INTERNATIONAL STANDARD

ISO/IEC/ IEEE 29148

First edition
2011-12-01

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 29148:2011(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat, the IEC Central Office and IEEE do not accept any liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies and IEEE members. In the unlikely event that a problem relating to it is found, please inform the ISO Central Secretariat or IEEE at the address given below.

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56
CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
E-mail inmail@iec.ch
Web www.iec.ch

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

Published in Switzerland

Contents

Page

Foreword

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

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

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of ISO/IEC JTC 1 is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is called to the possibility that implementation of this standard may require the use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. ISO/IEEE is not responsible for identifying essential patents or patent claims for which a license may be required, for conducting inquiries into the legal validity or scope of patents or patent claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance or a Patent Statement and Licensing Declaration Form, 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 ISO or the IEEE Standards Association.

ISO/IEC/IEEE 29148 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in cooperation with the Software & Systems Engineering Standards Committee of the IEEE Computer Society, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

Introduction

This International Standard provides a unified treatment of the processes and products involved in engineering requirements throughout the life cycle of systems and software. This International Standard is the result of harmonization of the following sources:

ISO/IEC 12207:2008 (IEEE Std 12207-2008), *Systems and software engineering — Software life cycle processes*

ISO/IEC 15288:2008 (IEEE Std 15288-2008), *Systems and software engineering — System life cycle processes*

ISO/IEC/IEEE 15289:2011, *Systems and software engineering — Content of life-cycle information products (documentation)*

ISO/IEC TR 19759, *Software Engineering — Guide to the Software Engineering Body of Knowledge (SWEBOK)*

IEEE Std 830, *IEEE Recommended Practice for Software Requirements Specifications*

IEEE Std 1233, *IEEE Guide for Developing System Requirements Specifications*

IEEE Std 1362, *IEEE Guide for Information Technology — System Definition — Concept of Operations (ConOps) Document*

ISO/IEC TR 24748-1, *Systems and software engineering — Life cycle management — Part 1: Guide for life cycle management*

ISO/IEC/IEEE 24765, *Systems and software engineering — Vocabulary*

Systems and software engineering — Life cycle processes — Requirements engineering

1 Scope

This International Standard

- specifies the required processes that are to be implemented for the engineering of requirements for systems and software products (including services) throughout the life cycle,
- gives guidelines for applying the requirements and requirements-related processes described in ISO/IEC 12207:2008 (IEEE Std 12207-2008) and ISO/IEC 15288:2008 (IEEE Std 15288-2008),
- specifies the required information items that are to be produced through the implementation of the requirements processes,
- specifies the required contents of the required information items, and
- gives guidelines for the format of the required and related information items.

This International Standard is applicable to

- those who use or plan to use ISO/IEC 15288 and ISO/IEC 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 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 15288:2008 (IEEE Std 15288-2008) and/or ISO/IEC 12207:2008 (IEEE Std 12207-2008),
- those who use or plan to use ISO/IEC/IEEE 15289:2011 on projects dealing with man-made systems, software-intensive systems, software and hardware products, and services related to those systems and products, regardless of project scope, product(s), methodology, size or complexity, and
- anyone performing requirements engineering activities to aid in ensuring that the information items developed during the application of requirements engineering processes conform to ISO/IEC/IEEE 15289:2011.

2 Conformance

2.1 Intended Usage

This International Standard provides guidance for the execution of the ISO/IEC 15288 and ISO/IEC 12207 processes that deal with requirements engineering. This International Standard 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 International Standard can claim conformance to the process provisions or to the information item provisions, or both.

2.2 Conformance to processes

This International Standard 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 International Standard are contained in 5.2.4, 5.2.5, 5.2.6, 5.2.7, and 6.1.

NOTE 1 If a user of this International Standard claims full conformance to ISO/IEC 15288:2008 (IEEE Std 15288-2008) and/or ISO/IEC 12207:2008 (IEEE Std 12207-2008), then by implication the user may claim conformance to the processes in this International Standard.

NOTE 2 A claim to tailored conformance to ISO/IEC 15288:2008 (IEEE Std 15288-2008) and/or ISO/IEC 12207:2008 (IEEE Std 12207-2008), does not necessarily imply conformance to the processes in this International Standard.

2.3 Conformance to information item content

This International Standard 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 International Standard means that

— the user produces the required information items stated in this International Standard, and

— the user demonstrates that the information items produced during the requirements engineering activities conform to the content requirements defined in this International Standard.

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

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

NOTE 2 In this International Standard, for simplicity of reference, each information item is described as if it were published as a separate document. However, information items will 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.

2.4 Full conformance

A claim of full conformance to this International Standard is equivalent to claiming conformance

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

— to the requirements-engineering-related processes of ISO/IEC 15288:2008 (IEEE Std 15288-2008) and ISO/IEC 12207:2008 (IEEE Std 12207-2008) cited in subclause 6.1,

— to the information items cited in Clause 7, and

— to the content requirements of the information items in Clause 9 and Annex A.

2.5 Tailored conformance

2.5.1 Processes

This International Standard does not make provision for tailoring processes. ISO/IEC 15288:2008 (IEEE Std 15288-2008), Annex A provides normative direction regarding the tailoring of system life cycle processes. ISO/IEC 12207:2008 (IEEE Std 12207-2008), Annex A provides normative direction regarding the tailoring of software life cycle processes.

2.5.2 Information items

When this International Standard is used as a basis for establishing a set of information items that do not qualify for full conformance, the clauses of this International Standard are selected or modified in accordance with the tailoring process prescribed in Annex D. 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.

3 Normative references

The following referenced documents are indispensable for the application 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 12207:2008 (IEEE Std 12207-2008), *Systems and software engineering — Software life cycle processes*

ISO/IEC 15288:2008 (IEEE Std 15288-2008), *Systems and software engineering — System life cycle processes*

ISO/IEC/IEEE 15289:2011, *Systems and software engineering — Content of life-cycle information products (documentation)*

4 Terms, definitions and abbreviated terms

4.1 Terms and definitions

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

4.1.1
acquirer
stakeholder that acquires or procures a product or service from a supplier

[ISO/IEC 15288:2008 (IEEE Std 15288-2008) and ISO/IEC 12207:2008 (IEEE Std 12207-2008)]

NOTE Other terms commonly used for an acquirer are buyer, customer, owner, and purchaser.

4.1.2
attribute
inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means

[ISO/IEC 25000:2005]

NOTE 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).

4.1.3
baseline
specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures

[ISO/IEC 15288:2008 (IEEE Std 15288-2008) and ISO/IEC 12207:2008 (IEEE Std 12207-2008)]

4.1.4
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

[ANSI/AIAA G-043-1992]

NOTE 1 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.

NOTE 2 It provides the basis for bounding the operating space, system capabilities, interfaces and operating environment.

4.1.5
condition
measurable qualitative or quantitative attribute that is stipulated for a requirement

4.1.6
constraint
externally imposed limitation on system requirements, design, or implementation or on the process used to develop or modify a system

NOTE A constraint is a factor that is imposed on the solution by force or compulsion and may limit or modify the design changes.

4.1.7
customer
organization or person that receives a product or service

[ISO/IEC 15288:2008 (IEEE Std 15288-2008) and ISO/IEC 12207:2008 (IEEE Std 12207-2008)]

NOTE Customers are a subset of stakeholders.

4.1.8
derived requirement
requirement deduced or inferred from the collection and organization of requirements into a particular system configuration and solution

4.1.9
developer
organization that performs development tasks (including requirements analysis, design, testing through acceptance) during a life cycle process

[ISO/IEC 12207:2008 (IEEE Std 12207-2008)]

NOTE Developers are a subset of stakeholders.

