INTERNATIONAL STANDARD

ISO/IEC/IEEE 29148

Second edition
2018-11

Systems and software engineering — Life cycle processes — Requirements engineering

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

Reference number
ISO/IEC/IEEE 29148:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Institute of Electrical and Electronics Engineers, Inc
3 Park Avenue, New York
NY 10016-5997, USA
Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

Contents

Page

Foreword

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

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 rules given in the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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

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

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

For an explanation 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.

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

This second edition cancels and replaces the first edition (ISO/IEC/IEEE 29148:2011), which has been technically revised.

Changes in this revision of ISO/IEC/IEEE 29148 were developed in response to the revision of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207. The purpose of these revisions is to accomplish the harmonization of the structures and contents of the two documents, while supporting the requirements of the assessment community.

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.

Introduction

This document provides a unified treatment of the processes and products involved in engineering requirements throughout the life cycle of systems and software. It provides details for the construct of well-formed textual requirements, to include characteristics and attributes, in the context of system and software engineering. This document also provides guidance for the implementation of requirements related processes from ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207. Finally, this document identifies information items related to requirements engineering and their content.

Systems and software engineering — Life cycle processes — Requirements engineering

1 Scope

This document:

— specifies the required processes implemented in the engineering activities that result in requirements for systems and software products (including services) throughout the life cycle;

— provides guidelines for applying the requirements and requirements-related processes described in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207;

— specifies the required information items produced through the implementation of the requirements processes;

— specifies the required contents of the required information items;

— provides guidelines for the format of the required and related information items.

This document is applicable to:

— those who use or plan to use ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 on projects dealing with man-made systems, software-intensive systems, software and hardware products, and services related to those systems and products, regardless of the project scope, product(s), methodology, size or complexity;

— anyone performing requirements engineering activities to aid in ensuring that their application of the requirements engineering processes conforms to ISO/IEC/IEEE 15288 and/or ISO/IEC/IEEE12207;

— those who use or plan to use ISO/IEC/IEEE 15289 on projects dealing with man-made systems, software-intensive systems, software and hardware products and services related to those systems and products, regardless of the project scope, product(s), methodology, size or complexity;

— anyone performing requirements engineering activities to aid in ensuring that the information items developed during the application of requirements engineering processes conforms to ISO/IEC/IEEE 15289.

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 15288:2015, *Systems and software engineering — System life cycle processes*

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

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC/IEEE 15288, ISO/IEC/IEEE 12207 and the following apply.

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

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

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

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

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

3.1.1
acquirer
stakeholder (3.1.28) that acquires or procures a product or service from a *supplier* (3.1.31)

Note 1 to entry: Other terms commonly used for an acquirer are buyer, *customer* (3.1.9), owner, purchaser or internal/organizational sponsor.

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.1]

3.1.2
attribute
inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means

Note 1 to entry: ISO 9000 distinguishes two types of attributes: a permanent characteristic existing inherently in something; and an assigned characteristic of a product, process, or system (e.g., the price of a product, the owner of a product).

[SOURCE: ISO/IEC 25000:2014, 4.1, modified — The original NOTE 1 has been removed; NOTE 2 has become Note 1 to entry.]

3.1.3
baseline
formally approved version of a configuration item, regardless of media, formally designated and fixed at a specific time during the configuration item's life cycle

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.10]

3.1.4
business requirements specification
structured collection of the *requirements* (3.1.19) (business or mission problem or opportunity definition, concepts, and required conditions of solutions) of the business or mission and its relation to the external environment

3.1.5
concept of operations
verbal and graphic statement, in broad outline, of an organization's assumptions or intent in regard to an operation or series of operations

Note 1 to entry: The concept of operations frequently is embodied in long-range strategic plans and annual operational plans. In the latter case, the concept of operations in the plan covers a series of connected operations to be carried out simultaneously or in succession. The concept is designed to give an overall picture of the organization operations. See also *operational concept* (3.1.16).

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

Note 3 to entry: The concept of operations includes how an enterprise or organization intends to employ available human and technological resources to achieve one or more outcomes.

[SOURCE: ANSI/AIAA G-043A-2012e]

3.1.6
condition
measurable qualitative or quantitative *attribute* (3.1.2) that is stipulated for a *requirement* (3.1.19) and that indicates a circumstance or event under which a requirement applies

3.1.7
constraint
externally imposed limitation on the system, its design, or implementation or on the process used to develop or modify a system

Note 1 to entry: A constraint is a factor that is imposed on the solution by force or compulsion and may limit or modify the design.

3.1.8
context of use
users (3.1.35), tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used

[SOURCE: ISO/IEC 25000:2014, 4.2]

3.1.9
customer
person or organization that could or does receive a product or a service that is intended for or required by this person or organization

Note 1 to entry: Customers are a subset of *stakeholders* (3.1.28).

Note 2 to entry: A customer can be internal or external to the organization.

[SOURCE: ISO 9000:2015, 3.2.4, modified — The original Note 1 to entry has become Note 2 to entry; a new Note 1 to entry has been introduced and the EXAMPLE has been removed.]

3.1.10
derived requirement
requirement (3.1.19) deduced or inferred from the collection and organization of requirements into a particular system configuration and solution

Note 1 to entry: The next higher level requirement is referred to as a "parent" requirement while the derived requirement from this parent is called a "child" requirement.

Note 2 to entry: A derived requirement is typically identified during the elicitation of *stakeholder* (3.1.28) requirements, requirements analysis, trade studies or *validation* (3.1.36).

3.1.11
developer
individual or organization that performs development activities (including requirements analysis, design, testing through acceptance) during the system or software life-cycle process

Note 1 to entry: Developers are a subset of *stakeholders* (3.1.28).

[SOURCE: ISO/IEC 25000:2014, 4.6, modified — Note 1 to entry has been added.]

3.1.12
document
uniquely identified unit of information for human use

EXAMPLE report, specification, manual or book in printed or electronic form.

Note 1 to entry: A document can be a single *information item* (3.1.14), or part of a larger *information item* (3.1.14).

Note 2 to entry: Documents include both paper and electronic documents.

Note 3 to entry: The use of the term document is not intended to preclude interpretation in the broader sense of "documentation", especially as it applies to model-based engineering or conformance.

[SOURCE: ISO/IEC/IEEE 15289:2017, 3.1.10, modified – Notes 2 and 3 to entry have been added.]

3.1.13
human systems integration
interdisciplinary technical and management processes for integrating human considerations within and across all system elements

[SOURCE: INCOSE SEHbk 4:2015]

3.1.14
information item
separately identifiable body of information that is produced, stored, and delivered for human use

[SOURCE: ISO/IEC/IEEE 15289:2017, 3.1.13, modified – Notes 1 and 2 to entry have been removed.]

3.1.15
level of abstraction
view at a specific level of detail in a description of a system

3.1.16
operational concept
verbal and graphic statement of an organization's assumptions or intent in regard to an operation or series of operations of a specific system or a related set of specific new, existing or modified systems

Note 1 to entry: The operational concept is designed to give an overall picture of the operations using one or more specific systems, or set of related systems, in the organization's operational environment from the *users*' (3.1.35) and *operators*' (3.1.18) perspective. It is what the enterprise or organization intends to achieve. See also *concept of operations* (3.1.5).

[SOURCE: ANSI/AIAA G-043A-2012e]

3.1.17
operational scenario
description of an imagined sequence of events or activities that includes the interaction of the product or service with its environment and *users* (3.1.35), as well as interaction among its product or service components when there is end-use significance

Note 1 to entry: Operational scenarios are used to evaluate the *requirements* (3.1.19) and design of the system and to verify and validate the system.

3.1.18
operator
individual or organization that performs the operations of a system

Note 1 to entry: The role of operator and the role of *user* (3.1.35) can be vested, simultaneously or sequentially, in the same individual or organization.

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

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

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.26]

3.1.19
requirement
statement which translates or expresses a need and its associated *constraints* (3.1.7) and *conditions* (3.1.6)

Note 1 to entry: Requirements exist at different levels in the system structure.

Note 2 to entry: A requirement is an expression of one or more particular needs in a very specific, precise and unambiguous manner.

Note 3 to entry: A requirement always relates to a system, software or service, or other item of interest.

3.1.20
requirements elicitation
use of systematic techniques, such as prototyping and structured surveys, to proactively identify and document *customer* (3.1.9) and end *user* (3.1.35) needs

3.1.21
requirements engineering
interdisciplinary function that mediates between the domains of the *acquirer* (3.1.1) and *supplier* (3.1.31) to establish and maintain the *requirements* (3.1.19) to be met by the system, software or service of interest

Note 1 to entry: Requirements engineering is concerned with discovering, eliciting, developing, analyzing, verifying, validating, communicating, documenting and managing requirements.

3.1.22
requirements management
activities that identify, document, maintain, communicate, trace and track *requirements* (3.1.19) throughout the life cycle of a system, product or service

3.1.23
requirements traceability
identification and documentation of the derivation path (upward) and allocation/flow-down path (downward) of *requirements* (3.1.19) in the requirements set

Note 1 to entry: One or more requirements upward from which a requirement is derived are called parent requirements. A requirement downward that is derived from one or more parent requirements is called a child requirement.

3.1.24
requirements traceability matrix
structured information artifact that links *requirements* (3.1.19) to their higher level requirements or needs or to lower level implementation

3.1.25
requirements validation
confirmation that *requirements* (3.1.19) (individually and as a set) define the right system as intended by the *stakeholders* (3.1.28)

[SOURCE: EIA 632:1999, modified]

3.1.26
requirements verification
confirmation by examination that *requirements* (3.1.19) (individually and as a set) are well-formed

Note 1 to entry: This means that a requirement or a set of requirements has been reviewed to help ensure the characteristics of good requirements are achieved and the requirements set is well organized.

[SOURCE: EIA 632:1999, modified]

3.1.27
software requirements specification
structured collection of the essential *requirements* (3.1.19) [functions, performance, design *constraints* (3.1.7) and *attributes* (3.1.2)] of the software and its external interfaces

[SOURCE: IEEE Std 1012:2016, modified — 'documentation' has become 'structured collection'.]

3.1.28
stakeholder
individual or organization having a right, share, claim or interest in a system or in its possession of characteristics that meet their needs and expectations

Note 1 to entry: Stakeholders include, but are not limited to, end *users* (3.1.35), end user organizations, supporters, *developers* (3.1.11), producers, trainers, maintainers, disposers, *acquirers* (3.1.1), *customers* (3.1.9), *operators* (3.1.18), supplier organizations, accreditors and regulatory bodies.

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.44, modified — EXAMPLE and Note 1 to entry have been removed; new Note 1 to entry has been added.]

3.1.29
stakeholder requirements specification
structured collection of the *requirements* (3.1.19) [characteristics, context, concepts, *constraints* (3.1.7) and priorities] of the *stakeholder* (3.1.28) and the relationship to the external environment

3.1.30
state
condition (3.1.6) that characterizes the behaviour of a function, subfunction or element at a point in time

Note 1 to entry: A state may be used to define sets of functions or system characteristics that can vary with time.

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.3958, modified — Definition 4 has been kept; all other definitions have been removed: new Note 1 to entry has replaced the original Note 1 to entry.]

3.1.31
supplier
organization or individual that enters into an agreement with the *acquirer* (3.1.1) for the supply of a product or service

Note 1 to entry: Suppliers are a subset of *stakeholders* (3.1.28).

Note 2 to entry: Other terms commonly used for supplier are contractor, producer, seller or vendor.

Note 3 to entry: The *acquirer* (3.1.1) and the supplier sometimes are part of the same organization.

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.45, modified — New Note 1 to entry has been added; other Notes to entry renumbered.]

3.1.32
system-of-interest
system whose life cycle is under consideration in the context of this document

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.48 modified — 'document' has replaced 'International Standard'.]

3.1.33
system requirements specification
structured collection of the *requirements* (3.1.19) [functions, performance, design *constraints* (3.1.7), and other *attributes* (3.1.2)] for the system and its operational environments and external interfaces

3.1.34
trade-off
decision-making actions that select from various *requirements* (3.1.19) and alternative solutions on the basis of net benefit to the *stakeholders* (3.1.28)

3.1.35
user
individual or group that interacts with a system or benefits from a system during its utilization

Note 1 to entry: The role of user and *operator* (3.1.18) may be vested, simultaneously or sequentially, in the same individual or organization.

Note 2 to entry: Users are a subset of *stakeholders* (3.1.28).

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.52, modified — Note 2 to entry has been added.]

3.1.36
validation
confirmation, through the provision of objective evidence, that the *requirements* (3.1.19) for a specific intended use or application have been fulfilled

Note 1 to entry: Validation in a system life cycle context is a set of activities ensuring and gaining confidence that a system is able to accomplish its intended use, goals and objectives. The right system has been built.

[SOURCE: ISO 9000:2015, 3.8.13, modified — Notes 1, 2 and 3 to entry have been removed; new Note 1 to entry has been added.]

3.1.37
verification
confirmation, through the provision of objective evidence, that specified *requirements* (3.1.19) have been fulfilled

Note 1 to entry: Verification in a system life cycle context is a set of activities that compares a product of the system life cycle against the required characteristics for that product. This may include, but is not limited to, specified requirements, design description and the system itself. The system has been built right.

[SOURCE: ISO 9000:2015, 3.8.12, modified — Notes 1, 2 and 3 to entry have been removed; new Note 1 to entry has been added.]

3.2 Abbreviated terms

BABOK	Business Analysis Book of Knowledge
BRS	Business Requirements Specification
ConOps	Concept of Operations
FSM	Functional Size Measurement
HSI	Human Systems Integration
MDD	Model Driven Development
MOP	Measures of Performance
OpsCon	Operational Concept
RTM	Requirements Traceability Matrix
SRS	Software Requirements Specification
StRS	Stakeholder Requirements Specification
SyRS	System Requirements Specification
TBD	To Be Determined

TBR	To be Resolved, To be Revised
TBS	To be Supplied, To be Specified.
TPM	Technical Performance Measure

4 Conformance

4.1 Intended usage

This document provides guidance for the execution of the ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 processes that deal with requirements engineering. This document also provides normative definition of the content and recommendations for the format of the information items or documentation that result from the implementation of these processes. Users of this document can claim conformance to the process provisions or to the information item provisions, or both.

4.2 Full conformance

A claim of full conformance to this document is equivalent to claiming conformance:

— to the provisions contained in 5.2.4, 5.2.5, 5.2.6, and 5.2.7;

— to the requirements-engineering-related processes of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 cited in 6.1;

— to the information items specified in Clause 7; and

— to the required content of the information items in Clause 9 and Annex A of this document.

4.3 Conformance to processes

This document provides requirements for a number of requirements engineering processes suitable for usage during the life cycle of a system, a product or a service.

The requirements for processes in this document are contained in 6.1.

4.4 Conformance to information item content

This document provides requirements for a number of requirements engineering information items to be produced during the life cycle of a system, a product or a service. A claim of conformance to the information item provisions of this document means that:

— the required information items stated in this document are produced; and

— it is demonstrated that the information items produced during the requirements engineering activities conform to the content requirements defined in this document.

The requirements for information items in this document are contained in Clause 7. The requirements for the content of information items in this document are contained in Clause 9 and Annex A.

NOTE 1 If a user of this document claims full conformance to ISO/IEC/IEEE 15289, it does not imply that the user can claim conformance to the information items and information item content in this document. The reason is that this document adds additional information items.

NOTE 2 In this document, for simplicity of reference, each information item is described as if it were published as a separate document. However, information items can be considered as conforming if they are unpublished but available in a repository for reference, divided into separate documents or volumes, or combined with other information items into one document.

4.5 Tailored conformance

4.5.1 Processes

This document does not make provision for tailoring life cycle processes. ISO/IEC/IEEE 15288:2015, Annex A provides normative direction regarding the tailoring of system life cycle process. ISO/IEC/IEEE 12207:2017, Annex A provides normative direction regarding the tailoring of software life cycle processes.

4.5.2 Information items

When this document is used as a basis for establishing a set of information items that do not qualify for full conformance, the clauses of this document shall be selected or modified in accordance with the tailoring process prescribed in Annex C. The tailored text, for which tailored conformance is claimed, is declared. Tailored conformance is achieved by demonstrating that requirements for the information items, as tailored, have been satisfied using the outcomes of the tailoring process as evidence.

5 Concepts

5.1 General

This clause presents concepts that apply to requirements statements themselves, and to the information items generated during the process of documenting requirements. The concepts apply to the properties of requirements at all levels of the system-of-interest. The concepts also apply to the processes used in the elicitation, analysis, allocation, documentation and management of requirements.

5.2 Requirements fundamentals

5.2.1 General

Requirements engineering is an interdisciplinary function that mediates between the domains of the acquirer and supplier or developer to establish and maintain the requirements to be met by the system, software or service of interest. Requirements engineering is concerned with discovering, eliciting, developing, analyzing, verifying (including verification methods and strategy), validating, communicating, documenting and managing requirements. The primary result of requirements engineering is sets of requirements, each set:

— being with reference to a defined system, software or service;

— enabling an agreed understanding between stakeholders (e.g., acquirers, users, customers, operators, suppliers);

— having been validated against real-world needs;

— able to be implemented; and

— providing a reference for verifying designs and solutions.

The collection of requirements may be represented in one or more requirements specifications (see Clauses 8 and 9 for specification templates and content).

5.2.2 Stakeholders

Stakeholders vary across projects when considered in the context of requirements engineering. A minimum set of stakeholders consists of users and acquirers (who may not be the same). Complex projects can impact many users and many acquirers, each with different concerns. It may be necessary to include two other groups as part of the minimum set of stakeholders. First, the organization or organizations developing, maintaining or operating the system or software has a legitimate interest in benefiting from the system. Second, regulatory authorities can have statutory, industry, or other external requirements demanding careful analysis.

5.2.3 Transformation of needs into requirements

Defining requirements begins with stakeholder needs (or goals, or objectives) that are refined and evolve before arriving as valid stakeholder requirements. Initial stakeholder concerns do not serve as stakeholder requirements, since they often lack definition, analysis and possibly consistency and feasibility. Using the Concept of Operations to aid the understanding of the stakeholder concerns at the organizational level and the System Operational Concept from the system perspective, requirements engineering leads stakeholders from those initial, often latent needs to a set of objectively adequate, structured and more formal statements of stakeholder requirements and goals. These statements are well-formed and meet the characteristics of 5.2.4, 5.2.5, and 5.2.6.

When the system of interest is a system element at the second or lower level in the overall system structure, it is also necessary to identify stakeholder needs and requirements at that level. These needs do not just apply to the topmost system in the system structure. There could be additional stakeholders that are not discovered until lower level architecture or design decisions are made.

The stakeholder-owned system requirements are then transformed into lower level system element requirements for the system-of-interest. Where risk due to technology or complexity is significant, requirements are genuinely changing because of change of need, or requirements were never properly defined in the first place, this transformation may involve significant iteration. The same processes are also applied recursively to lower physical level system elements that themselves are subject to design and development, generating again lower-level system element requirements in the process.

Clause 6 details the processes to perform stakeholder and system requirements definition. Clauses 7, 8, and 9 contain further guidance on information items associated with documenting the requirements. Annex A provides requirements for the content of a System Operational Concept and Annex B provides guidelines for the content of a Concept of Operations.

NOTE See ISO/IEC 26551 for additional guidance on requirements development techniques, including requirements reuse.

5.2.4 Requirements construct

Well-formed stakeholder requirements, system requirements and system element requirements shall be developed. This practice contributes to requirements validation with the stakeholders and helps ensure that the requirements accurately capture stakeholder needs.

A well-formed specified requirement contains one or more of the following:

— it shall be met or possessed by a system to solve a problem, achieve an objective or address a stakeholder concern;

— it is qualified by measurable conditions;

— it is bounded by constraints;

— it defines the performance of the system when used by a specific stakeholder or the corresponding capability of the system, but not a capability of the user, operator or other stakeholder; and

— it can be verified (e.g., the realization of the requirement in the system can be demonstrated).

NOTE 1 An additional consideration is that a requirement at lower levels aligns with the design of the higher physical-level system.

This description provides a means for distinguishing between requirements and the attributes of those requirements (conditions, assumptions and constraints).

A requirement is a statement that translates or expresses a need and its associated constraints and conditions. A requirement can be written in the form of a natural language or some other form of language. If expressed in the form of a natural language, the statement should include a subject and a verb, together with other elements necessary to adequately express the information content of the requirement. A requirement shall state the subject of the requirement (e.g., the system, the software, etc.), what shall be done (e.g., operate at a power level, provide a field for) or a constraint on the system. Figure 1 shows example syntax for requirements. Condition-action tables and use cases are other means of capturing requirements.

It is important to agree in advance on the specific keywords and terms that signal the presence of a requirement. A common approach is to stipulate the following.

— Requirements are mandatory binding provisions and use 'shall'.

— Non-requirements, such as descriptive text, use verbs such as ‘are’, ‘is’, and ‘was’. It is best to avoid using the term ‘must’, due to potential misinterpretation as a requirement.

— Statements of fact, futurity, or a declaration of purpose are non-mandatory, non-binding provisions and use 'will'. 'Will' can also be used to establish context or limitations of use.

— Preferences or goals are desired, non-mandatory, non-binding provisions and use 'should'. They are not requirements.

— Suggestions or allowances are non-mandatory, non-binding provisions and use 'may'.

— Use positive statements and avoid negative requirements such as ‘shall not’.

— Use active voice: avoid using passive voice, such as ’it is required that’.

— Avoid using terms such as ‘shall be able to’.

NOTE 2 Requirements in agile may use alternative formulations such as user stories without explicitly using the term ‘shall’. See ISO/IEC/IEEE 12207:2017, Annex H for additional discussion on agile application.

All terms specific to requirements engineering should be formally defined and applied consistently throughout all requirements of the system.

[Condition] [Subject] [Action] [Object] [Constraint of Action]

EXAMPLE: When signal x is received **[Condition]**, the system **[Subject]** shall set **[Action]** the signal x received bit **[Object]** within 2 seconds **[Constraint of Action].**

Or

[Condition] [Subject] [Action] [Object] [Constraint of Action]

EXAMPLE: At sea state 1 **[Condition]**, the Radar System **[Subject]** shall detect **[Action]** targets **[Object]** at ranges out to 100 nautical miles **[Constraint of Action]**.

Or

[Subject] [Action] [Constraint of Action]

EXAMPLE: The Invoice System **[Subject]** shall display pending customer invoices **[Action]** in ascending order of invoice due date **[Constraint of Action].**

Figure 1 — Examples of functional requirements syntax

Conditions are measurable qualitative or quantitative attributes that are stipulated for a requirement. They further qualify a requirement that is needed, and provide attributes that permit a requirement to be formulated and stated in a manner that can be validated and verified. Conditions may limit the options open to a designer. It is important to transform the stakeholder needs into stakeholder requirements without imposing unnecessary bounds on the solution space.

Constraints restrict the design solution or implementation of the systems engineering process. Constraints may apply across all requirements, may be specified in a relationship to a specific requirement or set of requirements, or may be identified as stand-alone requirements (i.e., not bounding any specific requirement).

Examples of constraints giving rise to requirements include:

— interfaces to already existing systems (e.g., format, protocol or content) where the interface cannot be changed;

— physical size limitations (e.g., a controller shall fit within a limited space in an airplane wing);

— laws of a particular country;

— available duration or budget;

— pre-existing technology platform;

— maintenance constraints; or

— user or operator capabilities and limitations.

Requirements may be ranked or weighted to indicate priority, timing or relative importance. Requirements in scenario form depict the system's action from a user's perspective.

NOTE 3 6.3.3 and 6.4.3 detail the process to define stakeholder and system requirements.

