life cycle processes**

**1 Scope**

**1.1 Overview**

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

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

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

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

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

**1.2 Purpose**

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

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

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

**1.3 Field of application**

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

1

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

**1.4 Limitations**

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

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

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

**2 Normative references**

There are no normative references in this document.

**3 Terms, definitions, and abbreviated terms**

**3.1 Terms and definitions**

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

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

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

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

— IEEE Standards Dictionary Online: available at http://ieeexplore.ieee.org/xpls/dictionary.jsp

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

**3.1.1**

**acquirer**

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

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

2

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**3.1.2**

**acquisition**

process of obtaining a system, product or service

**3.1.3**

**activity**

set of cohesive tasks of a process

**3.1.4**

**agile development**

**ISO/IEC/IEEE 12207:2017(E)**

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

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

**3.1.5**

**agreement**

mutual acknowledgement of terms and conditions under which a working relationship is conducted EXAMPLE Contract, memorandum of agreement.

**3.1.6**

**architecture**

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

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

**3.1.7**

**architecture framework**

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

EXAMPLE 1 Generalised Enterprise Reference Architecture and Methodologies (GERAM) [ISO 15704] is an architecture framework.

EXAMPLE 2 Reference Model of Open Distributed Processing (RM‐*ODP) [ISO/IEC* 10746] is an architecture framework. [SOURCE: ISO/IEC/IEEE 42010:2011]

**3.1.8**

**architecture view**

work product expressing the architecture of a system from the perspective of specific system concerns [SOURCE: ISO/IEC/IEEE 42010:2011]

**3.1.9**

**architecture viewpoint**

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

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

**3.1.10**

**audit**

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

3

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

**3.1.11**

**baseline**

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

[SOURCE: IEEE Std 828‐2012]

**3.1.12**

**business process**

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

**3.1.13**

**concept of operations**

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

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

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

[SOURCE: ANSI/AIAA G‐043A‐2012e]

**3.1.14**

**concern**

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

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

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

**3.1.15**

**configuration item**

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

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

**3.1.16**

**customer**

organization or person that receives a product or service

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

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

**3.1.17**

**design,** verb

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

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

**3.1.18**

**design,** noun

result of the process in 3.1.17

4

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

**3.1.19**

**design characteristic**

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

**3.1.20**

**enabling system**

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

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

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

**3.1.21**

**environment**

<system> context determining the setting and circumstances of all influences upon a system [SOURCE: ISO/IEC/IEEE 42010:2011]

**3.1.22**

**facility**

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

**3.1.23**

**incident**

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

**3.1.24**

**information item**

separately identifiable body of information that is produced, stored, and delivered for human use [SOURCE: ISO/IEC/IEEE 15289:2015]

**3.1.25**

**infrastructure**

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

**3.1.26**

**life cycle**

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

**3.1.27**

**life cycle model**

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

**3.1.28**

**operational concept**

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

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

5

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

[SOURCE: ANSI/AIAA G‐043A‐2012e]

**3.1.29**

**operator**

individual or organization that performs the operations of a system

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

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

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

**3.1.30**

**organization**

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

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

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

**3.1.31**

**party**

organization entering into an agreement

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

**3.1.32**

**problem**

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

**3.1.33**

**process**

set of interrelated or interacting activities that transforms inputs into outputs

**3.1.34**

**process outcome**

observable result of the successful achievement of the process purpose

**3.1.35**

**process purpose**

high‐level objective of performing the process and the likely outcomes of effective implementation of the process Note 1 to entry: The purpose of implementing the process is to provide benefits to the stakeholders.

**3.1.36**

**product**

result of a process

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

**3.1.37**

**project**

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

6

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

**3.1.38**

<project> **portfolio**

collection of projects that addresses the strategic objectives of the organization

**3.1.39**

**qualification**

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

**3.1.40**

**quality assurance**

part of quality management focused on providing confidence that quality requirements will be fulfilled [SOURCE: ISO 9000:2015]

**3.1.41**

**quality characteristic**

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

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

**3.1.42**

**quality management**

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

**3.1.43**

**release**

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

EXAMPLE Test release.

**3.1.44**

**requirement**

statement that translates or expresses a need and its associated constraints and conditions [SOURCE: ISO/IEC/IEEE 29148:2011, modified, NOTE has been removed.]

**3.1.45**

**resource**

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

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

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

**3.1.46**

**retirement**

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

**3.1.47**

**risk**

effect of uncertainty on objectives

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

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

7

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

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

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

**3.1.48**

**safety**

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

**3.1.49**

**security**

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

[SOURCE: NATO AEP‐67]

**3.1.50**

**service**

performance of activities, work, or duties

Note 1 to entry: A service is self‐contained, coherent, discrete, and can be composed of other services. Note 2 to entry: A service is generally an intangible product.

**3.1.51**

**software element**

system element that is software

**3.1.52**

**software engineering**

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

**3.1.53**

**software item**

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

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

**3.1.54**

**software product**

set of computer programs, procedures, and possibly associated documentation and data Note 1 to entry: A software product is a software system viewed as the output (product) resulting from a process.

**3.1.55**

**software system**

system for which software is of primary importance to the stakeholders

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

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

8

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

**3.1.56**

**software system element**

member of a set of elements that constitute a software system

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

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

**3.1.57**

**software unit**

atomic‐level software component of the software architecture that can be subjected to standalone testing Note 1 to entry: Some software units are separately compilable pieces of code.

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

**3.1.58**

**stage**

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

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

Note 2 to entry: Stages often overlap.

**3.1.59**

**stakeholder**

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

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

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

**3.1.60**

**supplier**

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

Note 1 to entry: Other terms commonly used for supplier are contractor, producer, seller, or vendor. Note 2 to entry: The acquirer and the supplier sometimes are part of the same organization.

**3.1.61**

**system**

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

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

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

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

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

**3.1.62**

**system element**

member of a set of elements that constitute a system

9

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

**3.1.63**

**system-of-interest**

**SOI**

system whose life cycle is under consideration

**3.1.64**

**system of systems**

**SoS**

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

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

**3.1.65**

**systems engineering**

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

**3.1.66**

**task**

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

**3.1.67**

**technical management**

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

**3.1.68**

**trade-off**

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

**3.1.69**

**traceability**

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

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

**3.1.70**

**user**

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

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

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

**3.1.71**

**validation**

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

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

10

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

**3.1.72**

**verification**

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

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

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

**3.2 Abbreviated terms**

CCB Configuration Control Board

CM Configuration Management

COTS Commercial‐Off‐The‐Shelf

FCA Functional Configuration Audit

FOSS Free and Open Source Software

GUI Graphical User Interface

NDI Non‐Developmental Items

QA Quality Assurance

PCA Physical Configuration Audit

PESTEL Political, Economic, Social, Technological, Environmental, and Legal

PMI Project Management Institute

PMP Project Management Plan

PRM Process Reference Model

SCM Software Configuration Management

SDP Software Development Plan

SEMP Systems Engineering Management Plan

SOI System‐of‐Interest

SoS System of Systems

SWOT Strengths, Weaknesses, Opportunities, Threats

WBS Work Breakdown Structure

**4 Conformance**

**4.1 Intended usage**

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

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

11

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

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

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

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

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

**4.2 Full conformance**

**4.2.1 Full conformance to outcomes**

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

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

**4.2.2 Full conformance to tasks**

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

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

**4.3 Tailored conformance**

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

12

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**5 Key concepts and application 5.1 Introduction**

**ISO/IEC/IEEE 12207:2017(E)**

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

NOTE Further elaboration of these concepts can be found in ISO/IEC TS 24748‐1, ISO/IEC TR 24748‐2, and ISO/IEC TR24748‐3 on the application of life cycle management.

**5.2 Software system concepts**

**5.2.1 Software systems**

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

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

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

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

a) defined boundaries encapsulate meaningful needs and practical solutions;

b) there is a hierarchical or other relationship between system elements;

c) an entity at any level in the system‐of‐interest can be viewed as a system;

d) a system comprises an integrated, defined set of subordinate system elements;

e) humans can be viewed as both users external to a system and as system elements (i.e., operators) within a system; and

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

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

**5.2.2 Software system structure**

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

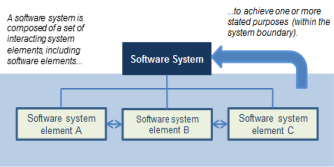
13

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

****

**Figure 1 — Software system and software system element relationship**

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

Software

System-of

Interest

System

Software System

System

Software

System

Element

System

System

Software

Software

Element

Element

System

Element

Element

System

System

Software Element

Software Element

System Software Element

Software System

System Element

Software Element

System Element

System Element

Software Element

Software Element

System Element

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

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

14

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

**5.2.3 Enabling systems**

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

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

System B in

operational

environment

Software system A

in operational

environment

*Interaction with*

*systems comprising the operational environment*

**Software system**

**of-interest**

*Interaction*

*with enabling*

*systems*

Software system C

in operational

environment

Enabling

software

system

X

Enabling

system

Y

Enabling

software

system

Z

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

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

15

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

NOTE Enabling systems in software development include software development and test environments for target platforms. **5.2.4 Life cycle processes for the software system**

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

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

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

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

**5.3 Organization and project concepts**

**5.3.1 Organizations**

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

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

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

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

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

16

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

**5.3.2 Organization and project-level adoption**

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

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

**5.4 Life cycle concepts**

**5.4.1 Software life cycle stages**

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

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

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

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

**5.4.2 Life cycle model for the software system**

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

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

17

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

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

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

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

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

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

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

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

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

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

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

18

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**5.5 Process concepts 5.5.1 Criteria for processes**

**ISO/IEC/IEEE 12207:2017(E)**

The determination of the life cycle processes in this document is based upon three basic principles: 1) Each life cycle process has strong relationships among its outcomes, activities and tasks. 2) The dependencies among the processes are reduced to the greatest feasible extent.

3) A process is capable of execution by a single organization in the life cycle.

**5.5.2 Description of processes**

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

a) The title conveys the scope of the process as a whole.

b) The purpose describes the goals of performing the process.

c) The outcomes express the observable results expected from the successful performance of the process. d) The activities are sets of cohesive tasks of a process.

e) The tasks are requirements, recommendations, or permissible actions intended to support the achievement of the outcomes.

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

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

Additional detail regarding this form of process description can be found in ISO/IEC TR 24774. **5.5.3 General characteristics of processes**

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

**5.5.4 Tailoring**

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

**5.6 Process groups**

**5.6.1 Introduction**

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

19

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

a) Agreement processes;

b) Organizational Project‐Enabling Processes;

c) Technical Management Processes; and

d) Technical Processes.

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

20

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**Agreement Processes**

**Software Life Cycle Processes**

**Technical Management**

**Project**

**Project**

**Processes**

**Processes**

**Processes**

**Project Planning Process (6.3.1)**

**ISO/IEC/IEEE 12207:2017(E)**

**Technical Processes**

**Business or Mission Analysis**

**Process (6.4.1) Acquisition Process (6.1.1)**

**Supply Process (6.1.2)**

**Organizational**

**Project-Enabling**

**Processes**

**Life Cycle Model Management Process (6.2.1)**

**Project - Enabling**

**Infrastructure Management Process (6.2.2)**

**Portfolio Management Process**

**Life Cycle Model**

**(6.2.3)**

**(Clause 6.2.1)**

**Infrastructure**

**Human Resource Management Infrastructure**

**Process (6.2.4)**

**Quality Management Process**

**Project Portfolio**

**(6.2.5)**

**Project Portfolio**

**(Clause 6.2.3)**

**Knowledge Management Process Human Resource**

**(6.2.6)**

**Project Assessment and Control Process (6.3.2)**

**Decision Management Process (Clause 6.3.2)**

**(Clause 6.3.2)**

**(6.3.3)**

**Risk Management Process (6.3.4)**

**Configuration Management Process (6.3.5)**

**Information Management Process (6.3.6)**

**Information Management**

**Information Management**

**Measurement Process (6.3.7) Process**

**Process**

**Quality Assurance Process (6.3.8)**

**Stakeholder Needs and**

**Requirements Definition Process (6.4.2)**

**Systems/Software Requirements Definition Process (6.4.3)**

**Architecture Definition Process (6.4.4)**

**Design Definition Process (6.4.5) System Analysis Process (6.4.6) Implementation Process (6.4.7) Integration Process (6.4.8) Verification Process (6.4.9) Transition Process (6.4.10) Validation Process (6.4.11) Operation Process (6.4.12) Maintenance Process (6.4.13) Disposal Process (6.4.14)**

**Figure 4 —Software life cycle processes**

**5.6.2 Agreement processes**

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

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

21

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

**5.6.3 Organizational project-enabling processes**

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

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

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

**5.6.4 Technical Management processes**

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

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

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

NOTE 2 Technical Management processes are applied during the performance of each Technical process. **5.6.5 Technical processes**

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

**5.7 Process application**

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

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

22

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

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

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

The changing nature of the influences on the software system (e.g., operational environment changes, new opportunities for system element implementation, modified structure and responsibilities in organizations) requires continual review of the selection and timing of process use. Process use in the life cycle can be dynamic, responding to the many external influences on the software system. The life cycle approach also allows for incorporating the changes in the next stage. The life cycle stages assist the planning, execution and management of life cycle processes in the face of this complexity in life cycles by providing comprehensible and recognizable high‐

level purpose and structure. The set of processes within a life cycle stage are applied with the common goal of satisfying the exit criteria for that stage or the entry criteria of the formal progress reviews within that stage.

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

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

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

**5.8 Process reference model**

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

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

23

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

**6 Software life cycle processes**

**6.1 Agreement processes**

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

The Agreement processes consist of the following:

a) Acquisition process – used by organizations for acquiring products or services; and

b) Supply process – used by organizations for supplying products or services.

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

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

**6.1.1 Acquisition process**

**6.1.1.1 Purpose**

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

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

**6.1.1.2 Outcomes**

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

**6.1.1.3 Activities and tasks**

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

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

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

24

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

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

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

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

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

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

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

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

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

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

b) **Advertise the acquisition and select the supplier.** This activity consists of the following tasks: 1) Communicate the request for the supply of a product or service to potential suppliers; and 2) Select one or more suppliers.