4.1.10
document
uniquely identified unit of information for human use, such as a report, specification, manual or book in printed or electronic form

[ISO/IEC 15289:2006]

4.1.11
human systems integration
interdisciplinary technical and management process for integrating human considerations with and across all system elements, an essential enabler to systems engineering practice

NOTE Adapted from INCOSE SEHbk 3.2:2010.

4.1.12
level of abstraction
view of an object at a specific level of detail

4.1.13
mode
set of related features or functional capabilities of a product

[IEEE Std 1362-1998]

4.1.14
operational concept
verbal and graphic statement of an organization's assumptions or intent in regard to an operation or series of operations of a system or a related set of systems

[ANSI/AIAA G-043-1992]

NOTE The operational concept is designed to give an overall picture of the operations using one or more specific systems, or set of related systems, in the organization's operational environment from the users' and operators' perspective. See also concept of operations.

4.1.15
operational scenario
description of an imagined sequence of events that includes the interaction of the product or service with its environment and users, as well as interaction among its product or service components

NOTE Operational scenarios are used to evaluate the requirements and design of the system and to verify and validate the system.

4.1.16
operator
entity that performs the operations of a system

[ISO/IEC 15288:2008 (IEEE Std 15288-2008) and ISO/IEC 12207:2008 (IEEE Std 12207-2008)]

NOTE The role of operator and the rule of user may be vested, simultaneously or sequentially, in the same individual or organization.

4.1.17
requirement
statement which translates or expresses a need and its associated constraints and conditions

NOTE Requirements exist at different tiers and express the need in high-level form (e.g. software component requirement).

4.1.18
requirements elicitation
process through which the acquirer and the suppliers of a system discover, review, articulate, understand, and document the requirements on the system and the life cycle processes

NOTE Adapted from ISO/IEC/IEEE 24765:2010.

4.1.19
requirements engineering
interdisciplinary function that mediates between the domains of the acquirer and supplier to establish and maintain the requirements to be met by the system, software or service of interest

NOTE Requirements engineering is concerned with discovering, eliciting, developing, analyzing, determining verification methods, validating, communicating, documenting, and managing requirements.

4.1.20
requirements management
activities that ensure requirements are identified, documented, maintained, communicated and traced throughout the life cycle of a system, product, or service

4.1.21
requirements traceability matrix
table that links requirements to their origin and traces them throughout the project life cycle

4.1.22
requirements validation
confirmation by examination that requirements (individually and as a set) define the right system as intended by the stakeholders

NOTE Adapted from EIA 632:1999.

4.1.23
requirements verification
confirmation by examination that requirements (individually and as a set) are well formed

NOTE 1 Adapted from EIA 632:1999.

NOTE 2 This means that a requirement or a set of requirements has been reviewed to ensure the characteristics of good requirements are achieved.

4.1.24
software requirements specification
structured collection of the requirements (functions, performance, design constraints, and attributes) of the software and its external interfaces

NOTE Adapted from IEEE Std 1012:2004.

4.1.25
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 Stakeholders include, but are not limited to, end users, end user organizations, supporters, developers, producers, trainers, maintainers, disposers, acquirers, customers, operators, supplier organizations, accreditors, and regulatory bodies.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008) and ISO/IEC 12207:2008 (IEEE Std 12207-2008)]

4.1.26
state
condition that characterizes the behaviour of a function/subfunction or element at a point in time

[ISO/IEC 26702]

4.1.27
supplier
organization or individual that enters into an agreement with the acquirer for the supply of a product or service

[ISO/IEC 15288:2008 (IEEE Std 15288-2008) and ISO/IEC 12207:2008 (IEEE Std 12207-2008)]

NOTE Suppliers are a subset of stakeholders.

4.1.28
system-of-interest
system whose life cycle is under consideration in the context of this International Standard

[ISO/IEC 15288:2008 (IEEE Std 15288-2008)]

4.1.29
system requirements specification
structured collection of the requirements (functions, performance, design constraints, and attributes) of the system and its operational environments and external interfaces

NOTE Adapted from IEEE Std 1233:1998 and IEEE Std 1012:2004.

4.1.30
trade-off
decision-making actions that select from various requirements and alternative solutions on the basis of net benefit to the stakeholders

4.1.31
user
individual or group that benefits from a system during its utilization

[ISO/IEC 15288:2008 (IEEE Std 15288-2008)]

NOTE 1 The role of user and operator may be vested, simultaneously or sequentially, in the same individual or organization.

NOTE 2 Users are a subset of stakeholders.

4.1.32
validation
confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled

[ISO 9000:2005]

NOTE Validation in a system life cycle context is the 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.

4.1.33
verification
confirmation, through the provision of objective evidence, that specified requirements have been fulfilled

[ISO 9000:2005]

NOTE 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.

4.2 Abbreviated terms

BRS	Business Requirements Specification
ConOps	Concept of Operations
FSM	Functional Size Measurement
HSI	Human Systems Integration
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

5 Concepts

5.1 Introduction

This clause presents concepts that apply to requirements 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 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, determining verification methods, validating, communicating, documenting, and managing requirements. The result of requirements engineering is a hierarchy of requirements that:

- enables an agreed understanding between stakeholders (e.g., acquirers, users, customers, operators, suppliers)

- is validated against real-world needs, can be implemented

- provides a basis of verifying designs and accepting solutions.

The hierarchy 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. Project requirements can necessitate including two other groups as part of the minimum set of stakeholders. First, the organization 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 intentions (referred to as needs, goals, or objectives), that evolve into a more formal statement before arriving as valid stakeholder requirements. Initial stakeholder intentions 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 intentions at the organizational level and the System Operational Concept from the system perspective, requirements engineering leads stakeholders from those initial intentions to structured and more formal stakeholder requirement statements. These statements are well-formed and meet the characteristics of subclauses 5.2.4, 5.2.5, and 5.2.6.

The stakeholder requirements are then transformed into system requirements for the system-of-interest, in accordance with subclauses 5.2.4, 5.2.5, and 5.2.6. Consistent practice has shown this process requires iterative and recursive steps in parallel with other life cycle processes through the system design hierarchy. The recursive application of the processes in Clause 6 will generate lower-level system element requirements.

Clause 6 details the processes to perform stakeholder requirements definition and requirements analysis. 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:—, *Software and systems engineering — Tools and methods of requirements engineering and management for product lines* 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 will contribute to requirements validation with the stakeholders, and ensure that the requirements accurately capture stakeholder needs.

A well-formed requirement is a statement that

- can be verified,
- has to be met or possessed by a system to solve a stakeholder problem or to achieve a stakeholder objective,
- is qualified by measurable conditions and bounded by constraints, and
- 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.

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

The following provides guidance on writing well-formed requirements. A requirement is a statement which translates or expresses a need and its associated constraints and conditions. This statement is written in a language which can take the form of a natural language. If expressed in the form of a natural language, the statement should comprise a subject, a verb and a complement. A requirement shall state the subject of the requirement (e.g., the system, the software, etc.) and what shall be done (e.g., operate at a power level, provide a field for). 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'.
- 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. However, 'will' can be construed as legally binding, so it is best to avoid using it for requirements.

- Preferences or goals are desired, non-mandatory, non-binding provisions and use 'should'.
- Suggestions or allowances are non-mandatory, non-binding provisions and use 'may'.
- 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.
- Use positive statements and avoid negative requirements such as ‘shall not'.
- Use active voice: avoid using passive voice, such as 'shall be able to select'.

