TECHNICAL SPECIFICATION

ISO/IEC TS 33061

First edition
2021-04

Information technology — Process assessment — Process assessment model for software life cycle processes

Reference number
ISO/IEC TS 33061:2021(E)

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

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work.

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

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO 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) or the IEClist of patent declarations received (see http://patents.iec.ch).

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering.*

This document cancels and replaces ISO/IEC 15504-5:2012, which has been technically revised.

The main changes compared to ISO/IEC 15504-5:2012 are as follows:

— all processes and their base practices are changed to reflect the ISO/IEC/IEEE 12207 processes;

— all process related process outputs and their descriptions are revised;

— this process assessment model includes a process quality attribute of process performance and can be used with other models of process quality, for instance capability as described in ISO/IEC 33020.

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

Introduction

The set of International Standards ISO/IEC 33001 to ISO/IEC 33099, termed the ISO/IEC 330xx family, defines the requirements and resources needed for process assessment. The overall architecture and content of the ISO/IEC 330xx family is described in ISO/IEC 33001. Several standards in the ISO/IEC 330xx family for process assessment are intended to replace and extend parts of the ISO/IEC 15504 series. Abstracts and previews of the ISO/IEC 330xx family of standards can be found on the ISO website.

This document defines a process assessment model for software life cycle processes, conformant with the requirements of ISO/IEC 33004, for use in performing a conformant assessment in accordance with the requirements of ISO/IEC 33002.

A process assessment model is related to one or more process reference models. The process reference model defined in ISO/IEC/IEEE 12207 is used as the basis for the process assessment model in this document.

A process assessment model incorporates a process measurement framework conformant with the requirements of ISO/IEC 33003 and is expressed as a process quality characteristic with a defined set of process attributes.

A process assessment model includes a set of assessment indicators. Process performance indicators address the process purpose and outcomes of each process in the process assessment model. Process quality indicators demonstrate the achievement of the process attributes in the process measurement framework. These indicators may also provide a reference source of practices when implementing a process improvement program.

The assessment indicators are used as a basis for collecting objective evidence to support an assessor's judgement in assigning ratings of the performance and quality of an implemented process. The set of indicators defined in this document are not intended to be an all-inclusive set and applicable in its entirety. Subsets appropriate to the context and scope of the assessment should be selected, and potentially augmented with additional indicators.

A process assessment is conducted according to a documented assessment process. A documented assessment process identifies the rating method to be used in rating process attributes and identifies or defines the aggregation method to be used in determining ratings.

ISO/IEC 33020 provides a process measurement framework for the assessment of process capability which may be incorporated as a process measurement framework in this document. ISO/IEC 33020:2019, Annex B includes a set of process quality indicators for each process attribute in the process measurement framework.

Information technology — Process assessment — Process assessment model for software life cycle processes

1 Scope

This document defines a process assessment model for software life cycle processes, conformant with the requirements of ISO/IEC 33004, for use in performing a conformant assessment in accordance with the requirements of ISO/IEC 33002.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC/IEEE 12207:2017, *Systems and software engineering — Software life cycle processes*

ISO/IEC 33001, *Information technology — Process assessment — Concepts and terminology*

ISO/IEC 33003:2015, *Information technology — Process assessment — Requirements for process measurement frameworks*

ISO/IEC 33004:2015, *Information technology — Process assessment — Requirements for process reference, process assessment and maturity models*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 33001 apply.

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

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

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

4 The process assessment model

4.1 General

This document provides a basis for a process assessment model that is two-dimensional. In one dimension, the process dimension, the processes are defined and classified into process groups together with the set of assessment indicators of process performance. In the other dimension, the quality dimension, for each process attribute in the process measurement framework a set of process quality indicators is defined for the selected process quality characteristic.

This document applies the software system concepts as defined in ISO/IEC/IEEE 12207. These concepts include software systems, software system architecture and enabling systems.

NOTE 1 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. (ISO/IEC/IEEE 12207)

NOTE 2 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. (ISO/IEC/IEEE 12207)

NOTE 3 The life cycle processes in this document are described in relation to a software system that is composed of interacting system elements (including software elements), each of which can be implemented to fulfil its respective specified requirements. Responsibility for the implementation of any system element may therefore be delegated to another party through an agreement. (ISO/IEC/IEEE 12207)

NOTE 4 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. (ISO/IEC/IEEE 12207)

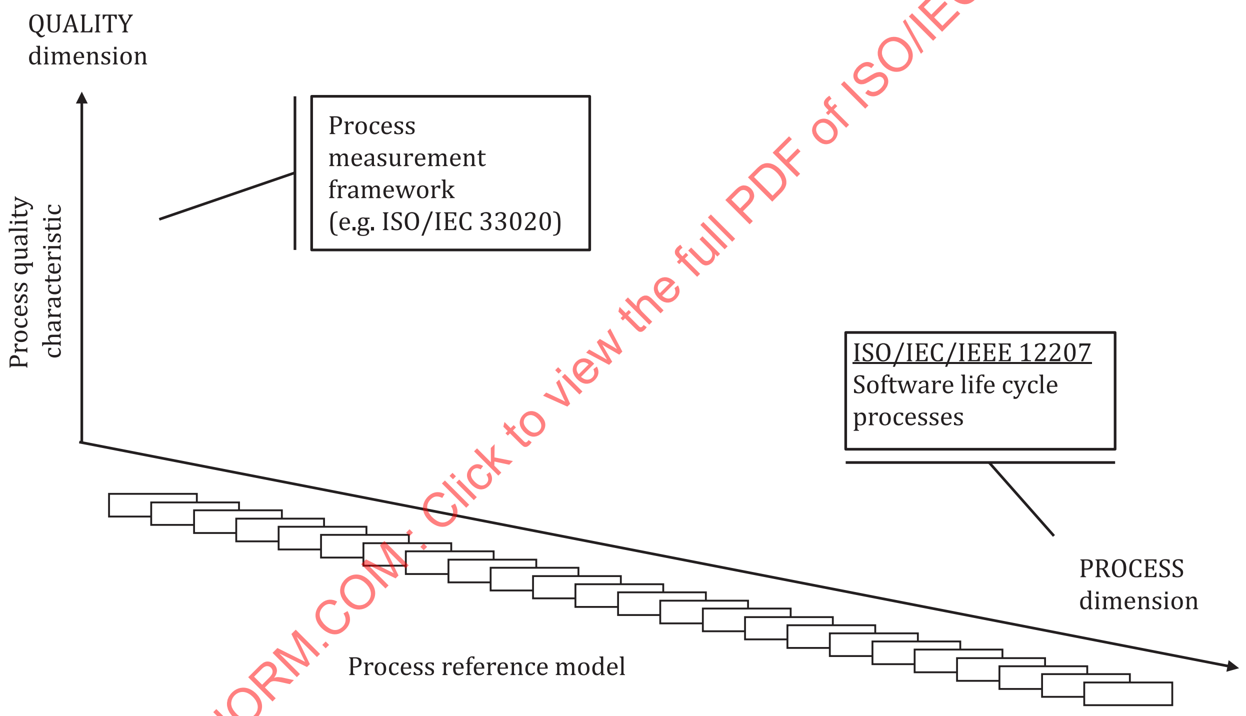

QUALITY
dimension
Process quality characteristic
Process measurement framework (e.g. ISO/IEC 33020)
ISO/IEC/IEEE 12207
Software life cycle processes
PROCESS
dimension
Process reference model

Figure 1 — Structure of the process assessment model

Figure 1 shows the process assessment model as a two-dimensional model, the process dimension with its relationship to ISO/IEC/IEEE 12207 software life cycle processes, and the quality dimension in relationship to a process measurement framework.

Users of this document may freely reproduce the detailed descriptions contained in the assessment model as part of any tool or other material to support the performance of process assessments, so that it can be used for its intended purpose.

4.2 Structure of the process assessment model

4.2.1 General

This subclause describes the detailed structure of the process assessment model and its key components.

The process dimension comprises the set of processes defined with process purpose and process outcomes together with a set of assessment indicators of process performance.

Processes included in a process reference model shall be in accordance with ISO/IEC 33004:2015, 5.4.

The processes in this document are derived directly from ISO/IEC/IEEE 12207 and meet the ISO/IEC 33004 requirements for process descriptions, process purposes and outcomes.

The quality dimension comprising a set of process attributes for a selected process quality characteristic is incorporated as a process measurement framework together with a set of process quality indicators.

NOTE ISO/IEC 33020 provides a process measurement framework for the assessment of process capability which can be incorporated into this document. ISO/IEC 33020 also includes a set of process quality indicators for each process attribute in the process measurement framework.

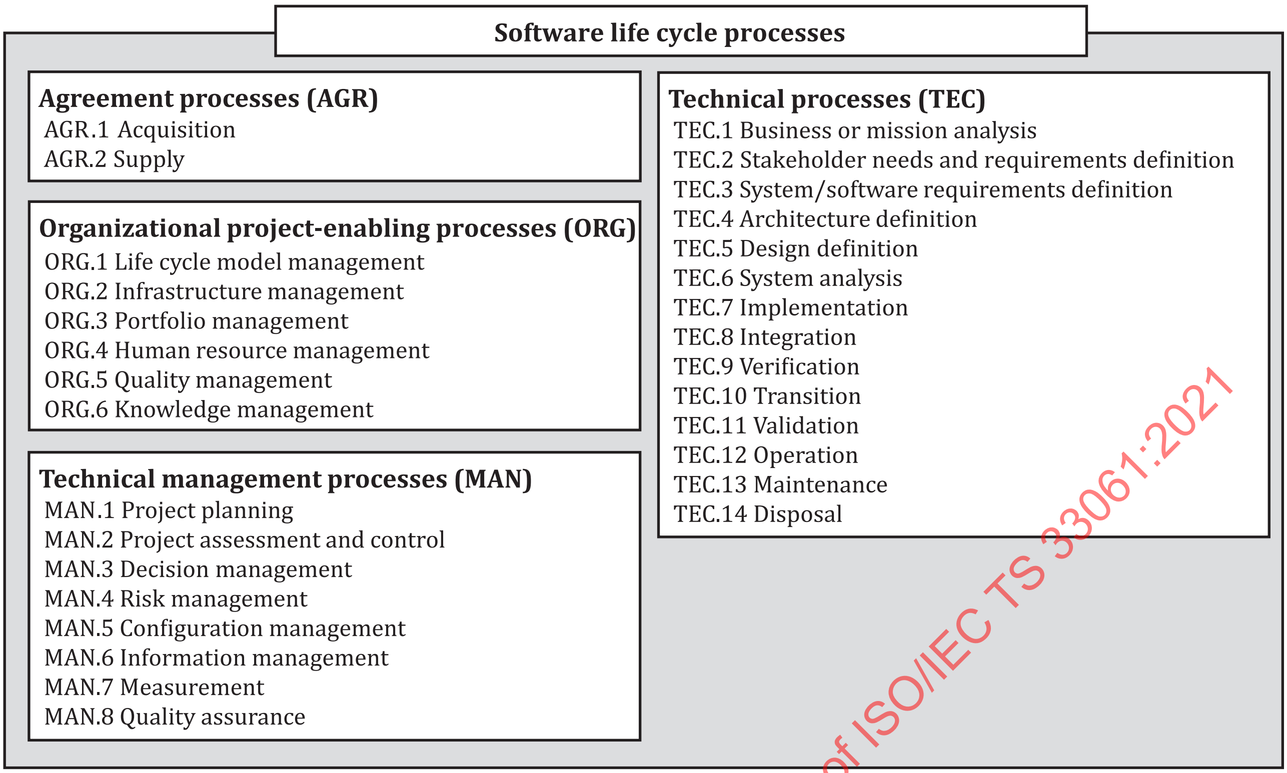
4.2.2 Processes

The process assessment model is based upon the software life cycle processes defined in ISO/IEC/IEEE 12207. The process reference model drawn from ISO/IEC/IEEE 12207 consists of the process purpose and outcomes defined in that standard. The purpose and outcomes are a statement of the goals of the performance of each process.