5.2.5 Characteristics of individual requirements

Each stakeholder, system and system element requirement shall possess the following characteristics.

— Necessary. The requirement defines an essential capability, characteristic, constraint and/or quality factor. If it is not included in the set of requirements, a deficiency in capability or characteristic will exist, which cannot be fulfilled by implementing other requirements. The requirement is currently applicable and has not been made obsolete by the passage of time. Requirements with planned expiration dates or applicability dates are clearly identified.

— Appropriate. The specific intent and amount of detail of the requirement is appropriate to the level of the entity to which it refers (level of abstraction appropriate to the level of entity). This includes avoiding unnecessary constraints on the architecture or design while allowing implementation independence to the extent possible.

  NOTE 1 While additional detailed information could still be important, the information is documented and communicated in some other form of documentation, such as the requirements attributes in 5.2.8 (e.g., rationale) in order to aid in design and implementation. Additionally, including design solutions in the requirements creates the risk that potential design solutions could be overlooked or eliminated. Examples include stating requirements that express an exact commercial system set or a system that can be bought rather than made, stating tolerances for items deep within the conceptual system, or establishing constraints that are not necessarily reflective of the parent requirement.

— Unambiguous. The requirement is stated in such a way so that it can be interpreted in only one way. The requirement is stated simply and is easy to understand.

— Complete. The requirement sufficiently describes the necessary capability, characteristic, constraint or quality factor to meet the entity need without needing other information to understand the requirement.

— Singular. The requirement states a single capability, characteristic, constraint or quality factor.

  NOTE 2 Although a single requirement consists of a single function, quality or constraint, it can have multiple conditions under which the requirement is to be met.

— Feasible. The requirement can be realized within system constraints (e.g., cost, schedule, technical) with acceptable risk.

— Verifiable. The requirement is structured and worded such that its realization can be proven (verified) to the customer's satisfaction at the level the requirements exists. Verifiability is enhanced when the requirement is measurable.

— Correct. The requirement is an accurate representation of the entity need from which it was transformed.

— Conforming. The individual items conform to an approved standard template and style for writing requirements, when applicable.

5.2.6 Characteristics of a set of requirements

There are certain characteristics that are considered for the set of stakeholder, system and system element requirements rather than for any individual requirement. These sets of requirements provide for a consistent solution that meets the stakeholder intentions and constraints. Each set of requirements for a system, software or service shall possess the following characteristics.

— Complete. The set of requirements stands alone such that it sufficiently describes the necessary capabilities, characteristics, constraints or quality factors to meet entity needs without needing further information. In addition, the set does not contain any To Be Defined (TBD), To Be Specified (TBS), or To Be Resolved (TBR) clauses. Resolution of the TBx designations may be iterative and there is an acceptable timeframe for TBx items, determined by risks and dependencies.

NOTE 1 To improve completeness, the following practices can be adopted:

i) include all requirements types relevant to the system under consideration;

ii) account for requirements in all stages of the life cycle; and

iii) involve all stakeholders in the requirements elicitation, capture, and analysis activity.

NOTE 2 It is common to need to include TBx designations during the evolution of the requirements definition, as the process is informed by system analysis results and trade-off decisions. However, the set of requirements cannot be considered complete until all the TBx designated requirements have been resolved.

NOTE 3 Adapted and open source software frequently have existing functions that are not utilized in the system of interest. For integrated systems, systems of systems and systems containing COTS components, the requirements for the solution of interest can still be 'complete'.

— Consistent. The set of requirements contains individual requirements that are unique, do not conflict with or overlap with other requirements in the set, and the units and measurement systems are homogeneous. The terminology used within the set of requirements is consistent, i.e. the same term is used throughout the set to mean the same thing.

— Feasible. The complete set of requirements can be realized within entity constraints (e.g., cost, schedule, technical) with acceptable risk.

NOTE 4 Feasible includes the concept of 'affordable'.

— Comprehensible. The set of requirements is written such that it is clear as to what is expected by the entity and its relation to the system of which it is a part.

— Able to be validated. It is practicable that satisfaction of the requirement set will lead to the achievement of the entity needs within constraints (e.g., cost, schedule, technical, legal and regulatory compliance).

Careful checking of the requirements set for these characteristics is critical to avoiding requirements changes and growth ('requirements creep') during the life cycle that will impact the cost, schedule or quality of the system.

5.2.7 Requirement language criteria

When writing textual requirements, implementing the following considerations will result in well-formed requirements employing the characteristics above.

Requirements should state 'what' is needed, not 'how'. Requirements should state what is needed for the system-of-interest and not include design decisions for it. However, as requirements are allocated and decomposed through the levels of the system, there can be recognition of design decisions/solution architectures defined at a higher level. This is part of the iterative and recursive application of the requirements, architecture and design processes.

Vague and general terms shall be avoided. They result in requirements that are often difficult or even impossible to verify or may allow for multiple interpretations. The following are types of unbounded or ambiguous terms:

— superlatives (such as 'best', 'most');

— subjective language (such as 'user friendly', 'easy to use', 'cost effective');

— vague pronouns (such as 'it', 'this', 'that');

— ambiguous terms such as adverbs and adjectives (such as 'almost always', 'significant', 'minimal') and ambiguous logical statements (such as ‘or’, ‘and/or’);

NOTE 1 Consider multiple requirements when encountering terms such as ‘or’, ‘and’, or ‘and/or’.

— open-ended, non-verifiable terms (such as 'provide support', 'but not limited to', 'as a minimum');

— comparative phrases (such as 'better than', 'higher quality');

— loopholes (such as 'if possible', 'as appropriate', 'as applicable');

— terms that imply totality (such as ‘all’, ‘always’, ‘never’, and ‘every’);

NOTE 2 It is very difficult to verify such requirements.

— incomplete references (not specifying the reference with its date and version number; not specifying just the applicable parts of the reference to restrict verification work).

All assumptions made regarding a requirement shall be documented and validated in one of the requirement's attributes in 5.2.8 (e.g., rationale) associated with a requirement or in an accompanying document. Include definitions as declarative statements, not requirements.

5.2.8 Requirements attributes

5.2.8.1 General

To support requirements analysis, well-formed requirements should include descriptive attributes defined to assist in identifying relevant requirements and to help in understanding and managing the requirements. The attribute information should be associated with the requirements in the selected requirements repository.

5.2.8.2 Examples of requirements attributes

Important examples of requirements attributes include:

— Identification. Each requirement should be uniquely identified (i.e., number, name tag, mnemonic). Identification can reflect linkages and relationships, if needed, or they can be separate from identification. Unique identifiers aid in requirements tracing. Once assigned, the identification is unique - it is never changed (even if the identified requirement changes) nor is it reused (even if the identified requirement is deleted).

— Version Number (and indication of the version of the requirement). This is to make sure that the correct version of the requirement is being implemented as well as to provide an indication of the volatility of the requirement. A requirement that has a lot of change could indicate a problem or risk to the project.

— Owner. The person or element of the organization that maintains the requirement, who has the right to say something about this requirement, approves changes to the requirement, and reports the status of the requirement.

— Stakeholder Priority. The priority of each requirement should be identified. This may be established through a consensus process among potential stakeholders. As appropriate, a scale such as 1-5 or a simple scheme such as High, Medium or Low, could be used for identifying the priority of each requirement. The priority is not intended to imply that some requirements are not necessary, but it may indicate what requirements are candidates for the trade space when decisions regarding alternatives are necessary. Prioritization needs to consider the stakeholders who need the requirements. This facilitates trading off requirements and balancing the impact of changes among stakeholders.

— Risk. A risk value assigned to each requirement based on risk factors. Requirements that are at risk include requirements that fail to have the set of characteristics that well-formed requirements should have. Failing to have these characteristics can result in the requirement not being implemented (fails system verification) and entity needs not being realized (fails system validation). Risk can also address feasibility/attainability in terms of technology, schedule, cost, politics, etc. If the technology needed to meet the requirement is new with a low maturity the risk is higher than if using a mature technology used in other similar projects. Risk may also be inherited from a parent requirement.

NOTE Additional guidance on risk factors can be found in ISO/IEC 16085.

— Rationale. The rationale for establishing each requirement should be captured. The rationale provides the reason that the requirement is needed and points to any supporting analysis, trade study, modelling, simulation or other substantive objective evidence.

— Difficulty. The assumed difficulty for each requirement should be noted (e.g., Easy/Nominal/Difficult). This provides additional context in terms of requirements breadth and affordability. It also helps with cost modelling.

— Type. Requirements vary in intent and in the kinds of properties they represent. Use of a type attribute aids in identifying relevant requirements and categorizing requirements into groups for analysis and allocation.

5.2.8.3 Examples of the requirements type attribute

Important examples of the requirements type attribute include:

— Functional/Performance. Functional requirements describe the system or system element functions or tasks to be performed by the system. Performance is an attribute of function. A performance requirement alone is an incomplete requirement. Performance is normally expressed quantitatively.

NOTE 1 There can be more than one performance requirement associated with a single function, functional requirement or task.

— Interface. Interface requirements are the definition of how the system is required to interact with external systems (external interface), or how system elements within the system, including human elements, interact with each other (internal interface). External interface requirements state characteristics required of the system, software or service at a point or region of connection of the system, software or service to the world outside of the item. They include, as applicable, characteristics such as location, geometry and what the interface is to be able to pass in each direction.

— Process Requirements. These are stakeholder, usually acquirer or user, requirements imposed through the contract or statement of work. Process requirements include: compliance with national, state or local laws, including environmental laws, administrative requirements, acquirer/supplier relationship requirements and specific work directives. Process requirements may also be imposed on a program by corporate policy or practice. System or system element implementation process requirements, such as mandating a particular design method, are usually captured in project agreement documentation such as contracts, statements of work and quality plans.

— Quality (Non-Functional) Requirements. – Include a number of the 'ilities' in requirements to include, for example, transportability, survivability, flexibility, portability, reusability, reliability, maintainability and security. The kinds of quality requirements (e.g., "ilities") should be identified prior to initiating the requirements activities. This should be tailored to the system(s) being developed. As appropriate, measures for the quality requirements should be included as well.

— Usability/Quality-in-Use Requirements (for user performance and satisfaction) – Provide the basis for the design and evaluation of systems to meet the user needs. Usability/Quality-in-Use requirements are developed in conjunction with, and form part of, the overall requirements specification of a system.

NOTE 2 Additional guidance on software quality requirements can be found in the ISO/IEC SQuaRE standards, especially ISO/IEC 25030, and in ISO/IEC 25010.

— Human Factors Requirements. – State required characteristics for the outcomes of interaction with human users (and other stakeholders affected by use) in terms of safety, performance, effectiveness, efficiency, reliability, maintainability, health, well-being and satisfaction. These include characteristics such as measures of usability, including effectiveness, efficiency and satisfaction; human reliability; freedom from adverse health effects.

NOTE 3 Additional guidance on human factors requirements can be found in 6.3.3.4.

5.3 Practical considerations

5.3.1 Application of iteration and recursion

Two forms of process application – iterative and recursive – are essential and useful for applying the processes defined in this document.

5.3.1.1 Iterative application of processes

When the application of the same process or set of processes is repeated on the same level of the system, the application is referred to as iterative. Iteration is not only appropriate but also expected. New information is created by the application of a process or set of processes. Typically this information takes the form of questions with respect to requirements, analyzed risks or opportunities. Such questions should be resolved before completing the activities of a process or set of processes. When re-application of activities or processes can resolve the questions, then it is useful to do so. Iteration can be required to determine that information with admissible quality is used prior to applying the next process or set of activities to a system-of-interest. In this case iteration adds value to the system to which the processes are being used. See Figure 2.

NOTE There can also be internal iteration within a process. This is not shown in Figure 2 for simplicity of the figure.

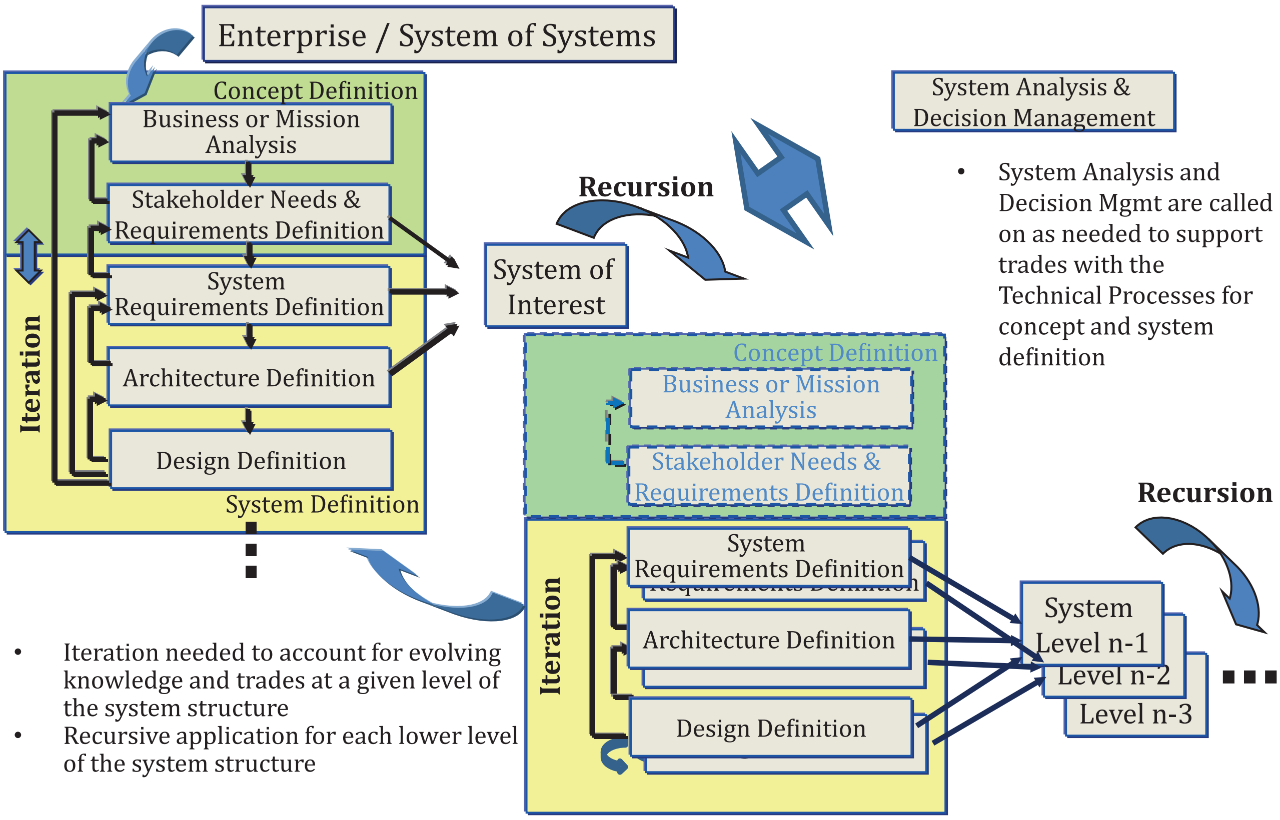

Figure 2 — Iterative and recursive application of processes

5.3.1.2 Recursive application of processes

When the same set of processes or the same set of process activities are applied to successive levels of system elements within the system structure, the application form is referred to as recursive. The outcomes from one application are used as inputs to the next lower (or higher) system in the system structure to arrive at a more detailed or mature set of outcomes. Such an approach adds value to successive systems in the system structure. Figure 2 illustrates the recursive application of processes to systems from the top down. The stakeholder requirements definition process may only be applied at the system-of-interest level. However, the requirements, architecture and design processes may be applied at each successive level of recursion.

NOTE The recursion can also be bi-directional, with requirements from the system requiring further analysis at the system-of-interest level.

5.3.2 Iteration and recursion in requirements engineering

Since different groups of stakeholders often view the system from differing levels of system structure, it is necessary to define and document requirements statements at lower, more detailed levels than just the overall system-of-interest. Allocating or distributing the system requirements to the system elements accomplishes this. The activity of allocating requirements to system elements is part of the Architecture Definition process and proceeds in parallel with the definition of the system architecture. There may be multiple iterations between the requirements processes and other processes in the life cycle (e.g., architecture, design) to resolve trade-offs between the requirements and architecture.

The main forms of appropriate iteration within requirements engineering include:

— purposeful iteration within requirements analysis, between analysis activities;

— planned iteration from downstream activities back to requirements analysis because of a predicted, significant, genuine rate of change of requirements that reflect change of need;

- planned or unplanned iteration from downstream activities back to requirements because of feasibility and balance issues arising from risk due to technology or implementation issues, or risk due to limited knowledge of them;
- unplanned iteration from downstream activities back to requirements because of other solution issues, such as changes to or defects in non-developmental system elements, or obsolescence of system elements;
- reverse engineering of requirements for reasons of regulatory compliance; and
- limited iteration from downstream activities back to requirements analysis because of the reality that requirements can never be perfect, nor is it cost-effective to try to make them so.

Recursion is a major strategy in the successful engineering of systems beyond trivial complexity. Design of the system of interest creates requirements on system elements, the design of which creates requirements on lower physical system elements, and so on. Thus the activities of requirements engineering addressed in this document can be, and usually should be, applied with respect to progressively subordinate solution elements.

One of the main objectives of architecting is to determine how to partition the system; that is, how to identify which requirements should be allocated to which system elements. As system elements are defined, additional requirements statements (called derived requirements) should be created to define relationships between the architectural elements of the system, to provide necessary clarity in the context of the lower levels of abstraction of the system elements, or to specify design constraints or performance levels on system elements. This is accomplished through recursive application of the requirements definition processes.

In addition, some requirements cannot be derived until some portion of the architecture or design evolves. Some requirements depend on how several system elements inter-operate. For example, the information throughput of a system is dependent on the interaction of the system hardware, software, personnel actions and environment. Recursive and iterative application of the requirements, architecture and design processes are used to capture these requirements.

Even where requirements engineering is well resourced, the level of analysis will seldom be uniformly applied. For example, early in the requirements definition processes experienced engineers are often able to identify where existing or off-the-shelf solutions can be adapted to the implementation of system elements. The requirements allocated to these may need less analysis, while others, for which a solution is less obvious, may need to be subjected to further, more detailed analysis. Critical requirements, i.e. requirements having high risk or impacting public safety, environment or health, should always be analyzed more rigorously.

NOTE 6.3.3 details the process to define requirements, including how to apply iteration and recursion to develop requirements fully, particularly with regards to requirements negotiation during the analysis, allocation and trading off between requirements.

5.4 Requirement information items

This subclause describes the relationship between the requirements processes and requirement information items by illustrating a typical application style in a project.

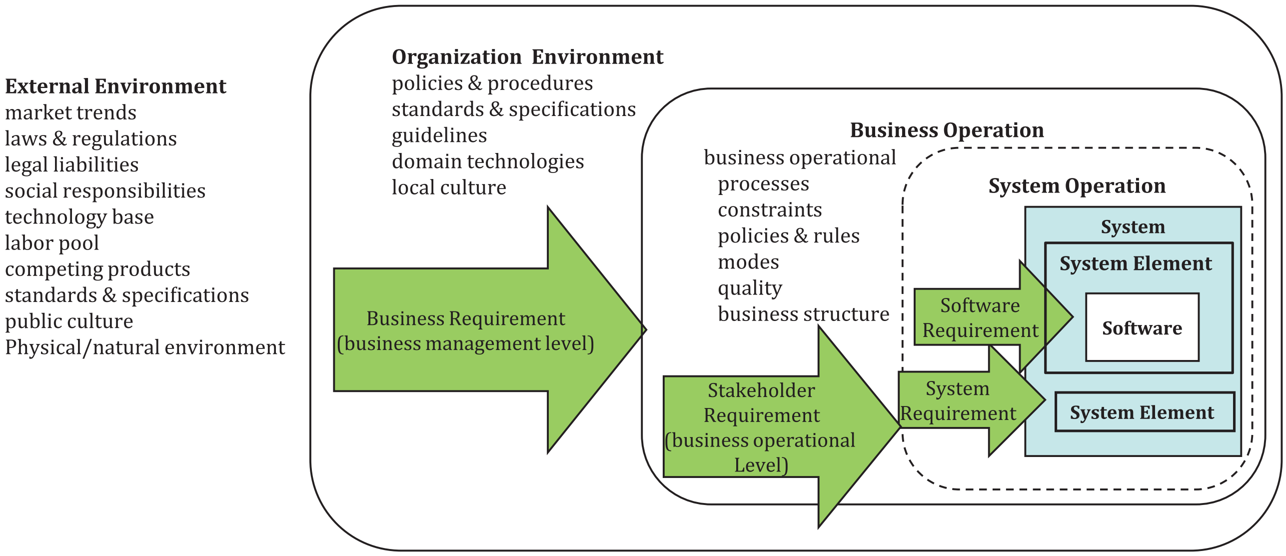

Figure 3 — Example of requirements scope in a business context

Requirements processes and their resultant specifications depend on the scope of the system for which the requirements are to be defined. Requirements for a system or system element to be developed or changed are subject to organization level requirements for the business or organizational operation. The requirements for the system or system element are allocated to lower level systems progressively. A typical view for the scope of a system and the corresponding requirements is illustrated in Figure 3.

NOTE 1 The term business is used even though it could apply to not-for-profit organizations such as in the public sector. Users of this document can replace each occurrence of the term business with the term organization or organizational depending on the users' environment.

The business requirements specification (BRS), the stakeholder requirements specification (StRS), the system requirements specification (SyRS) and the software requirements specification (SRS) are intended to represent different sets of requirement information items. The specifications correspond to the requirements in Figure 3 as follows: BRS – Business Requirement (business management level); StRS - Stakeholder Requirement (business operational level); SyRS - System Requirements; and SRS - Software Requirements. These information items can be applied to multiple specifications (instances) iteratively or recursively. An example of a sequence of requirements processes and specifications is illustrated in Figure 4.

EXAMPLE 1 The SyRS can be used for a system or a system element. The SyRS can also be used to specify software requirements.

EXAMPLE 2 The SRS can be used for lower level software requirements for software specific elements of a system or system element.

NOTE 2 There can be cases where projects are not driven by business requirements, but by mission requirements. An information item such as a Mission Requirements Specification could be considered in such cases.