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

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

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

Or

[Condition] [Action or Constraint] [Value]

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

Or

[Subject] [Action] [Value]

EXAMPLE: The Invoice System **[Subject]**, shall display pending customer invoices **[Action]** in ascending order **[Value]** in which invoices are to be paid.

Figure 1 — Examples of Requirement 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 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,
- 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 Subclauses 6.2.3 and 6.3.3 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 removed or deleted, a deficiency will exist, which cannot be fulfilled by other capabilities of the product or process. 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.

- Implementation Free. The requirement, while addressing what is necessary and sufficient in the system, avoids placing unnecessary constraints on the architectural design. The objective is to be implementation-independent. The requirement states what is required, not how the requirement should be met.

NOTE While such information may still be important, the information should be documented and communicated in some other form of documentation, such as the requirements attributes in subclause 5.2.8 (e.g., rationale) in order to aid in design and implementation. Additionally, including design solutions in the requirements may risk the possibility that potential design solutions may 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.

- Consistent. The requirement is free of conflicts with other requirements.

- Complete. The stated requirement needs no further amplification because it is measurable and sufficiently describes the capability and characteristics to meet the stakeholder's need.

- Singular. The requirement statement includes only one requirement with no use of conjunctions.

- Feasible. The requirement is technically achievable, does not require major technology advances, and fits within system constraints (e.g., cost, schedule, technical, legal, regulatory) with acceptable risk.

- Traceable. The requirement is upwards traceable to specific documented stakeholder statement(s) of need, higher tier requirement, or other source (e.g., a trade or design study). The requirement is also downwards traceable to the specific requirements in the lower tier requirements specification or other system definition artefacts. That is, all parent-child relationships for the requirement are identified in tracing such that the requirement traces to its source and implementation.

- Verifiable. The requirement has the means to prove that the system satisfies the specified requirement. Evidence may be collected that proves that the system can satisfy the specified requirement. Verifiability is enhanced when the requirement is measurable.

5.2.6 Characteristics of a set of requirements

There are certain characteristics that need to be considered for the set of stakeholder, system, and system element requirements rather than for any individual requirement. This will ensure that the set of requirements collectively provide for a feasible solution that meets the stakeholder intentions and constraints. Each set of requirements shall possess the following characteristics:

- Complete. The set of requirements needs no further amplification because it contains everything pertinent to the definition of the system or system element being specified. In addition, the set contains no 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 Some practices are recommended to improve completeness; include all requirements types; account for requirements in all stages of the life cycle; and involve all stakeholders in the requirements elicitation activity.

- Consistent. The set of requirements does not have individual requirements which are contradictory. Requirements are not duplicated. The same term is used for the same item in all requirements.
- Affordable. The complete set of requirements can be satisfied by a solution that is obtainable/feasible within life cycle constraints (e.g., cost, schedule, technical, legal, regulatory).
- Bounded. The set of requirements maintains the identified scope for the intended solution without increasing beyond what is needed to satisfy user needs.

Careful checking of the requirements set and the traceable architectural design 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, the following considerations will help ensure that good requirements characteristics are employed.

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 will be recognition of design decisions / solution architectures defined at a higher level. This is part of the iterative and recursive application of the requirements analysis and architectural 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 adverbs and adjectives (such as 'almost always', 'significant', 'minimal')
- 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')
- 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)
- Negative statements (such as statements of system capability not to be provided)

All assumptions made regarding a requirement shall be documented and validated in one of the requirement's attributes in subclause 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

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

5.2.8.1 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 has to be unique - it is never changed (even if the identified requirement changes) nor is it reused (even if the identified requirement is deleted).

- 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 will facilitate trading off requirements and balancing the impact of changes among stakeholders.

- Dependency. The dependency between requirements should be defined, when a dependency exists. Some requirements could have a low priority from one of the stakeholders' perspective, but nevertheless be essential for the success of the system. For example, a requirement to measure external ambient temperature could be essential to provide support to other requirements such as the maintenance of internal cabin temperature. This relationship should be identified so that if the primary requirement is removed, the supporting requirement can also be eliminated.

- Risk. Risk analysis techniques can be used to determine a grading for system requirements in terms of their consequences or degree of risk avoidance. Major risks are related to potential financial loss, potential missed business opportunity, loss of confidence by stakeholders, environmental impact, safety and health issues, and national standards or laws.

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

- Source. Each requirement should include an attribute that indicates the originator. Multiple sources may be considered creators of each requirement. Identifying the sources for each requirement support identifying which organizations(s) to consult for requirement clarification, deconfliction, modification, or deletion. The concept of ownership is related to source. Ownership applies to the origin of a requirement. The requirement source indicates where the requirement came from. For stakeholder requirements, the stakeholder who issues the requirement gains ownership. As requirements are further devolved through allocation and derivation, the responsibility to meet the requirement is passed to the appropriate product team.

- 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. This aids collecting requirements into groups for analysis and allocation.

5.2.8.2 Examples of the requirements type attribute

Important examples of the requirements type attribute include:

- Functional. Functional requirements describe the system or system element functions or tasks to be performed.

- Performance. A requirement that defines the extent or how well, and under what conditions, a function or task is to be performed. These are quantitative requirements of system performance and are verifiable individually. Note that there may be more than one performance requirement associated with a single function, functional requirement, or task.

  - 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.

- 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).

- Design Constraints. A requirement that limits the options open to a designer of a solution by imposing immovable boundaries and limits (e.g., the system shall incorporate a legacy or provided system element, or certain data shall be maintained in an on-line repository).

- 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.

- Non-Functional. These specify requirements under which the system is required to operate or exist or system properties. They define how a system is supposed to *be.* Quality requirements and human factors requirements are examples of this type.

  - Quality Requirements – Include a number of the 'ilities' in requirements to include, for example, transportability, survivability, flexibility, portability, reusability, reliability, maintainability, and security. The list of non-functional, quality requirements (e.g., “ilities”) should be developed prior to initiating the requirements document. This should be tailored to the system(s) being developed. As appropriate, measures for the quality requirements should be included as well.

NOTE 1 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 2 Additional guidance on human factors requirements can be found in subclause 6.2.3.2.

NOTE 3 There may be other ways to organize requirements types.

5.3 Practical considerations

5.3.1 Iteration and recursion of processes

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

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, analysed 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 ensure 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. The iterative application of processes is illustrated in Figure 2.

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

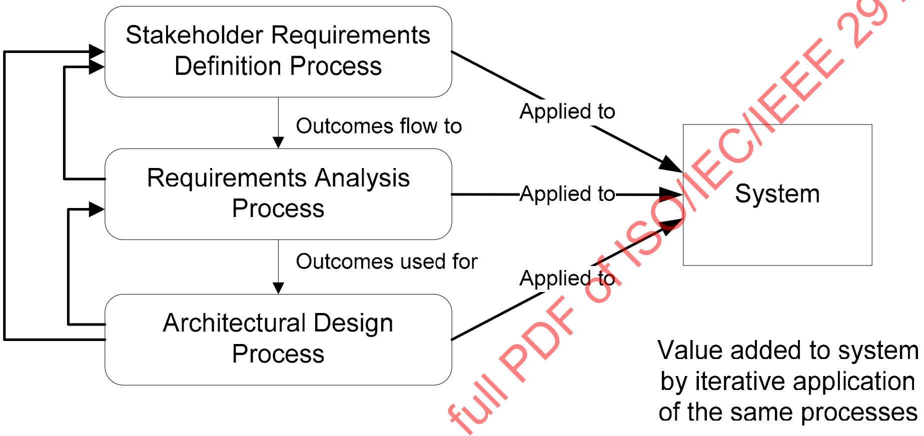
Stakeholder Requirements Definition Process
Outcomes flow to
Requirements Analysis Process
Outcomes used for
Architectural Design Process
Applied to
Applied to
Applied to
System
Value added to system by iterative application of the same processes

Figure 2 — Iterative 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 3 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 analysis and architectural design processes will be applied at each successive level of recursion.