In this document, ISO/IEC/IEEE 12207 serves as the process reference model for process assessments; any other use of the processes shall adhere to the requirements of ISO/IEC/IEEE 12207.

Figure 2 lists the processes from ISO/IEC/IEEE 12207 that are included in the process dimension of the process assessment model and shows their classification into process groups.

The process groups and their associated processes are described in Clause 5. The description of each process group includes a characterization of the processes it contains. In this process assessment model, each process belonging to a group is identified with a process identifier (ID) consisting of the group abbreviated name and the sequential number of the process in that group.



Software life cycle processes

Agreement processes (AGR)
AGR.1 Acquisition
AGR.2 Supply

Organizational project-enabling processes (ORG)
ORG.1 Life cycle model management
ORG.2 Infrastructure management
ORG.3 Portfolio management
ORG.4 Human resource management
ORG.5 Quality management
ORG.6 Knowledge management

Technical management processes (MAN)
MAN.1 Project planning
MAN.2 Project assessment and control
MAN.3 Decision management
MAN.4 Risk management
MAN.5 Configuration management
MAN.6 Information management
MAN.7 Measurement
MAN.8 Quality assurance

Technical processes (TEC)
TEC.1 Business or mission analysis
TEC.2 Stakeholder needs and requirements definition
TEC.3 System/software requirements definition
TEC.4 Architecture definition
TEC.5 Design definition
TEC.6 System analysis
TEC.7 Implementation
TEC.8 Integration
TEC.9 Verification
TEC.10 Transition
TEC.11 Validation
TEC.12 Operation
TEC.13 Maintenance
TEC.14 Disposal

Figure 2 — Process groups

4.2.3 Process dimension

Each process has a process identifier (ID) consisting of the process group abbreviated name and the sequential number of the process in that group. Each process is described by its name, purpose, outcomes, base practices, and process outputs.

Each process is described in terms of a purpose statement. These statements contain the unique functional objectives of the process when performed in a particular environment. A list of specific process outcomes is associated with each of the process purpose statements, as a list of expected positive results of the process performance.

4.2.4 Quality dimension

For the quality dimension, the minimum requirement is that the process is performed, i.e. the implemented process achieves its process purpose and the expected outcomes are observable.

Process attributes are features of a process that can be evaluated on a scale of achievement, providing a measure of the quality of the process and are applicable to all processes.

4.3 Assessment indicators

A process assessment model is based on the principle that the quality of a process can be assessed by demonstrating the achievement of process attributes on the basis of evidences related to assessment indicators.

There are two types of assessment indicators: process performance indicators and process quality indicators. Process performance indicators address the process purpose and outcomes of each process in the process dimension. Process quality indicators demonstrate the achievement of the process attributes in the quality dimension.

The process performance indicators are:

- — base practice (BP);
- — process output (PO).

The performance of base practices (BPs) provides an indication of the extent of achievement of the process purpose and process outcomes. The base practices correspond to ISO/IEC/IEEE 12207 activities and tasks. Process outputs (POs) are either used or produced (or both), when performing the process. Information items that are the key outputs of the process, are primarily used as performance indicators.

Annex A provides the list of process outputs associated with the processes in Clause 5. The process outputs are identified by categories. The process outputs are indicated by the process IDs.

Process quality indicators depend on the process quality characteristic of interest. The minimum requirement is that at least one of the process attributes shall comprise the achievement of the defined process purpose and process outcomes for the process; this is termed the process performance attribute (see ISO/IEC 33003:2015, 4.2.1). Other process quality attributes can be defined as needed.

The process performance and process quality indicators represent types of objective evidence that can be found in an instantiation of a process and therefore can be used to judge achievement of quality. Figure 3 shows how the assessment indicators are related to process performance and process quality.

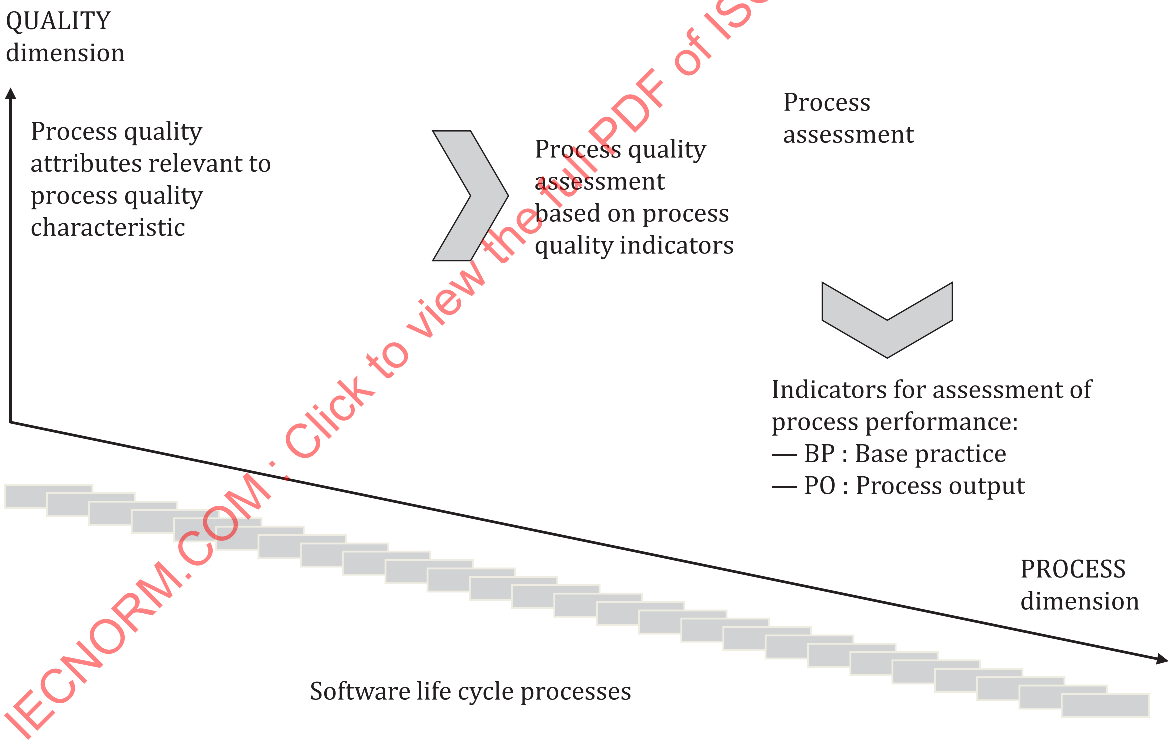


Figure 3 — Assessment indicators

5 The process dimension

5.1 General

The process dimension comprises the set of processes defined with process purpose and process outcomes together with a set of assessment indicators of process performance.

The individual processes each have a process identifier (ID) consisting of the process group abbreviated name and the sequential number of the process in that group and are described in terms of process name, process purpose, and process outcomes as defined in ISO/IEC/IEEE 12207. The process group descriptions are compiled from ISO/IEC/IEEE 12207.

In addition, the process dimension of the process assessment model provides information in the form of a set of:

a) base practices for the process providing a definition of the tasks and activities needed to accomplish the process purpose and fulfil the process outcomes; each base practice is associated to one or more process outcomes;

b) process outputs that are related to one or more process outcomes;

The process purposes, outcomes, base practices and process outputs associated with the processes are included in this clause. The base practices and process outputs constitute the set of indicators of process performance.

A documented assessment process and assessor judgment is needed to ensure that process context (application domain, business purpose, development methodology, size of the organization, etc.) is explicitly considered when using this information.

5.2 Agreement processes (AGR)

5.2.1 General

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

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

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;

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 (all parts) 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. (ISO/IEC/IEEE 12207)

5.2.2 Acquisition process

Process ID	**AGR.1**
Process name	**Acquisition process**

Process 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 is agreed to by both the acquirer and supplier.
Process 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.
Base practices	**AGR.1.BP1: Prepare for the acquisition.** [Outcome: a] 1) Define a strategy for how the acquisition will be conducted. 2) Prepare a request for the supply of a product or service that includes the requirements. **AGR.1.BP2: Advertise the acquisition and select the supplier.** [Outcome: b] 1) Communicate the request for the supply of a product or service to potential suppliers. 2) Select one or more suppliers. **AGR.1.BP3: Establish and maintain an agreement.** [Outcome: c, d] 1) Develop an agreement with the supplier that includes acceptance criteria. 2) Identify necessary changes to the agreement. 3) Evaluate impact of changes on the agreement. 4) Negotiate the agreement with the supplier. 5) Update the agreement with the supplier, as necessary. **AGR.1.BP4: Monitor the agreement.** [Outcome: d, e] 1) Assess the execution of the agreement. 2) Provide data needed by the supplier and resolve issues in a timely manner. **AGR.1.BP5: Accept the product or service.** [Outcome: d, e] 1) Confirm that the delivered product or service complies with the agreement. 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.
Process outputs	Acquisition approach [Outcome: a] Request for supply [Outcome: a] Acquisition agreement [Outcome: c] Acquisition agreement change request [Outcome: c] Accepted system or system element (e.g. software) [Outcome: d] Acquisition report [Outcome: e] Acquisition record [Outcome: e]

5.2.3 Supply process

Process ID	**AGR.2**
Process name	**Supply process**
Process purpose	The purpose of the Supply process is to provide an acquirer with a product or service that meets agreed requirements.
Process 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.
Base practices	**AGR.2.BP1: Prepare for the supply.** [Outcome: a] 1) Determine the existence and identity of an acquirer who has a need for a product or service. 2) Define a supply strategy. **AGR.2.BP2: Respond to a request for supply of products or services.** [Outcome: b] 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. **AGR.2.BP3: Establish and maintain an agreement.** [Outcome: c] 1) Negotiate an agreement with the acquirer that includes acceptance criteria. 2) Identify necessary changes to the agreement. 3) Evaluate impact of changes on the agreement. 4) Negotiate the agreement with the acquirer, as necessary. 5) Update the agreement with the acquirer, as necessary. **AGR.2.BP4: Execute the agreement.** [Outcome: d, e] 1) Execute the agreement according to the established project plans. 2) Assess the execution of the agreement. **AGR.2.BP5: Deliver and support the product or service.** [Outcome: d, e, f] 1) Deliver the product or service in accordance with the agreement criteria. 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.

Process outputs	Supply approach [Outcome: a] Request for supply response [Outcome: b] Supply agreement [Outcome: c] Supply agreement change request [Outcome: c] Supplied system [Outcome: d] Supply report [Outcome: e, f] Supply record [Outcome: e, f]

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. (ISO/IEC/IEEE 12207)

5.3 Organizational project-enabling processes (ORG)

5.3.1 General

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.

The organizational project-enabling processes help ensure the organization's capability to acquire and supply products or services through the initiation, support and control of projects. These processes provide resources and infrastructure necessary to support projects and help ensure the satisfaction of organizational objectives and established agreements. They are not intended to be a comprehensive set of business processes that enable strategic management of the organization's business.

5.3.2 Life cycle model management process

Process ID	**ORG.1**
Process name	**Life cycle model management process**
Process 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.

Process 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.
Base practices	**ORG.1.BP1: Establish the process.** [Outcome: a, b] 1) Establish policies and procedures for process management and deployment that are consistent with organizational strategies. 2) Establish the processes that implement the requirements of this document and that are consistent with organizational strategies. 3) Define the roles, responsibilities, accountabilities, and authorities to facilitate implementation of processes and the strategic management of life cycles. 4) Define business criteria that control progression through the life cycle. 5) Establish standard life cycle models for the organization that are comprised of stages, and define the purpose and outcomes for each stage. **ORG.1.BP2: Assess the process.** [Outcome: c] 1) Monitor process execution across the organization. 2) Conduct periodic reviews of the life cycle models used by the projects. 3) Identify improvement opportunities from assessment results. **ORG.1.BP3: Improve the process.** [Outcome: d] 1) Prioritize and plan improvement opportunities. 2) Implement improvement opportunities and inform relevant stakeholders.
Process outputs	Life cycle model management strategy [Outcome: a, c, d] Organization policy [Outcome: a, b] Organization procedure [Outcome: a, b] Life cycle models [Outcome: a, b] Organizational measurement needs [Outcome: c, d] Organizational performance data [Outcome: c, d] Life cycle model management report [Outcome: c, d] Life cycle model management record [Outcome: c, d]

5.3.3 Infrastructure management process

Process ID	**ORG.2**
Process name	**Infrastructure management process**
Process 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.