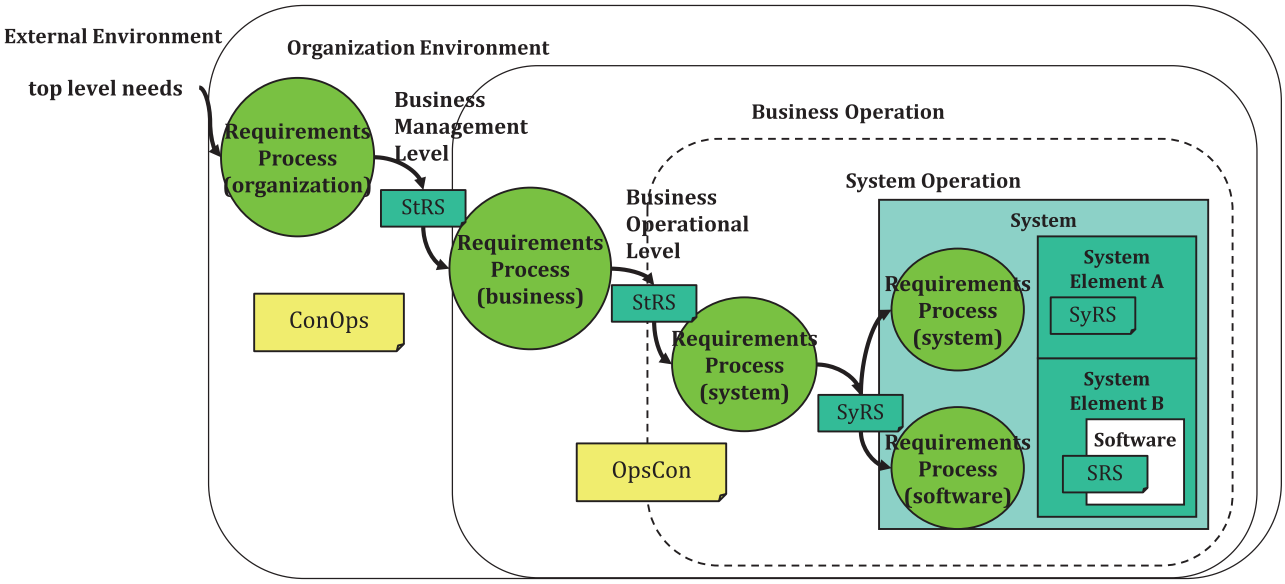

Figure 4 — Example of the relationships between requirements processes and specifications

The Concept of Operation (ConOps) and the System Operational Concept (OpsCon) are useful in eliciting requirements from various stakeholders in an organization and as a practical means to communicate and share the organization's intentions. The ConOps, at the organization level, addresses the leadership's intended way of operating the organization. It may refer to the use of one or more systems as 'black boxes'. The OpsCon addresses the specific system-of-interest from the user's view point.

Information items represented in the BRS, StRS, SyRS, SRS, ConOps, and OpsCon are interdependent. The development of these items requires interaction and cooperation, particularly in relation to the business processes, organizational practice and options for technical solutions.

Different types of systems can have parallel documentation for the various requirements they contain. However, in general, they will start with a BRS, an StRS and an SyRS, and include the software specifications as well as those for hardware and interfaces.

6 Processes

6.1 Requirement processes

The project shall implement the following requirements engineering processes as defined in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207, depending on the adherence to one or both of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207.

— Business or mission analysis process (ISO/IEC/IEEE 15288:2015, 6.4.1 or ISO/IEC/IEEE 12207:2017, 6.4.1).

— Stakeholder needs and requirements definition process (ISO/IEC/IEEE 15288:2015, 6.4.2 or ISO/IEC/IEEE 12207:2017, 6.4.2)

— System requirements definition process (ISO/IEC/IEEE 15288:2015, 6.4.3) or System/software requirements definition process (ISO/IEC/IEEE 12207:2017, 6.4.3)

6.1.1 Guidelines for processes

In this document, the requirements related processes are elaborated upon, in order to provide the user with additional planning and implementation guidance.

Beginning with 6.2, original ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 tasks relevant to this document are highlighted in a box, to show the reader the original text being elaborated. Tasks that are not relevant were omitted, but the original numbering from ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 was retained. The original source references are included in the lower right hand corner. The original ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 purposes and outcomes relevant to this document are used in their entirety, without any change, for the subset of processes that are relevant to requirements engineering.

In many cases, ISO/IEC/IEEE 15288 refers to systems while ISO/IEC/IEEE 12207 refers to software systems or systems/software. To avoid excessive repetition, this variation is shown in the boxed text by bracketed text for the wording in ISO/IEC/IEEE 12207, e.g., system [software system]. Where the variation is more significant, the entire task text is included for both source documents.

The principal processes are:

1) Business or Mission Analysis process;

2) Stakeholder Needs and Requirements Definition process; and

3) System [System/Software] Requirements Definition process.

These three processes result in a baseline set of requirements that flow into the Architecture and Design processes where the requirements are allocated, decomposed and traced to system elements. The Architecture and Design processes also include allocation of requirements that initiate the recursive and iterative application of the requirements processes. This is applied based on the project's system life cycle model definition as described in ISO/IEC/IEEE 24748-1. The Architecture and Design processes include allocation and decomposition of requirements that trigger the recursive application of the requirements processes for the definition of system element requirements and the iterative application of the System [System/Software] Requirements Definition process for derived requirements. There are also other technical and project processes that have requirements-related activities or tasks.

NOTE See ISO/IEC/IEEE 42020 for additional information on the architecture process.

6.2 Business or mission analysis process

6.2.1 Purpose

The purpose of the Business or Mission Analysis process is to define the business or mission problem or opportunity, characterize the solution space, and determine potential solution class(es) that could address a problem or take advantage of an opportunity.
[*ISO/IEC/IEEE 15288:2015, 6.4.1.1*]
[*ISO/IEC/IEEE 12207:2017, 6.4.1.1*]

6.2.2 Outcomes

As a result of the successful implementation of the Business or Mission Analysis process: a) The problem or opportunity space is defined. b) The solution space is characterized. c) Preliminary operational concepts and other concepts in the life cycle stages are defined. d) Candidate alternative solution classes are identified and analyzed. e) The preferred candidate alternative solution class(es) are selected. f) Any enabling systems or services needed for business or mission analysis are available. g) Traceability of business or mission problems and opportunities and the preferred alternative solution classes is established. [*ISO/IEC/IEEE 15288:2015*, *6.4.1.2*] [*ISO/IEC/IEEE 12207:2017*, *6.4.1.2*]

6.2.3 Activities and tasks

6.2.3.1 General

The project shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Business or Mission Analysis process.

6.2.3.2 Prepare for Business or Mission Analysis

This activity consists of the following tasks.

Review identified problems and opportunities in the organization strategy with respect to desired organization goals or objectives. [*ISO/IEC/IEEE 15288:2015*, *6.4.1.3* a) 1)] [*ISO/IEC/IEEE 12207:2017*, *6.4.1.3* a) 1)]

The organization strategy generally includes the intended direction and the business or mission objectives for the organization, including any problems or opportunities that should be addressed. By reviewing problems and opportunities with respect to the organization's business or mission, the organization can identify deficiencies or gaps in existing capabilities, systems, products or services. Problems and opportunities can also be identified and reviewed from a view of risk to human health, property or environmental preservation.

Define the business or mission analysis strategy. [*ISO/IEC/IEEE 15288:2015*, *6.4.1.3* a) 2)] [*ISO/IEC/IEEE 12207:2017*, *6.4.1.3* a) 2)]

The strategy includes approaches, milestones, resources and specific considerations required to conduct business or mission analysis and to help ensure the business or mission needs are elaborated and formalized into business/mission requirements. This also includes the approach to be employed to identify the problem space and the specific problem or opportunity, to characterize the solution space and the suitable solution classes and to select a solution class.

Identify and plan for the necessary enabling systems or services needed to support business or mission analysis.
[*ISO/IEC/IEEE 15288:2015*, *6.4.1.3* a) 3)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.1.3* a) 3)]

Enabling systems or services facilitate the life cycle activities of the system. In the case of business or mission analysis, they include business systems and repositories of the organization, business development and market analysis resources, and other systems or services that provide insight for the assessment and analysis of the problem space and solution space. This can also include resources or services that are external to the organization, such as business intelligence, market trend analysis or benchmarking repositories or services.

Obtain or acquire access to the enabling systems or services to be used.
[*ISO/IEC/IEEE 15288:2015*, *6.4.1.3* a) 4)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.1.3* a) 4)]

This can include scheduled or specified access to business data systems or other resources that are held internal or external to the organization.

6.2.3.3 Define the problem or opportunity space

This activity consists of the following tasks.

Analyze the problems and opportunities in the context of relevant trade-space factors.
[*ISO/IEC/IEEE 15288:2015, 6.4.1.3 b) 1)*]
Analyze customer complaints, problems and opportunities in the context of relevant trade-space factors.
[*ISO/IEC/IEEE 12207:2017, 6.4.1.3 b) 1)*]

In preparation for the identification and evaluation of solution classes, this task is associated with the understanding of the scope, basis or drivers of the identified problems or opportunities. The focus is on changes in mission requirements, business opportunities, capabilities, improvements in some quality or performance aspect, or some increase in efficiency. The trade space factors are the most critical criteria for determining the viability of an alternative. These factors define the problem in terms of what is important to the mission or business opportunity. The criteria may be technical parameters, such as measures of effectiveness, or business parameters, such as market share.

NOTE Improvements in quality include safety, security, accessibility and usability. Improvements in performance include reliability, user satisfaction and service level. Increases in efficiency include user efficiency, time efficiency, resource utilization or capacity.

Define the mission, business, or operational problem or opportunity.
[*ISO/IEC/IEEE 15288:2015*, *6.4.1.3* b) 2)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.1.3* b) 2)]

Every analysis task should begin with a concise statement of the opportunity or problem. While business or mission management may find it difficult to define the problem or opportunity concisely, the subsequent analysis is invariably likely to fail if the owners of the problem or opportunity cannot articulate it succinctly at the outset. In some organizations, the problem or opportunity may be captured in a concise statement, sometimes called a mission statement or strategic business objective.

6.2.3.4 Characterize the solution space

This activity consists of the following tasks.

Define preliminary operational concepts and other concepts in life cycle stages.
[*ISO/IEC/IEEE 15288:2015*, *6.4.1.3* c) 1)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.1.3* c) 1)]

Business or mission management prepare a number of preliminary life cycle concepts which are elaborated and refined in much more detail by stakeholders at the business operations level in the subsequent Stakeholder Needs and Requirements Definition process. Typical life-cycle concepts include:

a) The OpsCon outlines operational aspects of the system solution (new or evolved) in the context of the intended operation of the organization. It provides the lower-level operations-oriented concepts that address a part of the organization's ConOps.

b) The Acquisition Concept describes the way the system solution will be acquired including aspects such as stakeholder engagement, source of the solution, requirements definition, solicitation and contracting issues, design, production and verification.

c) The Deployment Concept describes the way the system solution will be validated, delivered and introduced into operations.

d) The Support Concept describes the desired support infrastructure and manpower considerations for supporting the system solution after it is deployed. A support concept addresses operating support, engineering support, maintenance support, supply support and training support.

e) The Retirement Concept describes the way the system will be removed from operation and retired, including the disposal of any hazardous materials used in or resulting from the process.

As these are typical examples of life cycle concepts, others may also be created to address a specific focus of the life cycle. The set of life cycle concepts, especially the OpsCon, are important in establishing the correct requirements to address the problem or opportunity.

NOTE The eventual system solution can be a new system, an evolution of an existing system or set of systems, an operational change to an existing system or set of systems, or some combination of these.

Identify candidate alternative solution classes that span the potential solution space.
[*ISO/IEC/IEEE 15288:2015*, *6.4.1.3* c) 2)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.1.3* c) 2)]

This activity identifies and describes the classes of solutions that can address the problem or opportunity. At the early stage of requirements engineering, business or mission management needs to be careful to describe their needs and requirements in logical terms — that is, in terms of the problem domain (the business and operational environment). For any problem domain, however, there is normally a range of potential solution classes within the solution domain. Each alternative solution class will represent a group of solutions that may represent a completely different type design and even a different type of project. A feasibility analysis is therefore an essential step to narrowing the solution domain so that the subsequent project can be managed effectively and efficiently.

6.2.3.5 Evaluate alternative solution classes

This activity consists of the following tasks:

Assess each alternative solution class. [*ISO/IEC/IEEE 15288:2015, 6.4.1.3 d) 1)*] [*ISO/IEC/IEEE 12207:2017, 6.4.1.3 d) 1)*]

Feasible alternatives are considered in terms of available resources such as money, time, personnel and materials. The trade space factors play a major role in the set of assessment parameters and decision criteria. The trade space factors are the most critical criteria for determining the viability of an alternative solution class. These factors define the problem in terms of what is important to the mission or business opportunity. The criteria may be technical parameters, such as measures of effectiveness, or business parameters, such as market share. For the establishment and conduct of the trade study and assessment of the alternatives, refer to ISO/IEC/IEEE 15288:2015, 6.3.4 and 6.4.6, or ISO/IEC/IEEE 12207:2017, 6.3.4 and 6.4.6.

Select the preferred alternative solution class(es). [*ISO/IEC/IEEE 15288:2015*, *6.4.1.3* d) 2)] [*ISO/IEC/IEEE 12207:2017*, *6.4.1.3* d) 2)]

For information on the execution of the decision analysis (trade study) and making the decision among the alternatives, refer to ISO/IEC/IEEE 15288:2015, 6.3.4 or ISO/IEC/IEEE 12207:2017, 6.3.4, Decision Management.

6.2.3.6 Manage the Business or Mission Analysis

This activity consists of the following tasks.

Maintain traceability of business or mission analysis. [*ISO/IEC/IEEE 15288:2015*, *6.4.1.3* e) 1)] [*ISO/IEC/IEEE 12207:2017*, *6.4.1.3* e) 1)]

Requirements traceability should be established and maintained to document how the business needs and requirements are intended to meet the business problems and opportunities and how they are related to preferred alternative solution classes and stakeholder needs and requirements. Business and mission needs and requirements need to be captured, traced, and maintained throughout the system life cycle and beyond. Use of a requirements management tool can facilitate this process. More discussion on the application of traceability can be found in 6.4.3.5 of this document under task 2.

Provide key [artifacts and] information items that have been selected for baselines. [*ISO/IEC/IEEE 15288:2015*, *6.4.1.3* e) 2)] [*ISO/IEC/IEEE 12207:2017*, *6.4.1.3* e) 2)]

Key information items and artifacts would include the preliminary life-cycle concepts, including: the OpsCon, the acquisition concept, the deployment concept, the support concept and the retirement concept. Additionally, the trade study reports and supporting analysis may also be considered to be key information items.

6.3 Stakeholder needs and requirements definition process

6.3.1 Purpose

The purpose of the Stakeholder Needs and Requirements Definition process is to define the stakeholder requirements for a system that can provide the capabilities needed by users and other stakeholders in a defined environment.

It identifies stakeholders, or stakeholder classes, involved with the system throughout its life cycle, and their needs. It analyzes and transforms these needs into a common set of stakeholder requirements that express the intended interaction the system will have with its operational environment and that are the reference against which each resulting operational capability is validated. The stakeholder requirements are defined considering the context of the system of interest with the interoperating systems and enabling systems.

[*ISO/IEC/IEEE 15288:2015*, *6.4.2.1*]

[*ISO/IEC/IEEE 12207:2017*, *6.4.2.1*]

6.3.2 Outcomes

As a result of the successful implementation of the Stakeholder Needs and Requirements Definition process:

a) Stakeholders of the system are identified.

b) Required characteristics and context of use of capabilities and concepts in the life cycle stages, including operational concepts, are defined.

c) Constraints on a system are identified.

d) Stakeholder needs are defined.

e) Stakeholder needs are prioritized and transformed into clearly defined stakeholder requirements.

f) Critical performance measures are defined.

g) Stakeholder agreement that their needs and expectations are reflected adequately in the requirements is achieved.

h) Any enabling systems or services needed for stakeholder needs and requirements are available.

i) Traceability of stakeholder requirements to stakeholders and their needs is established.

[*ISO/IEC/IEEE 15288:2015*, *6.4.2.2*]

[*ISO/IEC/IEEE 12207:2017*, *6.4.2.2*]

6.3.3 Activities and tasks

6.3.3.1 General

The project shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Stakeholder Needs and Requirements Definition process.

6.3.3.2 Prepare for Stakeholder Needs and Requirements Definition

This activity consists of the following tasks.

Identify the stakeholders who have an interest in the [software] system throughout its life cycle. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* a) 1)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* a) 1)]

It is best to identify all stages of the system life cycle, and then identify the individual stakeholders or stakeholder classes who have a legitimate interest in the system throughout its life cycle. Requirements elicited from a stakeholder can be dependent on the role, responsibility and position of the stakeholder in the organization. Identify all of the stakeholder classes that have a role or interest in the desired product or service. Then identify those stakeholders who have strong influence on goals, strategies, operations and the target system. The list of stakeholder classes is often modified with time as more is learned about the desired product or service. Representatives from each stakeholder class should be identified and include multi-level perspectives. Information gathered from only one stakeholder class, or only one level, is likely to be biased from a single perspective. A representative cross-section of stakeholders is necessary to provide the true picture of the problem or opportunity to be addressed.

Define the stakeholder needs and requirements definition strategy. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* a) 2)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* a) 2)]

The strategy includes approaches, milestones, resources and specific considerations required to elicit and capture the stakeholder needs and transform them into stakeholder requirements. This includes how to deal with opposing interests. In the case of opposing stakeholder needs, it is useful to start by looking for the areas where there is consensus on a subset of the needs and then build out from there to establish a common set of stakeholder requirements. In some cases, the diverse or opposing stakeholder needs may result in the realization that the solution may need to accommodate variants of requirements, which can be variants in operational, logical or physical aspects of the solution. The strategy addresses how to deal with the diverse and possibly opposing needs.

Identify and plan for the necessary enabling systems or services needed to support stakeholder needs and requirements definition. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* a) 3)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* a) 3)]

Enabling systems or services facilitate the life cycle activities of the system. In the case of stakeholder needs and requirements definition, they include tools and repositories that can be used to elicit needs and requirements from stakeholders and to capture, manage and transform this information. This can also include tools and repositories that maintain information about systems that provide inputs or receive outputs at the interfaces or that provide necessary capabilities for the needs, such as the baseline information for the organizations portfolio of systems.

Obtain or acquire access to the enabling systems or services to be used. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* a) 4)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* a) 4)]

This can include scheduled or specified access to business data systems or other resources that are held internal or external to the organization.

6.3.3.3 Define stakeholder needs

This activity consists of the following tasks.

Define context of use within the concept of operations and the preliminary life cycle concepts.
[*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* b) 1)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* b) 1)]

The ConOps describes an organization's assumptions or intent in regard to an operation or series of operations. It is often captured in long-range strategic plans and annual operational plans and can cover a series of connected operations that are carried out simultaneously or in succession. The concept gives an overall picture of the organization operations. It provides the basis for bounding the operating space, system capabilities, interfaces and operating environment. Sometimes, the context of use is captured using a Context of Use Description (see ISO/IEC 25063). Preliminary life cycle concepts are developed by the Business or Mission Analysis process.

Identify stakeholder needs.
[*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* b) 2)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* b) 2)]

Identification of stakeholder needs includes elicitation of needs directly from the stakeholder(s), identification of implicit stakeholder needs based on domain knowledge and context understanding, and documented gaps from previous activities. Needs often include measures of effectiveness. Functional analysis is often used to aid in the elicitation of needs. Also, quality characteristics of the quality model in ISO/IEC 25010 and quality model application to requirements definition in ISO/IEC 25030 are useful to elicit and identify quality requirements of non-functional requirements, which are often implicit stakeholder needs.

Prioritize and down-select needs.
[*ISO/IEC/IEEE 15288:2015, 6.4.2.3 b) 3)*]
[*ISO/IEC/IEEE 12207:2017, 6.4.2.3 b) 3)*]

Use the Decision Management process (ISO/IEC/IEEE 15288:2015, 6.3.4 or ISO/IEC/IEEE 12207:2017, 6.3.4) to assist in setting up the assessment and selection of the needs. Other processes may be used to provide insight for the assessment and selection, such as the Measurement process (ISO/IEC/IEEE 15288:2015, 6.3.7 or ISO/IEC/IEEE 12207:2017, 6.3.7) to provide quantitative insight, the System Analysis process (ISO/IEC/IEEE 15288:2015, 6.4.6 or ISO/IEC/IEEE 12207:2017, 6.4.6) to provide analysis results of specific parameters, or the Risk Management process (ISO/IEC/IEEE 15288:2015, 6.3.5 or ISO/IEC/IEEE 12207:2017, 6.3.5) to provide insight into the technical, cost, schedule or other risks or opportunities associated with the needs.

Define the stakeholder needs and rationale.
[*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* b) 4)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* b) 4)]

In most systems, there can be many sources of needs and ultimately, requirements, and it is essential that all potential sources are identified and evaluated for their impact on the system. Some of the common sources and issues that need to be dealt with are:

— Goals – The term ‘Goal’ (sometimes called ‘business concern’ or ‘critical success factor’) refers to the overall, high-level objectives of the system. Goals provide the motivation for a system but are often vaguely formulated. It is important to assess the value (relative to priority) and cost of goals.

— Mission profile – How will the system perform its mission? How will the system contribute to business or organizational operations?

— Operational scenarios – Are there any special scenarios that need to be accounted for? Scenarios can be used to define operational concepts and to bound the range of anticipated uses of system products, the intended operational environment and interfacing systems, platforms or products. Scenarios help identify requirements that might otherwise be overlooked. Decomposition of a scenario into smaller parts can help reveal activities that might lead to identifying requirements. A "Day-in-the-Life" analysis allows an analyst to walk through a use of the system in a typical usage cycle. This type of analysis might draw out requirements from a perspective other than traditional structured system thinking.

NOTE 1 5.2.3 describes the use of the Concept of Operations and the System Operational Concept as tools to elicit, document and capture the information needed to build requirements.

— Operational environment and context of use – Requirements are derived from the environment in which the system or software product will operate. Will it operate in hot or cold conditions, externally, or other equally restrictive conditions? What are the characteristics, timing and quantity (workload) of interactions with the system environment? Are there any timing constraints in a real-time system or interoperability constraints in a business environment such as constraints in operational hours? Other aspects of the environment (threats and interoperating systems) can also lead to requirements upon the system. These can greatly affect system feasibility and cost, and restrict design choices.

— Operational deployment – When will the system be used? Will it be deployed during the initial, middle or wrap up phases of a need?

— Performance – What are the critical system parameters to accomplish the mission?

— Effectiveness – How effective/efficient should the system be in performing its mission? What are the applicable measures of effectiveness? Does the system have to be available to perform its missions a minimum amount of time, such as 90-percent of the time?

— Operational life cycle – How long will the system's life time be? 20 years? 30 years? How many hours per year should the system operate?

— Organizational environment – Many systems are required to support an organization's process and this may be conditioned by the structure, culture, and internal politics of the organization. In general, new systems should not force unplanned change to the business process.

— User and operator characteristics – Who will be using or operating the system? How will they vary in role, skill level and expected workload? What are the expectations or constraints on their capability and availability? Should allowance be made for accessibility?

NOTE 2 There are very few systems for which there are no significant risks related to use, users, operators, maintainers or some source of human-system issues. See ISO 9241-220:—, E.5 for a description of the types of harm that can result from use of a system.

NOTE 3 See ISO/IEC TR 29138-1 for additional information on accessibility.

NOTE 4 For additional information on user needs see ISO/IEC 25064, and for identifying user and environmental characteristics see ISO 9241-11 and ISO/IEC 25063.

6.3.3.4 Develop the operational concept and other life cycle concepts

This activity consists of the following tasks.

Define a representative set of scenarios to identify all required capabilities that correspond to anticipated operational and other life cycle concepts. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* c) 1)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* c) 1)]

Scenarios can be used to define the concept documents and bound the range of anticipated uses of system products, the intended operational environment and interfacing systems, platforms or products. Scenarios help identify requirements that might otherwise be overlooked. Scenarios may help to establish critical and desired system performance thresholds and objectives for system performance parameters that are critical for system success. They may also establish those that are desired but may be subject to compromise in order to meet the critical parameters. Use case approaches can also be used to define concept documents. Under this approach, a set of actors (systems and classes of people that interact with the system) is identified, along with their goals, purposes and needs for the system. The use cases are analyzed to identify stakeholder requirements.

Different levels of abstraction or presentation mechanisms are often necessary to address the full range of stakeholders, including the acquirer, user and supplier.

A number of preliminary life cycle concepts that are initiated by the Business or Mission Analysis process are elaborated and refined in much more detail by stakeholders in the Stakeholder Needs and Requirements process. These include the OpsCon, Acquisition Concept, Deployment Concept, Support Concept, Retirement Concept and others created to address a specific focus of the life cycle.