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

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

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

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

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

25

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

2) Identify necessary changes to the agreement.

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

3) Evaluate impact of changes on the agreement.

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

4) Negotiate the agreement with the supplier.

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

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

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

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

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

1) Assess the execution of the agreement.

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

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

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

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

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

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

2) Provide payment or other agreed consideration.

3) Accept the product or service from the supplier, or other party, as directed by the agreement. 4) Close the agreement.

NOTE The project is closed by the Portfolio Management process.

26

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**6.1.2 Supply process 6.1.2.1 Purpose**

**ISO/IEC/IEEE 12207:2017(E)**

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

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

**6.1.2.2 Outcomes**

| As a result of the successful implementation of the Supply process:  a) An acquirer for a product or service is identified.  b) A response to the acquirer’s request is produced.  c) An agreement is established between the acquirer and supplier.  d) A product or service is provided.  e) Supplier obligations defined in the agreement are satisfied.  f) Responsibility for the acquired product or service, as directed by the agreement, is transferred. |
| --- |

**6.1.2.3 Activities and tasks**

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

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

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

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

2) Define a supply strategy.

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

b) **Respond to a request for supply of products or services**. This activity consists of the following tasks: 1) Evaluate a request for the supply of a product or service) to determine feasibility and how to respond. 2) Prepare a response that satisfies the solicitation.

c) **Establish and maintain an agreement.** This activity consists of the following tasks: 1) Negotiate an agreement with the acquirer that includes acceptance criteria.

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

27

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

2) Identify necessary changes to the agreement.

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

3) Evaluate impact of changes on the agreement.

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

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

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

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

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

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

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

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

2) Assess the execution of the agreement.

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

e) **Deliver and support the product or service.** This activity consists of the following tasks: 1) Deliver the product or service in accordance with the agreement criteria.

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

2) Provide assistance to the acquirer in support of the delivered product or service, per the agreement. 3) Accept and acknowledge payment or other agreed consideration.

4) Transfer the product or service to the acquirer, or other party, as directed by the agreement. 5) Close the agreement.

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

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

**6.2 Organizational Project-Enabling processes**

The Organizational Project‐Enabling processes help ensure the organization’s capability to acquire and supply products or services through the initiation, support and control of projects. These processes provide resources and infrastructure necessary to support projects and help ensure the satisfaction of organizational objectives and 28

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

The Organizational Project‐Enabling Processes consist of the following:

a) Life Cycle Model Management process;

b) Infrastructure Management process;

c) Portfolio Management process;

d) Human Resource Management process;

e) Quality Management process; and

f) Knowledge Management process.

**6.2.1 Life cycle model management process**

**6.2.1.1 Purpose**

| The purpose of the Life Cycle Model Management process is to define, maintain, and assure availability of policies, life cycle processes, life cycle models, and procedures for use by the organization with respect to the scope of this document.  This process provides life cycle policies, processes, models, and procedures that are consistent with the organization’s objectives, that are defined, adapted, improved and maintained to support individual project needs within the context of the organization, and that are capable of being applied using effective, proven methods and tools. |
| --- |

**6.2.1.2 Outcomes**

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

**6.2.1.3 Activities and tasks**

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

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

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

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

29

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

2) Establish the processes that implement the requirements of this document and that are consistent with organizational strategies.

3) Define the roles, responsibilities, accountabilities, and authorities to facilitate implementation of processes and the strategic management of life cycles.

4) Define business criteria that control progression through the life cycle.

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

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

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

3. The life cycle processes and activities are selected, tailored as appropriate and employed in a stage to fulfill the purpose and outcomes of that stage.

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

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

1) Monitor process execution across the organization.

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

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

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

3) Identify improvement opportunities from assessment results.

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

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

1) Prioritize and plan improvement opportunities.

2) Implement improvement opportunities and inform relevant stakeholders.

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

**6.2.2 Infrastructure Management process**

**6.2.2.1 Purpose**

| The purpose of the Infrastructure Management process is to provide the infrastructure and services to projects to support organization and project objectives throughout the life cycle.  This process defines, provides and maintains the facilities, tools, and communications and information technology assets needed for the organization’s business with respect to the scope of this document. |
| --- |

30

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**6.2.2.2 Outcomes**

**ISO/IEC/IEEE 12207:2017(E)**

| As a result of the successful implementation of the Infrastructure Management process: a) The requirements for infrastructure are defined.  b) The infrastructure elements are identified and specified.  c) Infrastructure elements are developed or acquired.  d) The infrastructure is available. |
| --- |

**6.2.2.3 Activities and tasks**

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

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

1) Define project infrastructure requirements.

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

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

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

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

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

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

1) Evaluate the degree to which delivered infrastructure resources satisfy project needs.

2) Identify and provide improvements or changes to the infrastructure resources as the project requirements change.

**6.2.3 Portfolio Management process**

**6.2.3.1 Purpose**

| The purpose of the Portfolio Management process is to initiate and sustain necessary, sufficient and suitable projects in order to meet the strategic objectives of the organization.  This process commits the investment of adequate organization funding and resources, and sanctions the authorities needed to establish selected projects. It performs continued assessment of projects to confirm they justify, or can be redirected to justify, continued investment. |
| --- |

31

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

**6.2.3.2 Outcomes**

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

**6.2.3.3 Activities and tasks**

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

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

1) Identify potential new or modified capabilities or missions.

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

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

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

3) Define projects, accountabilities and authorities.

4) Identify the expected goals, objectives, and outcomes of each project.

5) Identify and allocate resources for the achievement of project goals and objectives.

6) Identify multi‐project interfaces and dependencies to be managed or supported by each project.

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

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

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

32

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

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

1) Evaluate projects to confirm ongoing viability.

NOTE Viability includes the following criteria:

i) The project is making progress towards achieving established goals and objectives.

ii) The project is complying with project directives.

iii) The project is being conducted according to approved project life cycle policies, processes, and procedures.

iv) The project remains viable, as indicated by, for example, continuing need for the service, practicable product implementation, and acceptable investment benefits.

2) Act to continue or redirect projects that are satisfactorily progressing or can be expected to progress satisfactorily by appropriate redirection.

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

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

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

2) After completion of the agreement for products and services, act to close the projects. NOTE Closure is accomplished in accordance with organizational policies and procedures, and the agreement. **6.2.4 Human Resource Management process**

**6.2.4.1 Purpose**

| The purpose of the Human Resource Management process is to provide the organization with necessary human resources and to maintain their competencies, consistent with business needs.  This process provides a supply of skilled and experienced personnel qualified to perform life cycle processes to achieve organization, project, and stakeholder objectives. |
| --- |

**6.2.4.2 Outcomes**

| As a result of the successful implementation of the Human Resource Management process: a) Skills required by projects are identified.  b) Necessary human resources are provided to projects.  c) Skills of personnel are developed, maintained or enhanced.  d) Conflicts in multi‐project resource demands are resolved. |
| --- |

33

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

**6.2.4.3 Activities and tasks**

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

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

1) Identify skill needs based on current and expected projects.

2) Identify and record skills of personnel.

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

1) Establish skills development strategy.

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

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

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

3) Provide planned skill development.

4) Maintain records of skill development.

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

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

1) Obtain qualified personnel when skill deficits are identified.

NOTE This includes using outsourced resources.

2) Maintain and manage the pool of skilled personnel necessary to staff ongoing projects. 3) Make project assignments based on project and staff‐development needs.

4) Motivate personnel, e.g., through career development and reward mechanisms.

5) Control multi‐project management interfaces to resolve personnel conflicts.

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

**6.2.5 Quality Management process**

**6.2.5.1 Purpose**

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

**6.2.5.2 Outcomes**

| As a result of the successful implementation of the Quality Management process:  a) Organizational quality management policies, objectives, and procedures are defined and implemented. |
| --- |

34

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

**6.2.5.3 Activities and tasks**

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

NOTE Refer to ISO 9001:2015 for information and requirements to establish a Quality Management System. a) **Plan quality management.** This activity consists of the following tasks:

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

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

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

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

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

3) Define quality evaluation criteria and methods.

4) Provide resources and information for quality management.

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

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

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

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

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

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

c) **Perform corrective and preventive action.** This activity consists of the following tasks: 1) Plan corrective actions when quality management objectives are not achieved.

2) Plan preventive actions when there is a sufficient risk that quality management objectives will not be achieved.

3) Monitor corrective and preventive actions to completion and inform relevant stakeholders.

35

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

**6.2.6 Knowledge Management process**

**6.2.6.1 Purpose**

| The purpose of the Knowledge Management process is to create the capability and assets that enable the organization to exploit opportunities to re‐apply existing knowledge.  This encompasses knowledge, skills, and knowledge assets, including system elements. |
| --- |

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

**6.2.6.2 Outcomes**

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

**6.2.6.3 Activities and tasks**

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

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

1) Define the knowledge management strategy.

NOTE 1 This knowledge management strategy generally includes:

i) Identifying domains and their potential for the reapplication of knowledge.

ii) Plans for obtaining and maintaining knowledge, skills, and knowledge assets for their useful life.

iii) Characterization of the types of knowledge, skills, and knowledge assets to be collected and maintained.

iv) Criteria for accepting, qualifying, and retiring knowledge, skills, and knowledge assets. v) Procedures for controlling changes to the knowledge, skills, and knowledge assets.

vi) Plans, mechanisms, and procedures for protection, control, and access to classified or sensitive data and information.

vii) Mechanisms for storage and retrieval.

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

36

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

2) Identify the knowledge, skills, and knowledge assets to be managed.

3) Identify projects that can benefit from the application of the knowledge, skills, and knowledge assets. b) **Share knowledge and skills throughout the organization.** This activity consists of the following tasks:

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

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

2) Capture or acquire knowledge and skills.

3) Share knowledge and skills across the organization.

c) **Share knowledge assets throughout the organization.** This activity consists of the following tasks: 1) Establish a taxonomy to organize knowledge assets.

NOTE 1 The taxonomy includes the following:

i) Definition of the boundaries of domains and their relationships to others.

ii) Domain models capturing essential common and different features, capabilities, concepts, and functions.

iii) An architecture for a family of systems within the domain, including their common and different features.

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

2) Develop or acquire knowledge assets.

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

3) Share knowledge assets across the organization.

NOTE Automated search capabilities improve access to knowledge assets.

d) **Manage knowledge, skills, and knowledge assets.** This activity consists of the following tasks: 1) Maintain knowledge, skills, and knowledge assets.

2) Monitor and record the reuse of knowledge, skills, and knowledge assets.

3) Periodically reassess the currency of technology and market needs for the knowledge assets. NOTE Assess the business benefits which the organization gained through the use of knowledge management practices. **6.3 Technical Management processes**

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

37

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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

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

The Technical Management processes consist of the following:

a) Project Planning process;

b) Project Assessment and Control process;

c) Decision Management process;

d) Risk Management process;

e) Configuration Management process;

f) Information Management process;

g) Measurement process; and

h) Quality Assurance process.

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

**6.3.1 Project Planning process**

**6.3.1.1 Purpose**

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

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

**6.3.1.2 Outcomes**

| As a result of the successful implementation of the Project Planning process:  a) Objectives and plans are defined.  b) Roles, responsibilities, accountabilities, and authorities are defined.  c) Resources and services necessary to achieve the objectives are formally requested and committed. d) Plans for the execution of the project are activated. |
| --- |

38

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**6.3.1.3 Activities and tasks**

**ISO/IEC/IEEE 12207:2017(E)**

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

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

1) Identify the project objectives and constraints.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

39

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

2) Define achievement criteria for the life cycle stage decision gates, delivery dates and major dependencies on external inputs or outputs.

NOTE The time intervals between internal reviews are defined in accordance with organizational policy on issues such as business and system criticality, schedule and technical risks.

3) Define the costs and plan a budget.

NOTE Budgeted costs are based on the schedule, software size and complexity estimates, labor estimates, infrastructure costs, procurement items, acquired service and enabling system estimates, and budget reserves for risk management.

4) Define roles, responsibilities, accountabilities, and authorities.

NOTE This includes defining the project organization, staff acquisitions, and the development of staff skills. Authorities include, as appropriate, the legally responsible roles and individuals, e.g., design authorization, safety authorization, and those responsible for applicable certifications or accreditations.

5) Define the infrastructure and services required.

NOTE This includes defining the capacity needed, its availability and its allocation to project tasks. Infrastructure includes facilities, services, tools, communications, and information technology assets. The requirements for enabling systems and services for each life cycle stage are also specified.

6) Plan the acquisition of materials and enabling systems and services supplied from outside the project.

NOTE 1 This includes, as necessary, plans for solicitation, supplier selection, acceptance, contract administration and contract closure. The agreement processes are used for the planned acquisitions.

NOTE 2 ISO/IEC 27036, *Information security for supplier relationships*, provides guidance for acquisition of infrastructure and services.

7) Generate and communicate a plan for project and technical management and execution, including reviews.

NOTE 1 Technical planning for the software system is often captured in a Systems Engineering Management Plan (SEMP) or a Software Engineering Management Plan or a Software Development Plan (SDP). ISO/IEC/IEEE 24748‐5 provides more detail on software engineering technical management planning and includes an annotated outline for an SDP. Planning for the project is often captured in a Project Management Plan. ISO/IEC/IEEE 16326 provides more detail on project planning.

NOTE 2 The strategy activities and tasks from each of the other processes provide inputs and are integrated in the Project Planning process. The Project Assessment and Control process is used to help ensure that the plans are integrated, aligned, and feasible.

c) **Activate the project.** This activity consists of the following tasks:

1) Obtain approval to start the project.

NOTE Approval to start (authorization to proceed) is provided through the Portfolio Management process. 2) Submit requests and obtain commitments for necessary resources to perform the project. NOTE This includes access to enabling systems or services.

3) Implement project plans.