Process outcomes	As a result of the successful implementation of the infrastructure management process: a) The requirements for infrastructure are defined. b) The infrastructure elements are identified and specified. c) Infrastructure elements are developed or acquired. d) The infrastructure is available.
Base practices	**ORG.2.BP1: Establish the infrastructure.** [Outcome: a, b, c, d] 1) Define project infrastructure requirements. 2) Identify, obtain and provide infrastructure resources and services that are needed to implement and support projects. **ORG.2.BP2: Maintain the infrastructure.** [Outcome: a, b, c, d] 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.
Process outputs	Infrastructure management strategy [Outcome: b, c] Infrastructure requirements [Outcome: a] Organization infrastructure [Outcome: b, c, d] Infrastructure change requests [Outcome: b, c] Infrastructure management report [Outcome: d] Infrastructure management record [Outcome: d]

5.3.4 Portfolio management process

Process ID	**ORG.3**
Process name	**Portfolio management process**
Process 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.
Process 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.

Base practices	**ORG.3.BP1: Define and authorize projects.** [Outcome: a, b, c, d] 1) Identify potential new or modified capabilities or missions. 2) Prioritize, select and establish new business opportunities, ventures or undertakings. 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. 7) Specify the project reporting requirements and review milestones that govern the execution of each project. 8) Authorize each project to commence execution of project plans. **ORG.3.BP2: Evaluate the portfolio of projects.** [Outcome: e, f] 1) Evaluate projects to confirm ongoing viability. 2) Act to continue or redirect projects that are satisfactorily progressing or can be expected to progress satisfactorily by appropriate redirection. **ORG.3.BP3: Terminate projects.** [Outcome: f, g] 1) Where agreements permit, act to cancel or suspend projects whose disadvantages or risks to the organization outweigh the benefits of continued investments. 2) After completion of the agreement for products and services, act to close the projects.
Process outputs	Portfolio management strategy [Outcome: a, b, c, d, e, f, g] Project portfolio [Outcome: a, b] Project authorization [Outcome: c, d] Project direction [Outcome: e, f, g] Organization lessons learned [Outcome: e, f, g] Portfolio management report [Outcome: e, f, g] Portfolio management record [Outcome: e, f, g]

NOTE 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. (ISO/IEC/IEEE 12207)

5.3.5 Human resource management process

Process ID	**ORG.4**
Process name	**Human resource management process**
Process 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.

Process 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.
Base practices	**ORG.4.BP1: Identify skills.** [Outcome: a, c] 1) Identify skill needs based on current and expected projects. 2) Identify and record skills of personnel. **ORG.4.BP2: Develop skills.** [Outcome: a, c] 1) Establish skills development strategy. 2) Obtain or develop training, education or mentoring resources. 3) Provide planned skill development. 4) Maintain records of skill development. **ORG.4.BP3: Acquire and provide skills.** [Outcome: b, c, d] 1) Obtain qualified personnel when skill deficits are identified. 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.
Process outputs	Human resource management strategy [Outcome: a, b, c, d] Human resource requirements [Outcome: a] Qualified personnel [Outcome: b, c, d] Human resource management report [Outcome: c, d] Human resource management record [Outcome: c, d]

5.3.6 Quality management process

Process ID	**ORG.5**
Process name	**Quality management process**
Process 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.
Process outcomes	As a result of the successful implementation of the quality management process: a) Organizational quality management policies, objectives, and procedures are defined and implemented. b) Quality evaluation criteria and methods are established. c) Resources and information are provided to projects to support the operation and monitoring of project quality assurance activities. d) Quality assurance evaluation results are gathered and analysed. e) Quality management policies and procedures are improved based upon project and organizational results.

Base practices	**ORG.5.BP1: Plan quality management.** [Outcome: a, b, c] 1) Establish quality management policies, objectives, and procedures. 2) Define responsibilities and authority for implementation of quality management. 3) Define quality evaluation criteria and methods. 4) Provide resources and information for quality management. **ORG.5.BP2: Evaluate quality management.** [Outcome: b, d] 1) Gather and analyse quality assurance evaluation results, in accordance with the defined criteria. 2) Assess customer satisfaction. 3) Conduct periodic reviews of project quality assurance activities for compliance with the quality management policies, objectives, and procedures. 4) Monitor the status of quality improvements on processes, products, and services. **ORG.5.BP3: Perform corrective and preventive action.** [Outcome: d, e] 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.
Process outputs	Quality management strategy [Outcome: a, b] Quality management criteria and methods [Outcome: b] Quality management system [Outcome: a, b, c, d, e] Quality management corrective action [Outcome: c, e] Quality management report [Outcome: d, e] Quality management evaluation report [Outcome: d] Quality management record [Outcome: d, e]

NOTE Refer to ISO 9001 for information and requirements to establish a quality management system. (ISO/IEC/IEEE 12207)

5.3.7 Knowledge management process

Process ID	**ORG.6**
Process name	**Knowledge management process**
Process 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.
Process 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 analysed.

Base practices	**ORG.6.BP1: Plan knowledge management.** [Outcome: a, b] 1) Define the knowledge management strategy. 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. **ORG.6.BP2: Share knowledge and skills throughout the organization.** [Outcome: b, c] 1) Establish and maintain a classification for capturing and sharing knowledge and skills across the organization. 2) Capture or acquire knowledge and skills. 3) Share knowledge and skills across the organization. **ORG.6.BP3: Share knowledge assets throughout the organization.** [Outcome: b, c] 1) Establish a taxonomy to organize knowledge assets. 2) Develop or acquire knowledge assets. 3) Share knowledge assets across the organization. **ORG.6.BP4: Manage knowledge, skills, and knowledge assets.** [Outcome: b, c, d] 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.
Process outputs	Knowledge management strategy [Outcome: a] Knowledge management system [Outcome: b, c] Knowledge asset [Outcome: b] Knowledge management report [Outcome: d]

NOTE The re-application of existing knowledge is known as knowledge reuse and includes the reuse of knowledge about or from software elements. (ISO/IEC/IEEE 12207)

5.4 Technical management processes (MAN)

5.4.1 General

The technical management processes are concerned with managing the resources and assets allocated by organization management and with applying them to fulfil 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.

Typically, several projects co-exist in one organization. The technical management processes can be employed at a corporate level to meet internal needs.

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 fulfilment. Individual technical management processes may be invoked at any time in the life cycle and at any level in a hierarchy of projects, as required by plans or unforeseen events. The technical management processes are applied with a level of rigor and formality that depends on the risk and complexity of the project.

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

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.

5.4.2 Project planning process

Process ID	**MAN.1**
Process name	**Project planning process**
Process 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 on-going process that continues throughout a project, with regular revisions to plans.
Process outcomes	As a result of the successful implementation of the project planning process: a) Objectives and plans are defined. b) Roles, responsibilities, accountabilities, 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.

Base practices	**MAN.1.BP1: Define the project.** [Outcome: a, b] 1) Identify the project objectives and constraints. 2) Define the project scope as established in the agreement. 3) Define and maintain a life cycle model that is comprised of stages using the defined life cycle models of the organization. 4) Establish a work breakdown structure (WBS) based on the deliverable products or the evolving architecture of the software system. 5) Define and maintain the processes that will be applied on the project. **MAN.1.BP2: Plan project and technical management.** [Outcome: b, c] 1) Define and maintain a project schedule based on management and technical objectives and work estimates. 2) Define achievement criteria for the life cycle stage decision gates, delivery dates and major dependencies on external inputs or outputs. 3) Define the costs and plan a budget. 4) Define roles, responsibilities, accountabilities, and authorities. 5) Define the infrastructure and services required. 6) Plan the acquisition of materials and enabling systems and services supplied from outside the project. 7) Generate and communicate a plan for project and technical management and execution, including reviews. **MAN.1.BP3: Activate the project.** [Outcome: c, d] 1) Obtain approval for the project. 2) Submit requests and obtain commitments for necessary resources to perform the project. 3) Implement project plans.
Process outputs	Project objectives [Outcome: a] Project constraints [Outcome: a] Project plan (e.g. software development plan) [Outcome: a] Project budget [Outcome: b, c] Work breakdown structure [Outcome: b, c] Project schedule [Outcome: b, c] Project infrastructure needs [Outcome: c] Project human resources needs [Outcome: c] Acquisition need [Outcome: c] Project planning record [Outcome: d]

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. (ISO/IEC/IEEE 12207)

5.4.3 Project assessment and control process

Process ID	**MAN.2**
Process name	**Project assessment and control process**

Process 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. 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.
Process 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 analysed. 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.

Base practices	**MAN.2.BP1: Plan for project assessment and control.** [Outcome: a] 1) Define the project assessment and control strategy. **MAN.2.BP2: Assess the project.** [Outcome: b, c, d, e, f] 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. 5) Assess the adequacy and availability of resources. 6) Assess progress using measured achievement and milestone completion. 7) Conduct required management and technical reviews, audits and inspections. 8) Monitor critical processes and new technologies. 9) Analyse measurement results and make recommendations. 10) Record and provide status and findings from assessment tasks. 11) Monitor process execution within the project. **MAN.2.BP3: Control the project.** [Outcome: g, h, i, j] 1) Initiate necessary actions needed to address identified issues. 2) Initiate necessary project replanning. 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. 4) Authorize the project to proceed toward the next milestone or event, if justified.
Process outputs	Project assessment and control approach [Outcome: a] Project measurement needs [Outcome: a] Project performance data [Outcome: a] Project status report [Outcome: a, b, c, d, e, f] Project review result [Outcome d] Project control request [Outcome: g, h] Project change request [Outcome: g, h] Project authorization to proceed request [Outcome: i] Project lessons learned [Outcome: j] Project assessment and control record [Outcome: j]

5.4.4 Decision management process

Process ID	**MAN.3**
Process name	**Decision management process**
Process 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.

Process 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.
Base practices	**MAN.3.BP1: Prepare for decisions.** [Outcome: a] 1) Define a decision management strategy. 2) Identify the circumstances and need for a decision. 3) Involve relevant stakeholders in the decision-making in order to draw on experience and knowledge. **MAN.3.BP2: Analyse the decision information.** [Outcome: b, d] 1) Select and declare the decision management strategy for each decision. 2) Determine desired outcomes and measurable selection criteria. 3) Identify the trade space and alternatives. 4) Evaluate each alternative, against the criteria. **MAN.3.BP3: Make and manage decisions.** [Outcome: c, d] 1) Determine preferred alternative for each decision. 2) Record the resolution, decision rationale, and assumptions. 3) Record, track, evaluate and report decisions.
Process outputs	Decision management approach [Outcome: a] Decision register [Outcome: a, c, d] Decision management report [Outcome: b, c, d] Decision management record [Outcome: d]

NOTE 1 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. (ISO/IEC/IEEE 12207)

NOTE 2 When it is necessary to perform a detailed assessment of a parameter for one of the criteria, the system analysis process is employed to perform the assessment. (ISO/IEC/IEEE 12207)

5.4.5 Risk management process

Process ID	**MAN.4**
Process name	**Risk management process**
Process purpose	The purpose of the risk management process is to identify, analyse, 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.