Identify the interaction between users and the system.
[*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* c) 2)]
Identify the factors affecting interactions between users and the system.
[*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* c) 2)]

These factors can include the workplace environment, conditions of normal or unusual use and the users' expected skills and knowledge. System usability requirements depend on these factors.

Consideration of human systems integration (HSI) is an important concept within systems engineering. HSI focuses on the human over the system life cycle. It promotes a total system approach that includes humans, technology (hardware and software), the operational context and the necessary interfaces among the system elements to make them work in harmony. HSI brings human-centered disciplines (such as manpower, personnel, training, human factors, environment, health, safety, habitability and survivability) into the systems engineering process to improve the overall system design and performance. Incorporation of HSI considerations into requirements is contingent upon a clear understanding of the missions, functions, operational scenarios and tasks, user population and quality characteristic considerations. Requirements in the areas of user tasks and performance, manpower and training can primarily be defined through decomposition of the goals or missions of the system down to the level of task analyses to define characteristics of the user interface or front end analyses to determine training impacts.

NOTE See ISO TS 18152 for further information about human-systems integration processes.

6.3.3.5 Transform stakeholder needs into stakeholder requirements

This activity consists of the following tasks.

Identify the constraints on a system solution.
[*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* d) 1)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* d) 1)]

Constraints are one type of requirement. They may be imposed by:

— external or organization stakeholders (e.g., engineering plans, technical performance measures, technical maturity, regulations, life cycle costs or user and operator staffing constraints);

— external, interacting or enabling systems;

— activities from other life cycle phases and technical activities such as Transition, Operation and Maintenance;

— measures of effectiveness and suitability that reflect overall acquirer/user satisfaction (e.g., performance, safety, reliability, availability, maintainability and workload requirements).

Examples of constraints include:

1) the budget limit required by top management is a constraint for succeeding requirement processes;

2) the maintenance strategy developed for the system may impose conditions or constraints on requirements (repair times and/or spares levels may drive reliability values), or may define capability requirements directly (e.g., built-in-test functionality to support maintenance fault isolation).

Identify the stakeholder requirements and functions that relate to critical quality characteristics such as assurance, safety, security, environment, or health. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* d) 2)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* d) 2)]

Define stakeholder requirements consistent with life cycle concepts, scenarios, interactions, constraints, and critical quality characteristics. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* d) 3)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* d) 3)]

Critical quality characteristics are aspects of the system that are essential to help determine the integrity of the system and its operating environment.

As part of this task, it is important to identify and assess opportunities to reuse previously existing requirements. This includes identification of existing systems that provide similar functions or capabilities, specified functions or capabilities applicable to the new system-of-interest, and information on the extent of reusability.

NOTE 4 See ISO/IEC 26551 for additional guidance on requirements reuse.

Requirements and needs elicitation is an iterative activity. Consider several different techniques for identifying requirements and needs during elicitation to better accommodate the diverse set of requirements sources, including:

— structured workshops with brainstorming;

— interviews, questionnaires;

— observation of environment or work patterns (e.g., time and motion studies);

— technical documentation review;

— market analysis or competitive system assessment;

— simulations, prototyping, modelling;

— benchmarking processes and systems; and

— organizational analysis techniques (e.g., Strength – Weakness – Opportunity - Threat analysis, product portfolio)

System stakeholders can be authoritative sources for requirements of the system that represent their interests or area(s) of expertise. However, they usually are not familiar with how to transform their expertise into well-formed requirements statements. In addition to these human sources of requirements, important system requirements often are imposed by other systems in the environment that require some services of the system, or act to constrain the system, or even from fundamental characteristics of the application domain. There may also be safety or other constraints that drive system requirements.

A description of the user community (typically found in the organization concept of operations) may provide common understanding across the effort and validate the appropriateness of scenarios. A user description may cover the demographic group(s) to which a product will be marketed or the specific personnel categories that will be assigned to employ the system or otherwise benefit from its operation.

Involving the stakeholders in the verification of the stakeholder requirements (e.g., well-formed requirements) during stakeholder requirements and needs elicitation can also aid early validation by those stakeholders that the statements accurately capture their needs. Apply the characteristics and guidelines for building well-formed requirements statements provided in 5.2.

6.3.3.6 Analyze stakeholder requirements.

This activity consists of the following tasks.

Analyze the complete set of stakeholder requirements. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* e) 1)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* e) 1)]

Requirements should be analyzed for the characteristics defined in 5.2.5 and 5.2.6. Requirements should be prioritized and may be classified as described in 5.2.8. The use of checklists or standard templates helps in the review process.

If stakeholder requirements from existing or legacy systems have been identified as candidates for reuse, they should be analyzed for use based on factors such as applicability, feasibility, availability, quality, cost effectiveness, value and currency. While reusing requirements, a careful consistency check of reused requirements with the system-of-interest's specific requirements should be performed in order to assure consistency.

Define critical performance measures that enable the assessment of technical achievement. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* e) 2)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* e) 2)]

Identify the measures of performance and technical performance measures that are needed to assess achievement of each stakeholder requirement.

Feed back the analyzed requirements to applicable stakeholders to validate that their needs and expectations have been adequately captured and expressed. [*ISO/IEC/IEEE 15288:2015, 6.4.2.3 e) 3)*] [*ISO/IEC/IEEE 12207:2017, 6.4.2.3 e) 3)*]

It is normal for there to be one or more formally scheduled points in the requirements engineering process where the requirements are validated. The objective is to identify any problems before resources are committed to implementing a system solution for the requirements. Requirements validation is concerned with the process of examining the requirements set to help ensure that it defines the right system, i.e. the system that the stakeholder expects. The most common activities in requirements validation are conducting requirements reviews, simulation, and prototyping.

Requirements validation is subject to approval by the project authority and key stakeholders. This activity is invoked to confirm that the requirements properly reflect the stakeholder needs and to establish validation criteria, i.e. that we have the right requirements.

NOTE Additional guidance on validating usability requirements can be found in ISO/IEC TR 25060 and ISO/IEC 25062.

Resolve stakeholder requirements issues. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* e) 4)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* e) 4)]

It is important to continue to perform requirements negotiation during the analysis and allocation of requirements, because conflicts will occur. Negotiation might be needed among stakeholders requiring mutually incompatible features, or due to conflicts between desired performance requirements, constraints, available budget and delivery schedule. In most cases, it is necessary to consult with the stakeholder(s) to reach a consensus on an appropriate trade-off. It is often important for contractual reasons that such decisions are traceable to the stakeholder. Various analysis methods and conflict resolution techniques may be applicable to facilitate the resolution and are dependent on the specific situation.

Some organizations consider requirements negotiation to be part of requirements validation. The specific process subcategory is not important as long as the conflict resolution occurs as early as possible in the requirements analysis task.

6.3.3.7 Manage the stakeholder needs and requirements definition

This activity consists of the following tasks.

Obtain explicit agreement [with designated stakeholders] on the stakeholder requirements. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* f) 1)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* f) 1)]

Conducting requirements reviews is perhaps the most common means of both verification and validation of the requirements specification(s). A group of reviewers is constituted with a brief to look for errors, mistaken assumptions, lack of clarity, verifiability issues and deviation from standard practice. The composition of the group that conducts the review is important (at least one representative of the acquirer should be included for an acquirer-driven project, for example) and it may help to provide guidance on what to look for in the form of checklists.

Reviews may be conducted at any level of the system structure in the set of requirements. Various types of reviews may be applicable throughout the development and maintenance of the requirements, including technical reviews, inspections and walk-throughs. Effective early requirements review and validation can be achieved using low fidelity prototypes to obtain feedback from potential users of the system.

NOTE 1 Additional guidance on reviews can be found in IEEE Std 1028-2008.

NOTE 2 Discussion on prototyping and simulation is contained in 6.3.3.2.

The agreement on stakeholder requirements may be limited to individual stakeholders only agreeing on subsets of the stakeholder requirements. There may be conflicts among the totality of stakeholder requirements. It is the business of stakeholder requirements to record the totality of stakeholder needs even if there are conflicts among these needs. Agreement among stakeholders should happen on the basis of system requirements, not necessarily on the basis of stakeholder requirements.

Maintain traceability of stakeholder needs and requirements. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* f) 2)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* f) 2)]

Requirements traceability should be established and maintained to document how the formal requirements are intended to meet the stakeholder objectives and achieve stakeholder agreement. Stakeholder requirements need to be captured, traced and maintained throughout the system life cycle and beyond, and placed under configuration control. Use of a requirements management tool can facilitate this process. More discussion on the application of traceability can be found in 6.4.3.5 of this document under task 2.

NOTE 3 Additional guidance on placing information under configuration control can be found in ISO/IEC/IEEE 15288:2015, 6.3.5, and in 6.6.2.2 of this document.

NOTE 4 5.2.5 describes requirements traceability as it pertains to requirements engineering.

Consideration should be given to using a requirements management tool, especially for more complex projects. This tool should have the capability to trace linkages between requirements to show relationships. A requirements management tool is intended to facilitate and support the systematic managing of requirements throughout the project life cycle. This includes, but is not limited to, requirements elicitation, requirements analysis, requirements change management, requirements reuse and requirements quality assessment.

NOTE 5 Additional information and guidelines on requirements management tools can be found in ISO/IEC TR 24766.

The requirements repository should first be populated with the source documentation of the stakeholder needs, project constraints (such as from business policies/rules) and any other conditions that provide the basis for the total set of system requirements that will govern its design. Both the source and rationale for each requirement need to be captured.

The requirements repository should also include any requirements attributes, including the priority and criticality of the requirements. Additional information on requirements attributes can be found in 5.2.8.

Provide key [artifacts and] information items that have been selected for baselines. [*ISO/IEC/IEEE 15288:2015*, *6.4.2.3* f) 3)] [*ISO/IEC/IEEE 12207:2017*, *6.4.2.3* f) 3)]

Information items that can be output as part of the Stakeholder Requirements Definition process include:

— Stakeholder Requirements Specification;

— Concept of Operations; and

— System Operational Concept.

A key artifact would be the requirements repository.

Additional information on these requirements-related information items can be found in Clauses 7 to 9 and Annexes A and B.

6.4 System [System/Software] Requirements definition process

6.4.1 Purpose

The purpose of the System [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. [*ISO/IEC/IEEE 15288:2015*, *6.4.3.1*] [*ISO/IEC/IEEE 12207:2017*, *6.4.3.1*]

6.4.2 Outcomes

As a result of the successful implementation of the System [System/Software] Requirements Definition process: a) The system [or element] description, including system interfaces, functions and boundaries, for a system solution is defined. b) System[/software] requirements (functional, performance, process, non-functional, and interface) and design constraints are defined c) Critical performance measures are defined. d) The system[/software] requirements are analyzed. e) Any enabling systems or services needed for system[/software] requirements definition are available. f) Traceability of system[/software] requirements to stakeholder requirements is developed. [*ISO/IEC/IEEE 15288:2015*, *6.4.3.2*] [*ISO/IEC/IEEE 12207:2017*, *6.4.3.2*]

6.4.3 Activities and tasks

6.4.3.1 General

The project shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the System [System/Software] Requirements Definition process.

6.4.3.2 Prepare for System [System/Software] Requirements Definition

This activity consists of the following tasks.

Define the functional boundary of the system [software system or element] in terms of the behaviour and properties to be provided. [*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* a) 1)] [*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* a) 1)]

Scope problems can be minimized by establishing boundary conditions for the system, software system, element or service with the stakeholders before defining the system or software requirements. Three factors that affect the boundary conditions are:

— Organization – stakeholders should have an understanding of the organization in which the targeted system or software will be used and of the real mission or objective of that organization.

— Environment – stakeholders should be aware of the maturity of the domain of the system-of-interest, the certainty of interfaces between the system-of-interest and other systems in the operational environment, and the role of the system-of-interest relative to other systems in the operational environment.

— Constraints – stakeholders should consider the constraints that affect the life cycle of the system-of-interest, such as cost, schedule, political, environmental or operational.

Define the system[/software] requirements definition strategy. [*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* a) 2)] [*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* a) 2)]

The strategy includes approaches, milestones, resources and specific considerations required to identify and define the system or software requirements and manage the requirements through the life cycle.

Identify and plan for the necessary enabling systems or services needed to support system[/software] requirements definition.
[*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* a) 3)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* a) 3)]

Enabling systems or services facilitate the life cycle activities of the system. In the case of system or software requirements definition, they include tools and repositories that can be used to elicit system requirements from stakeholders and to capture, manage and transform this information.

Obtain or acquire access to the enabling systems or services to be used.
[*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* a) 4)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* a) 4)]

This can include scheduled or specified access to business data systems or other resources that are held internal or external to the organization.

6.4.3.3 Define system[/software] requirements

This activity consists of the following tasks.

Define each function that the system [software system or element] is required to perform.
[*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* b) 1)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* b) 1)]

As a better understanding is gained of the interactions and interfaces among the various functions and elements of the system or software, requirements are generated through combinations of performance and effectiveness analyses, trade studies, design development, interface definitions and cost/benefit assessments. The architecture can be used to identify functions, functional interactions and behaviours, functional flow items, etc.

Again, for the system or software, it is important to identify and assess opportunities to reuse previously existing requirements. This includes identification of existing systems that provide similar functions or capabilities, specified functions or capabilities applicable to the new system-of-interest and information on the extent of reusability.

NOTE See ISO/IEC 26551 for additional guidance on requirements reuse.

Identify required states or modes of operation of the software system.
[*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* b) 2)]

The characteristics of software states or modes of operation lead to the software functional requirements.

Define necessary implementation constraints.
[*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* b) 2)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* b) 3)]

It is important to validate constraints with stakeholders and to be certain they are fully understood and correct before evolving a set of system/software requirements and the architecture. In addition to operational scenarios and requirements, implementation constraints may also come from external drivers, such as interfacing systems in the operating environment, enabling systems or regulatory requirements.

Identify system requirements that relate to risks, criticality of the system, or critical quality characteristics.
[*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* b) 3)]
Identify requirements that relate to risks, criticality of the software system, or critical quality characteristics.
[*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* b) 4)]

Technical measures are used to provide insight into the progress of the system or system elements in achieving the technical parameters specified in the requirements. These include Measures of Performance (MOP) and Technical Performance Measures (TPM). An MOP is a measure that characterizes physical or functional attributes relating to the system operation. MOPs are measured under operational environment conditions. A TPM is a measure used to assess design progress, compliance to performance requirements, and technical risks for critical performance parameters. See ISO/IEC/IEEE 24748-2 for more information on these. The quality in use measures are used to determine whether a product meets the needs of specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction in a specified context of use in a realistic system environment.

The architecture can also be used to help identify critical quality characteristics and to identify areas of risk.

Define system requirements and rationale.
[*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* b) 4)]
Define system/software requirements and requirements attributes, including the following: i) Data elements, data structures and formats, and database or data retention requirements; ii) User interfaces and user documentation (information for users) and user training; iii) Interfaces with other systems and services; iv) Functions and non-functional characteristics, including critical quality characteristics and cost targets; v) Transition of operational processes and data from existing automated and manual systems, migration approach and schedule, software installation and acceptance of the product; and vi) Requirement attributes, such as rationale; priority; traceability to software system elements; test cases, and information items; methods of verification; inclusion in approved baselines; and evaluated risk.
[*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* b) 5)]

The architecture can be used as a source for identifying system requirements.

Again, both the source and rationale for each requirement need to be captured. The traceability should be updated and maintained to document how the formal system or software requirements, including derived requirements, are intended to meet the stakeholder requirements and objectives and achieve stakeholder agreement.

The choice of parameters for software specification depends on the software element, architecture and stakeholder needs.

Specifications are collections of requirements. They describe essential technical requirements for products, materials and the criteria for determining whether those requirements are met. Requirements specifications that are important as part of the System/[Software] Requirements Analysis Process may include:

— System Requirements Specification; and

— Software Requirements Specification.

9.5 and 9.6 contain detailed specification content for the system and software requirements specifications.

The benefits of documenting the system requirements and the software requirements include:

— it establishes the basis for agreement between the acquirers or suppliers on what the product is to do (in market driven projects, the user input may be provided by marketing);

— it forces a rigorous assessment of requirements before design can begin and reduces later redesign;

— it provides a realistic basis for estimating product costs, risks and schedules;

— organizations can use the specifications to develop validation and verification plans;

— it provides an informed basis for deploying a product to new users or new operational environments;

— it provides a basis for product enhancement.

6.4.3.4 Analyze system[/software] requirements

This activity consists of the following tasks.

Analyze the complete set of system[/software] requirements. [*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* c) 1)] [*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* c) 1)]

Again, for system/software requirements, it is important to verify that requirements are well formulated. Review all requirements for the characteristics of a good requirement and good set of requirements as described in 5.2.5 and 5.2.6.

The architecture can be used to help analyze the set of requirements to help ensure that all features and functions of the architecture are properly represented in the requirements.

If system/software requirements from existing or legacy systems have been identified as candidates for reuse, they should be analyzed for use based on factors such as applicability, feasibility, availability, quality, cost effectiveness, value and currency. While reusing requirements, a careful consistency check of reused requirements with the system-of-interest's specific requirements should be performed in order to assure consistency.

NOTE 1 See ISO/IEC 26551 for additional guidance on requirements reuse.

The classifications in 5.2.8 can help with this task. The 'prepare for verification' activity of ISO/IEC/IEEE 15288, 6.4.9.3 a) or ISO/IEC/IEEE 12207:2017, 6.4.9.3 a), should be used for the definition, planning, and execution of requirements verification.

Define critical performance measures that enable the assessment of technical achievement. [*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* c) 2)] [*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* c) 2)]

Identify the measures of performance and technical performance measures that are needed to assess achievement of each system requirement. The architecture can be used to help define critical performance measures. Sometimes the architecture will explicitly identify where measures are important for overall system suitability.

In addition to verification of the requirements, the following activity addresses validation of the requirements, individually and as a set, as properly representing the stakeholder needs.

Feed back the analyzed requirements to applicable stakeholders for review.
[*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* c) 3)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* c) 3)]

Requirements validation helps to ensure that stakeholder requirements have been correctly transformed into system requirements. Various techniques can be used, including stakeholder reviews, prototyping, modelling and simulation, conceptual modelling and formal modelling. The appropriate technique can vary based on the characteristics of the stakeholders, so multiple techniques may need to be employed so that all stakeholders are accounted for. Reviews are discussed in 6.3.3.7 of this document under task 1.

Stakeholder reviews are a common technique to validate requirements that can be implemented easily. The stakeholder reviews involve conducting an analysis of the requirements with a group that includes the key stakeholders to determine that the system requirements are complete, correct and consistent reflecting the intent of the stakeholder requirements. Checklists are often developed to aid the reviews to help ensure that all applicable categories of requirements have been considered and documented.

Prototyping is commonly employed for eliciting requirements, validating the interpretation of the system requirements, clarifying or examining requirement attributes and identifying any omitted requirements. The advantage of prototypes is that they provide a richer context for stakeholder evaluation and input, they can make it easier to interpret the assumptions, and they can provide useful feedback on why they are wrong. For example, the dynamic behaviour of a user interface can be better understood through an animated or static prototype than through textual description or graphical models. There are also some disadvantages, however. These include the cost of developing prototypes, potential erroneous assumptions and unwarranted expectations and quality problems with low fidelity prototypes. Effective early requirements review and validation can be achieved using the appropriate level of fidelity for prototypes when the purpose of the prototype is well understood. The level of fidelity and build quality should be based on the purpose of the prototype.

Modelling and simulation can be used to assist the stakeholder validation of the requirements. The advantage of modelling and simulations is that they can demonstrate interactions and allow for sensitivity analysis when the results are not what the stakeholder expected. Models and views from the Architecture Definition process can be used to assist in stakeholder validation of requirements.

NOTE 2 See ISO/IEC/IEEE 42020 for additional information on the role of modelling in developing the architecture and in helping to identify relevant requirements.

Static conceptual modelling is another technique that can be used. The purpose is to aid understanding of the problem rather than to initiate design of the solution. Hence, conceptual models comprise models of entities from the problem domain configured to reflect their real-world relationships and dependencies. There are several kinds of models that can be developed. These include data and control flows, state models, event traces, user interactions, object models, system context models and many others. The factors that influence the choice of model include:

— The nature of the problem: some types of application demand that certain aspects be analyzed particularly rigorously. For example, control flow and state models are likely to be more important for real-time systems than for an information system.

— Expertise: it is often more productive to adopt a modelling notation or method with which the supplier has experience. However, it may be appropriate or necessary to adopt a notation that is better supported by tools, imposed as a process requirement, or simply ‘better’.

— The process requirements of the acquirer: acquirers may impose a particular notation or method. This can conflict with the previous factor.

— The availability of methods and tools: notations or methods that are poorly supported by training and tools might not reach widespread acceptance even if they are better suited to particular types of problem.

Formal modelling that uses notations based upon discrete mathematics and that is traceable to logical reasoning has made an impact in some specialized domains. Formal modelling may be imposed by acquirers or standards or may offer compelling advantages to the analysis of certain critical functions or elements.

Resolve system requirements issues.
[*ISO/IEC/IEEE 15288:2015, 6.4.3.3 c) 4)*]
Identify and resolve issues, deficiencies, conflicts, and weaknesses within the complete set of requirements.
[*ISO/IEC/IEEE 12207:2017, 6.4.3.3 c) 4)*]

It is important to continue to perform requirements negotiation during the analysis and allocation of system requirements, because conflicts will occur. Requirements support a usable design when the requirements set is complete, consistent and accurate.

6.4.3.5 Manage system[/software] requirements

This activity consists of the following tasks.

Obtain explicit agreement on the system[/software] requirements.
[*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* d) 1)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* d) 1)]

Once again, conducting requirements reviews is perhaps the most common means of both verification and validation of the requirements and to facilitate agreement on requirements. Additionally, stakeholders should be prepared to reach and maintain the agreement through negotiation. Typically, the owner of the system/software requirements is responsible for leading the negotiation. Whoever is responsible should have the necessary negotiation skills.

Maintain traceability of the system[/software] requirements.
[*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* d) 2)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* d) 2)]

Requirements tracing is concerned with recovering the source of requirements and predicting the effects of requirements change. The traceability should include interface requirements. Tracing is fundamental to performing coverage analysis (to help ensure that all stakeholder requirements are met in the design and that each low-level requirement is justified); compliance analysis (to document that stakeholder requirements have been satisfied); and impact analysis when requirements change. A requirement should be traceable:

— to lower-level requirements (e.g., stakeholder to system to element and ultimately to hardware and software requirements);

NOTE In application, traceability down is accomplished through requirements allocation as the derived requirements are yet to be developed.

— to architecture (e.g., logical or physical);

— to system elements (e.g., software and hardware elements that implement the requirement);

— to verification/test entities that satisfy it, along with any supporting models and analysis; and

— upwards to the parent requirements from which it was derived or to the stakeholder needs from which it was transformed.

Each requirement should also be traceable upwards to the requirements and stakeholders that motivated it (from a software requirement back to the system requirement(s) that it helps satisfy, for example). In the case of requirements derived from trade or design studies, those derived requirements should be traceable back to the study from which they derive, and the study should be traceable back to the high-level requirements by which it was informed. Bi-directional traceability is a technique that can be used to:

— improve the integrity and accuracy of all requirements, from the system level all the way down to the lowest level system element;

— allow tracking of the requirements development and allocation with related measures such as requirements coverage, compliance and complexity;

— provide a means of documenting and reviewing the relationships between layers of requirements that capture certain aspects of the design; and

— support easier maintenance and change implementation of the system in the future.