**6.3.2 Project assessment and control process**

**6.3.2.1 Purpose**

| The purpose of the Project Assessment and Control process is to assess if the plans are aligned and feasible; determine the status of the project, technical and process performance; and direct execution to help ensure that the performance is according to plans and schedules, within projected budgets, to satisfy technical objectives. |
| --- |

40

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

| This process evaluates, periodically and at major events, the progress and achievements against requirements, plans and overall business objectives. Information is provided for management action when significant variances are detected. This process also includes redirecting the project activities and tasks, as appropriate, to correct identified deviations and variations from other technical management or technical processes. Redirection may include re‐planning as appropriate. |
| --- |

**6.3.2.2 Outcomes**

| As a result of the successful implementation of the Project Assessment and Control process: a) Performance measures or assessment results are available.  b) Adequacy of roles, responsibilities, accountabilities, and authorities is assessed.  c) Adequacy of resources is assessed.  d) Technical progress reviews are performed.  e) Deviations in project performance from plans are investigated and analyzed.  f) Affected stakeholders are informed of project status.  g) Corrective action is defined and directed, when project achievement is not meeting targets. h) Project replanning is initiated, as necessary.  i) Project action to progress (or not) from one scheduled milestone or event to the next is authorized. j) Project objectives are achieved. |
| --- |

**6.3.2.3 Activities and tasks**

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

a) **Plan for project assessment and control**. This activity consists of the following task: 1) Define the project assessment and control strategy.

NOTE The strategy identifies the expected Project Assessment and Control activities, including planned assessment methods and timeframes, and necessary management and technical reviews.

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

1) Assess alignment of project objectives and plans with the project context.

2) Assess management and technical plans against objectives to determine adequacy and feasibility.

3) Assess project and technical status against appropriate plans to determine actual and projected cost, schedule, and performance variances.

4) Assess the adequacy of roles, responsibilities, accountabilities, and authorities.

NOTE This includes assessment of the adequacy of personnel competencies to perform project roles and accomplish project tasks. Objective measures are used wherever possible, e.g., efficiency of resource use, project achievement.

5) Assess the adequacy and availability of resources.

41

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

NOTE Resources include infrastructure, personnel, funding, time, or other pertinent items. This task includes evaluating the reuse of existing processes and infrastructure resources, and confirming that intra‐organizational commitments are satisfied.

6) Assess progress using measured achievement and milestone completion.

NOTE This includes collecting and evaluating data for labor, material, service costs, and technical performance, as well as other technical data about objectives, such as affordability. These are compared against measures of achievement. This includes conducting effectiveness assessments to determine the adequacy of the evolving software system against requirements. It also includes the readiness of enabling systems to deliver their services when needed.

7) Conduct required management and technical reviews, audits and inspections.

NOTE These are formal or informal, and are conducted to determine readiness to proceed to the next stage of the life cycle or project milestone, to help ensure that project and technical objectives are being met, or to obtain feedback from stakeholders.

8) Monitor critical processes and new technologies.

NOTE This includes identifying and evaluating technology maturity and feasibility of technology insertion. Technology maturity is the readiness of a technology for operational use, and is often measured on a scale from low (exists as a concept only) to high (proven in operational use).

9) Analyze measurement results and make recommendations.

NOTE Measurement results are analyzed to identify deviations, variations or undesirable trends from planned values that include potential concerns, and to make appropriate recommendations for corrections or preventive actions. This includes, where appropriate, statistical analysis of measures that indicates trends, e.g., fault density to indicate quality of outputs, distribution of measured parameters that indicate process repeatability.

10) Record and provide status and findings from assessment tasks.

NOTE These are generally designated in the agreement, policies and procedures.

11) Monitor process execution within the project.

NOTE This includes the analysis of process measures and review of trends with respect to project objectives. Any improvement actions identified can be handled through the Quality Assurance process, the Quality Management process, or the Life Cycle Model Management process.

c) **Control the project.** This activity consists of the following tasks:

1) Initiate necessary actions needed to address identified issues.

NOTE 1 This task occurs when project or technical achievement is not meeting planned targets. This includes preventive, corrective, and problem resolution actions. Actions generally require replanning or reassignment of personnel, tools and infrastructure assets when inadequacy or unavailability has been detected, or when project or technical achievement exceeds targets or plan. They often impact the cost, schedule, or technical scope or definition. Actions sometimes require changes to the implementation and execution of the life cycle processes.

NOTE 2 Actions are recorded and reviewed to confirm their adequacy and timeliness.

2) Initiate necessary project replanning.

NOTE 1 Project replanning is initiated when project objectives or constraints have changed, or when planning assumptions are shown to be invalid.

NOTE 2 Any change that requires a change to the agreement between acquirer and supplier invokes the Acquisition and Supply processes.

3) Initiate change actions when there is a contractual change to cost, time or quality due to the impact of an acquirer or supplier request.

NOTE This includes consideration of modified terms and conditions for supply or initiating new supplier selection, which invokes the Acquisition and Supply processes.

42

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

4) Authorize the project to proceed toward the next milestone or event, if justified.

NOTE The Project Assessment and Control process is used to reach agreement on milestone completion. **6.3.3 Decision Management process**

**6.3.3.1 Purpose**

| The purpose of the Decision Management process is to provide a structured, analytical framework for objectively identifying, characterizing and evaluating a set of alternatives for a decision at any point in the life cycle and select the most beneficial course of action. |
| --- |

This process is used to resolve technical or project issues and respond to requests for decisions encountered during the software life cycle, in order to identify the alternative(s) that provides the preferred outcomes for the situation. The methods most frequently used for Decision Management are the trade study and engineering analysis. Each of the alternatives is assessed against the decision criteria (e.g., cost impact, schedule impact, programmatic constraints, regulatory implications, technical performance characteristics, critical quality characteristics, and risk). Results of these comparisons are ranked, via a suitable selection model, and are then used to decide on an optimal solution. Key study data (e.g., assumptions and decision rationale) are typically maintained to inform decision‐makers and support future decision‐making.

NOTE When it is necessary to perform a detailed assessment of a parameter for one of the criteria, the System Analysis process can be employed to perform the assessment.

**6.3.3.2 Outcomes**

| As a result of the successful implementation of the Decision Management process:  a) Decisions requiring alternative analysis are identified.  b) Alternative courses of action are identified and evaluated.  c) A preferred course of action is selected.  d) The resolution, decision rationale and assumptions are identified. |
| --- |

**6.3.3.3 Activities and tasks**

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

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

1) Define a decision management strategy.

NOTE A decision management strategy includes the identification of roles, responsibilities, accountabilities, and authorities. The strategy considers the need for obtaining information input and for returning a timely decision. It includes the identification of decision categories and a prioritization scheme. Decisions often arise as a result of an effectiveness assessment, a technical trade‐off, a problem needing to be solved, an action needed as a response to risk exceeding the acceptable threshold, or a new opportunity or approval for project progression to the next life cycle stage. Organization or project guidelines determine the degree of rigor and formality to apply to the decision analysis.

2) Identify the circumstances and need for a decision.

NOTE Problems or opportunities and the alternative courses of action that will resolve their outcome are recorded, categorized and reported.

3) Involve relevant stakeholders in the decision‐making in order to draw on experience and knowledge. NOTE It is good practice to identify the subject matter expertise needed for the analysis and the decision.

43

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

b) **Analyze the decision information.** This activity consists of the following tasks:

1) Select and declare the decision management strategy for each decision.

NOTE The degree of rigor required to resolve these problems or opportunities is determined, as well as the data and system analysis needed for evaluating the alternatives. Define the timeframe to reach a decision.

2) Determine desired outcomes and measurable selection criteria.

NOTE The desired value for quantifiable criteria and the threshold value(s) beyond which the attribute will be unsatisfactory are determined, as well as weighting factors for the criteria.

3) Identify the trade space and alternatives.

NOTE If a large number of alternatives exist, they are qualitatively screened to reduce alternatives to a manageable number for further detailed systems analysis. This screening is often based on qualitative assessments of such factors as risk, cost, schedule, and regulatory impacts.

4) Evaluate each alternative against the criteria.

NOTE The System Analysis process is used, as necessary, to quantify specific criteria for each trade alternative to be evaluated. This includes new design parameters, different architecture characteristics, and range of values for critical quality characteristics. The System Analysis process assesses the range of parameter variations in order to obtain a sensitivity analysis for each of the trade alternatives evaluated. These results are used to establish the feasibility of the various trade alternatives.

c) **Make and manage decisions.** This activity consists of the following tasks:

1) Determine preferred alternative for each decision.

NOTE Alternatives are evaluated quantitatively, using the selection criteria. The selected alternative generally provides an optimization of, or improvement in, an identified decision.

2) Record the resolution, decision rationale, and assumptions.

3) Record, track, evaluate and report decisions.

NOTE 1 This includes records of problems and opportunities and their disposition, as stipulated in agreements or organizational procedures and in a manner that permits auditing and learning from experience.

NOTE 2 This allows the organization to confirm that problems have been effectively resolved, that adverse trends have been reversed, and that advantage has been taken of opportunities.

**6.3.4 Risk Management process**

**6.3.4.1 Purpose**

| The purpose of the Risk Management process is to identify, analyze, treat and monitor the risks continually.  The Risk Management process is a continual process for systematically addressing risk throughout the life cycle of a system product or service. It can be applied to risks related to the acquisition, development, maintenance or operation of a system. |
| --- |

NOTE Risk is defined in ISO Guide 73:2009 as “The effect of uncertainty on objectives”. This has an attached Note 1, “An effect is a deviation from the expected — positive and/or negative”. A positive risk is commonly known as an opportunity, and can be addressed within the Risk Management process.

**6.3.4.2 Outcomes**

| As a result of the successful implementation of the Risk Management process:  a) Risks are identified. |
| --- |

44

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

| b) Risks are analyzed.  c) Risk treatment options are identified, prioritized, and selected.  d) Appropriate treatment is implemented.  e) Risks are evaluated to assess changes in status and progress in treatment. |
| --- |

**6.3.4.3 Activities and tasks**

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

NOTE ISO/IEC/IEEE 16085 provides a more detailed set of risk management activities and tasks. This risk management process is aligned with ISO 31000:2009 *Risk management — Principles and Guidelines*, and ISO Guide 73:2009, *Risk management — Vocabulary.* ISO 9001:2015 describes planning for risks and opportunities in 6.1.

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

1) Define the risk management strategy.

NOTE This includes the risk management process of supply chain suppliers and describes how risks from suppliers will be raised to the next level(s) for incorporation in the project risk process.

2) Define and record the context of the Risk Management process.

NOTE 1 This includes a description of stakeholders’ perspectives, risk categories, and a description (perhaps by reference) of the technical and managerial objectives, assumptions and constraints. The risk categories include the relevant technical areas of the software system and facilitate identification of risks across the product life cycle. As noted in ISO 31000, the aim of this step is to generate a comprehensive list of risks based on those events that might create, enhance, prevent, degrade, accelerate, or delay the achievement of objectives.

NOTE 2 Opportunities, which are one type of risk, provide potential benefits for the software system or project. Each of the opportunities pursued has associated risks that detract from the expected benefit. This includes the risks associated with not pursuing an opportunity, as well as the risk of not achieving the effects of the opportunity.

b) **Manage the risk profile.** This activity consists of the following tasks:

1) Define and record the risk thresholds and conditions under which a level of risk may be accepted. 2) Establish and maintain a risk profile.

NOTE The risk profile records: the risk management context; a record of each risk’s state including its likelihood of occurrence, consequences, and risk thresholds; the priority of each risk based on risk criteria supplied by the stakeholders; and the risk action requests along with the status of their treatment. The risk profile is updated when there are changes in an individual risk’s state. The priority in the risk profile is used to determine the application of resources for treatment.

3) Periodically provide the relevant risk profile to stakeholders based upon their needs. c) **Analyze risks.** This activity consists of the following tasks:

1) Identify risks in the categories described in the risk management context.

NOTE Risks are commonly identified through various analyzes, such as safety, reliability, security, and performance analyzes; technology, architecture, and readiness assessments; and trade studies. These risks are often identified early in the life cycle and continue into the utilization, support, and retirement of the software system. Additionally, risks are often identified through the analysis of measurements of the evolving software system.

2) Estimate the likelihood of occurrence and consequences of each identified risk.

NOTE Consequences of a risk typically involve technical, schedule, cost, or quality impacts.

45

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

3) Evaluate each risk against its risk thresholds.

4) For each risk that does not meet its risk threshold, define and record recommended treatment strategies and measures.

NOTE Risk treatment strategies include, but are not limited to, eliminating the risk, reducing its likelihood of occurrence or severity of consequence, or accepting the risk. Treatments also include taking or increasing risk in order to pursue an opportunity. Measures provide information about the effectiveness of the treatment alternatives.

d) **Treat risks.** This activity consists of the following tasks:

1) Identify recommended alternatives for risk treatment.

2) Implement risk treatment alternatives for which the stakeholders determine that actions should be taken to make a risk acceptable.

3) When the stakeholders accept a risk that does not meet its threshold, consider it a high priority and monitor it continually to determine if future risk treatment actions are necessary or if its priority has changed.

4) Once a risk treatment is selected, coordinate management action.

NOTE The Project Assessment and Control process can be applied.

e) **Monitor risks.** This activity consists of the following tasks:

1) Continually monitor risks and the risk management context for changes and evaluate the risks when their state has changed.

2) Implement and monitor measures to evaluate the effectiveness of risk treatments.

3) Continually monitor for the emergence of new risks and sources throughout the life cycle. **6.3.5 Configuration Management process**

**6.3.5.1 Purpose**

| The purpose of Configuration Management is to manage and control system elements and configurations over the life cycle. Configuration Management (CM) also manages consistency between a product and its associated configuration definition. |
| --- |

Software configuration management (SCM) applies to both the software system and its interfaces. The purpose of interface management is to agree with interface partners on the exchange of data through communications among software systems and services. Annex E (see E.5) provides an example of an Interface Management Process View.

Software configurations are changed through the controlled release of a new version. The purpose of a release is to authorize and effect the availability of a software feature, function, or system for a specific purpose, with or without restrictions to a subset of users.