Process outcomes	As a result of the successful implementation of the risk management process: a) Risks are identified. b) Risks are analysed. 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.
Base practices	**MAN.4.BP1: Plan risk management.** [Outcome: a, b, c] 1) Define the risk management strategy. 2) Define and record the context of the risk management process. **MAN.4.BP2: Manage the risk profile.** [Outcome: a, b] 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. 3) Periodically provide the relevant risk profile to stakeholders based upon their needs. **MAN.4.BP3: Analyse risks.** [Outcome: a, b, c, e] 1) Identify risks in the categories described in the risk management context. 2) Estimate the likelihood of occurrence and consequences of each identified risk. 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. **MAN.4.BP4: Treat risks.** [Outcome: c, d, e] 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. **MAN.4.BP5: Monitor risks.** [Outcome: a, c, e] 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.
Process outputs	Risk management approach [Outcome: a, b, c, d, e] Risk register [Outcome: a, b, c, d, e] Risk management report [Outcome: e] Risk management record [Outcome: e]

NOTE Risk is defined in ISO Guide 73 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 sometimes commonly known as an opportunity and addressed within the risk management process. (ISO/IEC/IEEE 12207)

5.4.6 Configuration management process

Process ID	**MAN.5**
Process name	**Configuration management process**
Process 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.
Process 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. e) Required configuration audits are completed. f) System releases and deliveries are controlled and approved.

Base practices	**MAN.5.BP1: Plan configuration management.** [Outcome: a, b] 1) Define a configuration management strategy. 2) Define the storage, archive and retrieval procedures for configuration items, CM artefacts, and records. **MAN.5.BP2: Perform configuration identification.** [Outcome: a, b] 1) Select the software system elements to be uniquely identified as configuration items subject to configuration control. 2) Identify the attributes of configuration items. 3) Define baselines through the life cycle. 4) Obtain acquirer and supplier agreement to establish a baseline. **MAN.5.BP3: Perform configuration change management.** [Outcome: c, d, f] 1) Identify and record requests for change and requests for variance. 2) Coordinate, evaluate, and disposition requests for change and requests for variance. 3) Track and manage approved changes to the baseline, requests for change and requests for variance. **MAN.5.BP4: Perform release control.** [Outcome: f] 1) Identify and record release requests, identifying the software system elements in a release. 2) Approve software system releases and deliveries. 3) Track and manage distribution of software system releases to specified environments or software deliveries. **MAN.5.BP5: Perform configuration status accounting.** [Outcome: d, e] 1) Develop and maintain the CM status information, for software system elements, baselines, and releases. 2) Capture, store and report configuration management data. **MAN.5.BP5: Perform configuration evaluation.** [Outcome: c, d, e,] 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 the functional and performance capabilities identified for the baseline. 5) Assess whether the operational software system elements conform to the approved configuration information. 6) Record the CM audit results and disposition action items.

Process outputs	Configuration management approach [Outcome: a] Configuration management system [Outcome: a, b, c, d, e, f] Configuration baseline [Outcome: b] Configuration management change request [Outcome c] Configuration management variance request [Outcome c, f] Configuration audit result [Outcome e] Configuration management report [Outcome: a, b, c, d, e, f] Configuration evaluation report [Outcome: d] Configuration management record [Outcome: a, b, c, d, e, f]

NOTE 1 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. ISO/IEC/IEEE 12207:2017, E.5 provides an example of an interface management process view. (ISO/IEC/IEEE 12207)

NOTE 2 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. (ISO/IEC/IEEE 12207)

5.4.7 Information management process

Process ID	**MAN.6**
Process name	**Information management process**
Process 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.
Process 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.

Base practices	**MAN.6.BP1: Prepare for information management.** [Outcome: a, b] 1) Define the strategy for information management. 2) Define the items of information that will be managed. 3) Designate authorities and responsibilities for information management. 4) Define the content, formats and structure of information items. 5) Define information maintenance actions. **MAN.6.BP2: Perform information management.** [Outcome: c, d, e] 1) Obtain, develop, or transform the identified items of information. 2) Maintain information items and their storage records and record the status of information. 3) Publish, distribute or provide access to information and information items to designated stakeholders. 4) Archive designated information. 5) Dispose of unwanted, invalid or unvalidated information.
Process outputs	Information management approach [Outcome: a, b] Information register [Outcome: a, b, c, d, e] Information management report [Outcome: e] Information management record [Outcome: c]

NOTE Managed information has these quality characteristics: unambiguous, complete, verifiable, consistent, modifiable, traceable, and presentable. (ISO/IEC/IEEE 12207)

5.4.8 Measurement process

Process ID	**MAN.7**
Process name	**Measurement process**
Process purpose	The purpose of the measurement process is to collect, analyse, and report objective data and information to support effective management and demonstrate the quality of the products, services, and processes.
Process 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 analysed and the results interpreted. e) Information items provide objective information that supports decisions.

Base practices	**MAN.7.BP1: Prepare for measurement.** [Outcome: a, b] 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. 4) Select and specify measures that satisfy the information needs. 5) Define data collection, analysis, access, and reporting procedures. 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. **MAN.7.BP2: Perform measurement**. [Outcome: c, d, e] 1) Integrate manual or automated procedures for data generation, collection, analysis and reporting into the relevant processes. 2) Collect, store, and verify data. 3) Analyse data and develop information items. 4) Record results and inform the measurement users.
Process outputs	Measurement approach [Outcome: a, b] Measurement register [Outcome: a, b, c, d, e] Measurement report [Outcome: d, e] Measurement record [Outcome: c]

NOTE Measures have these quality characteristics: verifiable, meaningful, actionable, timely, and cost-effective. (ISO/IEC/IEEE 12207)

5.4.9 Quality assurance process

Process ID	**MAN.8**
Process name	**Quality assurance process**
Process 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 (QA) 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.
Process 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.

Base practices	**MAN.8.BP1: Prepare for quality assurance.** [Outcome: a, b] 1) Define a quality assurance strategy. 2)Establish independence of quality assurance from other life cycle processes. **MAN.8.BP2: Perform product or service evaluations.** [Outcome: b, c] 1)Evaluate products and services for conformance to established criteria, contracts, standards, and regulations. 2)Monitor that verification and validation of the outputs of the life cycle processes are performed to determine conformance to specified requirements. **MAN.8.BP3: Perform process evaluations.** [Outcome: c] 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. **MAN.8.BP4: Manage QA records and reports.** [Outcome: d] 1)Create records and reports related to quality assurance activities. 2)Maintain, store, and distribute records and reports. 3)Identify incidents and problems associated with product, service, and process evaluations. **MAN.8.BP5: Treat incidents and problems.** [Outcome: d, e, f] 1) Record, analyse and classify incidents. 2) Identify selected incidents to associate with known errors or problems. 3) Record, analyse and classify problems. 4) Identify root causes and treatment of problems where feasible. 5) Prioritize treatment of problems (problem resolution) and track corrective actions. 6) Analyse trends in incidents and problems. 7) Identify improvements in processes and products that may prevent future incidents and problems. 8) Inform designated stakeholders of the status of incidents and problems. 9) Track incidents and problems to closure.
Process outputs	Quality assurance approach [Outcome: a, b] Quality assurance criteria and methods [Outcome: b] Quality assurance system [Outcome: a, b, c, d, e] Quality assurance corrective action [Outcome: e, f] Quality assurance report [Outcome: d, e] Quality assurance evaluation report [Outcome: c, d] Quality assurance record [Outcome: c, d, e, f]

NOTE 1 Outcomes a) through d) align with the outcomes of the quality management process activities and tasks. (ISO/IEC/IEEE 12207)

NOTE 2 IEEE 730-2014 provides additional detail. (ISO/IEC/IEEE 12207)

5.5 Technical processes (TEC)

5.5.1 General

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.

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.

NOTE 1 For software systems, these processes can be recursively applied at more inclusive or more detailed levels for software system definition and realization. (ISO/IEC/IEEE 12207)

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. (ISO/IEC/IEEE 12207)

NOTE 3 The concept of life cycle stages and the application of these processes in any stage are described in detail in ISO/IEC/IEEE 24748-1. It has a complete set of example stages and stage outcomes for the enactment of technical processes within a software life cycle. (ISO/IEC/IEEE 12207)

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 ISO/IEC/IEEE 12207:2017, E.5 for an example of an interface management process view. (ISO/IEC/IEEE 12207)

NOTE 5 ISO/IEC 27002 and ISO/IEC 27034 provide guidance for applying security concerns in the technical processes for software systems. See ISO/IEC/IEEE 12207:2017, E.6 for a sample software assurance process view. (ISO/IEC/IEEE 12207)

5.5.2 Business or mission analysis process

Process ID	**TEC.1**
Process name	**Business or mission analysis process**
Process 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 can address a problem or take advantage of an opportunity.

Process 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 analysed. 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.
Base practices	**TEC.1.BP1: Prepare for business or mission analysis.** [Outcome: a, f] 1) Review identified problems and opportunities in the organization strategy with respect to desired organization goals or objectives. 2) Define the business or mission analysis strategy. 3) Identify and plan for the necessary enabling systems or services needed to support business or mission analysis. 4) Obtain or acquire access to the enabling systems or services to be used. **TEC.1.BP2: Define the problem or opportunity space.** [Outcome: b, c] 1) Analyse the problems and opportunities in the context of relevant trade-space factors. 2) Define the mission, business, or operational problem or opportunity. **TEC.1.BP3: Characterize the solution space.** [Outcome: b, c, d] 1) Define preliminary operational concepts and other concepts in life cycle stages. 2) Identify candidate alternative solution classes that span the potential solution space. **TEC.1.BP4: Evaluate alternative solution classes.** [Outcome: d, e] 1) Assess each alternative solution class. 2) Select the preferred alternative solution class(es). **TEC.1.BP5: Manage the business or mission analysis.** [Outcome: g] 1) Maintain traceability of business or mission analysis. 2) Provide key artefacts and information items that have been selected for baselines.

Process outputs	Business or mission analysis approach [Outcome: a, b, c] Problem or opportunity statement [Outcome: a] Life cycle concepts [Outcome: c] Validation criteria [Outcome: b] Alternative solution classes [Outcome: b, d, e] Requirements imposed on enabling systems [Outcome f] Traceability mapping [Outcome: g] Business or mission analysis record [Outcome: a, b, c, d, e, f, g]

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. (ISO/IEC/IEEE 12207)

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. (ISO/IEC/IEEE 12207)

NOTE 3 In some domains, business or mission analysis relates to the concept of identifying and analysing 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 are often translated into target capabilities. As applicable within a given domain, the problem or opportunity space includes the target capabilities. (ISO/IEC/IEEE 12207)

5.5.3 Stakeholder needs and requirements definition process

Process ID	**TEC.2**
Process name	**Stakeholder needs and requirements definition process**
Process 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 analyses 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.

Process 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.

Base practices	**TEC.2.BP1: Prepare for stakeholder needs and requirements definition.** [Outcome: a, d, h] 1) Identify the stakeholders who have an interest in the software system throughout its life cycle. 2) Define the stakeholder needs and requirements definition strategy. 3) Identify and plan for the necessary enabling systems or services needed to support stakeholder needs and requirements definition. 4) Obtain or acquire access to the enabling systems or services to be used.
	TEC.2.BP2: Define stakeholder needs. [Outcome: b, d, e] 1) Define context of use within the concept of operations and the preliminary life cycle concepts. 2) Identify stakeholder needs. 3) Prioritize and down-select needs. 4) Define the stakeholder needs and rationale.
	TEC.2.BP3: Develop the operational concept and other life cycle concepts. [Outcome: b, d] 1) Define a representative set of scenarios to identify required capabilities that correspond to anticipated operational and other life cycle concepts. 2) Identify the factors affecting interactions between users and the system.
	TEC.2.BP4: Transform stakeholder needs into stakeholder requirements. [Outcome: c, d, e] 1) Identify the constraints on a system solution. 2)Identify the stakeholder requirements and functions that relate to critical quality characteristics, such as assurance, safety, security, environment, or health. 3) Define stakeholder requirements, consistent with life cycle concepts, scenarios, interactions, constraints, and critical quality characteristics.
	TEC.2.BP5: Analyse stakeholder requirements. [Outcome: e, f, g] 1) Analyse the complete set of stakeholder requirements. 2) Define critical performance measures that enable the assessment of technical achievement. 3) Feed back the analysed requirements to applicable stakeholders to validate that their needs and expectations have been adequately captured and expressed. 4) Resolve stakeholder requirements issues.
	TEC.2.BP6: Manage the stakeholder needs and requirements definition. [Outcome: g, i] 1) Obtain explicit agreement with designated stakeholders on the stakeholder requirements 2) Maintain traceability of stakeholder needs and requirements. 3) Provide key artefacts and information items that have been selected for baselines.