Provide key [artifacts and] information items that have been selected for baselines.
[*ISO/IEC/IEEE 15288:2015*, *6.4.3.3* d) 3)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.3.3* d) 3)]

The requirements shall be configuration controlled. The ancillary information recorded along with the requirements can include a summary rationale for each requirement, decisions, assumptions and a change history, along with the requirements categorization information described in 5.2.8. Once again, use of a requirements management tool facilitates a cumbersome and complex project of maintaining requirements traceability and configuration control.

Information items that can be output as part of the System Requirements Definition process include:

— System Requirements Specification; and

— Software Requirements Specification.

Additional information on these requirements-related information items can be found in Clauses 7 to 9.

6.5 Requirements engineering activities in other technical processes

6.5.1 Requirements activities in architecture definition

6.5.1.1 General

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.

NOTE Activities a), b), c), e), and f) are not included as there is no specific guidance related to requirements engineering.

6.5.1.2 Relate the architecture to design

This activity consists of the following tasks.

NOTE Tasks 1), 4), and 5) under this activity are not included, as there is no specific guidance related to requirements engineering.

Define the interfaces and interactions between system elements and with external systems.
[*ISO/IEC/IEEE 15288:2015, 6.4.4.3 d) 2)*]
Define the interfaces and interactions among the software system elements and external entities.
[*ISO/IEC/IEEE 12207:2017, 6.4.4.3 d) 2)*]

Interface requirements (mechanical, electrical, data and envelope) are important types of requirements that need to be thoroughly documented. They can be included in specifications or Interface Control Documents and need to be traceable to both sides of the interface. The interface requirements are incorporated into the architecture definition. Interface documents are shared by the programs involved with the interactions between systems. Models can also be used to control interfaces.

Partition, align and allocate requirements to architectural entities and system elements.
[*ISO/IEC/IEEE 15288:2015*, *6.4.4.3* d) 3)]
[*ISO/IEC/IEEE 12207:2017*, *6.4.4.3* d) 3)]

A candidate architecture is defined in terms of the requirements for the set of system elements from which the system is configured. It is important to establish and maintain the traceability between requirements and the architecture, including the system elements and interfaces. Verification and validation criteria for the system elements should be identified and recorded as derived requirements are generated.

NOTE The Architecture Definition process is expanded on in ISO/IEC/IEEE 42020 where six architecture processes are specified.

6.5.2 Requirements activities in verification

6.5.2.1 General

The purpose of the Verification Process is to provide objective evidence that a system or system element fulfils its specified requirements and characteristics.

NOTE 1 Additional guidance on verification can be found in ISO/IEC/IEEE 15288:2015, 6.4.9 or ISO/IEC/IEEE 12207:2017, 6.4.9.

NOTE 2 Activity b) is not included as there is no specific guidance related to requirements engineering.

6.5.2.2 Prepare for verification

This activity consists of the following task.

NOTE 1 Tasks 1), 2), and 4) through 7) under this activity are not included, as there is no specific guidance related to requirements engineering.

Select appropriate verification methods or techniques and associated criteria for every verification action.
[*ISO/IEC/IEEE 15288:2015, 6.4.9.3 a) 3)*]
[*ISO/IEC/IEEE 12207:2017, 6.4.9.3 a) 4)*]

This activity is facilitated by initially associating a verification method as requirements are created. Verification methods should be documented. Documentation may include requirements verification and traceability matrix or verification statements in a verification plan. A verification method defines how (including success criteria and closure approach), where and when each requirement's compliance can be proven for acquirer acceptance. A verification method is associated with each requirement to define activities that yield objective information to prove satisfaction of the requirement. A good verification method definition addresses some or all of the following content considerations.

— How – identify which verification method to be applied (see list below).

— Who – identify the organization or person with the lead responsibility for performing the verification, such as a contractor, subcontractor, vendor, product team or supplier.

— When – designate a time in the program plan when the verification is to be done. This should be an event-based, and not a calendar date, accomplishment.

— Where – specify any unique venue and environment needed for the verification activity.

There are four standard verification methods to use to obtain the objective evidence that the requirements have been fulfilled: inspection, analysis or simulation, demonstration and test.

Inspection **-** an examination of the item against applicable documentation to confirm compliance with requirements. Inspection is used to verify properties best determined by examination and observation (e.g., - paint colour, weight, etc.). Inspection is generally non-destructive and typically includes the use of sight, hearing, smell, touch and taste; simple physical manipulation; mechanical and electrical gauging; and measurement.

Good practice: Include identification of the document(s) or drawing(s) to use to make the comparison between what is required versus what is being inspected.

Analysis (including modelling and simulation) **-** use of analytical data or simulations under defined conditions to show theoretical compliance. Used where testing to realistic conditions cannot be achieved or is not cost-effective. Analysis (including simulation) may be used when such means establish that the appropriate requirement, specification, or derived requirement is met by the proposed solution. Analysis may also be based on 'similarity' by reviewing a similar item’s prior verification and confirming that its verification status can legitimately be transferred to the present system element. Similarity can only be used if the items are similar in design, manufacture and use; equivalent or more stringent verification specifications were used for the similar system element; and the intended operational environment is identical to or less rigorous than the similar system element.

Good practice: Identify the generic name of the analysis (like Failure Modes and Effects Analysis), analytical or computer tools, or numeric methods; the source of input data; and how raw data is to be analyzed. Review and agree with the acquirer that the analysis methods and tools, including simulations, are acceptable for the provision of objective proof or requirements compliance.

Demonstration **-** a qualitative exhibition of functional performance, usually accomplished with no or minimal instrumentation or test equipment. Demonstration uses a set of test activities with system stimuli selected by the supplier to show that system or system element response to stimuli is suitable or to show that operators can perform their allocated functions when using the system. Observations are made and compared with predetermined responses. Demonstration may be appropriate when requirements or specifications are given in statistical terms (e.g., mean time to repair, average power consumption, etc.).

Good practice: State who the witnesses should be for the purpose of collecting the evidence of success, what general steps are to be followed, and what special resources are needed, such as instrumentation, special test equipment or facilities, simulators, specific data gathering, or rigorous analysis of demonstration results.

Test **-** an action by which the operability, supportability, or performance capability of an item is quantitatively verified when subjected to controlled conditions that are real or simulated. These verifications often use special test equipment or instrumentation to obtain very accurate quantitative data for analysis.

Good practice: State who the witnesses should be for the purpose of collecting the evidence of success. Identify the test facility, test equipment, any unique resource needs and environmental conditions, required qualifications and test personnel, general steps that are to be followed, specific data to be collected, criteria for repeatability of collected data, and methods for analyzing the results.

NOTE 2 Certification is often included as an alternative method. Certification is a written assurance that the system or system element has been developed in accordance with the required standard and meets the requirements. This assures that the system or system element can perform its assigned functions to a negotiated standard. The development reviews and system verification and validation results form the basis for certification. Certification is generally performed by a third party against an accepted standard.

This information is included and documented in a Requirements Traceability Matrix (RTM) or a Verification Cross Reference Matrix (VCRM).

6.5.2.3 Manage results of verification

This activity consists of the following task.

NOTE 1 Tasks 1), 2), 3), and 5) under this activity are not included, as there is no specific guidance related to requirements engineering.

NOTE 2 Anomalies identified can result in requirement changes in any of the technical processes.

Maintain traceability of the verified [software] system elements.
[*ISO/IEC/IEEE 15288:2015, 6.4.9.3 c) 4)*]
[*ISO/IEC/IEEE 12207:2017, 6.4.9.3 c) 4)*]

Requirements Traceability is frequently used as a single point of accountability for tracing a requirement back to the source of the requirement and forward through the life cycle to assess that the requirement has been met. In requirements traceability, verification methods and information are associated with the requirements to indicate how the system or system element is to be verified to show it meets the requirements. As the system moves through the life cycle phases, traceability of the requirements to the work products should be added. It is important to include unique identifiers for each requirement.

6.5.3 Requirements activities in validation

6.5.3.1 General

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.

NOTE 1 Additional guidance on validation can be found in ISO/IEC/IEEE 15288:2015, 6.4.11 or ISO/IEC/IEEE 12207:2017, 6.4.11.

NOTE 2 Activity b) is not included as there is no specific guidance related to requirements engineering.

6.5.3.2 Prepare for validation

This activity consists of the following task.

NOTE Tasks 2) through 7) under this activity are not included, as there is no specific guidance related to requirements engineering.

Identify the validation scope and corresponding validation actions. [*ISO/IEC/IEEE 15288:2015, 6.4.11.3 a) 1)*] Identify the validation scope, including the characteristics of the software system, element, or artifact to be validated, and the expected results of validation. [*ISO/IEC/IEEE 12207:2017, 6.4.11.3 a) 1) i)*]

The system operational concept and baselined stakeholder requirements of the containing system, of which the system of interest is a system element, are part of the validation scope activity.

6.5.3.3 Manage results of validation

This activity consists of the following task.

NOTE Tasks 1), 2), 3), and 5) under this activity are not included as there is no specific guidance related to requirements engineering.

Maintain traceability of the validated system elements. [*ISO/IEC/IEEE 15288:2015, 6.4.11.3 c) 4)*] [*ISO/IEC/IEEE 12207:2017, 6.4.11.3* c)4)]

System validation confirms that the system, as built, satisfies stakeholder stated needs and requirements, that it is the right system. Traceability should be maintained and may be documented in a Requirements Traceability Matrix (RTM) or similar information item.

6.6 Requirements management

6.6.1 Management overview

Requirements management encompasses those tasks that record and maintain the evolving requirements and associated context and historical information from the requirements engineering activities. Requirements management also establishes procedures for defining, controlling and publishing the baseline requirements for all levels of the system-of-interest. Effective requirements management occurs within the context of an organization's project and technical processes as defined in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207.

Requirements are rarely static. Requirements that are likely to evolve should be identified and communicated to both acquirers and the technical community. A core subset of requirements may be frozen early. The impact of proposed new requirements are evaluated to help ensure that the initial intent of the requirements baseline is maintained or that changes to the intent are understood and accepted by the acquirer.

In almost all cases, requirements understanding continues to evolve as life cycle activities proceed. This often leads to the revision of requirements late in the life cycle. Perhaps the most crucial point of understanding about requirements engineering is that a significant proportion of the requirements *will* change. This is sometimes due to errors in the analysis, but it is frequently an inevitable consequence of change in the environment, such as changes in the acquirer’s operating or business environment, or in the market into which the system is sold.

However, care should be exercised in making requirements changes during the life cycle. While some may be unavoidable, excessive uncontrolled changes can result in 'requirements creep' that can result in cost overruns, schedule delays, design errors, buyer dissatisfaction or even cancellation of the project.

6.6.2 Change management

6.6.2.1 General

Whatever the cause of requirements changes, it is important to recognize the inevitability of change and adopt measures to mitigate the effects of change. Change has to be managed by ensuring that proposed changes go through a defined impact assessment, review and approval process, and by applying careful requirements tracing and version management. Hence, the requirements engineering process is not merely a front-end task, but spans the life cycle. In a typical project the activities of the requirements management evolve over time from elicitation to change management.

6.6.2.2 Configuration management

6.6.2.2.1 General

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

NOTE Activities a) and d) through f) are not discussed as there is no specific guidance related to requirements engineering.

6.6.2.2.2 Perform configuration identification

This activity consists of the following task.

NOTE Tasks 2) and 5) under this activity are not included, as there is no specific guidance related to requirements engineering.

Identify the system elements and information items that are configuration items.
[*ISO/IEC/IEEE 15288:2015, 6.3.5.3 b) 1)*]
Select the software system elements to be uniquely identified as configuration items subject to configuration control.
[*ISO/IEC/IEEE 12207:2017, 6.3.5.3 b) 1)*]

Establish system, system element, and information item identifiers.
[*ISO/IEC/IEEE 15288:2015, 6.3.5.3 b) 3)*]
Identify the attributes of configuration items.
[*ISO/IEC/IEEE 12207:2017, 6.3.5.3 b) 3)*]

The system operational concept and stakeholder, system, software and system element requirements are identified as information items for configuration control in the configuration management planning and should have unique identifiers.

Define baselines through the life cycle.
[*ISO/IEC/IEEE 15288:2015, 6.3.5.3 b) 4)*]
[*ISO/IEC/IEEE 12207:2017, 6.3.5.3 b) 4)*]

Commonly used baselines are the functional, allocated, developmental and product baselines. The baselines to be used for a given project, along with their associated levels of authority needed for change approval, are typically identified in the project's configuration management plan. These baselines are described as follows.

— Functional baseline (requirements baseline) establishes a common understanding of what the system is expected to do (i.e., the agreed system requirement specification and associated specifications such as external interface definitions). It defines the capabilities the customer expects to receive from the system. The functional baseline generally provides a basis of agreement between parties.

— Allocated baseline corresponds to the reviewed and versioned system element requirements specifications, including the interface requirements, at the physical level below the system of interest.

— Developmental baseline represents the evolving system and system element configurations at selected times during the life cycle. Change authority for this baseline typically rests primarily with the supplier organization.

— Product baseline corresponds to the detailed specifications and associated detailed design artifacts that represent the completed system.

6.6.2.2.3 Perform configuration change management

This activity consists of the following task.

NOTE 1 Tasks 1) through 3) under this activity are not included, as there is no specific guidance related to requirements engineering.

Track and manage approved changes to the baseline, Requests for Change, and Requests for Variance. [*ISO/IEC/IEEE 15288:2015, 6.3.5.3 c) 4)*] [*ISO/IEC/IEEE 12207:2017, 6.3.5.3 c) 3)*]

As changes are made to the operational concepts and stakeholder, system, software and system element requirements, the changes need to be formally captured and in documented baselines of the requirements along with the configuration information that identifies the specific changes and associated rationale. Requirements traceability should be maintained.

Requirements shall be configuration managed, in accordance with project and organization configuration management processes.

NOTE 2 ISO/IEC/IEEE 15288:2015, 6.3.5 and ISO/IEC/IEEE 12207:2017, 6.3.5 has additional information on configuration management.

6.6.2.3 Information management

6.6.2.3.1 General

The purpose of the Information Management process is to generate, obtain, confirm, transform, retain, retrieve, disseminate and dispose of information, to designated stakeholders.

6.6.2.3.2 Prepare for information management

This activity consists of the following task.

NOTE Tasks 1) and 3) through 5) under this activity are not included, as there is no specific guidance related to requirements engineering.

Define the items of information that will be managed. [*ISO/IEC/IEEE 15288:2015, 6.3.6.3 a) 2)*] [*ISO/IEC/IEEE 12207:2017, 6.3.6.3 a) 2)*]

The system operational concept document and stakeholder requirements specification, system requirements specification, software requirements specification, and other system element requirements specifications are identified as information items to be managed during the system life cycle.

6.6.2.3.3 Perform information management

This activity consists of the following task.

NOTE 1 Tasks 2) through 5) under this activity are not included, as there is no specific guidance related to requirements engineering.

Obtain, develop, or transform the identified items of information. [*ISO/IEC/IEEE 15288:2015*, *6.3.6.3* b) 1)] [*ISO/IEC/IEEE 12207:2017*, *6.3.6.3* b) 1)]

As the operational concept document and various requirements specifications are created reflecting the configuration baselines, the information items are provided to the designated authorities and responsibilities for information management. As the requirements are changed and new baselines are created, the revised information items are provided for information management.

The requirements information shall be managed in accordance with the organization's information management process.

NOTE 2 ISO/IEC/IEEE 15288:2015, 6.3.6 and ISO/IEC/IEEE 12207:2017, 6.3.6 have additional detail on information management.

6.6.3 Measurement for requirements

6.6.3.1 General

The purpose of the Measurement process is to collect, analyze and report objective data and information to support effective management and demonstrate the quality of the products, services and processes.

6.6.3.2 Prepare for measurement

This activity consists of the following tasks.

NOTE 1 Tasks 1), 6), and 7) under this activity are not included, as there is no specific guidance related to requirements engineering.

Describe the characteristics of the organization that are relevant to measurement[, such as business and technical objectives].
Identify and prioritize the information needs.
Select and specify measures that satisfy the information needs.
Define data collection, analysis, access and reporting procedures.
[*ISO/IEC/IEEE 15288:2015*, *6.3.7.3* a) 2) through 5)]
[*ISO/IEC/IEEE 12207:2017*, *6.3.7.3* a) 2) through 5)]

Requirements engineering, as a discipline, benefits from measuring requirements in both the process and the product contexts. More than one measure may be needed to provide the insight into the information needs for the requirements. Practice has consistently proven various useful measures, including:

— Requirements quality – existing requirements quality (e.g., against attributes of 5.2.5) versus expected quality feeds into estimating the requirements analysis effort, can be used to set a standard for release of requirements, can be used as an entry criterion into requirements and detailed design reviews, and can be used in contracting for requirements engineering services.

— Requirements quantity – requirements quantity can be used for measuring rate of progress of requirements engineering activities, estimating requirements completeness, and developing (or reusing) other requirements engineering metrics such as average hours per requirement in analysis or design.

— Requirements volatility - in the process context, requirements volatility can indicate an organization's requirements engineering process will not converge a collection of requirements into a well-formed set. In the product context, a high volatility value can indicate risk early by stakeholders failing to reach consensus on system requirements, putting significant risk on subsequent activities in the life cycle.

Other useful requirements measures include:

— requirements trends;

— requirements change rate and backlog;

— traceability measures (e.g., percentage of parents without children, average number of child requirements per parent – an indicator of design complexity);

— requirements verification;

— requirements validation; and

— TBD and TBR closure progress per plan.

Software requirements are used in software Functional Size Measurement (FSM) methods to assist with many aspects of managing software projects. FSM methods are organized into two parts: uses for project management and uses for forecasting and performance management. If FSM methods are to provide high-fidelity results, it is very important to achieve an accurate and complete allocation and derivation of a system's software requirements from the system requirements.

NOTE 2 ISO/IEC 14143-1 provides details of FSM concepts and their uses.

NOTE 3 ISO/IEC/IEEE 15288:2015, 6.3.7 and ISO/IEC/IEEE 12207:2017, 6.3.7 provide additional information on measurement process, as does ISO/IEC/IEEE 15939.

6.6.3.3 Perform measurement

This activity consists of the following tasks.

Integrate [manual or automated] procedures for data generation, collection, analysis and reporting into the relevant processes.
Collect, store, and verify data.
Analyze data and develop information items.
Record results and inform measurement users.
[*ISO/IEC/IEEE 15288:2015, 6.3.7.3 b) 1) through 4)*]
[*ISO/IEC/IEEE 12207:2017, 6.3.7.3 b) 1) through 4)*]

It is good practice to choose measures for which data are readily available through the life cycle. The data collection can then be integrated into the requirements related processes to obtain the data and insight on a regular basis as the requirements engineering proceeds. It is also good practice to review the analyzed requirements related measures collectively, looking for predictive trends and projections that can aid risk management.

The requirements measurement shall be managed in accordance with the organization's measurement process.

7 Information items

The project shall produce the following information items as part of the requirements engineering processes:

— Business requirement specification (BRS);

— Stakeholder requirements specification (StRS);

— System requirements specification (SyRS); and

— Software requirements specification (SRS), if adhering to ISO/IEC/IEEE 12207.

NOTE 1 The Software Requirements Specification can also be identified as the System/Software Requirements Specification.

The information items shall contain the content as defined in Clause 9 of this document.

NOTE 2 Multiple specification information items for each of the four document types can be produced in the project as discussed in 5.4. For example, the SyRS can be produced for systems and system elements.

NOTE 3 The four specification information items, BRS, StRS, SyRS and SRS can contain similar information items that could be considered as different views for the same product. For ease of use, this document presents typical contents of the four forms of the specification separately.

NOTE 4 The BRS and StRS can have different titles in different domains and can be contained in other information items, as long as the functionality of these specifications can be clearly referenced.

NOTE 5 ISO/IEC/IEEE 15289 provides guidance on identifying and planning the specific information items to be produced during systems and software life cycles.

The management of information items shall be performed by applying the Information Management process of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207.

The information items do not require physical documentation, so long as required content is easily available and logically organized.

EXAMPLE Model-Driven Development (MDD) approaches maintain almost all system information in a modelling tool. In this case, the information is contextually stored in the modelling tool's repository. The required information can be viewed in the model or extracted in report or table formats.

8 Guidelines for information items

8.1 Requirements information item outlines

This clause also provides recommended structure in the form of an outline for the resulting information items.

8.2 Business requirements specification

8.2.1 General

The Business Requirements Specification (BRS) describes the organization's motivation for why the system is being developed or changed, defines processes and policies/rules under which the system is used and documents the top-level requirements from the stakeholder perspective including expressing needs of users/operators/maintainers as derived from the context of use in a specific, precise and unambiguous manner. In a business environment, the BRS describes how the organization is pursuing new business or changing the current business in order to fit a new business environment, and how to utilize the system as a means to contribute to the business. The description includes, at the organization level, the organizational environment, goals and objectives, the business model, and the information environment, and, at the business operation level, the business operation model, business operation modes, business operational quality, organizational formation and concept of the proposed system. It is very important that the business management level should actively participate or lead the development of the business requirement specification.

The information elements of the BRS should be specified by the business. Business management should be responsible for the content of the specification. The BRS serves as the basis of the stakeholders' active participation in the requirement processes. For example, a business analyst or representative user from the business can review the BRS and discuss the business model or operation, business management can revise the BRS, or a system analyst can review the BRS and discuss potential technical solutions. Typical types of requirements included in the BRS are organizational requirements and business requirements.

NOTE 1 ISO/IEC/IEEE 15289 provides guidance to include business and organizational requirements in the system requirements specification. This document includes these requirements in the BRS since the contents are specified from the perspective of business management.

NOTE 2 The BRS is often identified with the stakeholder requirement specification (StRS) in many industries. Users of this document can combine the StRS with BRS according to the users' environment.

NOTE 3 The stakeholder requirements and business requirements are distinguished in *The Guide to the Business Analysis Body of Knowledge* (BABOK) as follows: Business Requirements are high-level statements of the goal, objectives, or needs of the enterprise. They describe why a project is initiated, what the project will achieve, and which metrics can be used to measure the project's success. Stakeholder Requirements are statements of the needs of a particular stakeholder or class of stakeholders. They describe the needs that a given stakeholder has and how that stakeholder will interact with a solution. Stakeholder Requirements serves as a bridge between Business Requirements and the various classes of solution requirements.

8.2.2 BRS example outline

The specific requirements clause of the BRS should be organized such that a consensus of the stakeholders agrees that the organization method aids understanding of the requirements. There is no one optimal organization for all projects. An example outline of a BRS created in an organizational/ business context is shown in Figure 5.

1. Introduction
- 1.1 Business purpose
- 1.2 Business scope
- 1.3 Business overview
- 1.4 Definitions
- 1.5 Major stakeholders

2. References

3. Business management requirements
- 3.1 Business environment
- 3.2 Mission, goals, and objectives
- 3.3 Business model
- 3.4 Information environment

4. Business operational requirements
- 4.1 Business processes
- 4.2 Business operational policies and rules
- 4.3 Business operational constraints
- 4.4 Business operational modes
- 4.5 Business operational quality
- 4.6 Business structure

5. Preliminary operational concept of proposed system
- 5.1 Preliminary operational concept
- 5.2 Preliminary operational scenarios

6. Other preliminary life-cycle concepts
- 6.1 Preliminary acquisition concept
- 6.2 Preliminary deployment concept
- 6.3 Preliminary support concept
- 6.4 Preliminary retirement concept

7 Project Constraints

8. Appendix
- 8.1 Acronyms and abbreviations

Figure 5 — Example BRS Outline

NOTE Detailed content of the BRS is found in 9.3.