**6.3.5.2 Outcomes**

| As a result of the successful implementation of the Configuration Management process:  a) Items requiring configuration management are identified and managed.  b) Configuration baselines are established.  c) Changes to items under configuration management are controlled.  d) Configuration status information is available. |
| --- |

46

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

| e) Required configuration audits are completed.  f) System releases and deliveries are controlled and approved. |
| --- |

**6.3.5.3 Activities and tasks**

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

NOTE ISO/IEC/IEEE 19770 provides procedures and requirements for an IT asset management system. a) **Plan configuration management.** This activity consists of the following tasks:

1) Define a configuration management strategy, including approaches for the following:

i) Governance of CM, including roles, responsibilities, accountabilities, and authorities, and use of configuration control (change control) boards; and

ii) Consideration of the level of risk and impact in approval of configuration baselines and regular and emergency change requests.

NOTE Regularly scheduled changes to apply software patches using approved procedures or check‐in and check‐out of unit‐tested software elements under development are typically performed automatically, or reviewed and approved daily as a matter of routine. In comparison, significant changes to the software system design with major impact on project cost and schedule can involve extensive analyzes, consultations with suppliers, stakeholder reviews, and approvals at the highest levels of the organization.

iii) Coordination of CM across the set of acquirer, supplier, and supply chain organizations for the life of the software system, or the extent of the agreement or project, as appropriate.

iv) Control of access and changes to and disposition of configuration items.

v) The necessary baselines to be established, including criteria or events for commencing configuration control and maintaining baselines of evolving configurations.

vi) Control of software licenses, data rights, and other intellectual property assets.

vii) Frequency, priorities, and content of software versions and releases.

viii) The audit strategy and the responsibilities for validating continuous integrity and security of the configuration definition information.

ix) Change management, including preparing stakeholders and especially users for changes in operational software systems and services.

NOTE 1 For complex software systems, trade‐off studies are performed, e.g., to select an appropriate automated tool to support SCM needs and scope as identified in the strategy.

NOTE 2 Additional guidance regarding configuration management activities can be found in ISO 10007, IEEE Std 828, and SAE ANSI/EIA‐649‐B.

NOTE 3 The SWEBOK, Guide to the Software Engineering Body of Knowledge, provides detailed discussion on SCM. This knowledge area addresses SCM in the context of a system, SCM project and process planning, SCM plan and outline, tool selection, subcontractor control, surveillance and other audits, software configuration items and relationships, software libraries, and Configuration Management process activities.

NOTE 4 The SCM strategy is commonly documented in a plan, e.g., a configuration management plan, or sometimes in a project’s SEMP, SDP, or Project Management Plan (PMP). The strategy planning for Configuration Management is coordinated through the Project Planning process. In establishing points to establish baselines and conduct audits, CM planning is aligned with the software life cycle. The frequency of recurring SCM activities aligns with the iteration of technical processes and stages. SCM planning typically includes deciding when to review Configuration Management planning, what conditions require updating the CM plan, and who is authorized to change the CM plans and items held in configuration control.

47

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

2) Define the storage, archive and retrieval procedures for configuration items, CM artifacts, and records.

NOTE The locations and conditions of storage for software system configuration items, such as source code and executable software, are established in accordance with designated levels of integrity, security and safety.

b) **Perform configuration identification.** This activity consists of the following tasks:

1) Select the software system elements to be uniquely identified as configuration items subject to configuration control.

EXAMPLE Configuration items subject to configuration control in software systems usually include system/software requirements specifications, interface specifications; product and system elements (e.g., software objects, hardware, and services) while under development, baseline configurations or software versions established for transition between stages and as released for operational use; master (“gold”) copies of source code or executable software for different platforms or versions; site‐specific configurations in operational use; and information items, such as agreements, architecture models, service descriptions and operational procedures; and items in enabling systems.

NOTE 1 Unique identification can be applied to software components, versions, or to individually licensed copies. The identifiers are in accordance with relevant standards and product sector conventions, such that the items under configuration control are unambiguously traceable to their supplier and to their specifications or equivalent recorded descriptions. Information items are often identified and managed separately from other configuration items.

NOTE 2 Software configuration identifiers facilitate traceability when more than one developer or maintenance programmer is working on the same software function, so that various branches of code can be successfully reassembled and tested.

NOTE 3 The ISO/IEC 19770 standard (multiple parts) provides an IT Asset management system for tracking software licenses.

2) Identify the attributes of configuration items.

NOTE 1 Attributes refers to item status, or physical or logical features useful for managing or maintaining the software system. Appropriate attributes can differ for hardware and software configuration items.

NOTE 2 Configuration attributes and identifiers can reflect a decomposition of the software system, so that configuration items are tracked at the level at which change needs to be controlled.

EXAMPLE Software from external suppliers can be tracked by its license and maintenance agreement, which can involve tracking to the location, number or size of systems where it is used or the number of concurrent users allowed. Software versions can be traced to the stakeholder requirements which they implement.

3) Define baselines through the life cycle.

NOTE 1 Baselines capture the evolving configuration states of software system elements at designated times or under defined circumstances. The content for the baselines is developed through the technical processes, but is formalized at a point in time through the Configuration Management process. Baselines form the basis for subsequent changes. Selected baselines typically become formalized between acquirer and supplier, depending on the practices of the industry and the contractual involvement of the acquirer in the configuration management process. There are generally three major types of baselines at the system level: functional baseline, allocated baseline, and product baseline. These vary by domain or local strategy.

NOTE 2 Software systems development often entails establishing multiple developmental baselines to address evolving software configuration needs at key points in the life cycle, e.g., to allow for simultaneous control of software versions during software design, prototype, integration, and test releases. This can involve distributed configuration management responsibilities and access limitations for archives, e.g., software development or test libraries and a master configuration support library.

4) Obtain acquirer and supplier agreement to establish a baseline.

NOTE The Project Assessment and Control process is used to reach agreement. When software is being developed for commercial or internal use, the acquirer or project sponsor can be the authority to approve a baseline.

c) **Perform configuration change management** This activity consists of the following tasks:

48

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

NOTE Configuration change management establishes procedures and methods for managing change to a baseline once it is established. This is sometimes referred to as configuration control. The term ‘change management’ is also used for managing change to organizational procedures and business workflow.

1) Identify and record Requests for Change and Requests for Variance.

NOTE A request for variance is often referred to as a deviation, waiver, or concession.

2) Coordinate, evaluate, and disposition Requests for Change and Requests for Variance.

NOTE 1 Evaluation commonly includes analysis of rationale and need versus impact on the software and interoperating systems, considering risks and opportunities, quality, users, schedule, and cost. A decision is made on whether to implement or deny the change request.

NOTE 2 Requests for Change and Requests for Variance are often under the formal control of a Configuration Control Board (CCB).

3) Track and manage approved changes to the baseline, Requests for Change and Requests for Variance.

NOTE 1 This task involves prioritization, tracking, scheduling, and closing changes. Changes are then made through the Technical Processes. These changes are verified or validated through the Verification and Validation processes, to help ensure that the approved changes have been correctly applied.

NOTE 2 Changes and rationales are typically recorded when approved and when completed.

d) **Perform release control.** This activity consists of the following tasks:

1) Identify and record release requests, identifying the software system elements in a release.

NOTE 1 The life cycle model helps determine the frequency of iterative or incremental software releases. The Integration process is used to select and configure a release package, software version, update or patch. These changes are verified or validated through the Verification or Validation processes. Changes are made through the Technical processes, particularly Transition.

EXAMPLE Releases for a software test, for software or system qualification or other formal tests, or for trial (beta) or operational use.

2) Approve software system releases and deliveries.

NOTE 1 Releases often involve prioritization, tracking, scheduling, and closing changes. Approval of a release for operational use can include acceptance of the verified and validated changes. Criteria for approval of a release often includes rollback plans or contingency plans in the event of an unsuccessful release.

NOTE 2 For software systems, automated version control tools can help ensure that only the correct source code versions are accessed, updated, tested and documented for approved changes by appropriate personnel, and released.

3) Track and manage distribution of software system releases to specified environments or software deliveries.

NOTE Master copies or copies of incremental changes to released software versions can be maintained for the life of the system or project in a controlled environment. Software suppliers often track delivered copies of licensed software to the acquirer in order to provide agreed software maintenance. The software system release is stored and distributed in accordance with agreement and with the policies of the organizations involved.

e) **Perform configuration status accounting.** This activity consists of the following tasks: 1) Develop and maintain the CM status information for software system elements, baselines, and releases.

NOTE 1 Configuration status accounting provides the data on the status of controlled products needed to make decisions regarding system elements throughout the product life cycle. For example, software status can include past, current, and planned progression through stages in the life cycle for software functions and completion of verification and validation activities for software elements. Configuration status information permits forward and backward traceability to other configuration states. Configuration status records are maintained through the software or project life cycle and then archived according to agreements, relevant legislation or organizational practice.

49

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

NOTE 2 The recording, retrieval and consolidation of the current configuration status and the status of previous configurations to confirm information correctness, timeliness, integrity and security is managed. Audits are performed to verify conformance of a baseline to an architecture view, interface control documents, software license agreements, and other agreement requirements.

2) Capture, store and report configuration management data.

f) **Perform configuration evaluation.** This activity consists of the following tasks:

1) Identify the need for CM audits, and schedule the events.

2) Verify the product configuration meets the configuration requirements by comparing requirements, constraints, and waivers (variances) with results of formal verification activities, which can involve sampling methods.

3) Monitor the incorporation of approved configuration changes.

4) Assess whether the software system meets functional and performance capabilities identified for the baseline.

NOTE This is sometimes called a functional configuration audit (FCA), which assures that the product configuration meets specified requirements.

5) Assess whether the operational software system elements conform to the approved configuration information.

NOTE This is sometimes called a physical configuration audit (PCA). For software items, criteria for the PCA can include whether specified configuration items are installed on designated systems according to software license or agreement.

6) Record the CM audit results and disposition action items.

**6.3.6 Information Management process**

**6.3.6.1 Purpose**

| The purpose of the Information Management process is to generate, obtain, confirm, transform, retain, retrieve, disseminate and dispose of information, to designated stakeholders.  Information management plans, executes, and controls the provision of information to designated stakeholders that is unambiguous, complete, verifiable, consistent, modifiable, traceable, and presentable. Information includes technical, project, organizational, agreement, and user information. Information is often derived from data records of the organization, system, process, or project. |
| --- |

NOTE Managed information has these quality characteristics: unambiguous, complete, verifiable, consistent, modifiable, traceable, and presentable.

**6.3.6.2 Outcomes**

| As a result of the successful implementation of the Information Management process:  a) Information to be managed is identified.  b) Information representations are defined.  c) Information is obtained, developed, transformed, stored, validated, presented, and disposed of. d) The status of information is identified.  e) Information is available to designated stakeholders. |
| --- |

50

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**6.3.6.3 Activities and tasks**

**ISO/IEC/IEEE 12207:2017(E)**

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

NOTE ISO/IEC/IEEE 15289 summarizes requirements for the content of life cycle process information items (documentation) and provides guidance on their development.

a) **Prepare for information management.** This activity consists of the following tasks: 1) Define the strategy for information management.

NOTE Information about the same topic can be developed in different ways at different points in the life cycle and for different audiences.

2) Define the items of information that will be managed.

NOTE This includes the information that will be managed during the software life cycle and possibly maintained for a defined period beyond. This is done according to organizational policy, agreements, or legislation.

3) Designate authorities and responsibilities for information management.

NOTE Due regard is paid to information and data legislation, security and privacy, e.g., ownership, agreement restrictions, rights of access to data and ownership of data, intellectual property and patents. Where restrictions or constraints apply, information is identified accordingly. Staff members with knowledge of such items of information are informed of their obligations and responsibilities.

4) Define the content, formats and structure of information items.

NOTE The information originates and terminates in many forms (e.g., audiovisual, textual, graphical, numerical) and mediums (e.g., electronic, printed, magnetic, optical). Organization constraints, e.g., infrastructure, inter‐organizational communications, and distributed project workings, are taken into account. Relevant information item standards and conventions are used according to policy, agreements and legislation constraints.

5) Define information maintenance actions.

NOTE Information maintenance includes status reviews of stored information for integrity, validity and availability. It also includes any needs for replication or transformation to an alternative medium, as necessary, either to retain infrastructure as technology changes so that archived media can be read or to migrate archived media to newer technology.

b) **Perform information management.** This activity consists of the following tasks:

1) Obtain, develop, or transform the identified items of information.

NOTE This includes collecting the data, information, or information items from appropriate sources (e.g., resulting from any life cycle process), and writing, illustrating, or transforming it into usable information for stakeholders. It includes reviewing, validating, and editing information per information standards.

2) Maintain information items and their storage records, and record the status of information.

NOTE 1 Information items are maintained according to their integrity, security and privacy requirements. The status of information items is maintained, (e.g., version description, date of issue or validity date, record of distribution, security classification). Legible information is stored and retained in such a way that it is readily retrievable.

NOTE 2 The source data and tools used to transform information, along with the resulting documentation is placed under configuration control in accordance with the Configuration Management process. ISO/IEC/IEEE 26531 provides requirements for content management systems useful for life cycle information and documentation.

3) Publish, distribute or provide access to information and information items to designated stakeholders.

NOTE Information is provided to designated stakeholders in an appropriate form, as required by agreed schedules or defined circumstances. Information items include documentation used for certification, accreditation, license or assessment ratings, as required.

51

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

4) Archive designated information.

NOTE Archiving is done in accordance with the audit, knowledge retention, and project closure purposes. The media, location and protection of the information are selected in accordance with the specified storage and retrieval periods, and with organization policy, agreements and legislation. Arrangements are put in place to retain necessary information items after project closure.

5) Dispose of unwanted, invalid or unvalidated information.

NOTE This is done according to organization policy, and security and privacy requirements.