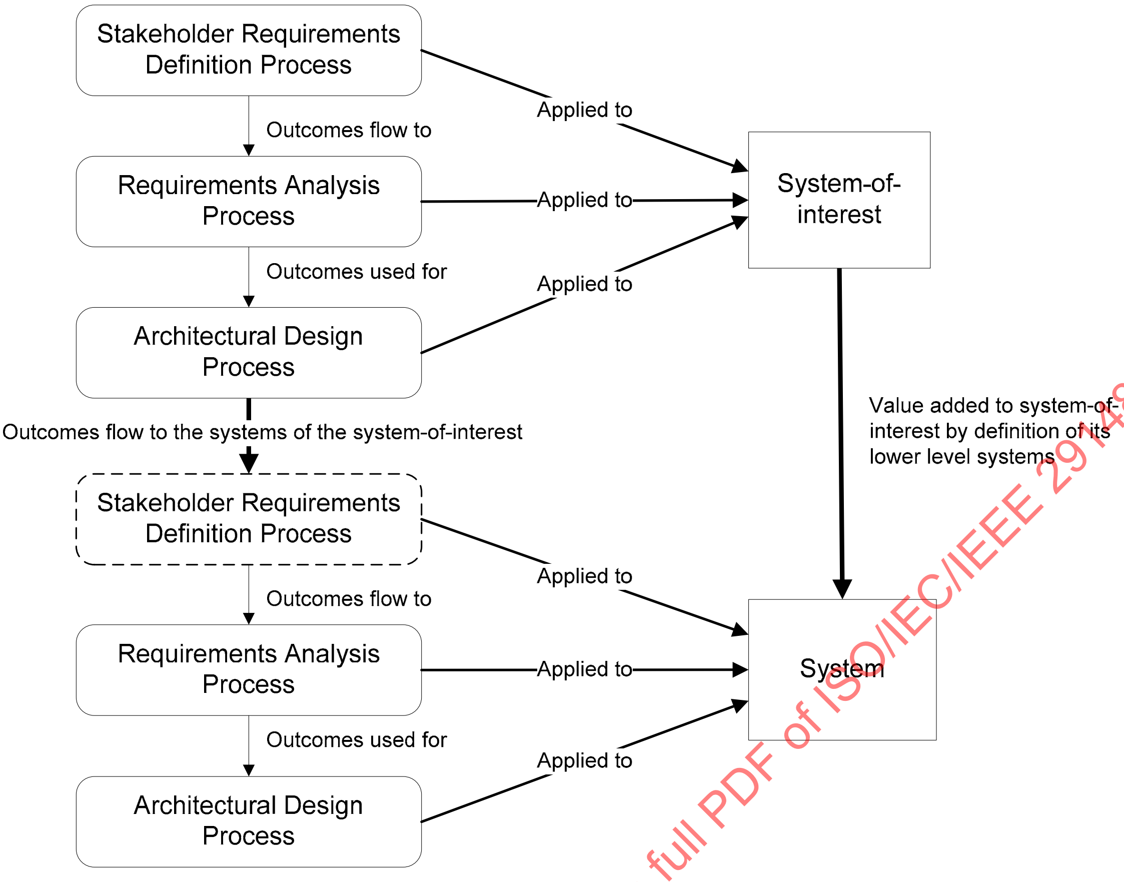
Stakeholder Requirements Definition Process
Outcomes flow to
Requirements Analysis Process
Outcomes used for
Architectural Design Process
Applied to
Applied to
Applied to
System-of-interest
Outcomes flow to the systems of the system-of-interest
Value added to system-of-interest by definition of its lower level systems
Stakeholder Requirements Definition Process
Outcomes flow to
Requirements Analysis Process
Outcomes used for
Architectural Design Process
Applied to
Applied to
Applied to
System

Figure 3 — Recursive application of processes

NOTE The recursion can also be bi-directional, with requirements from the system requiring further analysis at the system-of-interest level. This is not shown in Figure 3 for simplicity of the figure.

5.3.2 Iteration and recursion in requirements engineering

Since different groups of stakeholders often view the system from differing levels of abstraction, it is necessary to define and document requirements statements at lower, more detailed levels of abstraction than just the overall system-of-interest. This is accomplished by allocating or distributing the system requirements to the system elements. The activity of allocating requirements to system elements is part of the architectural design process and proceeds in parallel with the definition of the system architecture. There may be multiple iterations between the requirements analysis and architectural design processes to resolve trade-offs between the requirements and architecture (see Figure 2).

One of the main objectives of architectural design 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 analysis process (see Figure 3).

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 analysis and architectural 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 analysis process 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 rigorously.

NOTE Subclause 6.2.3.2 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.

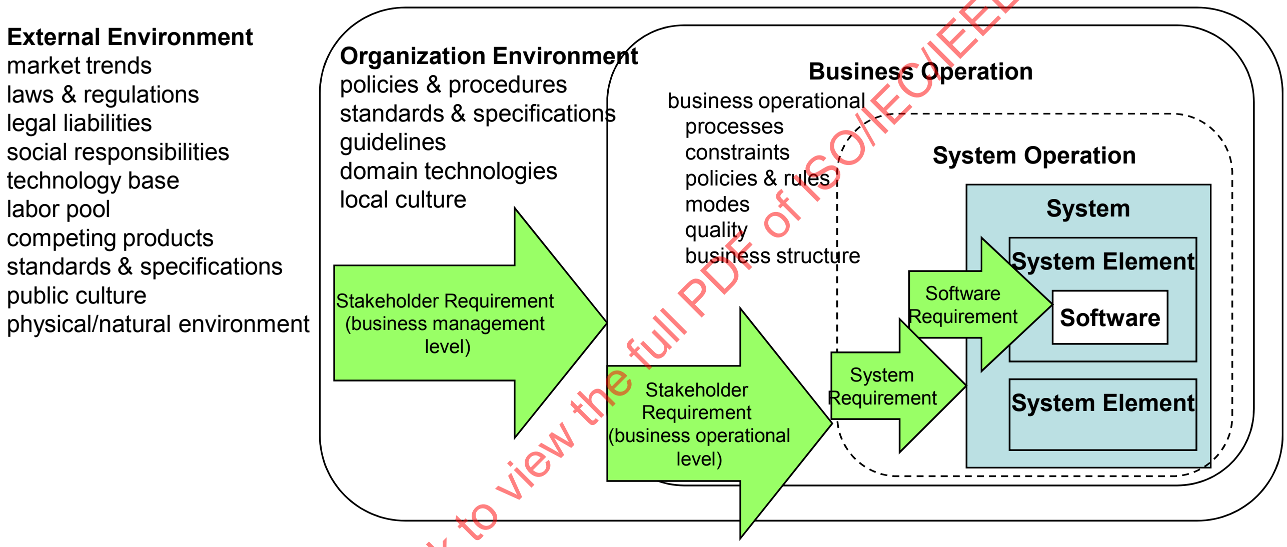

Figure 4 — Example of requirements scope in a business context

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 4.