Process outputs	Stakeholder needs and requirements definition approach [Outcome: a, b, c, d] Stakeholder identification [Outcome: a] Life cycle concepts [Outcome: b] Stakeholder requirements [Outcome: e, g] Validation criteria [Outcome: e, g] Critical performance measurement needs [Outcome: f] Critical performance data [Outcome: f] Requirements imposed on enabling systems [Outcome h] Traceability mapping [Outcome: i] Stakeholder needs and requirements definition record [Outcome: a, b, c, d, e, f, g, h, i]

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. (ISO/IEC/IEEE 12207)

5.5.4 System/software requirements definition process

Process ID	**TEC.3**
Process name	**System/software requirements definition process**
Process 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.
Process 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 analysed. 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.

Base practices	**TEC.3.BP1: Prepare for system/software requirements definition.** [Outcome: a, b, e] 1) Define the functional boundary of the software system or element in terms of the behaviour and properties provided. 2) Define the system/software requirements definition strategy. 3) Identify and plan for the necessary enabling systems or services needed to support system/software requirements definition. 4) Obtain or acquire access to the enabling systems or services to be used. **TEC.3.BP2: Define system/software requirements.** [Outcome: b, d] 1) Define each function that the software system or element is required to perform. 2) Identify required states or modes of operation of the software system. 3) Define necessary implementation constraints. 4) Identify requirements that relate to risks, criticality of the software system, or critical quality characteristics. 5) Define system/software requirements and requirements attributes. **TEC.3.BP3: Analyse system/software requirements.** [Outcome: c, d] 1) Analyse the complete set of system/software requirements. 2) Define critical performance measures that enable the assessment of technical achievement. 3) Feed back the analysed requirements to applicable stakeholders for review. 4) Identify and resolve issues, deficiencies, conflicts, and weaknesses within the complete set of requirements. **TEC.3.BP4: Manage system/software requirements.** [Outcome: d, f] 1) Obtain explicit agreement on the system/software requirements. 2) Maintain traceability of the system/software requirements. 3) Provide key artefacts and information items that have been selected for baselines.
Process outputs	System/software requirements definition approach [Outcome: a, b, c] System function model [Outcome: a] System/software requirements [Outcome: a, b, d] Verification criteria [Outcome: a, b, d] Critical performance measurement needs [Outcome: c] Critical performance data [Outcome: c] Requirements imposed on enabling systems [Outcome e] Traceability mapping [Outcome: f] System/software requirements definition record [Outcome: a, b, c, d, e, f]

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 analyses and tools support traceability of requirements between the software system and its elements. (ISO/IEC/IEEE 12207)

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. (ISO/IEC/IEEE 12207)

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. 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. (ISO/IEC/IEEE 12207)

5.5.5 Architecture definition process

Process ID	**TEC.4**
Process name	**Architecture definition process**
Process 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. Iteration of the architecture definition process with the business or mission analysis process, system 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.
Process 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, behaviours, 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.

Base practices	**TEC.4.BP1: Prepare for architecture definition.** [Outcome: a, b, c, h, j] 1) Review pertinent information and identify key drivers of the architecture. 2) Identify stakeholder concerns. 3) Define the architecture definition roadmap, approach, and strategy. 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. 6) Obtain or acquire access to the enabling systems or services to be used. **TEC.4.BP2: Develop architecture viewpoints.** [Outcome: b, c, d, e] 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. 3) Capture rationale for selection of framework(s), viewpoints and model types. 4) Select or develop supporting modelling techniques and tools. **TEC.4.BP3: Develop models and views of candidate architectures.** [Outcome: b, c, d, e, f, h] 1) Define the software system context and boundaries in terms of interfaces and interactions with external entities. 2) Identify architectural entities and relationships between entities that address key stakeholder concerns and critical software system requirements. 3) Allocate concepts, properties, characteristics, behaviours, functions, or constraints that are significant to architecture decisions of the software system to architectural entities. 4) Select, adapt, or develop models of the candidate architectures of the software system. 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. **TEC.4.BP4: Relate the architecture to design.** [Outcome: c, f, i, h] 1) Identify software system elements that relate to architectural entities and the nature of these relationships. 2) Define the interfaces and interactions among the software system elements and external entities. 3) Partition, align and allocate requirements to architectural entities and system elements. 4) Map software system elements and architectural entities to design characteristics. 5) Define principles for the software system design and evolution.

	TEC.4.BP5: Assess architecture candidates. [Outcome: g, h, i] 1) Assess each candidate architecture against constraints and requirements. 2) Assess each candidate architecture against stakeholder concerns using evaluation criteria. 3) Select the preferred architecture(s) and capture key decisions and rationale. 4) Establish the architecture baseline of the selected architecture. **TEC.4.BP6: Manage the selected architecture.** [Outcome: h, i, k] 1) Formalize the architecture governance approach and specify governance related roles and responsibilities, accountabilities, and authorities related to design, quality, security, safety. 2) Obtain explicit acceptance of the architecture by stakeholders. 3) Maintain concordance and completeness of the architectural entities and their architectural characteristics. 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. 6) Maintain traceability of the architecture. 7) Provide key artefacts and information items that have been selected for baselines.
Process outputs	Architecture definition approach [Outcome: a, h] System/software architecture model [Outcome: b, c, d, e, f, g, h, i] System/software architecture rationale [Outcome: b, c, d, e, f, g, h, i] System/software interface definition [Outcome: c, f] Critical performance measurement needs [Outcome: g] Critical performance data [Outcome: g] System/software architecture assessment report [Outcome: g] Requirements imposed on enabling systems [Outcome j] Traceability mapping [Outcome: k] Architecture definition record [Outcome: h]

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 for more information about architecture description and the uses and nature of architecture. (ISO/IEC/IEEE 12207)

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 behaviours 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. (ISO/IEC/IEEE 12207)

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 describes establishing a domain architecture for a product line. (ISO/IEC/IEEE 12207)

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. (ISO/IEC/IEEE 12207)

5.5.6 Design definition process

Process ID	**TEC.5**
Process name	**Design definition process**
Process purpose	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.
Process 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 artefacts 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.

Base practices	**TEC.5.BP1: Prepare for software system design definition.** [Outcome: a, c, g] 1) Define the design definition strategy, consistent with the selected life cycle model and anticipated design artefacts. 2) Select and prioritize design principles and design characteristics. 3) Identify and plan for the necessary enabling systems or services needed to support design definition. 4) Obtain or acquire access to the enabling systems or services to be used. **TEC.5.BP2: Establish designs related to each software system element.** [Outcome: b, c, d, e, f] 1) Transform architectural and design characteristics into the design of software system elements. 2) Define and prepare or obtain the necessary design enablers. 3) Examine design alternatives and feasibility of implementation. 4) Refine or define the interfaces among the software system elements and with external entities. 5) Establish the design artefacts. **TEC.5.BP3 Assess alternatives for obtaining software system elements.** [Outcome: b, c, e, f] 1) Determine technologies required for each element composing the software system. 2) Identify candidate alternatives for the software system elements. 3) Assess each candidate alternative against criteria developed from expected design characteristics and element requirements to determine suitability for the intended application. 4) Choose the preferred alternatives among candidate design solutions for the software system elements. **TEC.5.BP4 Manage the design.** [Outcome: a, b, h] 1) Capture the design and rationale. 2) Establish traceability between the detailed design elements, the system/software requirements, and the architectural entities of the software system architecture. 3) Determine the status of the software system and element design. 4) Provide key artefacts and information items that have been selected for baselines.

Process outputs	Design definition approach [Outcome: a, b, c] System/software design model [Outcome: a, c, d, f] System/software design rationale [Outcome: a, c, d, f] System/software interface definition [Outcome: d] System element (e.g. software) description [Outcome: a, b, d, e, f] Critical performance measurement needs [Outcome: e] Critical performance data [Outcome: e] System design assessment report [Outcome: e] Requirements imposed on enabling systems [Outcome g] Traceability mapping [Outcome: h] Design definition record [Outcome: f]

NOTE 1 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. ISO/IEC/IEEE 12207:2017, Annex H discusses software design using agile methods. During design and implementation, further process application refines allocation of evolving requirements among software elements. (ISO/IEC/IEEE 12207)

NOTE 2 The design definition process is driven by requirements that have been vetted through the architecture and more detailed analyses 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. (ISO/IEC/IEEE 12207)

NOTE 3 This process provides feedback to the software system architecture to consolidate or confirm the allocation, partitioning and alignment of architectural entities. (ISO/IEC/IEEE 12207)

NOTE 4 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. (ISO/IEC/IEEE 12207)

NOTE 5 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. (ISO/IEC/IEEE 12207)

5.5.7 System analysis process

Process ID	**TEC.6**
Process name	**System analysis process**

Process purpose	The purpose of the system analysis process is to provide a rigorous basis of data and information for technical understanding to aid decision-making across the life cycle. The system analysis process applies to the development of inputs needed for any technical assessment. It can provide confidence in the utility and integrity of system requirements, architecture, and design. System analysis covers a wide range of differing analytic functions, levels of complexity, and levels of rigor. It includes mathematical analysis, modelling, simulation, experimentation, and other techniques to analyse technical performance, system behaviour, feasibility, affordability, critical quality characteristics, technical risks, life cycle costs, and to perform sensitivity analysis of the potential range of values for parameters across all life cycle stages. It is used for a wide range of analytical needs concerning operational concepts, determination of requirement values, resolution of requirements conflicts, assessment of alternative architectures or system elements, and evaluation of engineering strategies (integration, verification, validation, and maintenance). Formality and rigor of the analysis will depend on the criticality of the information need or work product supported, the amount of information/data available, the size of the project, and the schedule for the results.
Process outcomes	As a result of the successful implementation of the system analysis process: a) System analyses needed are identified. b) System analysis assumptions and results are validated. c) System analysis results are provided for decisions. d) Any enabling systems or services needed for system analysis are available. e) Traceability of the system analysis results is established.
Base practices	**TEC.6.BP1: Define the system analysis strategy and prepare for system analysis.** [Outcome: a, c, d] 1) Identify the problem or question that requires analysis. 2) Identify the stakeholders of the analysis. 3) Define the scope, objectives, and level of fidelity of the analysis. 4) Select the methods to support the analysis. 5) Identify and plan for the necessary enabling systems or services needed to support the analysis. 6) Obtain or acquire access to the enabling systems or services to be used. 7) Collect the data and inputs needed for the analysis. **TEC.6.BP2: Perform system analysis.** [Outcome: b, c] 1) Identify and validate contexts and assumptions. 2) Apply the selected analysis methods to perform the required analysis. 3) Review the analysis results for quality and validity. 4) Establish conclusions and recommendations. 5) Record the results of the system analysis. **TEC.6.BP3: Manage the system analysis.** [Outcome: e] 1) Maintain traceability of the analysis results. 2) Provide key artefacts and information items that have been selected for baselines.