**6.3.7 Measurement process**

**6.3.7.1 Purpose**

| The purpose of the Measurement process is to collect, analyze, and report objective data and information to support effective management and demonstrate the quality of the products, services, and processes. |
| --- |

NOTE Measures have these quality characteristics: verifiable, meaningful, actionable, timely, and cost‐effective. **6.3.7.2 Outcomes**

| As a result of the successful implementation of the Measurement process:  a) Information needs are identified.  b) An appropriate set of measures, based on the information needs, is identified or developed. c) Required data is collected, verified, and stored.  d) The data is analyzed and the results interpreted.  e) Information items provide objective information that supports decisions. |
| --- |

**6.3.7.3 Activities and tasks**

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

NOTE 1 ISO/IEC 15939 provides a more detailed set of measurement activities and tasks that are aligned with the activities and tasks in this document.

NOTE 2 ISO 9001:2015 specifies Quality Management System requirements for measurement and monitoring. a) **Prepare for measurement.** This activity consists of the following tasks:

1) Define the measurement strategy.

2) Describe the characteristics of the organization that are relevant to measurement, such as business and technical objectives.

3) Identify and prioritize the information needs.

NOTE The information needs are based on the organization’s business objectives, the project objectives, identified risks, and other items related to project decisions. Measurements can relate to projects, processes, products, or decisions.

4) Select and specify measures that satisfy the information needs.

NOTE Measures are defined that are verifiable and cost‐effective.

5) Define data collection, analysis, access, and reporting procedures.

52

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

6) Define criteria for evaluating the information items and the Measurement process.

7) Identify and plan for the necessary enabling systems or services to be used.

b) **Perform measurement.** This activity consists of the following tasks:

1) Integrate manual or automated procedures for data generation, collection, analysis and reporting into the relevant processes.

NOTE This task can involve change impacts to other life cycle processes to accomplish procedural integration. 2) Collect, store, and verify data.

3) Analyze data and develop information items.

4) Record results and inform the measurement users.

NOTE The measurement analysis results are reported to relevant stakeholders in a timely, usable fashion to support decision making and assist in corrective actions, risk management, and improvements. Results are reported to decision process participants, technical and management review participants, and product and process improvement process owners.

**6.3.8 Quality Assurance process**

**6.3.8.1 Purpose**

| The purpose of the Quality Assurance process is to help ensure the effective application of the organization’s Quality Management process to the project.  Quality Assurance focuses on providing confidence that quality requirements will be fulfilled. Proactive analysis of the project life cycle processes and outputs is performed to assure that the product being produced will be of the desired quality and that organization and project policies and procedures are followed. |
| --- |

**6.3.8.2 Outcomes**

| As a result of the successful implementation of the Quality Assurance process:  a) Project quality assurance procedures are defined and implemented.  b) Criteria and methods for quality assurance evaluations are defined.  c) Evaluations of the project’s products, services, and processes are performed, consistent with quality management policies, procedures, and requirements.  d) Results of evaluations are provided to relevant stakeholders.  e) Incidents are resolved.  f) Prioritized problems are treated |
| --- |

NOTE Outcomes a) through d) align with the outcomes of the Quality Management process activities and tasks. **6.3.8.3 Activities and tasks**

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

NOTE IEEE Std 730‐2014, *Software Quality Assurance Processes*, provides additional detail.

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

53

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

1) Define a Quality Assurance strategy. The strategy is consistent with the organizational Quality Management policies and objectives and includes:

i) Priorities for applying Quality Assurance resources to processes and tasks that have the most significant impact on the quality of the delivered products and services;

ii) Defined roles, responsibilities, accountabilities, and authorities;

iii) Evaluation criteria and methods for processes, products, and services, including criteria for product or service acceptance;

iv) Activities appropriate to each supplier (including subcontractors);

v) Required verification, validation, monitoring, measurement, review, inspection, audit, and test activities specific to the products or services; and

vi) Problem resolution and process and product improvement activities.

NOTE In software projects, activities and tasks that have significant impact on product quality include obtaining agreement on new and changed requirements, performance of peer reviews and unit testing, analysis of problem reports and feedback from users; validating completion of corrective actions assigned at project milestone reviews, and root cause analysis of defects.

2) Establish independence of quality assurance from other life cycle processes.

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

b) **Perform product or service evaluations.** This activity consists of the following tasks:

1) Evaluate products and services for conformance to established criteria, contracts, standards, and regulations.

NOTE This task includes verifying if criteria for product or service acceptance are reflected in verification and validation activities. Derived system/software quality requirements are usually associated with quality characteristics during requirements definition processes. ISO/IEC 25010 and ISO/IEC 25030 provide additional information on system/software quality characteristics.

2) Monitor that verification and validation of the outputs of the life cycle processes are performed to determine conformance to specified requirements.

c) **Perform process evaluations.** This activity consists of the following tasks:

1) Evaluate project life cycle processes for conformance.

2) Evaluate tools and environments that support or automate the process for conformance. 3) Evaluate supplier processes for conformance to process requirements.

NOTE Consider items such as a collaborative software development environment, process measures that suppliers are required to provide, or a risk process that suppliers are required to use. This includes surveillance reviews of process implementation through the supply chain.

d) **Manage QA records and reports.** This activity consists of the following tasks:

1) Create records and reports related to quality assurance activities.

NOTE Create records and reports in accordance with organizational, regulatory, and project requirements, using the information management process.

2) Maintain, store, and distribute records and reports.

3) Identify incidents and problems associated with product, service, and process evaluations.

54

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

NOTE This includes the capture of lessons learned. Responsibilities for resolution are identified. e) **Treat incidents and problems.** This activity consists of the following tasks:

NOTE 1 In the terminology of quality management, problems are often described as “non‐conformities” when, if left untreated, they can cause the project to fail to meet its requirements.

NOTE 2 For additional information and examples of problem categories and priority classifications, see ISO/IEC TS 24748‐1:2016, Annex C.

1) Record, analyze and classify incidents.

2) Identify selected incidents to associate with known errors or problems.

3) Record, analyze and classify problems.

NOTE Analysis results include potential treatment options.

4) Identify root causes and treatment of problems where feasible.

5) Prioritize treatment of problems (problem resolution) and track corrective actions.

NOTE Implementation is done in the Technical processes after initiation by the Project Assessment and Control process. Organizational procedures for problem escalation can help focus resources on lagging problem resolutions.

6) Analyze trends in incidents and problems.

7) Identify improvements in processes and products that may prevent future incidents and problems.

NOTE The Risk Management process is used to treat risks and opportunities. The Life Cycle Model Management process is used to improve the organization’s processes.

8) Inform designated stakeholders of the status of incidents and problems.

9) Track incidents and problems to closure.

**6.4 Technical processes**

The Technical processes are used to define the requirements for a software system, to transform the requirements into an effective product, to permit consistent reproduction of the product where necessary, to use the product to provide the required services, to sustain the provision of those services, and to dispose of the product when it is retired from service.

The Technical processes define the activities that enable organization and project functions to optimize the benefits and reduce the risks that arise from technical decisions and actions. These activities enable software systems and services to possess the timeliness and availability, cost effectiveness, functionality, reliability, maintainability, producibility, usability, and other qualities required by acquiring and supplying organizations. They also enable products and services to conform to the expectations or legislated requirements of society, including health, safety, security, and environmental factors.

The Technical processes consist of the following:

a) Business or Mission Analysis process;

b) Stakeholder Needs and Requirements Definition process;

c) System/Software Requirements Definition process;

d) Architecture Definition process;

e) Design Definition process;

55

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

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; and

n) Disposal process.

NOTE 1 For software systems, these processes can be recursively applied at more inclusive or more detailed levels for software system definition and realization.

NOTE 2 For software systems, these processes are often performed concurrently, iterating between one another to establish a solution that has satisfactory trade‐offs with respect to requirements, critical performance measures, and critical quality characteristics. At any level of abstraction, requirements and models are made consistent via iterations of applicable technical processes. When requirements and models are not directly capable of being implemented, the technical processes are applied recursively at a more detailed level or through different system views.

NOTE 3 The concept of life cycle stages and the application of these processes in any stage are described in detail in ISO/IEC TS 24748‐1. It has a complete set of example stages and stage outcomes for the enactment of technical processes within a software life cycle.

NOTE 4 Interface management is a set of activities that cut across software engineering processes. These cross‐cutting activities of the Technical and Technical Management processes apply and track as a specific view of the processes and software system. See Annex E (E.5) for an example of an Interface Management process view.

NOTE 5 ISO/IEC 27002 *Code of practice for information security controls* and ISO/IEC 27034 *Application security*, provide guidance for applying security concerns in the Technical processes for software systems. See Annex E (E.6) for a sample Software Assurance Process View.

**6.4.1 Business or Mission Analysis process**

**6.4.1.1 Purpose**

| The purpose of the Business or Mission Analysis process is to define the business or mission problem or opportunity, characterize the solution space, and determine potential solution class(es) that could address a problem or take advantage of an opportunity. |
| --- |

NOTE 1 Business and Mission Analysis is related to the organization encompassing stakeholders concerned by the activities of the software life cycle. This process interacts with the organization’s strategy, which is generally outside the scope of ISO/IEC/IEEE 12207. The results of the organization’s strategic analysis include the organizational Concept of Operations, strategic goals and plans, new market or mission elements, and identified problems and opportunities. The organization's strategy establishes the context within which the business or mission analysis is performed. The organizational Concept of Operations relates to the leadership’s intended way of operating the organization. It describes the organization’s assumptions

and how it intends to use, acquire, or supply the system to be developed, existing systems, and possible future systems in support of an overall operation or series of operations of the business. In the case that the organization is the system‐of‐ interest, the organization’s strategy is part of the system definition.

NOTE 2 This process has application through the life of the software system solution and can be revisited if there are changes in the environment, needs, or other drivers.

NOTE 3 In some domains, Business or Mission Analysis relates to the concept of identifying and analyzing capabilities that are needed or desired by the organization. This process focuses on the necessary capabilities and interacts with the Portfolio Management process for identifying the trade space that can address the capability. The identified problems or opportunities

56

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

are often translated into target capabilities. As applicable within a given domain, the problem or opportunity space includes the target capabilities.

**6.4.1.2 Outcomes**

| As a result of the successful implementation of the Business or Mission Analysis process: a) The problem or opportunity space is defined.  b) The solution space is characterized.  c) Preliminary operational concepts and other concepts in the life cycle stages are defined. d) Candidate alternative solution classes are identified and analyzed.  e) The preferred candidate alternative solution class(es) are selected.  f) Any enabling systems or services needed for business or mission analysis are available.  g) Traceability of business or mission problems and opportunities and the preferred alternative solution classes is established. |
| --- |

**6.4.1.3 Activities and tasks**

The project shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Business or Mission Analysis process.

a) **Prepare for Business or Mission Analysis.** This activity consists of the following tasks:

1) Review identified problems and opportunities in the organization strategy with respect to desired organization goals or objectives.

NOTE This includes problems or opportunities with respect to the organization business or mission, vision, Concept of Operations, and other organization strategic goals and objectives. This includes identified deficiencies or gaps in existing capabilities, systems, products, or services.

2) Define the business or mission analysis strategy.

NOTE This includes the approach to be used to identify and define the problem space, characterize the solution space and select a solution class.

3) Identify and plan for the necessary enabling systems or services needed to support business or mission analysis.

NOTE This includes identification of requirements and interfaces for enabling systems. Enabling systems for business or mission analysis include the business systems and repositories of the organization or other accessible entities.

4) Obtain or acquire access to the enabling systems or services to be used.

NOTE The Validation process is used to objectively confirm that the enabling system achieves its intended use for its enabling functions.

b) **Define the problem or opportunity space.** This activity consists of the following tasks: 1) Analyze customer complaints, problems and opportunities in the context of relevant trade‐space factors.

NOTE 1 This analysis is focused on understanding the scope, basis, or drivers of the problems or opportunities, as opposed to the synthesis that is the focus of system analysis and decision management needed for trade studies. The focus here includes changes in mission requirements, business opportunities, capabilities, performance improvement, or lack of existing systems, security and safety improvement, factors such as cost and effectiveness, regulation changes, user dissatisfaction, and PESTEL factors (Political, Economic, Social, Technological, Environmental, and Legal). Relevant factors can be identified through external, internal, or SWOT (Strengths, Weaknesses, Opportunities and Threats) analysis.

57

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

NOTE 2 The outputs of the analysis are considered as part of the portfolio management decisions. 2) Define the mission, business, or operational problem or opportunity.

NOTE This definition includes the context and any key parameters, without regard to a specific solution, since the solution can be an operational change, a change to an existing product or service, or a new system.

c) **Characterize the solution space**. This activity consists of the following tasks:

1) Define preliminary operational concepts and other concepts in life cycle stages.

NOTE 1 This involves the identification of major stakeholder groups, such as customers, users, administrations, regulators, and system owners, that are defined in the Stakeholder Needs and Requirements Definition process.

NOTE 2 Preliminary life cycle concepts include preliminary acquisition concepts, preliminary deployment concepts, preliminary operational concepts, preliminary support concepts, and preliminary retirement concepts. Operational concepts include high level operational modes or states, operational scenarios, potential use cases, or usage within a proposed business strategy. These concepts enable feasibility analysis and evaluation of alternatives. These concepts are further refined within the Stakeholder Needs and Requirements Definition process.

NOTE 3 The operating environment can have known vulnerabilities associated with specific security threats and safety hazards. These vulnerabilities need to be understood in association with the product under development. The system and human interfaces are an element of the system assurance context and related vulnerabilities are examined in the context of mission‐critical threats.

2) Identify candidate alternative solution classes that span the potential solution space.

NOTE These classes can range from simple operational changes to various software system developments or modifications. This solution space can include the identification of existing assets, systems, and software products suitable for reuse, and changes in services that can address the need for operational or functional modifications. This includes deducing what potential expected services will be needed. The solution space characterization often invokes the Architecture Definition process for a user architecture viewpoint, resulting in architecture views (e.g., capability views, program views and operational views) as proposed by ISO/IEC/IEEE 42010.

d) **Evaluate alternative solution classes.** This activity consists of the following tasks: 1) Assess each alternative solution class.

NOTE 1 Each alternative solution class is assessed against defined criteria that are established based on the organization’s strategy. Feasibility of the solution class is one key decision criteria. The Portfolio Management process provides some criteria to be considered.