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

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 4 as follows: StRS - Stakeholder Requirement (business management level and 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 5.

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.

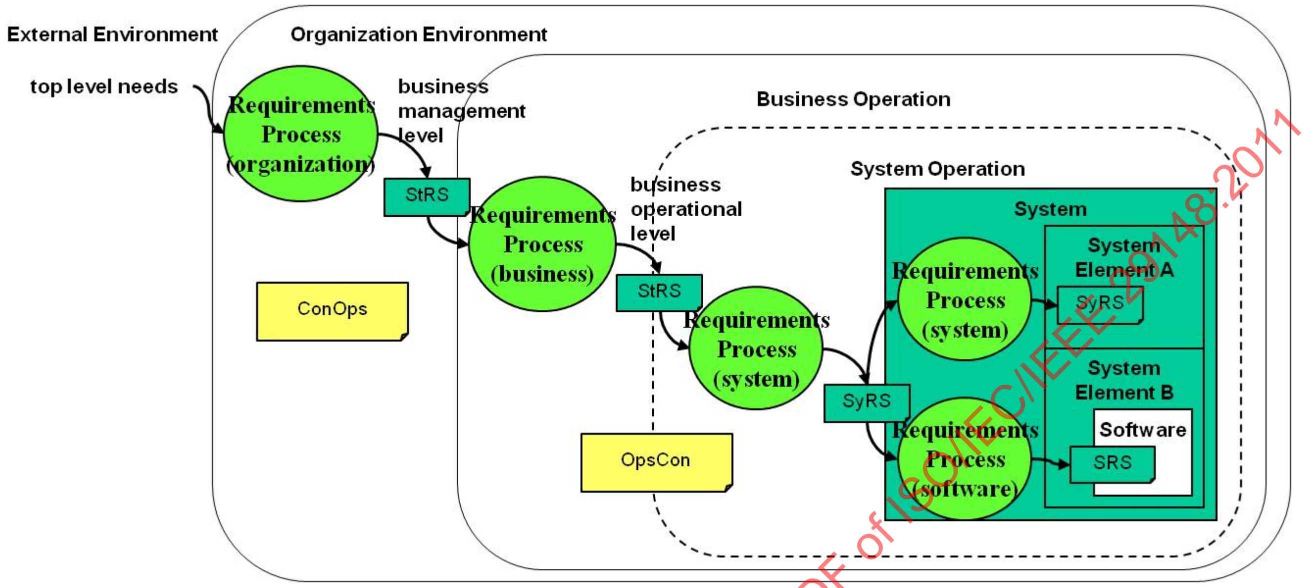
External Environment
top level needs
Organization Environment
Requirements Process (organization)
business management level
StRS
ConOps
Business Operation
Requirements Process (business)
business operational level
StRS
Requirements Process (system)
OpsCon
System Operation
SyRS
System
Requirements Process (system)
System Element A
SyRS
System Element B
Requirements Process (software)
Software
SRS

Figure 5 — Example of the sequence of 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 StRS, SyRS, SRS, ConOps, and OpsCon documents 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 may have parallel documentation for the various requirements they contain. However, in general, they will start with 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 15288:2008 (IEEE Std 15288-2008), *System life cycle processes*, and ISO/IEC 12207:2008 (IEEE Std 12207-2008), *Software life cycle processes*, depending on the adherence to one or both of ISO/IEC 15288 and ISO/IEC 12207:

- Stakeholder requirements definition process (ISO/IEC 15288:2008 (IEEE Std 15288-2008), subclause 6.4.1 or ISO/IEC 12207:2008 (IEEE Std 12207-2008), subclause 6.4.1)

- Requirements analysis process (ISO/IEC 15288:2008 (IEEE Std 15288-2008), subclause 6.4.2, or ISO/IEC 12207:2008 (IEEE Std 12207-2008), subclause 6.4.2)

- Software requirements analysis process (ISO/IEC 12207:2008 (IEEE Std 12207-2008), subclause 7.1.2) for the acquisition or supply of software products.

6.1.1 Guidelines for Processes

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

Beginning with subclause 6.2, original ISO/IEC 15288:2008 (IEEE Std 15288-2008) tasks relevant to this International Standard, 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 15288 was retained. The original source reference is included in the lower right hand corner. The original ISO/IEC 15288:2008 (IEEE Std 15288-2008) purposes and outcomes relevant to this International Standard are used in their entirety, without any change, for the subset of processes that are relevant to requirements engineering.

The principal processes are 1) Stakeholder Requirements Definition Process, and 2) Requirements Analysis Process (ISO/IEC 15288), or System Requirements Analysis Process (ISO/IEC 12207). These two processes result in a baseline set of requirements which flow into the architectural design process where the requirements are allocated, decomposed and traced to system elements. The architectural design process also includes allocation of requirements that initiates 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 TR 24748-1, *Systems and software engineering — Life cycle management — Part 1:Guide for life cycle management*. The architectural design process includes allocation and decomposition of requirements that triggers the recursive application of the requirements processes, for the definition of system element requirements and the iterative application of the requirements analysis process for derived requirements. There are also other technical and project processes that have requirements-related activities or tasks.

There are minor differences between the system and software activities. For the purposes of this International Standard, the clauses are generally titled after the systems engineering processes in ISO/IEC 15288.

The relationships between the processes in the two standards are shown in Annex C along with the mapping to the relevant clause in this International Standard. Tables C-1 and C-2 identify the two principal technical processes and their applicable activities in requirements engineering. Table C-3 identifies other technical activities related to requirements engineering.

6.2 Stakeholder requirements definition process

6.2.1 Purpose

The purpose of the Stakeholder Requirements Definition Process is to define the requirements for a system that can provide the services 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, expectations, and desires. It analyzes and transforms these 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 service is validated.

6.2.2 Outcomes

As a result of the successful implementation of the Stakeholder Requirements Definition Process:

a) The required system characteristics and context of use of the product functions and services, and operational concepts are specified.

b) The constraints on a system solution are defined.

c) Traceability of stakeholder requirements to stakeholders and their needs is achieved.

d) The stakeholder requirements are defined.

e) Stakeholder requirements for validation are identified.

6.2.3 Activities and tasks

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

6.2.3.1 Elicit stakeholder requirements.

This activity consists of the following tasks:

> 1) Identify the individual stakeholders or stakeholder classes who have a legitimate interest in the system throughout its life cycle.
>
> NOTE This includes, but is not limited to, users, operators, supporters, developers, producers, trainers, maintainers, disposers, acquirer and supplier organizations, parties responsible for external interfacing entities or enabling systems, regulatory bodies and members of society. Where direct communication is not practicable (e.g., for consumer products and services), representatives or designated proxy stakeholders are selected.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 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 will 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 to be solved'.

> 2) Elicit stakeholder requirements from the identified stakeholders.
>
> NOTE Stakeholder requirements describe the needs, wants, desires, expectations and perceived constraints of identified stakeholders. They are expressed in terms of a model that may be textual or formal, that concentrates on system purpose and behaviour, and that is described in the context of the operational environment and conditions. A product quality model and quality requirements, such as found in ISO/IEC 9126-1 and ISO/IEC 25030, may be useful for aiding this activity. Stakeholder requirements include the needs and requirements imposed by society, the constraints imposed by an acquiring organization and the capabilities and operational characteristics of users and operator staff. It is useful to cite sources, including solicitation documents or agreements, and, where possible, their justification and rationale, and the assumptions of stakeholders and the value they place on the satisfaction of their requirements. For key stakeholder needs, the measures of effectiveness are defined so that operational performance can be measured and assessed. If significant risks are likely to arise from issues (i.e. needs, wants, constraints, limits, concerns, barriers, factors or considerations) relating to people (users and other stakeholders) and their involvement in or interaction with a system at any time in the life cycle of that system, recommendations for identifying and treating human-system issues can be found in ISO PAS 18152, *A specification for the process assessment of human-system issues*.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 a) 2)]*

NOTE 1 Subclause 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. Annex A contains the essential elements for the System Operational Concept and Annex B contains this information for the Concept of Operations.