Process outputs	System analysis approach [Outcome a, b] Requirements imposed on enabling systems [Outcome d] Traceability mapping [Outcome: e] System analysis report [Outcome c] System analysis record [Outcome c, e]

NOTE The system analysis process can be employed for the entire software system or any element. This process is often used in conjunction with the decision management process. (ISO/IEC/IEEE 12207)

5.5.8 Implementation process

Process ID	**TEC.7**
Process name	**Implementation process**
Process purpose	The purpose of the implementation process is to realize a specified system element. This process transforms requirements, architecture, design, including interface, into actions that create a system element according to the practices of the selected implementation technology, using appropriate technical specialties or disciplines. This process results in a system element that satisfies specified system requirements (including allocated and derived requirements), architecture, and design.
Process outcomes	As a result of the successful implementation of the implementation process: a) Implementation constraints that influence the requirements, architecture, or design are identified. b) A system element is realized. c) A system element is packaged or stored. d) Any enabling systems or services needed for implementation are available. e) Traceability is established.

Base practices	**TEC.7.BP1: Prepare for implementation.** [Outcome: a, d] 1) Define an implementation strategy. 2) Identify constraints from the implementation strategy and implementation technology on the system/software requirements, architecture characteristics, design characteristics, or implementation techniques. 3) Identify and plan for the necessary and distinct software environments, including enabling systems or services needed to support implementation. 4) Obtain or acquire access to the software environments and other enabling systems or services. **TEC.7.BP2: Perform implementation.** [Outcome: b, c] 1) Realize or adapt software elements, according to the strategy, constraints, and defined implementation procedures. 2) Realize or adapt hardware elements of software systems. 3) Realize or adapt service elements of software systems. 4) Evaluate software unit and affiliated data or other information according to the implementation strategy and criteria. 5) Package and store the software system element. 6) Record objective evidence that the software system element meets requirements. **TEC.7.BP3: Manage results of implementation.** [Outcome: e] 1) Record implementation results and anomalies encountered. 2) Maintain traceability of the implemented software system elements. 3) Provide key artefacts and information items that have been selected for baselines.
Process outputs	Implementation approach [Outcome a, b, c, d, e] Constraints on solution [Outcome a] System element (e.g. software) [Outcome b, c] System element (e.g. software) description [Outcome b, c] Requirements imposed on enabling systems [Outcome d] Traceability mapping [Outcome: e] Implementation report [Outcome a, b, c, d, e] Implementation record [Outcome a, b, c, d, e]

NOTE 1 For software systems, the purpose of the implementation process is to realize a software system element. Software system elements can include hardware, software, and services. For software implementation, this process transforms specified designs, behaviour, interfaces and implementation constraints into actions that create a software system element implemented as a software product or service, also known as a "software item". Software implementation results in a software element that satisfies specified requirements through verification and stakeholder requirements through validation. Software implementation includes various combinations of construction (coding of newly built software elements), acquisition of new software packages (e.g. from open source or a commercial or organizational source) or re-use of existing elements (with or without modification). (ISO/IEC/IEEE 12207)

NOTE 2 Software implementation commonly involves use of the agreement processes to obtain non-developmental items (NDI), such as hardware and operating systems (the platform) or enabling systems and services. Software implementation is usually performed concurrently with software integration. Implementation is typically performed along with all of the technical management processes and many of the technical processes, especially:

a) the verification process, which provides objective evidence that the software implementation fulfils its specified requirements and identifies anomalies (errors, defects, faults) in implementation-related information items, (e.g. system/software requirements, architecture, design, or other descriptions), processes, software elements, items, units;

b) the validation process, which confirms that the implementation fulfils requirements for a specific intended use of a software work product. (ISO/IEC/IEEE 12207)

5.5.9 Integration process

Process ID	**TEC.8**
Process name	**Integration process**
Process purpose	The purpose of the integration process is to synthesize a set of system elements into a realized system (product or service) that satisfies system/software requirements, architecture, and design. This process assembles the implemented system elements. Interfaces are identified and activated to enable interoperation of the system elements as intended. This process integrates the enabling systems with the system-of-interest to facilitate interoperation.
Process outcomes	As a result of the successful implementation of the integration process: a) Integration constraints that influence system requirements, architecture, or design, including interfaces, are identified. b) Approach and checkpoints for the correct operation of the assembled interfaces and system functions are defined. c) Any enabling systems or services needed for integration are available. d) A system composed of implemented system elements is integrated. e) The interfaces between the implemented system elements that compose the system are checked. f) The interfaces between the system and the external environment are checked. g) Integration results and anomalies are identified. h) Traceability of the integrated system elements is established.

Base practices	**TEC.8.BP1: Prepare for integration.** [Outcome: a, b, c] 1) Define the integration strategy. 2) Identify and define criteria for integration and points at which the correct operation and integrity of the interfaces and the selected software system functions will be verified. 3) Identify and plan for the necessary enabling systems or services needed to support integration. 4) Obtain or acquire access to the enabling systems or services to be used. 5) Identify constraints for integration to be incorporated in the system/software requirements, architecture or design. **TEC.8.BP2: Perform integration.** Successively integrate software system element configurations until the complete system is synthesized. [Outcome: d, e, f] 1) Obtain implemented software system elements in accordance with agreed schedules. 2) Integrate the implemented elements. 3) Check that the integrated software interfaces or functions run from initiation to an expected termination within an expected range of data values. **TEC.8.BP3: Manage results of integration.** [Outcome: g, h] 1) Record integration results and anomalies encountered. 2) Maintain traceability of the integrated software system elements. 3) Provide key artefacts and information items that have been selected for baselines.
Process outputs	Integration approach [Outcome b] Constraints on solution [Outcome a] Integrated system or system element (e.g. software) [Outcome d] System/software interface definition [Outcome: e, f] Requirements imposed on enabling systems [Outcome c] Traceability mapping [Outcome: g] Integration report [Outcome g] Integration record [Outcome a, b, c, d, e, f, g, h]

NOTE 1 Software system integration iteratively combines implemented software system elements to form complete or partial system configurations in order to build a product or service. Software integration is typically performed daily or continuously during development and maintenance stages, using automated tools. Continuous integration involves frequent inclusion or replacement and archiving of items in software libraries under CM control. (ISO/IEC/IEEE 12207)

NOTE 2 Interfaces are defined by the architecture definition and design definition processes. The integration process coordinates with these other processes to check that the interface definitions, as implemented and integrated, are adequate and that they take into account the integration needs. (ISO/IEC/IEEE 12207)

5.5.10 Verification process

Process ID	**TEC.9**
Process name	**Verification process**

Process purpose	The purpose of the verification process is to provide objective evidence that a system or system element fulfils its specified requirements and characteristics. The verification process identifies the anomalies (errors, defects, or faults) in any information item (e.g. system/software requirements or architecture description), implemented system elements, or life cycle processes using appropriate methods, techniques, standards or rules. This process provides the necessary information to determine resolution of identified anomalies.
Process outcomes	As a result of the successful implementation of the verification process: a) Constraints of verification that influence the requirements, architecture, or design are identified. b) Any enabling systems or services needed for verification are available. c) The system or system element is verified. d) Data providing information for corrective actions is reported. e) Objective evidence that the realized system fulfils the requirements, architecture and design is provided. f) Verification results and anomalies are identified. g) Traceability of the verified system elements is established.
Base practices	**TEC.9.BP1: Prepare for verification.** [Outcome: a, b] 1) Define the verification strategy. 2) Identify constraints from the verification strategy to be incorporated in the system/software requirements, architecture, or design. 3) Define the purpose, conditions and conformance criteria for each verification action. 4) Select appropriate verification methods or techniques and associated criteria for verification actions, such as inspection, analysis, demonstration, or testing. 5) Identify and plan for the necessary enabling systems or services needed to support verification. 6) Obtain or acquire access to the enabling systems or services to be used to support verification. **TEC.9.BP2: Perform verification.** [Outcome: c, e, f] 1) Define the verification procedures, each supporting one or a set of verification actions. 2) Perform the verification procedures. **TEC.9.BP3: Manage results of verification.** [Outcome: d, e, f, g] 1) Review verification results and anomalies encountered and identify follow-up actions. 2) Record incidents and problems during verification and track their resolution. 3) Obtain stakeholder agreement that the software system or element meets the specified requirements. 4) Maintain traceability of the verified software system elements. 5) Provide key artefacts and information items that have been selected for baselines.

Process outputs	Verification approach [Outcome a, b, c, d, e, f, g] Constraints on solution [Outcome a] Verification criteria [Outcome: e] Verified system/software [Outcome c] Requirements imposed on enabling systems [Outcome b] Traceability mapping [Outcome: g] Verification report [Outcome d, e, f] Verification record [Outcome a, b, c, d, e, f, g]

NOTE 1 Verification can be performed across all technical processes. The verification process is typically used at key points in a software system's life cycle to demonstrate that the requirements (including functional and non-functional requirements) have been met, or that process outcomes have been achieved or process activities have been performed. Different domains and engineering or development communities can identify the milestones, verification strategies and criteria differently. (ISO/IEC/IEEE 12207)

NOTE 2 For software systems, the verification process is typically instantiated for the following purposes:

a) to confirm that a software work product or service properly reflects the specified requirements (often called software verification);

b) to confirm that the integrated software product meets its defined requirements (often called software qualification testing);

c) to confirm that the implementation of each system/software requirement is tested for compliance and that the software system is ready for delivery (often called system qualification testing). (ISO/IEC/IEEE 12207)

NOTE 3 The verification process determines that the "product is built right". The validation process determines that the "right product is built". (ISO/IEC/IEEE 12207)

NOTE 4 ISO/IEC/IEEE 29119 provides detailed processes and techniques for verification performed through testing. IEEE 1012-2012 provides additional details about these processes for systems, software, hardware, and interfaces being developed, maintained, or reused. (ISO/IEC/IEEE 12207)

NOTE 5 The SWEBOK, Guide to the Software Engineering Body of Knowledge, provides detailed discussion on software testing. This knowledge area addresses fundamentals, terminology, issues, techniques, application, process planning, measures, tools, practical considerations, and references. The guide also discusses software verification and validation in terms of software quality management processes, and identifies methods and techniques that support both verification and validation. The SWEBOK also addresses topics such as software construction for verification and software engineering models and methods support. (ISO/IEC/IEEE 12207)

5.5.11 Transition process

Process ID	**TEC.10**
Process name	**Transition process**
Process purpose	The purpose of the transition process is to establish a capability for a system to provide services specified by stakeholder requirements in the operational environment. This process moves the system in an orderly, planned manner into the operational status, such that the system is functional, operable and compatible with other operational systems. It installs a verified system, together with relevant enabling systems, e.g. planning system, support system, operator training system, user training system, as defined in agreements. This process is used at each level in the system structure and in each stage to complete the criteria established for exiting the stage. It includes preparing applicable storage, handling, and shipping enabling systems.

Process outcomes	As a result of the successful implementation of the transition process: a) Transition constraints that influence system/software requirements, architecture, or design are identified. b) Any enabling systems or services needed for transition are available. c) The site is prepared. d) The system, as installed in its operational location, is capable of delivering its specified functions. e) Operators, users and other stakeholders necessary to the system utilization and support are trained. f) Transition results and anomalies are identified. g) The installed system is activated and ready for operation. h) Traceability of the transitioned elements is established.
Base practices	**TEC.10.BP1: Prepare for software system transition.** [Outcome: a, b, c] 1) Define a strategy for managing software releases and other software system transitions. 2) Identify and define facility, site, communications network, or target environment changes needed for software system installation or transition. 3) Identify information needs and arrange for user documentation and training of operators, users, and other stakeholders necessary for system utilization and support. 4) Prepare detailed transition information, such as plans, schedules, and procedures. 5) Identify system constraints from transition to be incorporated in the software system requirements, architecture or design. 6) Identify and plan for the necessary enabling systems or services needed to support transition. 7) Obtain or acquire access to the enabling systems or services to be used. **TEC.10.BP2: Perform the transition.** [Outcome: c, d, e, g] 1) Prepare the site of operation or virtual environment in accordance with installation requirements. 2) Deliver the software system or element for installation at the correct location and time. 3) Install the product in its physical or virtual operational location and interface to its environment. 4) Provide user documentation and training for the operators, users, and other stakeholders necessary for product utilization and support. 5) Perform activation and check-out. **TEC.10.BP3: Manage results of transition.** [Outcome: f, h] 1) Record transition results and anomalies encountered. 2) Record transition incidents and problems and track their resolution. 3) Maintain traceability of the transitioned software system elements. 4) Provide key artefacts and information items that have been selected for baselines.