NOTE 2 The System Analysis process is used to assess the value of each criterion for each alternative solution class. Structured affordability trade‐offs are recommended. Including cost as a criterion will aid affordability decisions. The assessment of alternatives can include modeling, simulation, analytical techniques, or expert judgment to understand the risks, feasibility and value of the alternative candidate solution classes.

2) Select the preferred alternative solution class(es).

NOTE The Decision Management process is used to evaluate alternatives and to guide selection. Selected alternatives are validated in the context of the organization’s strategy. Feedback on risks, feasibility, market factors, and alternatives is provided for use in updating the organization’s strategy.

e) **Manage the business or mission analysis**. This activity consists of the following tasks: 1) Maintain traceability of business or mission analysis.

NOTE Through the life cycle, bidirectional traceability is maintained between the business and mission problems and opportunities and the preferred alternative solution classes with the organizational strategy, stakeholder needs and requirements, and system analysis results supporting decisions.

2) Provide key artifacts and information items that have been selected for baselines.

NOTE The Configuration Management process is used to establish and maintain configuration items and baselines. This process identifies candidates for the baseline, and the Information Management process controls the information items.

58

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

**6.4.2 Stakeholder Needs and Requirements Definition process**

**6.4.2.1 Purpose**

| The purpose of the Stakeholder Needs and Requirements Definition process is to define the stakeholder requirements for a system that can provide the capabilities needed by users and other stakeholders in a defined environment.  It identifies stakeholders, or stakeholder classes, involved with the system throughout its life cycle, and their needs. It analyzes and transforms these needs into a common set of stakeholder requirements that express the intended interaction the system will have with its operational environment and that are the reference against which each resulting operational capability is validated. The stakeholder requirements are defined considering the context of the system‐of‐interest with the interoperating systems and enabling systems. |
| --- |

NOTE The *SWEBOK, Guide to the Software Engineering Body of Knowledge*, Software Requirements knowledge area discusses software requirements fundamentals (e.g., definition, types, properties, quality characteristics) and other topics, such as stakeholders, requirements elicitation, analysis, and management that provide additional guidance for software systems.

**6.4.2.2 Outcomes**

| As a result of the successful implementation of the Stakeholder Needs and Requirements Definition process: a) Stakeholders of the system are identified.  b) Required characteristics and context of use of capabilities and concepts in the life cycle stages, including operational concepts, are defined.  c) Constraints on a system are identified.  d) Stakeholder needs are defined.  e) Stakeholder needs are prioritized and transformed into clearly defined stakeholder requirements. f) Critical performance measures are defined.  g) Stakeholder agreement that their needs and expectations are reflected adequately in the requirements is achieved.  h) Any enabling systems or services needed for stakeholder needs and requirements are available. i) Traceability of stakeholder requirements to stakeholders and their needs is established. |
| --- |

**6.4.2.3 Activities and tasks**

The project shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Stakeholder Needs and Requirements Definition process.

a) **Prepare for Stakeholder Needs and Requirements Definition.** This activity consists of the following tasks: 1) Identify the stakeholders who have an interest in the software system throughout its life cycle.

NOTE This includes individuals and classes of stakeholders who are users, operators, supporters, developers, producers, trainers, maintainers, disposers, acquirer and supplier organizations, parties responsible for external interfacing entities, regulatory bodies, and others who have a legitimate interest in the system. Where direct communication is not practicable (e.g., for consumer products and services), representatives or designated proxy stakeholders are selected.

2) Define the stakeholder needs and requirements definition strategy.

59

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

NOTE Some stakeholders have interests that oppose the acquirer’s interests (e.g., market competitors, hackers, terrorists) or oppose each other. When the stakeholder interests oppose each other, but do not oppose the software system, this process is intended to gain consensus among the stakeholder classes to establish a common set of acceptable requirements. The intent or desires of those that oppose the acquirers, or detractors of the system, are addressed through the Risk Management process, threat analyzes of the System Analysis process, or the system/software requirements for security, adaptability, or resilience. In this case, the stakeholder needs are not satisfied, but rather addressed in a manner to help ensure system assurance and integrity if actions from the detractors are encountered.

3) Identify and plan for the necessary enabling systems or services needed to support stakeholder needs and requirements definition.

NOTE This includes identification of requirements and interfaces for the enabling systems. Enabling systems for stakeholder needs and requirements definition include tools for facilitation and requirements management.

4) Obtain or acquire access to the enabling systems or services to be used.

NOTE The Validation process is used to objectively confirm that the enabling system achieves its intended use for its enabling functions.

b) **Define stakeholder needs.** This activity consists of the following tasks:

1) Define context of use within the concept of operations and the preliminary life cycle concepts.

NOTE Context of use is often captured using a Context of Use Description [ISO/IEC 25063]. Preliminary life cycle concepts are developed by the Business or Mission Analysis process.

2) Identify stakeholder needs.

NOTE 1 Identification of stakeholder needs includes elicitation of needs directly from the stakeholders, identification of implicit stakeholder needs based on domain knowledge and context understanding, and documented gaps from previous activities. Needs often include measures of effectiveness. Functional analysis is often used to aid the elicitation of needs. Also quality characteristics of the quality model in ISO/IEC 25010 and quality model application to requirements analysis in ISO/IEC 25030 are useful to elicit and identify quality requirements of non‐functional requirements, which are often implicit stakeholder needs.

NOTE 2 The *SWEBOK, Guide to the Software Engineering Body of Knowledge*, Software Requirements knowledge area discusses some additional techniques for eliciting and clarifying software requirements, such as, prototyping, observation, user stories to determine required functionality, data mining, and analyzing competitors’ products.

NOTE 3 Stakeholder needs describe the needs, wants, desires, expectations and perceived constraints of identified stakeholders. Understanding stakeholder needs for the minimum security and privacy requirements necessary for the operational environment minimizes the potential for disruption in plans, schedules, and performance. If significant issues are likely to arise relating to users and other stakeholders and their involvement in or interaction with a software system, recommendations for identifying and treating human‐system issues can be found in ISO TS 18152.

3) Prioritize and down‐select needs.

NOTE The Decision Management process is typically used to support prioritization. The System Analysis process is used to analyze needs for feasibility or other factors.

4) Define the stakeholder needs and rationale.

NOTE Needs concentrate on system purpose and behavior, and are described in the context of the operational environment and conditions. It is useful to trace needs to their sources and rationale.

c) **Develop the operational concept and other life cycle concepts.** This activity consists of the following tasks:

NOTE Other life cycle concepts can include acquisition concepts, deployment concepts, support concepts, security concepts, and retirement concepts. In this activity, the preliminary life cycle concepts defined within the Business or Mission Analysis process are further developed in the context of specific stakeholder needs, as associated scenarios and interactions are defined. See ISO/IEC/IEEE 29148:2011 Clauses 5 and 6 for more information on operational concepts, and ISO/IEC/IEEE 29148:2011 Annex A for an annotated outline of a System Operational Concept.

60

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

1) Define a representative set of scenarios to identify the required capabilities that correspond to anticipated operational and other life cycle concepts.

NOTE 1 Scenarios are used to analyze the operation of the system in its intended environment in order to identify additional needs or requirements that perhaps have not been explicitly identified by any of the stakeholders, e.g., legal, regulatory and social obligations The context of use of the system is identified and analyzed, including the activities that users perform to achieve system objectives, the relevant characteristics of the users (e.g., expected training and knowledge, frequency of system use, responsibilities, accessibility concerns), the physical environment (e.g., available light, temperature) and any equipment to be used (e.g., protective or communication equipment). The social and organizational influences on users that affect system use or constrain its design are analyzed when applicable. Scenarios centered on attackers, their environments, tools, techniques, and capabilities are key considerations for operational concept development. Scenarios are prioritized in order to reflect the weighted importance of the various operational needs.

NOTE 2 These scenarios often motivate updates to the operational or other life cycle concepts. Abuse and failure scenarios highlight the need for additional functional requirements (or more specific derived requirements) to mitigate risks that are identified in the abuse or failure scenarios.

2) Identify the factors affecting interactions between users and the system.

i) Anticipated physical, mental, and learned capabilities of the users;

ii) Workplace, environment and facilities, including other equipment in the context of use; iii) Normal, unusual, and emergency conditions; and

iv) Operator and user recruitment, training and culture.

NOTE 1 Usability requirements take into account human capabilities and skills limitations. Where possible, applicable standards, e.g., ISO 9241, and accepted professional practices are used.

NOTE 2 If usability is important, usability requirements are planned, specified, and implemented through the life cycle processes. Refer to ISO TS 18152 for information on human‐system issues and ISO/IEC 25060:2010 for information on usability.

d) **Transform stakeholder needs into stakeholder requirements.** This activity consists of the following tasks:

1) Identify the constraints on a system solution.

NOTE These constraints can result from 1) instances or areas of stakeholder‐defined solution; 2) implementation decisions made at higher levels of system hierarchical structure; 3) required use of defined enabling, legacy, or interfacing systems, system elements, resources, and staff; or 4) stakeholder‐defined affordability objectives. Include those that are unavoidable consequences of existing agreements, management decisions and technical decisions.

2) Identify the stakeholder requirements and functions that relate to critical quality characteristics, such as assurance, safety, security, environment, or health.

NOTE 1 See ISO/IEC/IEEE 15026 for additional information on system and software assurance.

NOTE 2 Identifying safety risks facilitates the identification of safety requirements and functions. Safety risks include those associated with methods of operations and support, health and safety, threats to property and environmental influences. Use applicable standards, and accepted professional practices. For example, IEC 61508:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems*, provides detailed requirements.

NOTE 3 Identifying security risks facilitates the identification of additional security requirements and functions. If warranted, include applicable areas of system security, such as physical, procedural, communications, computers, programs, data and emissions. This includes access and damage to protected personnel, properties and information, compromise of sensitive information, and denial of approved access to property and information. This also includes the required security functions, such as mitigation and containment, referencing applicable standards and accepted professional practices where mandatory or relevant. See Annex E (E.6) for a life cycle software assurance view.

NOTE 4 See ISO/IEC 25030 for further information regarding quality characteristics from a quality in use perspective.

61

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

3) Define stakeholder requirements, consistent with life cycle concepts, scenarios, interactions, constraints, and critical quality characteristics.

NOTE 1 See ISO/IEC/IEEE 29148:2011 Clauses 5 and 6 for more information on stakeholder requirements, and Clauses 8 and 9 for a description of and an annotated outline for a Stakeholder Requirements Specification.

NOTE 2 The stakeholder requirements are reviewed at key decision times in the life cycle to help ensure that account is taken of changes in needs.

NOTE 3 The stakeholder requirements are recorded in a form suitable for requirements management through the life cycle. These records establish the stakeholder requirements baseline, and retain changes of need and their origin throughout the software life cycle. These records are the basis for traceability to decisions made by the Business or Mission Analysis process as well as stakeholder needs, system requirements, and subsequent software system elements.

NOTE 4 The stakeholder requirements are the basis of the validation criteria for the software system and its elements. e) **Analyze stakeholder requirements.** This activity consists of the following tasks:

1) Analyze the complete set of stakeholder requirements.

NOTE 1 Stakeholder requirements are analyzed for characteristics of individual requirements, as well as characteristics of the set of requirements. Potential analysis characteristics include that the requirements are necessary, implementation‐ free, unambiguous, consistent, complete, singular, feasible, traceable, verifiable, affordable, and bounded. ISO/IEC/IEEE 29148 provides additional information on characteristics of requirements.

NOTE 2 The System Analysis process is used to assess feasibility and affordability. The Verification and Validation processes are used in the review of stakeholder requirements.

2) Define critical performance measures that enable the assessment of technical achievement.

NOTE This includes defining technical and quality measures and critical performance parameters associated with each effectiveness measure identified in the stakeholder requirements. The critical performance measures (e.g., measures of effectiveness and measures of suitability) are defined, analyzed and reviewed to help ensure stakeholder requirements are met and to help ensure identification of project cost, schedule or performance risk associated with any non‐

compliance. ISO/IEC 15939 provides a process to identify, define and use appropriate measures. INCOSE TP‐2003‐020‐01, *Technical Measurement,* provides information on the selection, definition and implementation of critical performance measures. The ISO/IEC 25000 series of standards provides relevant quality measures.

3) Feed back the analyzed requirements to applicable stakeholders to validate that their needs and expectations have been adequately captured and expressed.

4) Resolve stakeholder requirements issues.

NOTE This includes requirements that violate the characteristics for individual requirements or the set of requirements as defined in ISO/IEC/IEEE 29148.

f) **Manage the stakeholder needs and requirements definition.** This activity consists of the following tasks: 1) Obtain explicit agreement with designated stakeholders on the stakeholder requirements.

NOTE This includes confirming that stakeholder requirements are expressed correctly, comprehensible to originators, and that the resolution of conflict in the requirements has not corrupted or compromised stakeholder intentions.

2) Maintain traceability of stakeholder needs and requirements.

NOTE Through the life cycle, bidirectional traceability is maintained between the stakeholder needs and requirements, organizational strategy, and business and mission problems and opportunities. Stakeholder requirements are traced to the system/software requirements during the System/Software Requirements Definition process. Traceability is often maintained using an appropriate data repository.

3) Provide key artifacts and information items that have been selected for baselines.

NOTE The Configuration Management process is used to establish and maintain configuration items and baselines. This process identifies candidates for the baseline, and the Information Management process controls the information items.

62

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

For this process, the stakeholder needs, stakeholder requirements, and operational concept are typical information items that are baselined.

**6.4.3 System/Software requirements definition process**

**6.4.3.1 Purpose**

| The purpose of the System/Software Requirements Definition process is to transform the stakeholder, user‐ oriented view of desired capabilities into a technical view of a solution that meets the operational needs of the user.  This process creates a set of measurable system requirements that specify, from the supplier’s perspective, what characteristics, attributes, and functional and performance requirements the system is to possess, in order to satisfy stakeholder requirements. As far as constraints permit, the requirements should not imply any specific implementation. |
| --- |

NOTE 1 From a high‐level view of the software system, this process can be used to define the overall requirements of the system. As the software system is decomposed into elements, each element, in turn, is treated as a system, function, or set of functions and this process can be used to further specify requirements. Requirements analyzes and tools support traceability of requirements between the software system and its elements.