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.

In most systems, there will be many sources of requirements and it is essential that all potential sources are identified and evaluated for their impact on the system. Some of the common sources of requirements 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.

- Operational environment and context of use – Requirements will be 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 3 See ISO/IEC TR 29138-1:2009, *Information technology — Accessibility considerations for people with disabilities — Part 1: User needs summary*, for additional information on accessibility.

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, *Information technology — Tools and methods of requirements engineering and management for product lines* for additional guidance on requirements reuse.

Requirements elicitation is an iterative activity. Consider several different techniques for identifying requirements during the elicitation task 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
- Organizational analysis techniques (e.g., Strength – Weakness – Opportunity - Threat analysis, product portfolio)

System stakeholders will 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 regulatory 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 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 subclause 5.2.

6.2.3.2 Define stakeholder requirements.

This activity consists of the following tasks:

1) Define the constraints on a system solution that are unavoidable consequences of existing agreements, management decisions and technical decisions.

NOTE These may result from 1) instances or areas of stakeholder-defined solution 2) implementation decisions made at higher levels of system hierarchical structure 3) required use of defined enabling systems, resources and staff.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 b) 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, and 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).

> 2) Define a representative set of activity sequences to identify all required services that correspond to anticipated operational and support scenarios and environments.
>
> NOTE Scenarios are used to analyze the operation of the system in its intended environment in order to identify requirements that may not have been formally specified by any of the stakeholders, e.g., legal, regulatory and social obligations. The context of use of the system is identified and analyzed. Include in the context analysis the activities that users perform to achieve system objectives, the relevant characteristics of the end-users of the system (e.g., expected training, degree of fatigue), the physical environment (e.g., available light, temperature) and any equipment to be used (e.g., protective or communication equipment). The social and organizational influences on users that could affect system use or constrain its design are analyzed when applicable.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 b) 2)]*

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 will often be necessary to address the full range of stakeholders, including the acquirer, user, and supplier.

3) Identify the interaction between users and the system.

NOTE Usability requirements are determined, establishing, as a minimum, the most effective, efficient, and reliable human performance and human-system interaction. The interaction should take into account human capabilities and skills limitations. Where possible, applicable standards, e.g., ISO 9241, and accepted professional practices are used in order to define:

i) Physical, mental, and learned capabilities;

ii) Work place, environment and facilities, including other equipment in the context of use;

iii) Normal, unusual, and emergency conditions;

iv) Operator and user recruitment, training and culture;

If usability is important, usability requirements should be planned, specified, and implemented through the life cycle processes, and the following standards or technical reports may be applicable:

- ISO 9241-11:1998, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability*;
- ISO 13407:1999, Ergonomics — *Ergonomics of human-system interaction — Human-centred design process for interactive systems.*

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 b) 3)]

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 which 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-centred 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 only 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 ISO 13407:1999 has been replaced by ISO 9241-210, *Ergonomics of human-system interaction - part 210: Human-centred design for interactive systems.*

4) Specify health, safety, security, environment and other stakeholder requirements and functions that relate to critical qualities.

NOTE Identify safety risk and, if warranted, specify requirements and functions to provide safety. This includes risks associated with methods of operations and support, health and safety, threats to property and environmental influences. Use applicable standards, e.g., IEC 61508, and accepted professional practices. Identify security risk and, if warranted, specify all applicable areas of system security, including physical, procedural, communications, computers, programs, data and emissions. Identify functions that could impact the security of the system, including access and damage to protected personnel, properties and information, compromise of sensitive information, and denial of approved access to property and information. Specify the required security functions, including mitigation and containment, referencing applicable standards and accepted professional practices where mandatory or relevant.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 b) 4)]

Critical qualities are aspects of the system that are essential to ensure the integrity of the system and its operating environment.

6.2.3.3 Analyze and maintain stakeholder requirements.

This activity consists of the following tasks:

1) Analyze the complete set of elicited requirements.

NOTE Analysis includes identifying and prioritizing the conflicting, missing, incomplete, ambiguous, inconsistent, incongruous or unverifiable requirements.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 c) 1)]

Requirements should be analyzed for the characteristics defined in subclauses 5.2.5 and 5.2.6. Requirements should be prioritized and may be classified as described in subclause 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, then 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.

2) Resolve requirements problems.

NOTE This includes requirements that cannot be realized or are impractical to achieve.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 c) 2)]

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.

3) Feed back the analyzed requirements to applicable stakeholders to ensure that the needs and expectations have been adequately captured and expressed.

NOTE Explain and obtain agreement to the proposals to resolve conflicting, impractical and unrealisable stakeholder requirements.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 c) 3)]

4) Establish with stakeholders that their requirements are expressed correctly.

NOTE This includes confirming that stakeholder requirements are comprehensible to originators and that the resolution of conflict in the requirements has not corrupted or compromised stakeholder intentions.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 c) 4)]

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 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.

Conducting requirements reviews is perhaps the most common means of both verification and validation of the requirements document(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 abstraction 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, *Standard for Software Reviews and Audits*.

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

> 5) Record the stakeholder requirements in a form suitable for requirements management through the life cycle and beyond.
>
> NOTE These records establish the stakeholder requirements baseline, and retain changes of need and their origin throughout the system life cycle. They are the basis for traceability to the system requirements and form a source of knowledge for requirements for subsequent system entities.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 c) 5)]*

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 Additional information and guidelines on requirements management tools can be found in ISO/IEC TR 24766:2009 – *Guide for requirement engineering tool capabilities.*

The requirements repository should first be populated with the source documentation of the stakeholder needs, project constraints (such as from business policies/rules or regulatory requirements), 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 needs to be captured.

Requirements documents that may be output as part of the Stakeholder Requirements Definition process include:

- Stakeholder Requirements Specification
- Concept of Operations
- System Operational Concept

Additional information on these requirements-related documents can be found in Clauses 7 through 9 and Annex A and B.

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 subclause 5.2.8.

> 6) Maintain stakeholder requirements traceability to the sources of stakeholder need.
>
> NOTE The stakeholder requirements are reviewed at key decision times in the life cycle to ensure that account is taken of any changes of need.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.1.3 c) 6)]*

Initial 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 subclause 6.3.3.2 of this International Standard under task 3.

NOTE 1 Additional guidance on placing information under configuration control can be found in ISO/IEC 15288, subclause 6.3.5, and in subclause 6.5.2.1 of this International Standard.

NOTE 2 Subclause 5.2.5 describes requirements traceability as it pertains to requirements engineering.

6.3 Requirements analysis process

6.3.1 Purpose

The purpose of the Requirements Analysis Process is to transform the stakeholder, requirement-driven view of desired services into a technical view of a required product that could deliver those services.

This process builds a representation of a future system that will meet stakeholder requirements and that, as far as constraints permit, does not imply any specific implementation. It results in measurable system requirements that specify, from the supplier's perspective, what characteristics it is to possess and with what magnitude in order to satisfy stakeholder requirements.