8.3 Stakeholder requirements specification

8.3.1 General

The Stakeholder Requirements Specification (StRS) describes the organization's motivation for why the system is being developed or changed, defines processes and policies/rules under which the system is used and documents the top-level requirements from the stakeholders' perspective including expressing needs of users/operators/maintainers as derived from the context of use in a specific, precise and unambiguous manner. In the context described in the BRS, the StRS describes how the organization will utilize the system as a means to contribute to the business.

The information elements of the StRS should be specified by the stakeholders. The stakeholders should be responsible for the content of the specification. The StRS serves as the basis of the stakeholders' active participation in the requirement processes. Typical types of stakeholder requirements included in the StRS are organizational requirements, business requirements and user requirements.

NOTE 1 ISO/IEC/IEEE 15289 provides guidance to include user (stakeholder) requirements in the system requirements specification. This document includes these requirements in the StRS since the contents are specified from the stakeholders' perspective. They can be succeeded in the SyRS by addressing technical concerns.

NOTE 2 The StRS is often identified with the business requirement specification (BRS) in many industries. Users of this document can combine the StRS with the BRS according to the users' environment.

8.3.2 StRS example outline

The specific requirements clause of the StRS should be organized such that a consensus of the stakeholders agrees that the organization method aids understanding of the requirements. There is no one optimal organization for all projects. An example outline of an StRS created in an organizational/ business context is shown in Figure 6.

1. Introduction
- 1.1 Stakeholder purpose
- 1.2 Stakeholder scope
- 1.3 Overview
- 1.4 Definitions
- 1.5 Stakeholders

2. References

3. Business management requirements
- 3.1 Business environment
- 3.2 Mission, goals, and objective
- 3.3 Business model
- 3.4 Information environment

4. System operational requirements
- 4.1 System processes
- 4.2 System operational policies and rules
- 4.3 System operational constraints
- 4.4 System operational modes and states

5. User requirements

6. Detailed Life-cycle concepts of proposed system
- 6.1 Operational concept
- 6.2 Operational scenarios
- 6.3 Acquisition concept
- 6.4 Deployment concept
- 6.5 Support concept
- 6.6 Retirement concept

7 Project Constraints

8. Appendix
- 8.1 Acronyms and abbreviations

Figure 6 — Example StRS Outline

NOTE Detailed content of the StRS is found in 9.4.

8.4 System requirements specification

8.4.1 General

The System Requirements Specification (SyRS) identifies the technical requirements for the selected system-of-interest and usability for the envisaged human-system interaction. It defines the high-level system requirements from the domain perspective, along with background information about the overall objectives for the system, its target environment and a statement of the constraints, assumptions and non-functional requirements. It may include conceptual models designed to illustrate the system context, usage scenarios, the principal domain entities, data, information and workflows.

The purpose of the SyRS is to provide a description of what the system should do, in terms of the system's interactions or interfaces with its external environment. The SyRS should completely describe all inputs, outputs and required relationships between inputs and outputs. An SyRS has traditionally been viewed as a document that communicates the requirements of the acquirer to the technical community who will specify and build the system. The collection of requirements that constitutes the specification and its representation acts as the bridge between the two groups and needs to be understandable by both the acquirer and the technical community. One of the most difficult tasks in the creation of a system is that of communicating to all of the subgroups within both groups, especially in one document. This type of communication generally requires different formalisms and languages.

This document suggests a distinction between this structured collection of information and the way in which it is presented to its various audiences. The presentation of the SyRS should take a form that is appropriate for its intended use. This can be a paper document, models, prototypes, other non-paper document representations or any combination. All of these representations can be derived from this one SyRS to meet the needs of a specific audience. However, care should be taken to be certain that each of these presentations is traceable to a common source of system requirements information. The audience should be made aware that this structured collection of information remains the one definitive source for resolving ambiguities in the particular presentation chosen.

Generally, process requirements (how to develop or construct the system) should be contained in contract documentation such as a Statement of Work, not in a requirements specification. If included in a specification, they should be clearly identified as process requirements.

The SyRS presents the results of the definition of need, the system operational concept, the system architecture and the system requirements analysis tasks. As such, it is a description of what the system's acquirers expect it to do for them, the system's expected environment, the system's usage profile, performance parameters, expected quality and effectiveness and verification activities.

8.4.2 SyRS example outline

The specific requirements section of an SyRS should be organized such that a consensus of the stakeholders agrees that the organization method aids understanding of the requirements. There is no one optimal organization for all projects. An example outline of an SyRS is shown in Figure 7.

1. Introduction
1.1 System purpose
1.2 System scope
1.3 System overview
1.3.1 System context
1.3.2 System functions
1.3.3 User characteristics
1.4 Definitions
2. References

3. System requirements
3.1 Functional requirements
3.2 Usability requirements
3.3 Performance requirements
3.4 Interface requirements
3.4.1 External interface requirements
3.4.2 Internal interface requirements
3.5 System operations
3.6 System modes and states
3.7 Physical characteristics
3.8 Environmental conditions
3.9 Security requirements
3.10 Information management requirements
3.11 Policy and regulation requirements
3.12 System life cycle sustainment requirements
3.13 Packaging, handling, shipping and transportation requirements
4. Verification
(parallel to subsections in Section 3)
5. Appendices
5.1 Assumptions and dependencies
5.2 Acronyms and abbreviations

NOTE This SyRS outline can be used, with tailoring, for subordinate specifications for system elements, even those that include software.

Figure 7 — Example SyRS Outline

NOTE Detailed content of the SyRS is found in 9.5.

8.5 Software requirements specification

8.5.1 General

The software requirements specification (SRS) is a specification for a particular software product, program, or set of programs that performs certain functions in a specific environment. The SRS may be written by one or more representatives of the supplier, one or more representatives of the acquirer, or by both.

It is important to consider the part that the SRS plays in the total project plan. The software may contain essentially all the functionality of the project or it may be part of a larger system. In the latter case typically there would be a requirement specification that states the interfaces between the system and its software portion, and places external performance and functionality requirements upon the software portion. Of course, the SRS should then agree with and expand upon these system requirements. The SRS indicates the precedence and criticality of requirements. The SRS defines all of the required capabilities of the specified software product to which it applies, as well as documenting the conditions and constraints under which the software has to perform, and the intended verification approaches for the requirements

8.5.2 SRS example outline

The specific requirements clause of the SRS should be organized such that a consensus of the system stakeholders agrees that the organization method aids understanding of the requirements. There is no one optimal organization for all systems. An example outline for an SRS is in Figure 8.

1. Introduction
- 1.1 Purpose
- 1.2 Scope
- 1.3 Product overview
  - 1.3.1 Product perspective
  - 1.3.2 Product functions
  - 1.3.3 User characteristics
  - 1.3.4 Limitations
- 1.4 Definitions

2. References

3. Requirements
- 3.1 Functions
- 3.2 Performance requirements
- 3.3 Usability requirements
- 3.4 Interface requirements
- 3.5 Logical database requirements
- 3.6 Design constraints
- 3.7 Software system attributes
- 3.8 Supporting information

4. **Verification**

(parallel to subsections in Section 3)

5. Appendices
- 5.1 Assumptions and dependencies
- 5.2 Acronyms and abbreviations

Figure 8 — Example SRS Outline

NOTE 1 Detailed content of the SRS is found in 9.6.

Examples of organizational approaches to requirements in an SRS include:

— System mode - some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, degraded or emergency.

— User class - some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers and firefighters.

— Objects - objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods or processes.

— Feature - a feature is an externally desired service by the system that may require a sequence of inputs to effect the desired result. For example, in a telephone system, features include local call, call forwarding and conference call. Each feature is generally described in a sequence of stimulus-response pairs.

— Stimulus - some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

— Response - some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating pay checks, all functions associated with generating a current list of employees, etc.

— Functional hierarchy - when none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

NOTE 2 There are many notations, methods and automated support tools available to aid in the documentation of SRS requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts can prove helpful; when organizing by object, object-oriented analysis can prove helpful; when organizing by feature, stimulus-response sequences can prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries can prove helpful.

9 Information item content

9.1 General

This clause states the normative content of the required information items.

NOTE Information content in 9.2 is generally applicable to items of information maintained in document form.

9.2 General content

9.2.1 Identification

Include the following identification matter:

a) title; and

b) revision notice.

The title and a revision notice uniquely identify the document. Revision information may include the project name, version number of the document, date of release, approved signature, a list of sub-clauses that have been changed in the current version of the document and a list of version numbers and dates of release of all previous versions of the document.

9.2.2 Front matter

Include the following front matter:

a) a table of contents;

b) a list of figures; and

c) a list of tables.

9.2.3 Definitions

Provide definitions for any words or phrases that have special meaning beyond normal dictionary definitions.

9.2.4 References

Include the following information regarding references:

a) provide a complete list of all documents referenced elsewhere;

b) identify each document by title, report number (if applicable), date and publishing organization; and

c) specify the sources from which the references can be obtained.

This information may be provided by reference to an appendix or to another document. The references information should be subdivided into a ’Compliance’ section, containing references to those cited documents containing requirements which are included by that citation and a ‘Guidance’ section, containing reference to those cited documents containing information, but no requirements.

9.2.5 Acronyms and abbreviations

Spell out or define all acronyms and abbreviations used in the documents.

NOTE This information can be provided by reference to one or more appendixes in the documents or by reference to other documents.

9.3 Business requirements specification (BRS) content

9.3.1 BRS overview

This subclause defines the normative content of the business requirements specification (BRS). The project shall produce the following information item content in accordance with the project’s policies with respect to the business requirements specification. Organization of the content such as the order and section structure may be selected in accordance with the project’s information management policies.

9.3.2 Business purpose

Describe at the organization level the reason and background for which the organization is pursuing new business or changing the current business in order to fit a new management environment. In this context it should describe how the proposed system will contribute to meeting business objectives.

9.3.3 Business scope

Define the business domain under consideration by:

a) identifying the business domain by name;

b) defining the range of business activities included in the business domain concerned. The scope can be defined in terms of divisions in the organization and external entities that relate directly to the business activities, or functions to be performed by the business activities. It is helpful to show environmental entities that are outside of the scope;

c) describing the scope of the system being developed or changed. The description includes assumptions on which business activities are supported by the system.

9.3.4 Business overview

Describe major internal divisions and external entities of the business domain concerned and how they are interrelated. A diagrammatic description is recommended.

9.3.5 Major Stakeholders

List the major stakeholders or the classes of stakeholders and describe how they will influence the organization and business, or can be related to the development and operation of the system.

9.3.6 Business environment

Define external and internal environmental factors that should be taken into consideration in understanding the new or existing business and eliciting the stakeholder requirements for the system to be developed or changed. The environmental factors should include possible influences to the business and consequently the system from external conditions like market trends, laws and regulations, social responsibilities and technology base.

9.3.7 Mission, goals and objectives

Describe the business results to be obtained through or by the proposed system.

9.3.8 Business model

Describe methods by which the business mission is expected to be achieved. The description should be concentrated on the methods supported by the system to be developed or changed with the items such as product and services, geographies and locales, distribution channels, business alliance and partnership, and finance and revenue model.

NOTE Detailed discussions and definitions of the business model elements can be found in Business Motivation Model (BMM) Specification by OMG.

9.3.9 Information environment

Describe the overall strategy for the organization level decisions on common bases for multiple information systems. It should include the following items.

a) **project portfolio** – when multiple system projects are running or planned to pursue the same business goal, the priority, relative positioning and possible constraints come from the portfolio management strategy.

b) **long term system plan** – when common system infrastructure or architecture has been decided or planned, it should be described as constraints on possible design decisions.

c) **database configuration** – an organization level database configuration plan and possible constraints on availability and accessibility of organization global data should be specified.

9.3.10 Business processes

Provide description of the procedures of business activities and possible system interfaces within the processes. The purpose of this information item is to represent how and in which context the system supports the business activities. In general, business processes make an ordered structure with decomposition and classification. Each business process should be uniquely named and numbered in the structure. The description of the individual business process should be represented as a diagram representing a sequence of activities.

9.3.11 Business operational policies and rules

Describe logical propositions applied in conducting the business processes. The propositions may be conditions to start, branch and terminate the sequence of the business activities in the business processes; criteria for judgment in the business processes; or formula to evaluate a quantity, which will likely be addressed in functional requirements in the SyRS and SRS. The policies and rules shall be uniquely named and numbered, and shall be referenced in the description of the business processes.

9.3.12 Business operational constraints

Describe conditions to be imposed in conducting the business process. The conditions may be on a performance constraint (e.g., the process shall be finished within a day after the triggering event occurs), or may be from a management requisite such as 'every occurrence of the process shall be monitored and recorded'.

9.3.13 Business operational modes

Describe methods to conduct the business operation in an unsteady state, for example, a state when business operations might be extremely busy due to some intensive occurrence of events. An unsteady state of business operation includes a manual operation mode when the proposed system is not available due to some unexpected situation like an accident or natural disaster.

9.3.14 Business operational quality

Define the level of quality required for the business operation. For example, a business process may address required urgency with higher priority than the reliability of the business process.

NOTE This can include high-level objectives for usability and quality in use (effectiveness, efficiency, satisfaction and freedom from risk from use, see ISO 9241-220:—, 9.3.1.1).

9.3.15 Business structure

Identify and describe the structures in the business relevant to the system, such as organizational structure (divisions and departments), role and responsibility structures, geographic structures and resource sharing structures. There may be a need to align the system functions to these structures and to support future structural changes.

9.3.16 High-level operational concept

Describe the proposed system in a high-level manner, indicating the operational features that are to be provided without specifying design details. The following information should be included:

a) operational policies and constraints;

b) description of the proposed system;

c) modes of system operation;

d) user classes and other involved personnel; and

e) support environment.

NOTE Detailed discussion of the information item content for the System Operational Concept Document is in Annex A, particularly A.2.5.

9.3.17 High-level operational scenarios

Describe examples of how users/operators/maintainers will interact with the system in important contexts of use. The high-level scenarios are described for an activity or a series of activities of business processes supported by the system. The scenarios should be uniquely named and numbered and should be referenced in the description of the business processes in 9.3.9.

NOTE More information for the context of use and the usability requirements can be found in ISO/IEC 25030, ISO/IEC TR 25060, ISO 9241-220 and ISO 9241-210.

9.3.18 Other high-level life-cycle concepts

Describe how the system of interest is to be acquired, deployed, supported and retired.

9.3.19 Project constraints

Describe constraints to performing the project within cost and schedule.

9.4 Stakeholder requirements specification (StRS) content

9.4.1 StRS overview

This clause defines the normative content of the stakeholder requirements specification (StRS). The project shall produce the following information item content in accordance with the project's policies with respect to the stakeholder requirements specification. Organization of the content such as the order and section structure may be selected in accordance with the project's information management policies.

9.4.2 Stakeholder purpose

Describe at the organization level the reason and background for which the organization is pursuing a new business or changing the current business in order to fit a new management environment. In this context it should describe how the proposed system will contribute to meeting business objectives.

9.4.3 Stakeholder scope

Define the business domain under consideration by:

a) identifying the business domain by name;

b) defining the range of business activities included in the business domain concerned. The scope can be defined in terms of divisions in the organization and external entities that relate directly to the business activities, or functions to be performed by the business activities. It is helpful to show environmental entities which are outside of the scope;

c) describing the scope of the system being developed or changed. The description includes assumptions on which business activities are supported by the system.

9.4.4 Overview

Describe major internal divisions and external entities of the business domain concerned and how they are interrelated. A diagrammatic description is recommended.

9.4.5 Stakeholders

List the stakeholders or the classes of stakeholders and describe how they are related to the development and operation of the system.

9.4.6 Business environment

Define external and internal environmental factors that should be taken into consideration in understanding the new or existing business and eliciting the stakeholder requirements for the system to be developed or changed. The environmental factors should include possible influences to the business and consequently the system from external conditions like market trends, laws and regulations, social responsibilities, and technology base.

9.4.7 Mission, goals and objectives

Describe the business results to be obtained through or by the proposed system.

9.4.8 Business model

Describe methods by which the business goal is expected to be achieved. The description should be concentrated on the methods supported by the system to be developed or changed with the items such as product and services, geographies and locales, distribution channels, business alliance and partnership, and finance and revenue model.

NOTE Detailed discussions and definitions of the business model elements can be found in Business Motivation Model (BMM) Specification by OMG.

9.4.9 Information environment

Describe the overall strategy for the organization level decisions on common bases for multiple information systems. It should include the following items.

a) **project portfolio** – when multiple system projects are running or planned to pursue the same business goal, the priority, relative positioning and possible constraints come from the portfolio management strategy.

b) **long term system plan** – when common system infrastructure or architecture has been decided or planned, it should be described as constraints on possible design decisions.

c) **database configuration** – an organization level database configuration plan and possible constraints on availability and accessibility of organization global data should be specified.

9.4.10 System processes

Provide a description of how and in which context the system supports the business activities. In general, system processes flow from the ordered structure of the business processes with decomposition and classification. Each system process should be uniquely named and numbered in the structure. The description of the individual system process should be represented as a diagram representing a sequence of activities.

9.4.11 System operational policies and rules

Describe how the business operational policies and rules will likely be addressed in functional requirements in the SyRS and SRS. The policies and rules shall be uniquely named and numbered, and shall be referenced in the description of the business processes.

9.4.12 Operational constraints

Describe system conditions and functional requirements to be imposed on the system in conducting the business process. The conditions may result in a performance requirement in the SyRS.

9.4.13 System operational modes and states

Describe operational modes and states to support system operation.

9.4.14 System operational quality

Define the level of quality required for the system operation, such as performance, compatibility, reliability, security, maintainability and portability. For example, a process can address required urgency with higher priority than the reliability of the process.

NOTE See ISO/IEC 25010 for additional guidance on quality requirements.

9.4.15 User requirements

User requirements are requirements for use that provide the basis for design and evaluation of systems to meet identified user needs. User requirements can include requirements for use-related quality (including usability) that specify the intended outcomes and associated quality criteria, user-system interaction requirements that specify the required interaction to achieve the intended outcomes, and any constraints that could limit the freedom of design and implementation of solutions to satisfy the user requirements. The user requirements can be used as a basis for the operational scenarios that specify how to meet these requirements when interacting with the system.

The context of use specified for a design (i.e., the context in which the system is to be used) should be specified as part of the user requirements specification to clearly identify the conditions under which the requirements apply. Usability requirements and objectives for the system include measurable effectiveness, efficiency and satisfaction criteria in specific contexts of use.

NOTE 1 For more information about the context of use see ISO/IEC 25030 and ISO 9241-11. See ISO 20282-1 for more information about context of use for everyday products.

NOTE 2 Additional material on user needs and user requirements can be found in ISO/IEC TR 25060, ISO/IEC 25064, ISO 9241-210 and ISO 9241-220.

9.4.16 Operational concept

Describe the proposed system in a high-level manner, indicating the operational features that are to be provided without specifying design details. The following information should be included:

a) operational policies and constraints;

b) description of the proposed system;

c) modes of system operation;

d) user classes and other involved personnel; and

e) support environment.

NOTE Detailed discussion of the information item content for the System Operational Concept Document is in Annex A, particularly A.2.

9.4.17 Operational scenarios

Describe examples of how users/operators/maintainers will interact with the system in important contexts of use. The scenarios are described for an activity or a series of activities of business processes supported by the system. The scenario should be uniquely named and numbered and should be referenced in the description of the business processes in 9.3.10.

NOTE More information for the context of use and the usability requirements can be found in ISO/IEC 25030, ISO/IEC TR 25060, ISO 9241-220, and ISO 9241-210.

9.4.18 Other detailed concepts of proposed system

Describe the detailed content of the concepts for acquisition, deployment, support and retirement.

9.4.19 Project constraints

If appropriate, describe constraints to performing the project within cost and schedule.

9.5 System requirements specification (SyRS) content

9.5.1 SyRS overview

This clause defines the normative content of a system requirements specification (SyRS). The project shall produce the following information item content in accordance with the project's policies with respect to the system requirements specification. Organization of the content such as the order and section structure may be selected in accordance with the project's information management policies.

9.5.2 System purpose

Define the reason(s) for which the system is being developed or modified.

9.5.3 System scope

Define the scope of the system under consideration by:

a) identifying the system to be produced by name;

b) referring to and stating the results of the earlier finalized needs analysis, in the form of a brief but clear expression of the user's problem(s). It explains what the system will and will not do to satisfy those needs;

c) describing the application of the system being specified. As a portion of this, it should describe all relevant top-level benefits, objectives and goals as precisely as possible.

9.5.4 System overview

9.5.4.1 System context

Describe at a general level the major elements of the system, to include human elements and how they interact. The system overview includes appropriate diagrams and narrative to provide the context of the system, defining all significant interfaces crossing the system's boundaries.

9.5.4.2 System functions

Describe major system capabilities, conditions and constraints.

9.5.4.3 User characteristics

Identify each type of user/operator/maintainer of the system (by function, location, type of device), the number in each group, the nature of their use of the system and their characteristics and capabilities.

NOTE Where appropriate, the user characteristics of the SyRS and SRS are consistent.

9.5.5 Functional requirements

Define functional requirements applicable to system operation.

9.5.6 Usability requirements

Define usability and quality in use requirements and objectives for the system that can include measurable effectiveness, efficiency, satisfaction criteria and avoidance of harm that could arise from use in specific contexts of use.

9.5.7 Performance requirements

Define the critical performance conditions and their associated capabilities by including such considerations as:

a) dynamic actions or changes that occur (e.g., rates, velocities, movements and noise levels);

b) quantitative criteria covering endurance capabilities of the equipment required to meet the user needs under stipulated environmental and other conditions, including minimum total life expectancy. Indicate required operational session duration and planned utilization rate;

c) performance requirements for the operational phases and modes.

9.5.8 System interface requirements

Specify requirements for interfaces among system elements and with external entities. Interfaces among system elements should include interfaces with the human element. Interfaces with external entities should include other systems.

Define any interdependencies or constraints associated with the interfaces (e.g., communication protocols, special devices, standards, fixed formats). Each interface may represent a bidirectional flow of information. A graphic representation of the interfaces can be used when appropriate for the sake of clarity.

9.5.9 System operations

9.5.9.1 Human system integration requirements

Reference applicable documents and specify any special or unique requirements, e.g., constraints on allocation of functions to personnel and communications and personnel/equipment interactions.

Define requirements for any specific areas, stations or equipment that would require concentrated human engineering attention due to the sensitivity of the operation or criticality of the task (i.e., those areas where the effects of human error would be particularly serious).

NOTE ISO 9241-220 contains a formalized model that can be used in the specification, assessment and improvement of the human-centred processes in system development and operation.

9.5.9.2 Maintainability requirements

Specify the quantitative maintainability requirements that apply to maintenance in the planned maintenance and support environment. Examples are as follows.

a) Time (e.g., mean and maximum downtime, reaction time, turnaround time, mean and maximum times to repair, mean time between maintenance actions).

b) Rate (e.g., maintenance staff hours per specific maintenance action, operational ready rate, maintenance time per operating hour, frequency of preventative maintenance).

c) Maintenance complexity (e.g., number of people and skill levels, variety of support equipment, removing/replacing/repairing components).

d) Maintenance action indices (e.g., maintenance costs per operating hour, staff hours per overhaul).

e) Accessibility to components within systems and to parts within components.

9.5.9.3 Reliability requirements

Specify the system reliability requirements in quantitative terms, including the conditions under which the reliability requirements are to be met. This may also include the reliability apportionment model to support allocation of reliability values assigned to system functions for their share in achieving desired system reliability.