NOTE 2 The *SWEBOK, Guide to the Software Engineering Body of Knowledge*, Software Requirements knowledge area discusses software requirements definition, analysis, modelling, specification, validation, management and other topics that provide additional guidance for software systems.

NOTE 3 The wording of the outcomes of the System/Software Requirements Definition process differs slightly from the outcomes in the System Requirements Definition process of ISO/IEC/IEEE 15288:2015. Use of the wording “system/software requirements” emphasize the applicability of this document to software systems, having software requirements and systems requirements. This is intended to assist users who define systems requirements and software requirements hierarchically or in different stages.

**6.4.3.2 Outcomes**

| As a result of the successful implementation of the System/Software Requirements Definition process:  a) The system or element description, including interfaces, functions and boundaries, for a system solution is defined.  b) System/software requirements (functional, performance, process, non‐functional, and interface) and design constraints are defined.  c) Critical performance measures are defined.  d) The system/software requirements are analyzed.  e) Any enabling systems or services needed for system/software requirements definition are available. f) Traceability of system/software requirements to stakeholder requirements is developed. |
| --- |

**6.4.3.3 Activities and tasks**

The project shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the System/Software Requirements Definition process.

a) **Prepare for System/Software Requirements Definition.** This activity consists of the following tasks:

1) Define the functional boundary of the software system or element in terms of the behavior and properties provided.

NOTE The functional boundary definition is partly based on the context of use and operational scenarios defined in the frame of the Stakeholder Needs and Requirements Definition process. This includes the software system’s stimuli (input)

63

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

and its responses to users and external systems, and an analysis and description of the required interactions between the software system and its operational environment in terms of interface properties and constraints, such as procedural flows, calling orders, data formats and flows, throughput, and timing. This establishes the expected software system behavior, expressed in quantitative terms, at its boundary. For software, boundaries are commonly expressed in Application Program Interfaces (API) and a graphical user interface (GUI) or interface files or services, including data formats. Annex E (E.5) provides an interface management view of the life cycle processes.

2) Define the system/software requirements definition strategy.

NOTE This includes the approach to be used to identify and define, and manage the system/software requirements with the selected life cycle model, e.g., evolutionary, incremental or iterative. Many factors can influence the strategy, e.g., complexity of the software system and information and functions to be managed; need for ready access and common understanding by multiple team members; degree of collaborative involvement by the acquirer or user representatives throughout the development stage; whether the project involves a new development, a modification, re‐use or integration of existing systems; and process documentation requirements including period of retention. The life cycle model will influence when and how often the system/software requirements definition will be done. Annex H describes the progressive development of requirements in projects using agile methods.

3) Identify and plan for the necessary enabling systems or services needed to support system/software requirements definition.

NOTE This includes identification of requirements and interfaces for the enabling systems. Enabling systems for requirements definition include tools for facilitation and requirements management. Tools for software requirements management which are integrated with software development, test, and CM can simplify tracking and expedite software construction. Plans for enabling systems and description of modeling techniques used in support of the System/Software Requirements Definition process can be incorporated into an SDP.

4) Obtain or acquire access to the enabling systems or services to be used.

NOTE The Validation process is used to objectively confirm that the enabling system achieves its intended use for its enabling functions.

b) **Define system/software requirements.** This activity consists of the following tasks: 1) Define each function that the software system or element is required to perform.

NOTE 1 Software functions can be described in use cases, user stories, or scenarios, and involve the transformation of data and information to achieve user needs (stakeholder requirements). In some cases, functions are derived from analysis of critical quality characteristics, such as performance, security or availability (e.g., system diagnosing function or highly frequent data backup function for reliability).

NOTE 2 Enabling functions that are required to support the system‐of‐interest in achieving its functionality are also identified and defined concurrently with the function of the system‐of‐interest. This is necessary to help ensure that the enabling functions in the system environment are identified and accounted for.

2) Identify required states or modes of operation of the software system.

NOTE 1 States or modes of operation can be modeled and represented in multiple modeling techniques and perspectives to give a sufficiently complete description of the desired system or element requirements.

NOTE 2 Conditions for performance of functions often involve interoperability across functions or elements. For example, some software requirements (e.g., a performance timing limit) can be allocated across multiple software system elements, affecting treatment of the requirement in a test case or regression test.

3) Define necessary implementation constraints.

NOTE For software elements, this includes the implementation decisions that are allocated from architecture definition at higher levels in the software system, introduced by stakeholder requirements, or solution limitations. Implementation constraints include the conditions under which the system is to be capable of performing the function, the conditions under which the system is to commence performing that function (input) and the conditions under which the system is to cease performing that function (output).

4) Identify requirements that relate to risks, criticality of the software system, or critical quality characteristics.

64

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

NOTE 1 Non‐functional requirements and critical quality characteristics in software systems commonly include those related to health, safety, security and software assurance, reliability, availability and supportability (maintainability), and time constraints for throughput and performance. The System Analysis process can be used to determine appropriate values for performance requirements, considering the anticipated cost of achieving them and their impact on operation and use of the system.

NOTE 2 Analysis and definition of safety considerations include those requirements relating to methods of operation and maintenance, environmental influences, and risk of personnel injury. It also includes expressing each safety‐related function and its associated integrity, in terms of the necessary risk reduction and allocation to designated safety‐related systems. Applicable standards are used concerning functional safety, e.g., IEC 61508, and environmental protection, e.g., ISO 14001. Analysis includes security considerations, e.g., those related to compromise and protection of sensitive information, data and material. The security‐related risks are defined, including administrative, personnel, physical, computer, communication, network, emission and environment factors using, as appropriate, applicable security standards. Refer to ISO/IEC/IEEE 15026‐4 for system and software assurance guidance. ISO/IEC 27036 provides guidance for information security requirements for the outsourcing of products and services. ISO 25030 provides guidance for external system quality factors and characteristics. Annex E (E.6) provides a software assurance view of the life cycle processes.

NOTE 3 For software systems intended for human interaction, human‐factors engineering (ergonomics) specifications are considered. For systems that have usability requirements, recommendations for obtaining a desired level of usability can be found in ISO/FDIS 9241‐220, *Ergonomics of human-system interaction — Part 220: Processes for enabling, executing and assessing human-centred design within organizations*.

5) Define system/software requirements and requirements attributes, including the following: i) Data elements, data structures and formats, and database or data retention requirements; ii) User interfaces and user documentation (information for users) and user training; iii) Interfaces with other systems and services;

iv) Functions and non‐functional characteristics, including critical quality characteristics and cost targets;

v) Transition of operational processes and data from existing automated and manual systems, migration approach and schedule, software installation and acceptance of the product; and

vi) Requirement attributes, such as rationale; priority; traceability to software system elements, test cases, and information items; methods of verification; inclusion in approved baselines; and evaluated risk.

NOTE 1 Requirements definition involves iterative and recursive steps in parallel with other life cycle processes. Depending on the life cycle model that is being employed, it is useful to compare the resources to be spent in assuring initial correctness of requirements versus the resource needed to evolve requirements based on verification and validation results.

NOTE 2 The system/software requirements and attributes are recorded with a level of detail and in a form suitable for requirements management through the life cycle. See ISO/IEC/IEEE 29148:2011 Clauses 5 and 6 for more information on requirements, and Clauses 8 and 9 for a description of and an annotated outline for a System Requirements Specification and a Software Requirements Specification.

c) **Analyze system/software requirements.** This activity consists of the following tasks: 1) Analyze the complete set of system/software requirements.

NOTE 1 Requirements are analyzed for characteristics of individual requirements, as well as characteristics of the set of requirements. Potential analysis characteristics include that the requirements are necessary, implementation‐free, unambiguous, consistent, complete, singular, feasible, traceable, verifiable, affordable, and bounded. The Verification process is used to determine if requirements meet the attributes and characteristics of good requirements. In some cases, the technical and economic feasibility of validating and verifying alternative formulations of requirements is evaluated. ISO/IEC/IEEE 29148 provides additional information on characteristics of requirements.

NOTE 2 The System Analysis process can be used to assess feasibility, affordability, balance and other requirements characteristics. The System Analysis process is used to determine appropriate values for requirement parameters, considering the estimated cost, schedule, and technical performance of the software system.

65

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

NOTE 3 Anticipating that some requirements can be achieved incrementally or even deferred or waived, requirements can be prioritized.

2) Define critical performance measures that enable the assessment of technical achievement.

NOTE This includes defining technical and quality measures and critical performance parameters associated with each effectiveness measure identified in the software system element requirements. The critical performance measures (e.g., measures of performance and technical performance measures) are analyzed and reviewed to help ensure system/software requirements are met and to help ensure identification of project cost, schedule or performance risk associated with any non‐compliance. ISO/IEC 15939 provides a process to identify, define and use appropriate measures. INCOSE TP‐2003‐020‐01, *Technical Measurement*, provides information on the selection, definition and implementation of critical performance measures. The ISO/IEC 25000 series of standards provides relevant quality measures.

3) Feed back the analyzed requirements to applicable stakeholders for review.

NOTE Feedback helps validate that the specified requirements have been adequately captured and expressed. Confirmation is made that they are a necessary and sufficient response to stakeholder requirements and a necessary and sufficient input to other processes, in particular software architecture, design, and verification. The Validation process is used to determine if the system/software requirements address the users’ needs.

4) Identify and resolve issues, deficiencies, conflicts, and weaknesses within the complete set of requirements.

NOTE This includes requirements that are not verifiable, ambiguous, violate the characteristics for individual requirements, or are inconsistent with others in the set of requirements. Resolution of issues with requirements can be iterative within certain life cycle models.

d) **Manage system/software requirements.** This activity consists of the following tasks:

NOTE Maintaining system/software requirements includes defining, recording, and controlling the baseline, typically under formal configuration management, along with managing changes resulting from the application of other life cycle processes such as architecture or design.

1) Obtain explicit agreement on the system/software requirements.

NOTE This includes confirming that system/software requirements are expressed correctly, comprehensible to originators and implementers, and that the resolution of conflict in the requirements is consistent with stakeholder decisions.

2) Maintain traceability of the system/software requirements.

NOTE Through the life cycle, bidirectional traceability is maintained between the system/software requirements and the stakeholder requirements, architectural entities, interface definitions, analysis results, verification methods or techniques, and allocated, decomposed, and derived requirements. Traceability allows verification that achievable stakeholder requirements are met by one or more system or element requirements, and such requirements meet or contribute to meeting at least one stakeholder requirement. Traceability is often facilitated by an appropriate data repository or integrated development and test infrastructure.

3) Provide key artifacts and information items that have been selected for baselines.

NOTE The Configuration Management process is used to establish and maintain configuration items and baselines. This process identifies candidates for the baseline, and the Information Management process controls the information items, such as requirements specifications. For this process, the system/software requirements are typical artifacts that are baselined.

**6.4.4 Architecture Definition process**

**6.4.4.1 Purpose**

| The purpose of the Architecture Definition process is to generate system architecture alternatives, to select one or more alternative(s) that frame stakeholder concerns and meet system requirements, and to express this in a set of consistent views. |
| --- |

66

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

| Iteration of the Architecture Definition process with the Business or Mission Analysis process, System/Software Requirements Definition process, Design Definition process, and Stakeholder Needs and Requirements Definition process is often employed so that there is a negotiated understanding of the problem to be solved and a satisfactory solution is identified. The results of the Architecture Definition process are widely used across the life cycle processes. Architecture definition may be applied at many levels of abstraction, highlighting the relevant detail that is necessary for the decisions at that level. |
| --- |

NOTE 1 System architecture deals with fundamental principles, concepts, properties, and characteristics and their incorporation into the system‐of‐interest. Architecture definition has more uses than as merely a driver or part of design. Refer to ISO/IEC/IEEE 42010:2011 for more information about architecture description and the uses and nature of architecture.

NOTE 2 The Architecture Definition process supports identification of stakeholders and their concerns. As the process unfolds, insights are gained into the relation between the requirements specified for the software system and the emergent properties and behaviors of the system that arise from the interactions and relations between the system elements. An effective architecture is as design‐agnostic as possible to allow for maximum flexibility in the design trade space. Even for a single‐product software system, the design of the product will likely change over time while the architecture remains constant. An effective architecture also highlights and supports trade‐offs for the Design Definition process and possibly other processes, such as Portfolio Management, Project Planning, System/Software Requirements Definition, and Verification.

NOTE 3 Architecture Definition can apply to a product line rather than a single software system. A product line architecture describes the structural properties for building a group of related systems with common components and interrelationships. In product line architectures, the architecture necessarily spans several designs. The architecture serves to make the product line cohesive and helps ensure compatibility and interoperability across the product line. ISO/IEC 26550:2013 describes establishing a domain architecture for a product line.

NOTE 4 The *SWEBOK (Guide to the Software Engineering Body of Knowledge*) Software Requirements, Software Design and Software Engineering Models and Methods knowledge areas discuss key aspects of software architecture in relationship to the system, as well as with respect to iteration with design.

**6.4.4.2 Outcomes**

| As a result of the successful implementation of the Architecture Definition process:  a) Identified stakeholder concerns are addressed by the architecture.  b) Architecture viewpoints are developed.  c) Context, boundaries, and external interfaces of the system are defined.  d) Architecture views and models of the system are developed.  e) Concepts, properties, characteristics, behaviors, functions, or constraints that are significant to architecture decisions of the system are allocated to architectural entities.  f) System elements and their interfaces are identified.  g) Architecture candidates are assessed.  h) An architectural basis for processes throughout the life cycle is achieved.  i) Alignment of the architecture with requirements and design characteristics is achieved. j) Any enabling systems or services needed for architecture definition are available.  k) Traceability of architecture elements to stakeholder and system/software requirements is developed. |
| --- |

**6.4.4.3 Activities and tasks**

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

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

67

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

1) Review pertinent information and identify key drivers of the architecture.