6.3.2 Outcomes

As a result of the successful implementation of the Requirements Analysis Process:

a) The required characteristics, attributes, and functional and performance requirements for a product solution are specified.

b) Constraints that will affect the architectural design of a system and the means to realize it are specified.

c) The integrity and traceability of system requirements to stakeholder requirements is achieved.

d) A basis for verifying that the system requirements are satisfied is defined.

6.3.3 Activities and tasks

The project shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Requirements Analysis Process.

6.3.3.1 Define system requirements.

This activity consists of the following tasks:

1) Define the functional boundary of the system in terms of the behavior and properties to be provided.

NOTE This includes the system's stimuli and its responses to user and environment behavior, and an analysis and description of the required interactions between the system and its operational environment in terms of interface constraints, such as mechanical, electrical, mass, thermal, data, and procedural flows. This establishes the expected system behavior, expressed in quantitative terms, at its boundary.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.2.3 a) 1)]

Scope problems can be minimized by establishing boundary conditions for the system (or service) with the stakeholders before defining the system requirements. Three factors which affect the boundary conditions are:

— Organization – stakeholders should have an understanding of the organization in which the targeted system 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, and operational.

2) Define each function that the system is required to perform.

NOTE 1 This includes how well the system, including its operators, is required to perform that function, the conditions under which the system is to be capable of performing the function, the conditions under which the system is to commence performing that function and the conditions under which the system is to cease performing that function.

NOTE 2 Conditions for the performance of functions may incorporate reference to required states and modes of operation of the system. System requirements depend heavily on abstract representations of proposed system characteristics and may employ multiple modeling techniques and perspectives to give a sufficiently complete description of the desired system requirements.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.2.3 a) 2)]

As a better understanding is gained of the interactions and interfaces among the various functions and elements of the system, requirements are generated through combinations of performance and effectiveness analyses, trade studies, design development, interface definitions, and cost/benefit assessments.

Again, for the system, 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, *Software and systems engineering — Tools and methods of requirements engineering and management for product lines* for additional guidance on requirements reuse.

3) Define necessary implementation constraints that are introduced by stakeholder requirements or are unavoidable solution limitations.

NOTE This includes the implementation decisions that are allocated from design at higher levels in the structure of the system.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.2.3 a) 3)]

It is important to validate constraints with stakeholders and ensure they are fully understood and correct before evolving a set of system requirements and the architectural design. 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.

4) Define technical and quality in use measures that enable the assessment of technical achievement.

NOTE This includes defining critical performance parameters associated with each effectiveness measure identified in the stakeholder requirements. The critical performance measures are analyzed and reviewed to ensure stakeholder requirements are met and to ensure identification of project cost, schedule or performance risk associated with any non-compliance. ISO/IEC 15939 provides a process to identify, define and use appropriate measures. The ISO/IEC 9126 series of standards provides relevant quality measures.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.2.3 a) 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 26702 and ISO/IEC TR 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.

5) Specify system requirements and functions, as justified by risk identification or criticality of the system, that relate to critical qualities, such as health, safety, security, reliability, availability and supportability.

NOTE This includes analysis and definition of safety considerations, including those relating to methods of operation and maintenance, environmental influences and personnel injury. It also includes each safety related function and its associated safety integrity, expressed in terms of the necessary risk reduction, is specified and allocated to designated safety-related systems. Applicable standards are used concerning functional safety, e.g., IEC 61508, and environmental protection, e.g., ISO 14001. Analyze security considerations including those related to compromise and protection of sensitive information, data and material. The security-related risks are defined, including, but not limited to, administrative, personnel, physical, computer, communication, network, emission and environment factors using, as appropriate, applicable security standards.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.2.3 a) 5)]

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 requirements, including derived requirements, are intended to meet the stakeholder requirements and objectives and achieve stakeholder agreement.

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 Requirements Analysis Process may include:

- System Requirements Specification
- Software Requirements Specification

Subclauses 9.4 and 9.5 contain detailed specification content for the system and software requirements specifications.

The benefits of documenting both 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.

6.3.3.2 Analyze and maintain system requirements.

This activity consists of the following tasks:

1) Analyze the integrity of the system requirements to ensure that each requirement, pairs of requirements or sets of requirements possess overall integrity.

NOTE Each system requirement statement is checked to establish that it is unique, complete, unambiguous, consistent with all other requirements, implementable and verifiable. Deficiencies, conflicts and weaknesses are identified and resolved within the complete set of system requirements. The resulting system requirements are analyzed to confirm that they are complete, consistent, achievable (given current technologies or knowledge of technological advances) and expressed at an appropriate level of detail. Refer to *ISO/IEC 26702:2007, IEEE Standard for Application and Management of the Systems Engineering Process* for additional detailed guidance regarding the attributes and qualities of good requirements.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.2.3 b) 1)]

Again, for system 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 subclauses 5.2.5 and 5.2.6.

If system requirements from existing or legacy systems have been identified as candidates for reuse, then 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 See ISO/IEC 26551, *Software and systems engineering — Tools and methods of requirements engineering and management for product lines* for additional guidance on requirements reuse.

The classifications in subclause 5.2.8 may help with this task. The 'plan verification' process of ISO/IEC 15288, should be used for the definition, planning, and execution of requirements verification (ISO/IEC 15288:2008 (IEEE Std 15288-2008), subclause 6.4.6.3 a).

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.

> 2) Feed back the analyzed requirements to applicable stakeholders to ensure that the specified system requirements adequately reflect the stakeholder requirements to address the needs and expectations.
>
> NOTE Confirmation is made that they are a necessary and sufficient response to stakeholder requirements and a necessary and sufficient input to other processes, in particular architectural design.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.2.3 b) 2)]*

Requirements validation ensures that stakeholder requirements have been correctly transformed into system requirements. Various techniques may be used, including stakeholder reviews, prototyping, modelling and simulation, conceptual modelling, and formal modelling. The appropriate technique may vary based on the characteristics of the stakeholders, so multiple techniques may need to be employed to ensure accounting for all stakeholders. Reviews are discussed in subclause 6.2.3.3 of this International Standard under task 4.

Stakeholder reviews are a common technique to validate requirements that can be implemented easily. The stakeholder reviews involve conducting a 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 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.

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 may 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.

3) Demonstrate traceability between the system requirements and the stakeholder requirements.

NOTE Maintain mutual traceability between the system requirements and the stakeholder requirements, i.e. all achievable stakeholder requirements are met by one or more system requirements, and all system requirements meet or contribute to meeting at least one stakeholder requirement. The system requirements are held in an appropriate data repository that permits traceability to stakeholder needs and architectural design.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.2.3 b) 3)]

Requirements tracing is concerned with recovering the source of requirements and predicting the effects of requirements. The traceability should include interface requirements. Tracing is fundamental to performing coverage analysis (to 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)

- to architectural design (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.

A 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.

4) Maintain throughout the system life cycle the set of system requirements together with the associated rationale, decisions and assumptions.

[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.2.3 b) 4)]

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

6.4 Requirements engineering activities in other technical processes

6.4.1 Requirements in architectural design

The purpose of the Architectural Design Process is to synthesize a solution that satisfies system requirements.

6.4.1.1 Define the architecture

This activity consists of the following tasks:

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

> 2) Partition the system functions identified in requirements analysis and allocate them to elements of system architecture. Generate derived requirements as needed for the allocations.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.3.3 a) 2)]*

An architectural design solution 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 architectural design, including the system elements and interfaces. Verification and validation criteria for the system elements should be identified and recorded as derived requirements are generated.