9.5.9.4 Other quality requirements

Define how the system will implement other quality requirements such as compatibility and portability.

9.5.10 System modes and states

If the system can exist in various operational modes or states define these and, as appropriate, use diagrams. Define modes and states requirements.

9.5.11 Physical characteristics

9.5.11.1 Physical requirements

Include constraints on weight, volume and dimension. Include the construction characteristics of where the system will be installed, requirements for materials to be used in the item or service covered by this specification, and requirements covering nameplates and system markings, interchangeability of equipment and workmanship.

9.5.11.2 Adaptability requirements

Define requirements for growth, expansion, capability and contraction. For example, if the system will require future network bandwidth, the applicable hardware should be specified with extra card slots to accommodate new network cards as demand increases.

9.5.12 Environmental conditions

Include environmental conditions to be encountered by the system. The following areas should be addressed: natural environment (e.g., wind, rain, temperature, humidity, flora, fauna, fungus, mold, sand, salt spray, dust, radiation, chemical, airborne contaminants and immersion); induced environment (e.g., motion, shock, noise, electromagnetic, thermal); electromagnetic signal environment; self-induced environment (e.g., motion, shock, noise, electromagnetic, thermal); threat; and cooperative environment. Consideration should also be given to legal/regulatory, political, economic, social and business environments.

9.5.13 System security requirements

Define the system security requirements related to both the facility that houses the system and operational security requirements of the system itself. One example of security requirements might be to specify the security and privacy requirements, including access limitations to the system, such as existence of log-on procedures and passwords and of data protection and recovery methods. This could include the factors that would protect the system from accidental or malicious access, use, modification, destruction or disclosure. Especially in safety-critical embedded systems this might incorporate a distributed log or history of data sets, the assignment of certain functions to different single systems, or the restriction of communications between some areas of the system.

9.5.14 Information management requirements

Define the requirements for the system's management of information that it receives, generates or exports. Examples include types and amounts of information the system is required to receive and store, any proprietary or other protections levied on the information the system deals with, and what backup and archiving requirements exist for the information.

9.5.15 Policy and regulation requirements

Derive requirements from organizational policies and business practices that will affect the operation or performance of the system. Derive requirements from relevant external regulations. Examples of requirements include multilingual support, labour policies and protection of personnel information.

Specify derived requirements for health and safety criteria, including those basic to the design of the system, with respect to equipment characteristics, methods of operation and environmental influences such as toxic systems and electromagnetic radiation.

9.5.16 System life cycle sustainment requirements

Outline quality activities, such as review and measurement collection and analysis, to help realize a quality system. Life cycle sustainment also includes provision of facilities needed to provide operational- and depot-level support, spares, sourcing and supply, provisioning, technical documentation and data, support-personnel training, initial cadre training and initial contractor-logistics support.

9.5.17 Packaging, handling, shipping and transportation requirements

Define requirements imposed on the system to make certain that it can be packaged, handled, shipped, transported and stored within its intended operational context.

9.5.18 Verification

Provide the verification approaches and methods planned to qualify the system or system element. The information elements for verification are recommended to be given in a parallel manner with the information elements in 9.5.5 to 9.5.17.

9.5.19 Assumptions and dependencies

List any assumptions and dependencies applicable to the system requirements that should be taken into account in the allocation and derivation of lower-level system requirements.

9.6 Software requirements specification (SRS) content

9.6.1 SRS overview

This clause defines the normative content of the software requirements specification (SRS). The project shall produce the following information item content in accordance with the project's policies with respect to the software requirements specification. Organization of the content such as the order and section structure may be selected in accordance with the project's information management policies.

9.6.2 Purpose

Delineate the purpose of the software to be specified.

9.6.3 Scope

Describe the scope of the software under consideration by:

a) identifying the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.);

b) explaining what the software product(s) will do;

c) describing the application of the software being specified, including relevant benefits, objectives and goals; and

d) being consistent with similar statements in higher-level specifications (e.g., a system requirements specification), if they exist.

9.6.4 Product perspective

Define the system's relationship to other related products.

If the product is an element of a larger system, relate the requirements of that larger system to the functionality of the product covered by the SRS.

If the product is an element of a larger system, identify the interfaces between the product covered by the SRS and the larger system of which the product is an element.

Consider a block diagram showing the major elements of the larger system, interconnections and external interfaces.

Describe how the software operates within the following constraints:

a) system interfaces;

b) user interfaces;

c) hardware interfaces;

d) software interfaces;

e) communications interfaces;

f) memory;

g) operations;

h) site adaptation requirements; and

i) interfaces with services.

9.6.4.1 System interfaces

List each system interface and identify the functionality of the software to accomplish the system requirement and the interface description to match the system.

9.6.4.2 User interfaces

Specify the logical characteristics of each interface between the software product and its users.

NOTE A style guide for the user interface can provide consistent rules for organization, coding and interaction of the user with the system.

9.6.4.3 Hardware interfaces

Specify the logical characteristics of each interface between the software product and the hardware elements of the system. This includes configuration characteristics (number of ports, instruction sets, etc.). It also covers such matters as what devices are to be supported, how they are to be supported, and protocols. For example, terminal support may specify full-screen support as opposed to line-by-line support.

9.6.4.4 Software interfaces

Specify the use of other required software products (e.g., a data management system, an operating system or a mathematical package), and interfaces with other application systems (e.g., the linkage between an accounts receivable system and a general ledger system).

For each required software product, specify:

a) name;

b) mnemonic;

c) specification number;

d) version number; and

e) source.

NOTE It is acceptable to specify required platforms or operating systems, but rarely feasible to require a specific version. Typically, a version number most recent version or any currently maintain version can be specified for software.

For each interface, specify:

a) discussion of the purpose of the interfacing software as related to this software product;

b) definition of the interface in terms of message content and format. It is not necessary to detail any well-documented interface, but a reference to the document defining the interface is required.

9.6.4.5 Communications interfaces

Specify the various interfaces to communications such as local network protocols.

9.6.4.6 Memory constraints

Specify any applicable characteristics and limits on primary and secondary memory.

9.6.4.7 Operations

Specify the normal and special operations required by the user such as:

a) the various modes of operations in the user organization (e.g., user-initiated operations);

b) periods of interactive operations and periods of unattended operations;

c) data processing support functions; and

d) backup and recovery operations.

NOTE This is sometimes specified as part of the User Interfaces section.

9.6.4.8 Site adaptation requirements

The site adaptation requirements include:

a) definition of the requirements for any data or initialization sequences that are specific to a given site, mission or operational mode (e.g., grid values, safety limits, etc.);

b) specification of the site or mission-related features that should be modified to adapt the software to a particular installation.

9.6.4.9 Interfaces with services

Specify interactions with services, e.g., Software as a Service (SaaS) or cloud services.

9.6.5 Product functions

Provide a summary of the major functions that the software will perform. For example, an SRS for an accounting program may use this part to address customer account maintenance, customer statement and invoice preparation without mentioning the vast amount of detail that each of those functions requires.

Sometimes the function summary that is necessary for this part can be taken directly from the section of the higher-level specification (if one exists) that allocates particular functions to the software product.

Use cases, user stories and scenarios are also used to describe product functions.

Note that for the sake of clarity:

a) the product functions should be organized in a way that makes the list of functions understandable to the acquirer or to anyone else reading the document for the first time.

b) textual or graphical methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.

9.6.6 User characteristics

Describe those general characteristics of the intended groups of users of the product including characteristics that may influence usability, such as educational level, experience, disabilities and technical expertise. This description should not state specific requirements, but rather should state the reasons why certain specific requirements are later specified in specific requirements in 9.6.9.

NOTE 1 Where appropriate, the user characteristics of the SyRS and SRS are consistent.

NOTE 2 For additional information on context of use and user needs, see ISO/IEC 25063 and ISO/IEC 25064.

9.6.7 Limitations

Provide a general description of any other items that will limit the supplier's options, including:

a) regulatory requirements and policies;

b) hardware limitations (e.g., signal timing requirements);

c) interfaces to other applications;

d) parallel operation;

e) audit functions;

f) control functions;

g) higher-order language requirements;

h) signal handshake protocols (e.g., XON-XOFF, ACK-NACK);

i) quality requirements (e.g., reliability);

j) criticality of the application;

k) safety and security considerations;

l) physical/mental considerations; and

m) limitations that are sourced from other systems, including real-time requirements from the controlled system through interfaces.

9.6.8 Assumptions and dependencies

List each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but any changes to these factors can affect the requirements in the SRS. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would have to change accordingly.

9.6.9 Apportioning of requirements

Apportion the software requirements to software elements. For requirements that will require implementation over multiple software elements, or when allocation to a software element is initially undefined, this should be so stated. A cross-reference table by function and software element should be used to summarize the apportionments.

Identify requirements that may be delayed until future versions of the system (e.g., blocks and/or increments).

9.6.10 Specified requirements

Specify the software system requirements to a level of detail sufficient for software design, development and verification of the software increment or release in process.

The requirements should:

a) be stated in conformance with all the characteristics described in 5.2 of this document;

b) be cross-referenced to earlier versions or related documents;

c) be uniquely identifiable;

d) describe every input (stimulus) into the software system, every output (response) from the software system, and all functions performed by the software system in response to an input or in support of an output.

9.6.11 External interfaces

Define all inputs into and outputs from the software system. The description should complement the interface descriptions in 9.6.4.1 through 9.6.4.5, and should not repeat information there.

Each interface defined should include the following content:

a) name of item;

b) description of purpose;

c) source of input or destination of output;

d) valid range, accuracy and/or tolerance;

e) units of measure;

f) timing;

g) relationships to other inputs/outputs;

h) data formats;

i) command formats; and

j) data items or information included in the input and output.

9.6.12 Functions

Define the fundamental actions that have to take place in the software in accepting and processing the inputs and in processing and generating the outputs, including:

a) validity checks on the inputs;

b) exact sequence of operations;

c) responses to abnormal situations, including:

    1) overflow;

    2) communication facilities;

    3) hardware faults and failures; and

    4) error handling and recovery;

d) effect of parameters;

e) relationship of outputs to inputs, including:

    1) input/output sequences; and

    2) formulas for input to output conversion.

It may be appropriate to partition the functional requirements into sub-functions or sub-processes. This does not imply that the software design will also be partitioned that way.

9.6.13 Usability requirements

Define usability and quality in use requirements and objectives for the software system that can include measurable effectiveness, efficiency, satisfaction criteria and avoidance of harm that could arise from use in specific contexts of use.

NOTE Additional guidance on usability requirements can be found in ISO/IEC TR 25060.

9.6.14 Performance requirements

Specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole.

Static numerical requirements may include the following:

a) the number of terminals to be supported;

b) the number of simultaneous users to be supported; and

c) the amount and type of information to be handled.

Static numerical requirements are sometimes identified under a separate section entitled Capacity.

Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions.

The performance requirements should be stated in measurable terms.

For example,

95 % of the transactions shall be processed in less than 1 s.

rather than,

An operator shall not have to wait for the transaction to complete.

NOTE Numerical limits applied to one specific function are normally specified as part of the processing subparagraph description of that function.

9.6.15 Logical database requirements

Specify the logical requirements for any information that is to be placed into a database, including:

a) types of information used by various functions;

b) frequency of use;

c) accessing capabilities;

d) data entities and their relationships;

e) integrity constraints;

f) security; and

g) data retention requirements.

9.6.16 Design constraints

Specify constraints on the system design imposed by external standards, regulatory requirements or project limitations.

9.6.17 Standards compliance

Specify the requirements derived from existing standards or regulations, including:

a) report format;

b) data naming;

c) accounting procedures; and

d) audit tracing.

For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database shall be recorded in a trace file with before and after values.

9.6.18 Software system attributes

Specify the required attributes of the software product. The following is a partial list of examples:

a) Reliability - specify the factors required to establish the required reliability of the software system at the time of delivery.

b) Availability - specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery and restart.

c) Security - specify the requirements to protect the software from accidental or malicious access, use modification, destruction or disclosure. Specific requirements in this area could include the need to:

   1) utilize certain cryptographic techniques;

   2) keep specific log or history data sets;

3) assign certain functions to different modules;

4) restrict communications between some areas of the programme;

5) check data integrity for critical variables; and

6) assure data privacy.

d) Maintainability - specify attributes of software that relate to the ease of maintenance of the software itself. These may include requirements for certain modularity, interfaces or complexity limitation. Requirements should not be placed here just because they are thought to be good design practices.

e) Portability - specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems, including:

1) percentage of elements with host-dependent code;

2) percentage of code that is host dependent;

3) use of a proven portable language;

4) use of a particular compiler or language subset; and

5) use of a particular operating system.

9.6.19 Verification

Provide the verification approaches and methods planned to qualify the software. The information items for verification are recommended to be given in a parallel manner with the information items in 9.6.10 to 9.6.18.

9.6.20 Supporting information

Additional supporting information to be considered includes:

a) sample input/output formats, descriptions of cost analysis studies or results of user surveys;

b) supporting or background information that can help the readers of the SRS;

c) a description of the problems to be solved by the software; and

d) special packaging instructions for the code and the media to meet security, export, initial loading or other requirements.

The SRS should explicitly state whether or not these information items are to be considered part of the requirements.

Annex A
(normative)

System operational concept

A.1 Overview

A System Operational Concept (OpsCon) document describes what the system will do (not how it will do it) and why (rationale). An OpsCon is a user-oriented document that describes system characteristics of the to-be-delivered system from the user's viewpoint. The OpsCon document is used to communicate overall quantitative and qualitative system characteristics to the acquirer, user, supplier and other organizational elements.

Users of this document should consider producing separate System OpsCon and SyRS documents. The OpsCon document can then focus on all necessary requirements specifically from the user's point of view, and can use vocabulary and illustration tools that are familiar to the user's experience and knowledge base. The benefits include: definition of the issues, constraints and opportunities related to human involvement with the system, production of an estimate of what has not been specified, clarification of the constraints, opportunities and degree of flexibility required of the system, and setting of priorities for the requirements. However, there is nothing in this document that would preclude producing an information item combining the System OpsCon and the SyRS.

The project shall produce the following information item in accordance with the project's policies with respect to the System Operational Concept. The information item content in A.2.7 shall be produced in the course of producing the Stakeholder Requirements Specification as referred to in 9.3.16 and 9.3.17.

NOTE 1 A separate OpsCon document enables the engineering team during development of the SyRS to focus more on the technical community and the vocabulary and knowledge base of the system suppliers and users.

NOTE 2 Similarities exist between the Concept of Operations and the System Operational Concept. The Concept of Operations describes the view from the organization regarding the intent and assumptions for operation(s). It captures a broad picture of the organizational level purpose. See Annex B.

A.2 Operational concept document (OpsCon)

A.2.1 General

Describe each of the essential elements of an OpsCon document. Each version of an OpsCon document based on this guide should contain a title and a revision notice that uniquely identifies the document. Revision information may include the project name, version number of the document, date of release, approval signatures, a list of clauses that have been changed in the current version of the document, and a list of version numbers and dates of release of all previous versions of the document. The approved OpsCon document should be placed under configuration control.

The preface of an OpsCon document provides information that the writer wants the reader to know prior to reading the document. The preface should include the purpose of the document, the scope of activities that resulted in its development, who wrote the document and why, the intended audience for the document and the expected evolution of the document.

A table of contents, a list of figures and a list of tables should be included in every OpsCon document.

A.2.2 Scope

A.2.2.1 General

Provide an overview of the OpsCon document and the system to which it applies.

A.2.2.2 Identification

Include the identifying number, title and abbreviation (if applicable) of the system or subsystem to which this OpsCon applies. If related OpsCon documents for an overall system have been developed in a rank ordered structure or network manner, the position of this document relative to other OpsCon documents should be described.

A.2.2.3 Document overview

Summarize and expand on the purposes of motivations for the OpsCon document. The intended audience for the document should also be mentioned. Describe any security or privacy considerations associated with use of the OpsCon. Outline the remaining parts of this guide. The purposes of an OpsCon document will, in most cases, be:

— to communicate the user's needs for and expectations of the proposed system to the acquirer and/or supplier; or

— to communicate the acquirer's or supplier's understanding of the users' need and how the system shall operate to fulfil those needs.

However, an OpsCon document might also serve other purposes, such as building consensus among several user groups, among several acquirer organizations, and/or among several suppliers.

The audience of an OpsCon document can be a variety of people:

— Users might read it to determine whether their needs and desires have been correctly specified by their representative or to verify the suppliers understanding of their needs.

— Acquirers might read it to acquire knowledge of the user's needs and/or supplier's understanding of those needs.

— Suppliers will typically use the OpsCon document as a basis for system life cycle activities, and to familiarize new team members with the problem domain and the system to which the OpsCon applies.

A.2.2.4 System overview

Briefly state the purpose of the proposed system or subsystem to which the OpsCon applies. Describe the general nature of the system, and identify the project sponsors, user agencies, supplier organizations, support agencies, certifiers or certifying bodies, and the operating centres or sites that will run the system. Also identify other documents relevant to the present or proposed system. A graphical overview of the system is strongly recommended. This can be in the form of a context diagram, a top-level object diagram or some other type of diagram that depicts the system and its environment. Documents that might be cited include, but are not limited to: the project authorization, relevant technical documentation, significant correspondence, documents concerning related projects, risk analysis reports and feasibility studies.

A.2.3 Referenced documents

List the document number, title, revision and date of all documents referenced in the OpsCon document. Also, identify the source for all documents not available through normal channels.

A.2.4 Current system or situation

A.2.4.1 General

Describe the system or situation (either automated or manual) as it currently exists. If there is no current system on which to base changes, describe the situation that motivates the proposed system. In this case, the following sections may be tailored as appropriate to describe the motivating situation. Introduce the problem domain. This enables readers to better understand the reasons for the desired changes and improvements.

A.2.4.2 Background, objectives and scope

Provide an overview of the current system or situation, including as applicable, background, mission, objectives and scope. In addition to providing the background for the current system, this section should provide a brief summary of the motivation for the current system. Examples of motivations for a system might include automation of certain tasks or countering of certain threat situations. The goals for the current system should also be defined, together with the strategies, solutions, tactics, methods and techniques used to accomplish them. The modes of operation, classes of users and interfaces to the operational environment define the scope of the proposed system, which are summarized in this section and defined in greater detail in subsequent sections.

A.2.4.3 Operational policies and constraints

Describe any operational policies and constraints that apply to the current system or situation. Operational policies are predetermined management decisions regarding the operations of the current system, normally in the form of general statements or understandings that guide decision-making activities. Policies limit decision-making freedom but do allow for some discretion. Operational constraints are limitations placed on the operations of the current system. Examples of operational constraints include the following:

a) a constraint on the hours of operation of the system, perhaps limited by access to secure terminals;

b) a constraint on the number of personnel available to operate the system;

c) a constraint on the computer hardware (for example, shall operate on computer X);

d) a constraint on the operational facilities, such as office space.

A.2.4.4 Description of the current system or situation

Provide a description of the current system or situation, including the following, as appropriate:

a) the operational environment and its characteristics;

b) major system elements and the interconnection among those elements;

c) interfaces to external systems or procedures;

d) capabilities, functions/services and features of the current system;

e) charts and accompanying descriptions depicting inputs, outputs, data flows, control flows and manual and automated processes sufficient to understand the current system or situation from the user's point of view;

f) cost of system operations;

g) operational risk factors;

h) performance characteristics, such as speed, throughput, volume, frequency, workload;

i) quality attributes, such as: availability, correctness, efficiency, expandability, flexibility, interoperability, maintain-ability, portability, reliability, reusability, supportability, survivability and usability;

j) provisions for safety, security, privacy, integrity and continuity of operations in emergencies; and

k) logistics requirements to support system.

Since the purpose of this section is to describe the current system and how it operates, it is appropriate to use any tools and/or techniques that serve this purpose. It is important that the description of the system be simple enough and clear enough that all intended readers of the document can fully understand it. It is also important to keep in mind that the OpsCon document is written using the users' terminology.

Graphical tools should be used wherever possible, especially since OpsCon documents should be understandable by several different types of readers. Useful graphical tools include, but are not limited to, work breakdown structures (WBS), N2 charts showing functional or physical interfaces, sequence or activity charts, functional flow block diagrams, structure charts, allocation charts, data flow diagrams (DFD), object diagrams, context diagrams, storyboards and entity-relationship diagrams.

The description of the operational environment should identify, as applicable, the facilities, equipment, computing hardware, software, personnel and operational procedures used to operate the existing system. This description should be as detailed as necessary to give the readers an understanding of the numbers, versions, capacity, etc., of the operational equipment being used. For example, if the current system contains a database, the capacity of the storage unit(s) should be specified, provided the information exerts an influence on the users' operational capabilities. Likewise, if the system uses communication links, the capacities of those links should be specified if they exert influence on factors such as user capabilities, response time or throughput.

Those aspects of safety, security and privacy that exert an influence on the operation or operational environment of the current system should be described.

The information in this section should be organized as appropriate to the system or situation, as long as a clear description of the existing system is achieved. If parts of the descriptions are voluminous, they can be included in an appendix or incorporated by reference. An example of material that might be included in an appendix would be a data dictionary. An example of material to be included by reference might be a detailed manual of operational policies and procedures for the current system.

A.2.4.5 Modes of operation for the current system or situation

Describe the various modes of operation for the current system or situation (e.g., operational, degraded, maintenance, training, emergency, alternate-site, peacetime, wartime, ground-based, flight, active and idle modes). All of the modes that apply to all classes of users should be included. Important modes to include are degraded, backup and emergency modes, if such exist. This is especially true if these modes involve different geographical sites and equipment that have significant impacts on the operational aspects of the system.

This section can be further divided into lower-level sections, one for each mode described. System processes, procedures and capabilities or functions should be related to each mode, as appropriate, perhaps using a cross-reference matrix.

A.2.4.6 User classes and other involved personnel

A user class is distinguished by the ways in which users interact with the system. Factors that distinguish a user class include common responsibilities, skill levels, work activities and modes of interaction with the system. Different user classes may have distinct operational scenarios for their interactions with the system. In this context, a user is anyone who interacts with the existing system, including operational users, data entry personnel, system operators, operational support personnel, software maintainers and trainers.

This section can be organized further, as follows, if it is helpful in communicating the content.

A.2.4.6.1 Organizational structure

Describe the existing organizational structures of the various user groups and user classes that are involved with the current system. Organizational charts are useful graphic tools for this purpose.

A.2.4.6.2 Profiles of user classes

Provide a profile of each user class for the current system. If some users play several roles, each role should be identified as a separate user class.

Each user class for the current system, including operators and maintainers, should be described in a separate section. Each of these should provide a description of the user class, including responsibilities, education, background, skill level, activities and modes of interaction with the current system.

A.2.4.6.3 Interactions among user classes

Describe interactions among the various user classes involved with the current system, in particular, interactions among user groups, operators and maintainers. Interactions that occur among the users of the system and between users and non-users, both within the organization and across organizational boundaries, if they are relevant to the operation of the existing system, should be described. Informal as well as formal interactions should be included.

A.2.4.6.4 Other involved personnel

Describe other personnel who will not directly interact with the system, but who have an influence on, and are influenced by, the present system. Examples include executive managers, policy makers and the user's clients. Although these individuals do not have hands-on interaction with the system, they may significantly influence and be influenced by, the new or modified system.

A.2.4.7 Support environment

Describe the support concepts and support environment for the current system, including the support agency or agencies, facilities, equipment, support software, repair or replacement criteria, maintenance levels and cycles, and storage, distribution and supply methods.