Process outputs	Transition approach [Outcome a] Constraints on solution [Outcome a] Installed system/software [Outcome c, d, g] Requirements imposed on enabling systems [Outcome b] Traceability mapping [Outcome: h] Transition report [Outcome f] Transition record [Outcome a, b, c, d, e, f, g, h]

NOTE 1 For software systems, the purpose of the transition process is to establish a capability for a system to provide services in a different environment. The transition process is often used for recurring deployments of software to different environments, e.g. from a development environment to a test or maintenance environment, or between various test environments, or from one operational environment to another (e.g. rehosting or use of cloud services). Transitions to backup or contingent sites are typically planned and rehearsed for business continuity and disaster recovery. Transition for software systems can involve the physical relocation of hardware, the installation and activation or deactivation of physical or virtual infrastructure or enabling systems in different locations, or no change to the physical infrastructure. Transition can involve changes to the data sources, data structure, or updates or upgrades of functional software. Transition includes recurring scheduled or emergency patches and fixes for security and other concerns. Transition can involve transfer between organizations and also encompasses the addition of a large group of new users to an existing software system or service. Transition to a new system often is performed concurrently with retirement and disposal of an existing system, entailing data migration from the old system to its replacement. (ISO/IEC/IEEE 12207)

NOTE 2 Transition can involve knowledge transfer using the knowledge management process. (ISO/IEC/IEEE 12207)

5.5.12 Validation process

Process ID	**TEC.11**
Process name	**Validation process**
Process purpose	The purpose of the validation process is to provide objective evidence that the system, when in use, fulfils its business or mission objectives and stakeholder requirements, achieving its intended use in its intended operational environment. The objective of validating a system or system element is to acquire confidence in its ability to achieve its intended mission, or use, under specific operational conditions. Validation is ratified by stakeholders. This process provides the necessary information so that identified anomalies can be resolved by the appropriate technical process where the anomaly was created.
Process outcomes	As a result of the successful implementation of the validation process: a) Validation criteria for stakeholder requirements are defined. b) The availability of services required by stakeholders is confirmed. c) Constraints of validation that influence the requirements, architecture, or design are identified. d) The system or system element is validated. e) Any enabling systems or services needed for validation are available. f) Validation results and anomalies are identified. g) Objective evidence that the realized system or system element satisfies stakeholder needs is provided. h) Traceability of the validated system elements is established.

Base practices	**TEC.11.BP1: Prepare for validation.** [Outcome: a, b, c, e] 1) Define the validation strategy. 2) Identify system constraints from the validation strategy to be incorporated in the stakeholder requirements. 3) Define the purpose, conditions and conformance criteria for each validation action. 4) Select appropriate validation methods or techniques and associated criteria for each validation action. 5) Identify and plan for the necessary enabling systems or services needed to support validation. 6) Obtain or acquire access to the enabling systems or services to be used to support validation. **TEC.11.BP2: Perform validation.** [Outcome: b, d, f] 1) Define the validation procedures, each supporting one or a set of validation actions. 2) Perform the validation procedures in the defined environment. **TEC.11.BP3: Manage results of validation.** [Outcome: f, g, h] 1) Review validation results and anomalies encountered and identify follow-up actions. 2) Record incidents and problems during validation and track their resolution. 3) Obtain stakeholder agreement that the system or system element meets the stakeholder needs. 4) Maintain traceability of the validated system elements. 5) Provide key artefacts and information items that have been selected for baselines.
Process outputs	Validation approach [Outcome a, c] Constraints on solution [Outcome c] Validation criteria [Outcome: a] Validated system/software [Outcome b, d] Requirements imposed on enabling systems [Outcome e] Traceability mapping [Outcome: h] Validation report [Outcome f, g] Validation record [Outcome a, b, e, f, g]

NOTE 1 The validation process is typically used at key points in a product’s life cycle to demonstrate that the product’s requirements for stakeholder intended operational use have been met. Validation is also applicable to the software engineering artefacts (viewed as software system elements). Different domains and engineering or development communities can identify the milestones, validation strategies and criteria differently. For software systems, highly iterative life cycle models often feature frequent involvement by the acquirer, user representative, or other stakeholders to validate, e.g. the priority of requirements for inclusion in an iteration, the usability of the software interface through prototypes, and the suitability of the software for performing business tasks and fulfilling the operational concept. (ISO/IEC/IEEE 12207)

NOTE 2 For software systems, the following are purposes of the validation process:

a) to confirm that the requirements for a specific intended use of the software work product are fulfilled (often called software validation);

b) to achieve confidence (especially with an acquirer or customer) that the delivered product meets stakeholder requirements and is fit for use (often called software acceptance testing). (ISO/IEC/IEEE 12207)

NOTE 3 The validation process determines that the "right product is built". The verification process determines that the "product is built right". (ISO/IEC/IEEE 12207)

NOTE 4 Acceptance criteria, as used for acceptance testing, include criteria to determine whether the delivered product is fit to use or not. Acceptance criteria for acceptance can be specified and agreed between two parties, i.e., an acquirer and a supplier, and included in the stakeholder requirements. (ISO/IEC/IEEE 12207)

NOTE 5 IEEE 1012-2012 provides detailed requirements. The SWEBOK, Guide to the Software Engineering Body of Knowledge, discusses software verification and validation in terms of software quality management processes, and contains methods and techniques that support both verification and validation. The SWEBOK also addresses topics such as requirements and model validation. (ISO/IEC/IEEE 12207)

5.5.13 Operation process

Process ID	**TEC.12**
Process name	**Operation process**
Process purpose	The purpose of the operation process is to use the system to deliver its services. This process establishes requirements for and assigns personnel to operate the system, and monitors the services and operator-system performance. In order to sustain services, it identifies and analyses operational anomalies in relation to agreements, stakeholder requirements and organizational constraints.
Process outcomes	As a result of the successful implementation of the operation process: a) Operation constraints that influence system/software requirements, architecture, or design are identified. b) Any enabling systems, services, and material needed for operation are available. c) Trained, qualified operators are available. d) System product services that meet stakeholder requirements are delivered. e) System product performance during operation is monitored. f) Support to the customer is provided.

Base practices	**TEC.12.BP1: Prepare for operation.** [Outcome: a, b, c] 1) Define an operation strategy. 2) Identify system constraints from operation to be incorporated in changes to the system/software requirements, architecture, design, implementation, or transition. 3) Identify and plan for the necessary enabling systems or services needed to support operation. 4) Obtain or acquire access to the enabling systems or services to be used. 5) Identify or define training and qualification requirements for personnel needed for software system operation. 6) Depending on the need for human intervention and control of operations, assign trained, qualified personnel to be operators. **TEC.12.BP2: Perform operation.** [Outcome: d, e, f] 1) Use the software system in its intended operational environment. 2) Apply materials and other resources, as required, to operate the software system and sustain its services. 3) Monitor software system operation. 4) Consistent with the operational strategy, develop and, where feasible, automate operational procedures to minimize the risk of operational anomalies. 5) Consistent with the operational strategy, analyse measurements to confirm that service performance is within acceptable parameters. 6) Perform contingency operations, if necessary. **TEC.12.BP3: Manage results of operation.** [Outcome: e] 1) Record results of operation and anomalies encountered. 2) Record operational incidents and problems and track their resolution. 3) Maintain traceability of the operational services and configuration items. 4) Provide key artefacts and information items that have been selected for baselines. **TEC.12.BP4: Support the customer.** [Outcome: d, e, f] 1) Provide assistance and consultation to the customers and users to resolve complaints, incidents, problems, and service requests. 2) Record and monitor requests and subsequent actions for support. 3) Determine the degree to which delivered software system or services satisfy the needs of the customers and users.

Process outputs	Operation approach [Outcome a] Constraints on solution [Outcome a] Trained operator [Outcome c] Operational system/software [Outcome d, f] Requirements imposed on enabling systems [Outcome b] Traceability mapping [Outcome: e] Operation report [Outcome e] Operation record [Outcome a, b, c, d, e, f]

NOTE 1 The operation process typically aims to control or reduce the cost of operations while sustaining an acceptable or improved level of service. Software systems can have dedicated infrastructure, but are typically operated in distributed environments where other software systems and services (e.g. the internet) are active. The security, availability, and operational performance of the software system-of-interest are thus a matter of concern within a larger system of systems. It can include coordination with pre-existing, concurrent or continuing services delivered by other systems that provide identical or similar services. (ISO/IEC/IEEE 12207)

NOTE 2 ISO/IEC 20000-1 is a service management system standard that specifies requirements for the design, transition, delivery and improvement of managed operational services, and supports the operation process to achieve its purpose. (ISO/IEC/IEEE 12207)

5.5.14 Maintenance process

Process ID	**TEC.13**
Process name	**Maintenance process**
Process purpose	The purpose of the maintenance process is to sustain the capability of the system to provide a service. This process monitors the system's capability to deliver services, records incidents for analysis, takes corrective, adaptive, perfective and preventive actions and confirms restored capability.
Process outcomes	As a result of the successful implementation of the maintenance process: a) Maintenance constraints that influence system requirements, architecture, or design are identified. b) Any enabling systems or services needed for maintenance are available. c) Replacement, repaired, or revised system elements are made available. d) The need for changes to address corrective, perfective, or adaptive maintenance is reported. e) Failure and lifetime data, including associated costs, is determined.

Base practices	**TEC.13.BP1: Prepare for maintenance.** [Outcome: a, b] 1) Define a maintenance strategy. 2) For non-software elements, define a logistics strategy throughout the life cycle, including acquisition and operational considerations: the number and type of replacement elements to be stored, their storage locations and conditions, their anticipated replacement rate, and their storage life and renewal frequency. 3) Identify constraints from maintenance to be incorporated in the system/software requirements, architecture, or design. 4) Identify trades such that the system and associated maintenance and logistics actions results in a solution that is affordable, operable, supportable, and sustainable. 5) Identify and plan for the necessary enabling systems or services needed to support maintenance. 6) Obtain or acquire access to the enabling systems or services to be used. **TEC.13.BP2: Perform maintenance.** [Outcome: c, d, e] 1) Review stakeholder requirements, complaints, events, incident and problem reports to identify corrective, adaptive, perfective and preventive maintenance needs. 2) Analyse the impact of maintenance changes on data structures, data, and related software functions, user documentation, and interfaces. 3) Upon encountering unexpected faults that cause a software system failure, restore the system to operational status. 4) Implement the procedures for correction of flaws (defects) and errors, or for replacement or upgrade of system elements. 5) Perform preventive maintenance by replacing, patching, augmenting, or upgrading software system elements, to improve the performance of a software system that is projected to reach unacceptable service levels, e.g. lack of capacity due to increases in demand or stored data, or to avoid unacceptable operating conditions, e.g. running with outdated security software. 6) Identify when adaptive or perfective maintenance is required. **TEC.13.BP3: Perform logistics support.** [Outcome: b, c] 1) Obtain resources to support the software system through its life cycle or the project's life (acquisition logistics). 2) Monitor the quality and availability of replacement elements and enabling systems, their delivery mechanisms and their continued integrity during storage. 3) Implement mechanisms for software system or element distribution, including packaging, handling, storage and communications or transportation needed for items during the life cycle. 4) Confirm that logistics actions to fulfil software system or element supportability requirements or achieve operational readiness are planned and implemented. **TEC.13.BP4: Manage results of maintenance and logistics**. [Outcome: d, e] 1) Record incidents and problems, including their resolutions, and significant maintenance and logistics results. 2) Identify and record trends of incidents, problems, and maintenance and logistics actions. 3) Maintain traceability of the system elements being maintained. 4) Provide key artefacts and information items that have been selected for baselines. 5) Monitor and measure customer satisfaction with system and maintenance support.