NOTE 1 Key drivers are identified by reviewing: (a) market studies, industry projections, competitor product plans, and scientific findings; (b) organizational strategies, organizational‐level concept of operations, organizational policies and directives, regulatory and legal constraints, and stakeholder requirements; (c) mission or business concept of operations, system‐of‐interest and related system operational concept, operational environment, technology roadmaps, and system/software requirements, and (d) other factors that impact the suitability of the software system through its life cycle. This analysis of key drivers typically builds from the Business or Mission Analysis, Stakeholder Requirements Definition, and System/software Requirements Definition processes.

NOTE 2 Key drivers of the architecture can include architecture styles and patterns, elements, principles such as replaceable components, feasibility of implementation and integration; availability of COTS and open source components; data sources for data‐intensive systems; and performance implications. The effect of choosing various design elements can be lessened if the software system is properly architected.

2) Identify stakeholder concerns.

NOTE 1 Stakeholders are initially identified in the Stakeholder Needs and Requirements process. Additional stakeholders are usually identified during the Architecture Definition process. Stakeholder concerns related to architecture include system integrity concerns that the software system will be compromised intentionally or unintentionally via a threat agent or cause accidents as a safety hazard. Stakeholder. expectations or constraints are often associated with the system’s life cycle stages, such as utilization (e.g., availability, security, effectiveness, usability, interoperability with existing systems, availability or risks to data in the system), support (e.g., the supportability of the system over its projected life‐span, obsolescence management), evolution of the software system and its environment (e.g.,

adaptability, scalability, survivability), production (e.g., distribution, testability), and retirement (e.g., sensitive data eradication or retention).

NOTE 2 Concerns affecting software system architecture include data sources and performance implications for data‐ intensive systems, and constraints on the use of outsourced, existing, newly developed, proprietary, commercially available, or open source software elements, including software licensing. While software architecture is ideally design‐ agnostic, the feasibility of implementing the architecture in an affordable software system is a significant constraint for most systems.

3) Define the Architecture Definition roadmap, approach, and strategy.

NOTE This includes the identification of opportunities to communicate with designated stakeholders, the definition of architecture review activities, evaluation approach and criteria, measurement approach, and measurement methods (refer to the Measurement process). The roadmap shows how the architecture will evolve to an envisioned end state and often has a longer timeframe than for the current system‐of‐interest. The approach is the manner in which the work will be accomplished, such as how to engage with stakeholders, how to vet the results, or where to do the work. The strategy deals with the systematic plan of action for implementing the approach consistent with the roadmap.

4) Define architecture evaluation criteria based on stakeholder concerns and key requirements.

5) Identify and plan for the necessary enabling systems or services needed to support the Architecture Definition process.

NOTE This includes identification of requirements and interfaces for the enabling systems and services. Enabling systems for architecture definition can include tools for collaboration and architecture development, and architecture reuse repositories for artifacts such as architecture patterns, models, and reference architectures.

6) Obtain or acquire access to the enabling systems or services to be used.

NOTE The Validation process is used to objectively confirm that the enabling system achieves its intended use for its enabling functions. The Infrastructure Management process supports reuse of enabling systems.

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

1) Select, adapt, or develop viewpoints and model kinds based on stakeholder concerns. 2) Establish or identify potential architecture framework(s) to be used in developing models and views.

NOTE Some architecture frameworks identify stakeholders and their concerns, and relevant viewpoints that address those concerns, while other architecture frameworks are more general in their guidance. Viewpoints specify the kinds of models to be used and how the resulting models can be used to generate architecture views. Refer to ISO/IEC/IEEE 42010 for more information on architecture framework and architecture description practices.

68

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

3) Capture rationale for selection of framework(s), viewpoints and model kinds.

4) Select or develop supporting modelling techniques and tools.

NOTE Both the SWEBOK and ISO/IEC TR 24748‐3 describe modeling techniques that support Architecture Definition and Design Definition of software elements.

c) **Develop models and views of candidate architectures.** This activity consists of the following tasks:

1) Define the software system context and boundaries in terms of interfaces and interactions with external entities.

NOTE This task is mainly based on the outcomes of the Business or Mission Analysis process, and is performed concurrently with the Stakeholder Needs and Requirements Definition process. It consists of identifying the entities external to the software system (i.e., existing and projected systems, products, and services that constitute the system context) and defining the boundaries of the software system (i.e., interactions with these external entities through the interfaces that cross the boundaries). The external entities include the necessary enabling systems. The Architecture Definition process defines interfaces to the extent needed to support essential architectural decisions and understanding. These interface definitions are then refined by the Design Definition process.

2) Identify architectural entities and relationships between entities that address key stakeholder concerns and critical software system requirements.

NOTE Architecture is not necessarily concerned with all requirements, but rather only with those system/software requirements that drive the architecture. On the other hand, the Design Definition process addresses and takes into account all the requirements. Sometimes, through the Architecture Definition process there will be requirements that are deemed to be inappropriate, unaffordable, or unsuitable. These are requirements issues that are resolved through iteration of the System/Software Requirements Definition process. It is also important that the architecture addresses key stakeholder concerns since not all of these will be captured in requirements.

3) Allocate concepts, properties, characteristics, behaviors, functions, or constraints that are significant to architecture decisions of the software system to architectural entities.

NOTE The items being allocated can be physical, logical, or conceptual.

4) Select, adapt, or develop models of the candidate architectures of the software system.

NOTE It is common to use models in architecture definition. The models used are those that best address key stakeholder concerns. Refer to ISO/IEC/IEEE 42010 for how this can be done. Historically, it has been common to use logical and physical models in architecture definition. Information on logical and other models is provided in Annex F.

5) Compose views from the models in accordance with identified viewpoints to express how the architecture addresses stakeholder concerns and meets stakeholder and system/software requirements.

6) Harmonize the architecture models and views with each other.

NOTE Correspondence rules from frameworks are one way to establish harmony between views. See ISO/IEC/IEEE 42010.

d) **Relate the architecture to design.** This activity consists of the following tasks:

1) Identify software system elements that relate to architectural entities and the nature of these relationships.

NOTE Sometimes the software system elements are initially notional until Design Definition has occurred since this depends on the actual design(s) to be done. Sometimes a “reference architecture” is created using these notional system elements as a means to convey architectural intent and to check for design feasibility.

2) Define the interfaces and interactions among the software system elements and external entities.

NOTE This is defined at level of detail necessary to convey the architectural intent and can be further refined in the Design Definition process.

3) Partition, align and allocate requirements to architectural entities and system elements.

69

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

4) Map software system elements and architectural entities to design characteristics. 5) Define principles for the software system design and evolution.

EXAMPLE Principles can include interoperability, use of selected design patterns, ease of replacing and upgrading system elements, or security levels.

e) **Assess architecture candidates.** This activity consists of the following tasks:

1) Assess each candidate architecture against constraints and requirements.

2) Assess each candidate architecture against stakeholder concerns using evaluation criteria. NOTE The System Analysis process and the Risk Management process can be used to support this task. 3) Select the preferred architecture(s) and capture key decisions and rationale.

NOTE The Decision Management process can be used to support this task.

4) Establish the architecture baseline of the selected architecture.

NOTE The architecture baseline is composed of models, views and other relevant architecture descriptions. f) **Manage the selected architecture.** This activity consists of the following tasks:

1) Formalize the architecture governance approach and specify governance‐related roles and responsibilities, accountabilities, and authorities related to design, quality, security, and safety.

2) Obtain explicit acceptance of the architecture by stakeholders.

NOTE The Validation process is used to confirm that the architecture models and views reflect stakeholder requirements, that stakeholder concerns are addressed, and to help ensure that future iterations of software system architecture better address stakeholder concerns.

3) Maintain concordance and completeness of the architectural entities and their architectural characteristics.

NOTE The entities to be checked are not only technical. These are also, for example, legal, economical, organizational and operational entities that are normally part of stakeholder requirements and concerns.

4) Organize, assess and control evolution of the architecture models and views to help ensure that the architectural intent is met and the architectural vision and key concepts are correctly implemented.

5) Maintain the architecture definition and evaluation strategy.

NOTE This includes updating the architecture based upon technological (e.g., obsolescence), implementation, or operational experience. This includes the management of external and internal interfaces that are defined at this level of software system decomposition.

6) Maintain traceability of the architecture.

NOTE Throughout the life cycle, traceability is often maintained among the architectural entities or elements (models, views, and viewpoints), the requirements (including allocated, decomposed, and derived) and stakeholder concerns, software system design, interface definitions, analysis results, and verification methods or techniques.

7) Provide key artifacts and information items that have been selected for baselines.

NOTE The Configuration Management process is used to establish and maintain configuration items and baselines. This process identifies candidates for the baseline. The Information Management process controls the information items, such as architecture descriptions (architecture models, architecture views, evaluations, and traceability).

70

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**6.4.5 Design Definition process 6.4.5.1 Purpose**

**ISO/IEC/IEEE 12207:2017(E)**

| The purpose of the Design Definition process is to provide sufficient detailed data and information about the system and its elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture. |
| --- |

For software systems, design activities typically iterate with activities in System/Software Requirements Definition and Architecture Definition. Design definition is typically applied iteratively and incrementally to develop a detailed design, including software elements, interfaces, databases, and user documentation. Software design is usually concurrent with software implementation, integration, verification, and validation. Annex H discusses software design using agile methods. During design and implementation, further process application refines allocation of evolving requirements among software elements.

NOTE 1 The Design Definition process is driven by requirements that have been vetted through the architecture and more detailed analyzes of feasibility. Architecture focuses on suitability, viability, and desirability, whereas design focuses on compatibility with technologies and other design elements and feasibility of implementation and integration. An effective architecture is as design‐agnostic as possible to allow for maximum flexibility in the design trade space.

NOTE 2 This process provides feedback to the software system architecture to consolidate or confirm the allocation, partitioning and alignment of architectural entities.

**6.4.5.2 Outcomes**

| As a result of the successful implementation of the Design Definition process:  a) Design characteristics of each system element are defined.  b) System/software requirements are allocated to system elements.  c) Design enablers necessary for design definition are selected or defined.  d) Interfaces between system elements composing the system are defined or refined.  e) Design alternatives for system elements are assessed.  f) Design artifacts are developed.  g) Any enabling systems or services needed for design definition are available.  h) Traceability of the design characteristics to the architectural entities of the system architecture is established. |
| --- |

NOTE Design definition considers applicable technologies and their contribution to the system solution. Design provides the ‘implement‐to’ level of the definition, such as drawings, state diagrams, stories, and detailed design descriptions. For software elements, this process can result in a detailed design description that can be verified against requirements and the software architecture. Even if the software design is not fully specified in a formal description, it is sufficiently detailed to permit software implementation (construction) and test planning.

**6.4.5.3 Activities and tasks**

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

NOTE The SWEBOK, *Guide to the Software Engineering Body of Knowledge*, provides detailed discussion on software design. This knowledge area addresses fundamentals, key issues, design strategies and methods, and design notations.

a) **Prepare for software system design definition.** This activity consists of the following tasks:

71

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.

**ISO/IEC/IEEE 12207:2017(E)**

1) Define the design definition strategy, consistent with the selected life cycle model and anticipated design artifacts.

NOTE The software design strategy can include initial or incremental decomposition into system elements; creation of various views of automated procedures, data structures and control systems; selection of design patterns, or progressively more detailed definition of objects and their relationships.

2) Select and prioritize design principles and design characteristics.

NOTE Design principles include controlling ideas such as abstraction, modularization and encapsulation, separation of interface and implementation, concurrency, and persistence of data. Security considerations include the principle of least privilege, layered defenses, restricted access to system services, and other considerations to minimize and defend the system attack surface. Design characteristics include, for example, availability, fault tolerance and resilience, scalability,

usability, capacity and performance, testability, portability, and affordability.

3) Identify and plan for the necessary enabling systems or services needed to support design definition.

NOTE This includes identification of requirements and interfaces for the enabling systems. Enabling systems for design definition include selection of software and system platforms, programming languages, design notations and tools for collaboration and design development, design reuse repositories (for product lines, design patterns, and design artifacts), and design standards.

4) Obtain or acquire access to the enabling systems or services to be used.

NOTE The Validation process is used to objectively confirm that the enabling system achieves its intended use for its enabling functions.

b) **Establish designs related to each software system element.** This activity consists of the following tasks: 1) Transform architectural and design characteristics into the design of software system elements.

NOTE Characteristics apply to physical and logical system elements, such as database structures, provisions for memory and storage, software processes and controls, external interfaces such as user interfaces, or services. ISO 9241‐ 210 provides human centred design/ergonomic design guidelines.

2) Define and prepare or obtain the necessary design enablers.

NOTE Design enablers include models, equations, algorithms, calculations, formal expressions and values of parameters, patterns, and heuristics, which are associated with design characteristics using adequate representation such as drawings, logical diagrams, flowcharts, coding conventions, logic patterns, information models, business rules, user profiles, scenarios, use cases or user stories, and tables of metrics and their values, e.g., function points or user story points.

3) Examine design alternatives and feasibility of implementation.

NOTE 1 For the software system and software elements, typically reuse, adaptation, outsourced service, or new development are examined.

NOTE 2 Assess the feasibility of realizing design characteristics. If warranted by assessment results, examine other alternative design options or perform trade‐offs in the architecture or requirements when design characteristics are impractical to implement.

4) Refine or define the interfaces among the software system elements and with external entities.

NOTE Interfaces are identified and defined in the Architecture Definition process (see 6.4.4) to the level or extent needed for the architecture intent and understanding. These are refined in the Design Definition process based on the design characteristics, interfaces, and interactions of software elements with other elements composing the software system and with external entities. Additional interfaces are sometimes identified and defined that were not addressed in the architecture definition.

5) Establish the design artifacts.

NOTE This task formalizes the design characteristics of the software system elements through dedicated artifacts, depending on the implementation technology. Examples of artifacts include prototypes, data models, pseudocode, entity‐ relationship diagrams, use cases, user role and privilege matrixes, interface specifications, service descriptions, and

72

© ISO/IEC 2017 – All rights reserved

© IEEE 2017 – All rights reserved

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on November 16,2021 at 19:41:38 UTC from IEEE Xplore. Restrictions apply.