A.2.5 Justification for and nature of changes

A.2.5.1 General

Describe the shortcomings of the current system or situation that motivate development of a new system or modification of an existing system. Provide a transition from the discussion of the current system or situation, to the description of the proposed system. If there is no current system on which to base changes, this section should so indicate and provide justification for the features of the new system.

A.2.5.2 Justification for changes

This section should:

a) briefly summarize new or modified aspects of the user needs, missions, objectives, environments, interfaces, personnel or other factors that require a new or modified system;

b) summarize the deficiencies or limitations of the current system or situation that make it unable to respond to new or changed factors; and

c) provide justification for a new or modified system:

   1) if the proposed system is to meet a new opportunity, describe the reasons why a new system should be developed to meet this opportunity;

   2) if the proposed system improves a current operation, describe the rationale behind the decision to modify the existing system (e.g., to reduce life cycle costs or improve personnel efficiency); and

   3) if the proposed system implements a new functional capability, explain why this function is necessary.

A.2.5.3 Description of desired changes

Summarize new or modified capabilities, functions, processes, interfaces and other changes needed to respond to the factors identified in A.2.4.1. Changes should be based on the current system. If there is no existing system on which to base changes, summarize the capabilities to be provided by a new system. This description should include the following, as appropriate.

— Capability changes. Description of the functions and features to be added, deleted and modified in order for the new or modified system to meet its objectives and requirements.

— System processing changes. Description of the changes in the process or processes of transforming data that will result in new output with the same data, the same output with new data, or both.

— Interface changes. Description of changes in the system that will cause changes in the interfaces and changes in the interfaces that will cause changes in the system.

— Personnel changes. Description of changes in personnel caused by new requirements, changes in user classes, or both.

— Environment changes. Description of changes in the operational environment that will cause changes in the system functions, processes, interfaces, or personnel and/or changes that should be made in the environment because of changes in the system functions, processes, interfaces or personnel.

— Operational changes. Description of changes to the user's operational policies, procedures, methods or daily work routines caused by the above changes.

— Support changes. Description of changes in the support requirements caused by changes in the system functions, processes, interfaces, or personnel and/or changes in the system functions, processes, interfaces or personnel caused by changes in the support environment.

— Other changes. Description of other changes that will impact the users, but that do not fit under any of the above categories.

A.2.5.4 Priorities among changes

Identify priorities among the desired changes and new features. Each change should be classified as essential, desirable or optional. Desirable and optional changes should be prioritized within their classes. If there is no existing system on which to base changes, this section should classify and prioritize the features of the proposed system.

— Essential features. Features that shall be provided by the new or modified system. The impacts that would result if the features were not implemented should be explained for each essential feature.

— Desirable features. Features that should be provided by the new or modified system. Desirable features should be prioritized. Reasons why the features are desirable should be explained for each desirable feature.

— Optional features. Features that might be provided by the new or modified system. Optional features should be prioritized. Reasons why the features are optional should be explained for each optional feature.

Classifying the desired changes and new features into essential, desirable and optional categories is important to guide the decision-making process during the life cycle of the proposed system. This information is also helpful in cases of budget or schedule cuts or overruns, since it permits determination of which features have to be finished and which ones can be delayed or omitted.

A.2.5.5 Changes considered but not included

Identify changes and new features considered but not included in A.2.4.2 and the rationale for not including them. By describing changes and features considered but not included in the proposed system, the authors document the results of their analysis activities. This information can be useful to other personnel involved with the system, whether it be users, acquirers or suppliers should they want to know if a certain change or feature was considered, and if so, why it was not included. In software especially, there are few, if any, outward signs of what has been changed, improved or is still unsafe or unsecure (e.g., in certain scenarios or workarounds).

A.2.5.6 Assumptions and constraints

Describe any assumptions or constraints applicable to the changes and new features identified. This should include all assumptions and constraints that will affect users during the life cycle of the new or modified system. An assumption is a condition that is taken to be true. An example of an assumption is that the system workload will double over the next two years, thus a new system with higher performance is required. A constraint is an externally imposed limitation placed on the new or modified system or the processes used. Examples of constraints include external interface requirements and limits on schedule and budget.

A.2.6 Concepts for the proposed system

A.2.6.1 General

Describe the proposed system that results from the desired changes specified in A.2.4. Describe the proposed system in a high-level manner, indicating the operational features that are to be provided without specifying design details. Methods of description to be used and the level of detail in the description will depend on the situation. The level of detail should be sufficient to fully explain how the proposed system is envisioned to operate in fulfilling users' needs and buyer's requirements. In some cases, it may be necessary to provide some level of design detail in the OpsCon. The OpsCon should not contain design specifications, but it may contain some examples of typical design strategies, for the purpose of clarifying operational details of the proposed system. In the event that actual design constraints need to be included in the description of the proposed system, they shall be explicitly identified as required to avoid possible misunderstandings.

NOTE If some of the features of the proposed system are the same as the features of the original system, the comment 'No change' appears after the section number and name.

A.2.6.2 Background, objectives and scope

Provide an overview of the new or modified system, including, as applicable, background, mission, objectives and scope. In addition to providing the background for the proposed system, this section should provide a brief summary of the motivation for the system. Examples of motivations for a system might include automation of certain tasks or taking advantage of new opportunities. The goals for the new or modified system should also be defined, together with the strategies, solutions, tactics, methods and techniques proposed to achieve those goals. The modes of operation, classes of users and interfaces to the operational environment define the scope of the proposed system, which are summarized in this section and defined in greater detail in subsequent sections.

A.2.6.3 Operational policies and constraints

Describe operational policies and constraints that apply to the proposed system. Operational policies are predetermined management decisions regarding the operation of the new or modified system, normally in the form of general statements or understandings that guide decision-making activities. Policies limit decision-making freedom, but do allow for some discretion. Operational constraints are limitations placed on the operations of the proposed system. Examples of operational constraints include the following:

— a constraint on the hours of operations of the system, perhaps limited by access to secure terminals;

— a limiting constraint on the number of personnel available to operate the system;

— a limiting constraint on the computer hardware (e.g., shall operate on computer X); and

— a limiting constraint on the operational facilities, such as office space.

A.2.6.4 Description of the proposed system

This section contains the major portion of the description of the proposed system. It provides a description of the proposed system, including the following, as appropriate:

a) the operational environment and its characteristics;

b) major system elements and the interconnections among these elements;

c) interfaces to external systems or procedures;

d) capabilities or functions of the proposed system;

e) charts and accompanying descriptions depicting inputs, outputs, data flow, and manual and automated processes sufficient to understand the proposed system or situation from the user's point of view;

f) cost of systems operations;

g) operational risk factors;

h) performance characteristics, such as speed, throughput, volume, frequency;

i) quality attributes, such as: reliability, availability, correctness, efficiency, expandability, flexibility, interoperability, maintainability, portability, reusability, supportability, survivability and usability; and

j) provisions for safety, security, privacy, integrity, and continuity of operations in emergencies.

Since the purpose of this section is to describe the proposed system and how it should operate, it is appropriate to use any tools and/or techniques that serve that purpose. See A.2.4.4 for further discussion.

A.2.6.5 Modes of operation

Describe the various modes of operation for the proposed system (for example, regular, degraded, maintenance, training, emergency, alternate-site, peacetime, wartime, ground-based, flight, active and idle modes). Include all of the modes that apply to all user classes. Important modes to include are degraded, backup and emergency modes, if such exist. This is especially true if these modes involve different geographical sites and equipment that have significant impacts on the system.

This section can be further divided into lower-level sections, one for each mode described. System processes, procedures and capabilities or functions should be related to each mode.

A.2.6.6 User classes and other involved personnel

A.2.6.6.1 General

A user class is distinguished by the ways in which the users interact with the system. Factors that distinguish a user class include responsibilities, skill level, work activities and mode of interaction with the system. Different user classes may have distinct operational scenarios for their interactions with the system. In this context, a user is anyone who will interact with the proposed system, including operational users, data entry personnel, system operators, operational support personnel, maintainers and trainers. This section can be further divided into lower-level sections if it is helpful in communicating the content.

A.2.6.6.2 Organizational structure

Describe the organizational structures of the various user groups and user classes that are to be involved with the proposed system. Organizational charts are useful graphic tools for this purpose.

A.2.6.6.3 Profiles of user classes

Provide a profile of each user class for the proposed system. If some users play several roles, each role should be identified as a separate user class.

Each user class for the proposed system, including operators, maintainers and trainers, should be described in a separate section. Each section should provide a description of the user class, including responsibilities, education, background, skill level, activities and envisioned modes of interaction with the proposed system.

A.2.6.6.4 Interactions among user classes

Describe interactions among the various user classes that may be involved with the proposed system. In particular, interaction among user groups, operators and maintainers should be described. Interactions that will occur among the users of the proposed system and between users and non-users, both within the organization and across interfacing organizations, if they are relevant to the operation of the proposed system, should be described. Informal as well as formal interactions should be included.

A.2.6.6.5 Other involved personnel

Describe other personnel who will not directly interact with the system, but who have an influence on, and are influenced by, the present system. Examples include executive managers, policy makers and the user’s clients. Although these individuals do not have hands-on interaction with the system, they may significantly influence and be influenced by the new or modified system.

A.2.6.7 Support environment

Describe the support concepts and support environment for the proposed system, including the support agency or agencies, facilities, equipment, support software, repair or replacement criteria, maintenance levels and cycles, and storage, distribution, and supply methods.

A.2.7 Operational scenarios

A scenario is a step-by-step description of how the proposed system should operate and interact with its users and its external interfaces under a given set of circumstances. Scenarios should be described in a manner that will allow readers to walk through them and gain an understanding of how all the various parts of the proposed system function and interact. The scenarios tie together all parts of the system, the users and other entities by describing how they interact. Scenarios may also be used to describe what the system should not do.

Scenarios should be organized into sections and subsections, each describing an operational sequence that illustrates the roles of the system, its interactions with users and interactions with other systems. Operational scenarios should be described for all operational modes and all classes of users identified for the proposed system. Each scenario should include events, actions, stimuli, information and interactions as appropriate to provide a comprehensive understanding of the operational aspects of the proposed system. Prototypes, storyboards and other media, such as video or hypermedia presentations, may be used to provide part of this information.

In most cases, it may be necessary to develop several variations of each scenario, including one for normal operation, one for stress load handling, one for exception handling, one for degraded mode operation, etc.

Scenarios play several important roles. The first is to bind together all of the individual parts of a system into a comprehensible whole. Scenarios help understand how all the pieces interact to provide operational capabilities. The second role of scenarios is to provide operational details for the proposed system. This enables better understanding of the users' roles, how the system should operate and the various operational features to be provided.

Scenarios can also support the development of simulation models that help in the definition and allocation of derived requirements, identification and preparation of prototypes to address key issues. In addition, scenarios can serve as the basis for the first draft of the users' manual and as the basis for developing acceptance test plans. Scenarios are also useful for the acquirer and the supplier to verify that the system design will satisfy the stakeholders' needs and expectations.

Scenarios can be presented in several different ways. One approach is to specify scenarios for each major processing function of the proposed system. Using this approach, this section would contain one section for each process. Each section would then contain several more lower-level sections, one for each scenario supported by that process. An alternative approach is to develop thread-based scenarios, where each scenario follows one type of transaction type through the proposed system. In this case, each section would contain one scenario for each interaction type, plus scenarios for degraded, stress loaded and back-up modes of operation. Other alternatives include following the information flow through the system for each user capability, following the control flows or focusing on the objects and events in the system.

Scenarios are an important element of an OpsCon and should therefore receive substantial emphasis. The number of scenarios and level of detail specified will be proportional to the perceived risk and the criticality of the project.

A.2.8 Summary of impacts

A.2.8.1 General

Describe the operational impacts of the proposed system on the users, the suppliers and the operations and maintenance organizations. Also describe the temporary impacts on users, acquirers, suppliers and the operations and maintenance organizations during the period of time when the new system is being developed, installed or trained.

This information is provided in order to allow all affected organizations to prepare for the changes that will be brought about by the new system and to allow for planning of the impacts on the acquirer, user groups and the operations and maintenance organizations during the development of, and transition to the new system.

A.2.8.2 Operational impacts

Further divide this section into lower-level sections to describe the anticipated operational impacts on the user, support and operations or maintenance organizations during the operation of the proposed system. These impacts may include the following:

— interfaces with primary or alternate computer operating centres;

- changes in procedure;
- use of new data sources;
- changes in quantity, type and timing of data to be input into the system;
- changes in data retention requirements;
- new modes of operation based on emergency, disaster or accident conditions;
- new methods for providing input data if the required data are not readily available;
- changes in operational budget; and
- changes in operational risks.

A.2.8.3 Organizational impacts

Further divide this section to describe the anticipated operational impacts on the user, development and support or maintenance organization during the operation of the proposed system. These impacts may include the following:

- modification of responsibilities;
- addition or elimination of job positions;
- training or retraining users;
- changes in numbers, skill levels, position identifiers or locations of personnel; and
- numbers and skill levels of personnel needed for contingency operation at one or more alternate sites following an emergency, disaster or accident.

A.2.8.4 Impacts during development

Further divide this section to describe the anticipated impacts on the user, development and support or maintenance agency or agencies during the development project for the proposed system. These impacts may include the following:

- involvement in studies, meetings and discussions prior to award of the contract;
- user and support involvement in reviews and demonstrations, evaluation of initial operating capabilities and evolving versions of the system, development or modification of databases and required training;
- parallel operation of the new and existing systems; and
- operational impacts during system testing of the proposed system.

A.2.9 Analysis of the proposed system

A.2.9.1 General

Provide an analysis of the benefits, limitations, disadvantages and alternatives considered for the proposed system.

A.2.9.2 Benefits

Provide a qualitative (and to the extent possible, quantitative) summary of the benefits to be provided by the proposed system. This summary should include the below items, as applicable. In each case, the benefits should be related to deficiencies identified.

— New capabilities. Additional new features or functionality.

— Enhanced capabilities. Upgrades to existing capabilities.

— Deleted capabilities. Unused, obsolete, confusing or dangerous capabilities removed.

— Improved performance. Better response time, reduced storage requirements, improved quality, decreased system/user workload, etc.

A.2.9.3 Disadvantages and limitations

Provide a qualitative (and to the extent possible, quantitative) summary of the disadvantages and/or limitations of the proposed system. Disadvantages might include the need to retrain personnel, rearrange work spaces or change to a new style of user interface, limitations might include features desired by users but not included, degradation of existing capabilities to gain new capabilities or greater-than-desired response time for certain complex operations.

A.2.9.4 Alternatives considered

Describe major alternatives considered, the trade-off analysis results and the rationale for the decisions reached. In the context of an OpsCon document, alternatives are operational alternatives and not design alternatives, except to the extent that design alternatives may be limited by the operational capabilities desired in the new system. This information can be useful to determine, now and at later times, whether a given approach was analyzed and evaluated, or why a particular approach or solution was rejected. This information would probably be lost if not recorded.

A.2.10 Appendices

To facilitate ease of use and maintenance of the OpsCon document, some information may be placed in appendices to the document. Charts and classified data are typical examples. Each appendix should be referenced in the main body of the document where that information would normally have been provided. Appendices may be bound as separate documents for easier handling.

A.2.11 Glossary

A glossary should be maintained and updated during the processes of concept analysis and development of the OpsCon document. Include an alphabetical listing of all acronyms and abbreviations, along with their meanings as used in this document, and a list of any terms and definitions needed to understand the document. To avoid unnecessary work due to misinterpretations, all definitions should be reviewed and agreed upon by all involved parties.

Annex B
(informative)

Concept of operations

B.1 Overview

This Annex provides guidance for the Concept of Operations (ConOps) in terms of information item content. The ConOps is not a requirements specification required in this document, but is intended to help users of this document in the requirements elicitation task as practical documentation to understand the background of the project and communicate the overall business and system characteristics.

The ConOps, at the organization level, addresses the leadership's intended way of operating the organization. It may refer to the use of one or more systems, as black boxes, to forward the organization’s goals and objectives. The ConOps describes the organization’s assumptions or intent in regard to an overall operation or series of operations of the business with using the system to be developed, existing systems and possible future systems. It is frequently embodied in long-range strategic plans and annual operational plans. The ConOps serves as a basis for the organization to direct the overall characteristics of the future business and systems, for the project to understand its background, and for the users of this document to implement the stakeholder requirements elicitation. The ConOps should include the following information item content.

B.2 Concept of operation content

B.2.1 Purpose

Describe the purpose by describing the current status of the organization, the goal of the long-range strategic plan and gaps between them.

B.2.2 Scope

Describe the scope by specifying the organization’s business domains to which it applies.

B.2.3 Strategic plan

Describe the long-range strategic plan for the organization’s business and resultant systems. This includes description when the current business of the organization should be changed and what systems should be implemented and operated. The plan may contain options with priority.

B.2.4 Effectiveness

Estimate the effectiveness expected by implementing the plan.

B.2.5 Overall operation

B.2.5.1 Context

Describe an overall view for the organization’s business, which business function or organizational unit is covered by which existing systems, and which is to be covered by planned systems. For more detailed description, major data resources and flows may be mapped onto the view.

B.2.5.2 Systems

List and outline the existing systems and systems to be developed in the future.

B.2.5.3 Organizational unit

List and relate with each other the organizational units that play roles in business operation, systems operation and their management.

B.2.6 Governance

B.2.6.1 Governance policies

Describe policies or principles that will govern any critical business and technological decisions in implementing the plan.

B.2.6.2 Organization

Describe the organization unit that is responsible for the governance and describe policies on organizational structure and human resource in system development.

B.2.6.3 Investment plan

Describe plans on possible investment for the development of systems and depict how the policies should be managed.

B.2.6.4 Information asset management

Describe policies of how information assets are managed and define which organizational unit is responsible for the management.

B.2.6.5 Security

Describe policies on security.

B.2.6.6 Business continuity plan

Describe policies on business continuity plan.

B.2.6.7 Compliance

Describe policies on regulation and any applicable regulatory compliance.

Annex C
(normative)

Tailoring policies

C.1 General

This Annex provides requirements for the tailoring of the information item contents in this document. Tailoring is not a requirement for conformance to the document. In fact, tailoring is not permitted if a claim of 'full conformance' is to be made. If a claim of 'tailored conformance' is made, then the following process shall be applied to perform the tailoring.

C.2 Information item tailoring process

C.2.1 Purpose

The purpose of the tailoring process is to adapt the information item contents of this document to satisfy particular circumstances or factors that:

a) surround an organization that is employing this document in an agreement;

b) influence a project that is required to meet an agreement in which this document is referenced; and

c) reflect the needs of an organization in order to supply products or services.

C.2.2 Outcomes

As a result of successful implementation of the tailoring process, modified or new information item contents are defined to achieve the purpose of the requirements specifications provided in this document.

C.2.3 Activities and tasks

If this document is tailored, the organization or project shall implement the following activities:

a) Identify and document the circumstances that influence tailoring. These influences include, but are not limited to:

   1) stability of and variety in operational environments;

   2) novelty, size and complexity;

   3) starting date and duration of utilization;

   4) emerging technology opportunities;

   5) profile if budget and organizational resources available;

   6) availability of the services of enabling systems; and

   7) the need to conform to other standards.

b) Obtain input from all parties affected by the tailoring decisions. This includes, but may not limited:

   1) the stakeholders;

   2) the interested parties to an agreement mode by the organization; and

   3) the contributing organizational functions.

c) Select the information item contents that require tailoring and delete them.

NOTE 1 Irrespective of tailoring, organizations and projects are always permitted to create additional information items contents beyond those required for conformance to this document.

NOTE 2 An organization or project may encounter a situation where there is the desire to modify the information item contents. Modification is performed by deleting the information item contents (making the appropriate claim of tailored conformance) and, with careful consideration of consequences, creating information item contents that describe additional information beyond those of the tailored document.

Bibliography

[1] ISO 9241-11, *Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts*

[2] ISO 9241-210, *Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*

[3] ISO 9241-220[1], *Ergonomics of human-system interaction — Part 220: Processes for enabling, executing and assessing human-centred design within organizations*

[4] ISO/IEC 14143-1, *Information technology — Software measurement — Functional size measurement — Part 1: Definition of concepts*

[5] ISO/IEC/IEEE 15289:2017, *Systems and software engineering — Content of systems and software life cycle process information products (documentation)*

[6] ISO/IEC 16085, *Systems and software engineering — Life cycle processes — Risk management*

[7] ISO/TS 18152, *Ergonomics of human-system interaction — Specification for the process assessment of human-system issues*

[8] ISO 20282-1, *Ease of operation of everyday products — Part 1: Design requirements for context of use and user characteristics*

[9] ISO/IEC/IEEE 24748-1, *Systems and software engineering — Life cycle management — Part 1: Guidelines for life cycle management*

[10] ISO/IEC/IEEE 24748-2[2], *Systems and software engineering — Life cycle management — Part 2: Guidelines for the application of ISO/IEC 15288 (Systems life cycle processes)*

[11] ISO/IEC/IEEE 24765:2017, *Systems and software engineering — Vocabulary*

[12] ISO/IEC/TR 24766, *Information technology — Systems and software engineering — Guide for requirements engineering tool capabilities*

[13] ISO/IEC 25000:2014, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*

[14] ISO/IEC 25010, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*

[15] ISO/IEC 25030, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Quality requirements*

[16] ISO/IEC/TR 25060, *Systems and software engineering — Systems and software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability: General framework for usability-related information*

[17] ISO/IEC 25062, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability test reports*

[18] ISO/IEC 25063, *Systems and software engineering — Systems and software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability: Context of use description*

1) Under preparation.

2) Under preparation.

[19] ISO/IEC 25064, *Systems and software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability: User needs report*

[20] ISO/IEC 26551, *Software and systems engineering — Tools and methods for product line requirements engineering*

[21] ISO/IEC/TR 29138-1, *Information technology — Accessibility considerations for people with disabilities — Part 1: User needs summary*

[22] ISO/IEC/IEEE 42020[3)], *Systems and software engineering — Architecture processes*

[23] ISO 9000:2015, *Quality management systems — Fundamentals and vocabulary*

[24] ANSI/AIAA G-043A-2012, *Guide for the Preparation of Operational Concept Documents, American Institute of Aeronautics and Astronautics, July 2012*

[25] EIA 632:1999, *Processes for Engineering a System, Electronics and Information Technology Association, January 1999*

[26] IEEE Std 1012-2016, *IEEE Standard for System, Software and Hardware Verification and Validation*

[27] Handbook Systems Engineering Version 4, International Council on Systems Engineering, 2015

[28] The Guide to the Business Analysis Body of Knowledge (BABOK). Version 2.0, International Institute of Business Analysis, March 2009

3) Under preparation.

IEEE notices and abstract

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board

445 Hoes Lane

Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws._They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at http://ieeexplore.ieee.org/xpl/standards.jsp or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

The list of IEEE participants can be accessed at the following URL: https://standards.ieee.org/project/29148.html.

Abstract and keywords

This document contains provisions for the processes and products related to the engineering of requirements for systems and software products and services throughout the life cycle. It defines the construct of a good requirement, provides attributes and characteristics of requirements, and discusses the iterative and recursive application of requirements processes throughout the life cycle. This document provides additional guidance in the application of requirements engineering and management processes for requirements-related activities in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288. Information items applicable to the engineering of requirements and their content are defined. The content of this document can be added to the existing set of requirements-related life cycle processes defined by ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288, or can be used independently.

Keywords: requirement, requirement characteristics, concept of operations, ConOps, operational concept, OpsCon, prototyping, software requirements specification, system, system requirements specification, SyRS.

ICS 35.080

ISBN 978-1-5044-5302-8 STD23404 (PDF); 978-1-5044-5303-5 STDPD23404 (Print)

Price based on 92 pages