Process outputs	Maintenance approach [Outcome a] Constraints on solution [Outcome a] Trained maintainer [Outcome b] Maintained system/software [Outcome c] Requirements imposed on enabling systems [Outcome b] Traceability mapping [Outcome: d] Maintenance report [Outcome d, e] Maintenance record [Outcome a, b, c, d, e]

NOTE 1 For software systems, the maintenance process makes corrections, changes, and improvements to deployed software systems and elements. The software systems maintenance approach differs for systems that are freely available, in wide commercial distribution, or operating in a small number of controlled environments. The need for software system maintenance can arise from multiple causes other than latent system defects, such as changes to interfaced systems or infrastructure, evolving security threats, and technical obsolescence of system elements and enabling systems over the system life cycle. Often the extension of capability, mid-life upgrade, or evolution of legacy systems becomes a new software system development project that will apply the set of processes within an appropriate life cycle. If so, the portfolio management process is the starting point to initiate the work. In other cases, software system maintenance is performed as a continuing series of prioritized work items, possibly on a level of effort basis. Maintenance of software system elements can include hardware, software, and services, such as communication or web services. Maintenance is closely connected with the configuration management process and software asset management and is performed concurrently with the other technical processes. (ISO/IEC/IEEE 12207)

NOTE 2 ISO/IEC/IEEE 14764 and ISO/IEC 16350 provide additional detail. The SWEBOK, Guide to the Software Engineering Body of Knowledge, Software Maintenance knowledge area discusses software maintenance fundamentals, key issues, measurement, techniques, maintenance process and support activities, and tools. The guide also discusses models, techniques and measures that support software reliability. (ISO/IEC/IEEE 12207)

5.5.15 Disposal process

Process ID	**TEC.14**
Process name	**Disposal process**
Process purpose	The purpose of the disposal process is to end the existence of a system element or system for a specified intended use, appropriately handle replaced or retired elements, and to properly attend to identified critical disposal needs (e.g. per an agreement, per organizational policy, or for environmental, legal, safety, security aspects). This process deactivates, disassembles and removes the system or any of its system elements from the specific use. It addresses any waste products, consigning them to a final condition and returning the environment to its original or an acceptable condition. The waste products can be in-process resulting during any life cycle stage, e.g. waste materials during fabrication. This process destroys, stores or reclaims system elements and waste products in an environmentally sound manner, in accordance with legislation, agreements, organizational constraints and stakeholder requirements. Disposal includes preventing expired, non-reusable, or inadequate elements from getting back into the supply chain. Where required, it maintains records in order that the health of operators and users, and the safety of the environment, can be monitored. When part of the system will continue to be in use in a modified form, the disposal process helps ensure the proper handling of the portion being retired.

Process outcomes	As a result of the successful implementation of the disposal process: a) Disposal constraints are provided as inputs to requirements, architecture, design, and implementation. b) Any enabling systems or services needed for disposal are available. c) The system elements or waste products are destroyed, stored, reclaimed or recycled in accordance with requirements, e.g. safety and security requirements. d) The environment is returned to its original or an agreed state. e) Records of disposal actions and analysis are available.
Base practices	**TEC.14.BP1: Prepare for disposal.** [Outcome: a, b] 1) Define a disposal strategy for the software system, to include each system element and to identify and address critical disposal needs. 2) Identify constraints on disposal for the system/software requirements, architecture and design characteristics, or implementation techniques. 3) Identify and plan for the necessary enabling systems or services needed to support disposal. 4) Obtain or acquire access to the enabling systems or services to be used. 5) Specify containment facilities, storage locations, inspection criteria and storage periods, if the software system or data is to be stored, consistent with security and environmental considerations. 6) Define preventive methods to preclude disposed elements and materials that should not be repurposed, reclaimed or reused from re-entering the supply chain. **TEC.14.BP2: Perform disposal.** [Outcome: c] 1) Deactivate the software system or element to prepare it for removal. 2) Remove the software system, its elements, its data, and non-reusable material from use or production for appropriate disposition and action. 3) Withdraw impacted operating staff from the software system or system element and record relevant operating knowledge. 4) Reuse, recycle, recondition, overhaul, archive, or destroy designated software system elements. 5) Conduct destruction of the system elements, as necessary, to reduce the amount of waste treatment or to make the waste easier to handle. **TEC.14.BP3: Finalize the disposal.** [Outcome: d, e] 1) Confirm that detrimental health, safety, security, and environmental conditions following disposal have been identified and treated. 2) Return the environment to its original state or to a state that specified by agreement. 3) Archive information gathered through the lifetime of the product to permit audits and reviews in the event of long-term hazards to health, safety, security and the environment, and to permit future software system creators and users to build a knowledge base from experience.

Process outputs	Disposal approach [Outcome a] Constraints on solution [Outcome a] Disposed system/software [Outcome c, d] Requirements imposed on enabling systems [Outcome b] Traceability mapping [Outcome: c] Disposal report [Outcome a, c] Disposal record [Outcome e]

NOTE 1 Disposal of software systems encompasses the termination of services and disposal of software elements, stored data, media and firmware, information items, and associated hardware elements that will not be reused or transitioned to another system. The disposal process is intended to be applicable in any stage of a software systems life cycle. For software, the disposal process applies throughout the life cycle to source code or executable copies of the software, personally identifiable or controlled data used in the software system, and associated information items, retained under centralized configuration control or distributed for use, e.g. disposing of prototypes in early life cycle stages, and decommissioning elements replaced from modifications during utilization/deployment and support stages. When the system-of-interest is being modified for technology or capability upgrades, only the impacted elements are deactivated and removed. (ISO/IEC/IEEE 12207)

NOTE 2 The business or mission analysis process and decision management process are typically applied to address the impact on stakeholders of system disposal and potential new system capabilities. (ISO/IEC/IEEE 12207)

6 The quality dimension

A process assessment model shall incorporate a process measurement framework conformant with the requirements of ISO/IEC 33003 and is expressed as a process quality characteristic with a defined set of process attributes. At minimum, a process measurement framework includes a process quality attribute of process performance, which is needed to demonstrate that the process achieves its expected process outcomes. Other process quality attributes may be added over the process performance attribute.

NOTE 1 ISO/IEC 33020 provides a process measurement framework for the assessment of process capability which can be incorporated into this document. ISO/IEC 33020 also includes a set of process quality indicators for each process attribute in the process measurement framework.

The assessment indicators are used as a basis for collecting objective evidence to support an assessor's judgement in assigning ratings of the performance and quality of an implemented process. The set of indicators defined in this document are not intended to be an all-inclusive set and applicable in its entirety. Subsets appropriate to the context and scope of the assessment should be selected, and potentially augmented with additional indicators.

A process assessment is conducted according to a documented assessment process. A documented assessment process identifies the rating method to be used in rating process attributes and identifies or defines the aggregation method to be used in determining ratings.

NOTE 2 ISO/IEC 33020 includes a process attribute rating scale, process attribute rating method, and aggregation method which can provide a suitable basis for use for incorporating into any documented assessment process.

Annex A
(informative)

Process outputs

A.1 Process output descriptions

The process outputs associated with the processes in Clause 5 are described in Table A.1. The descriptions are exemplary. The process outputs are identified by categories which are defined in A.2. The corresponding processes of the process outputs are indicated by the process IDs.

NOTE For further guidance, ISO/IEC/IEEE 15289 addresses the content for life cycle process information items (documentation).

Table A.1 — Process output descriptions

Process output	Process output description	Category	Output of
Accepted system or system element (e.g. software)	System element or system is transferred from supplier to acquirer and the product or service is available to the project. [INCOSE SE Handbook 2015]	product	AGR.1
Acquisition agreement	The formal agreement between an acquirer and a supplier. Informally, commitments or agreements may be specified between an acquirer and a supplier of the same organization (sometimes called a memorandum of understanding). [Adapted from ISO/IEC/IEEE 15289]	agreement	AGR.1
Acquisition agreement change request	Requests from an acquirer to change an agreement with a supplier.	request	AGR.1
Acquisition approach	Specific approach to acquiring products and services that is based on considerations of supply sources, acquisition methods, requirements specification types, contract or agreement types, and related acquisition risks. [ISO/IEC/IEEE 24765]	plan	AGR.1
Acquisition need	The identification of a need that cannot be met within the organization encountering the need or a need that can be met in a more economical way by a supplier. [INCOSE SE Handbook 2015]	specification	MAN.1
Acquisition record	Permanent, readable form of data, information, or knowledge related to acquisition. [INCOSE SE Handbook 2015]	record	AGR.1
Acquisition report	An account prepared for interested parties in order to communicate the status, results, and outcomes of the acquisition activities. [INCOSE SE Handbook 2015]	report	AGR.1
Alternative solution classes	Identifies and describes the classes of solutions that may address the problem or opportunity. [INCOSE SE Handbook 2015]	description	TEC.1
Architecture definition approach	Approaches, schedules, resources, and specific considerations required to define the selected system/software architecture that satisfies the requirements. [Adapted from INCOSE SE Handbook 2015]	plan	TEC.4
Architecture definition record	Permanent, readable form of data, information, or knowledge related to architecture definition. [INCOSE SE Handbook 2015]	record	TEC.4

Table A.1 *(continued)*

Process output	**Process output description**	**Category**	**Output of**
Business or mission analysis approach	Approaches, schedules, resources, and specific considerations required to conduct business or mission analysis and ensure business needs are elaborated and formalized into business requirements. [INCOSE SE Handbook 2015]	plan	TEC.1
Business or mission analysis record	Permanent, readable form of data, information, or knowledge related to business or mission analysis. [INCOSE SE Handbook 2015]	record	TEC.1
Configuration audit result	Artefacts that are expected through conduct of configuration audit and that can be considered elements of exit criteria [Adapted from IEEE 15288.2-2014]	record	MAN.5
Configuration baseline	Configuration information formally designated at a specific time during the life of a product, product component, service, or service component. [ISO/IEC/IEEE 24765]	record	MAN.5
Configuration evaluation report	Provides results of configuration management evaluations. It includes evaluation criteria. [Adapted from ISO/IEC/IEEE 15289]	report	MAN.5
Configuration management approach	Describes the responsible organization for authorizing and performing configuration management activities, and their relationship with other organizations. [Adapted from ISO/IEC/IEEE 15289]	plan	MAN.5
Configuration management change request	Identifies a problem, maintenance need, or desired improvement and requests modifications. The requested change may affect a contract, configuration item, system, service, hardware, software, interface, asset, or documentation. [Adapted from ISO/IEC/IEEE 15289]	request	MAN.5
Configuration management record	Permanent, readable form of data, information, or knowledge related to configuration management. [INCOSE SE Handbook 2015]	record	MAN.5
Configuration management report	Provides the status of controlled configuration items, including baselines, release identifiers, and location of the configuration item master version. For deactivated systems/software, it contains information about system/software disposal to trace potential future environmental, safety, or security impacts. [Adapted from ISO/IEC/IEEE 15289]	report	MAN.5
Configuration management system	System used to support and enable configuration management.	product	MAN.5
Configuration management variance request	Request to accept a configuration item or other designated item which, during production or after having been submitted for inspection, is found to depart from specified requirements, but is nevertheless considered suitable for use as is or after rework by an approved method. [Adapted from ISO/IEC/IEEE 24765]	request	MAN.5
Constraints on solution	Externally imposed limitation on system requirements, design, or implementation or on the process used to develop or modify a system/software. [Adapted from ISO/IEC/IEEE 29148]	specification	TEC.7, TEC.8, TEC.9, TEC.10, TEC.11, TEC.12, TEC.13, TEC.14
Critical performance data	Data provided for the identified system of interest (including software as a system) measurement needs. [Adapted from INCOSE SE Handbook 2015]	data	TEC.2, TEC.3, TEC.4, TEC.5