> 3) Define and document the interfaces between system elements and at the system boundary with external systems.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.3.3 a) 3)]*

Interface requirements should be identified through appropriate interface specifications such as an Interface Control Document. The interface requirements are incorporated into the architectural design solution. These documents are shared by the programs involved with the interactions between systems.

6.4.1.2 Analyze and evaluate the architecture

This activity consists of the following tasks:

NOTE Tasks 1), 3), and 4) under this activity are not included, as there is no specific guidance related to requirements engineering.

> 2) Determine which system requirements are allocated to operators.
>
> NOTE 1 This determination takes account of the context of use factors and considers, as a minimum, the following factors for the most effective, efficient and reliable human-machine interaction:
>
> i) Limitations of human capabilities;
>
> ii) Human actions critical to safety and how the consequences of error are addressed;
>
> iii) Integration of human performance into systems and their operation.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.3.3 b) 2)]*

NOTE Additional guidance on architectural design can be found in ISO/IEC 15288:2008 (IEEE Std 15288-2008), subclause 6.4.3, Architectural design process.

6.4.2 Requirements in verification

The purpose of the Verification Process is to confirm that the specified design requirements are fulfilled by the system.

NOTE Additional guidance on verification can be found in ISO/IEC 15288:2008 (IEEE Std 15288-2008), subclause 6.4.6, Verification process.

6.4.2.1 Plan verification

This activity consists of the following tasks:

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

> 2) Define a verification plan based on system requirements.
>
> NOTE The plans account for the sequence of configurations defined in the integration strategy and, where appropriate, take account of disassembly strategies for fault diagnosis. The schedule typically defines risk-managed verification steps that progressively build confidence in compliance of the fully configured product.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.6.3 a) 2)]*

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 will 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 will be applied (see list below)
- Who – Identify the organization/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)/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 then the similar system element.

> Good practice: Identify the generic name of the analysis (like Failure Modes Effects Analysis), analytical/computer tools, and/or numeric methods; the source of input data; and how raw data will be analyzed. Ensure agreement 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 will be for the purpose of collecting the evidence of success, what general steps will 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 will 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 will be followed, specific data to be collected, criteria for repeatability of collected data, and methods for analyzing the results.

NOTE Certification is often included as a fifth method of verification. 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 an updated Requirements Traceability Matrix (RTM).

6.4.2.2 Perform verification

This activity consists of the following tasks:

NOTE Tasks 1), 3), and 4) under this activity are not included, as there is no specific guidance related to requirements engineering.

> 2) Conduct verification to demonstrate compliance to the specified design requirements.
>
> NOTE Non-compliance identifies the existence of random faults and/or design errors, and corrective actions are initiated as appropriate. Verification is undertaken in a manner, consistent with organizational constraints, such that uncertainty in the replication of verification actions, conditions and outcomes is minimized. Approved records of verification actions and outcomes are made.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.6.3 b) 2)]*

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 will 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.

6.4.3 Requirements in validation

The purpose of the Validation Process is to provide objective evidence that the services provided by a system when in use comply with stakeholders' requirements, achieving its intended use in its intended operational environment.

6.4.3.1 Plan validation

This activity consists of the following task:

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

> 2) Prepare a validation plan.
>
> NOTE Validation is based on the stakeholder requirements. Where appropriate, define validation steps, e.g., various operational states, scenarios and missions that progressively build confidence in the conformance of the installed system and assist diagnosis of any discrepancies. Methods and techniques needed to implement the validation strategy are specified, as are the purpose, conditions, and conformance criteria for each validation. Where stakeholder requirements cannot be specified comprehensively or change frequently, repeated validation of (often rapidly developed) increments in system evolution may be employed to refine stakeholder requirements and mitigate risks in the correct identification of need, e.g., ISO 13407 describes an iterative life cycle that involves users.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.8.3 a) 2)]*

The system operational concept and baselined stakeholder requirements are inputs to the validation plan activity.

6.4.3.2 Perform validation

This activity consists of the following task:

NOTE Tasks 1), 3), 4), and 5) under this activity are not included, as there is no specific guidance related to requirements engineering.

> 2) Conduct validation to demonstrate conformance of services to stakeholder requirements.
>
> NOTE Validation is undertaken in a manner, consistent with organizational constraints, such that uncertainty in the replication of validation actions, conditions and outcomes is minimized. Objectively record and approve validation actions and results. Validation may also be conducted to confirm that the system not only satisfies all operational, functional and usability requirements, but also satisfies the often less formally expressed, but sometimes overriding, attitudes, experience and subjective tests that comprise customer satisfaction.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.4.8.3 b) 2)]*

Requirements validation is subject to approval by the project authority and key stakeholders. This process is invoked during the Stakeholders Requirements Definition Process to confirm that the requirements properly reflect the stakeholder needs and to establish validation criteria, i.e. that we have the right requirements. System validation confirms that the system, as built, satisfies stakeholder stated needs and requirements, that it is the right system. Requirements traceability should be maintained, and may be documented in a Requirements Traceability Matrix (RTM).

NOTE 1 Additional guidance on validation can be found in ISO/IEC 15288:2008 (IEEE Std 15288-2008), subclause 6.4.8, Validation process.

NOTE 2 Additional guidance on validating usability requirements can be found in ISO/IEC 25060 and ISO/IEC 25062.

6.5 Requirements management

6.5.1 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 15288 and ISO/IEC 12207.

Requirements are rarely static. Although from the development management perspective, it is desirable to freeze a set of requirements permanently, it is rarely possible. 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 be evaluated to 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.

6.5.2 Change management

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.

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 corresponds to the reviewed system requirements, including the external interface descriptions.

- Allocated baseline corresponds to the reviewed system element requirements specifications including the interface requirements specification.

- 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 completed system.

6.5.2.1 Configuration management

The purpose of the Configuration Management Process is to establish and maintain the integrity of all identified outputs of a project or process and make them available to concerned parties.

6.5.2.1.1 Plan configuration management

This activity consists of the following task:

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

> 2) Identify items that are subject to configuration control.
>
> NOTE Items are distinguished by unique, durable identifiers or markings, where appropriate. The identifiers are in accordance with relevant standards and product sector conventions, such that the items under configuration control are unambiguously traceable to their specifications or equivalent, documented descriptions.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.3.5.3 a) 2)]*

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

6.5.2.1.2 Perform configuration management

This activity consists of the following task:

> 1) Maintain information on configurations with an appropriate level of integrity and security.
>
> NOTE This includes taking into account the nature of the items under configuration control. Configuration descriptions conform, where possible, to product or technology standards. Ensure that configuration information permits forward and backward traceability to other baselined configuration states. Consolidate the evolving configuration states of configuration items to form documented baselines at designated times or under defined circumstances. Record the rationale for the baseline and associated authorizations in configuration baseline data. Maintain configuration records through the system life cycle and archive them according to agreements, relevant legislation or best industry practice.
>
> 2) Ensure that changes to configuration baselines are properly identified, recorded, evaluated, approved, incorporated, and verified.
>
> NOTE Consolidate the evolving configuration states of configuration items to form documented baselines at designated times or under defined circumstances. Record the steps of configuration, the rationale for the baseline and associated authorizations in configuration baseline data. Maintain configuration records through the system life cycle and archive them according to agreements, relevant legislation or best industry practice. Manage the recording, retrieval and consolidation of the current configuration status and the status of all preceding configurations to confirm information correctness, timeliness, integrity and security. Perform audits to verify conformance of a baseline to drawings, interface control documents and other agreement requirements.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.3.5.3 b)]*

As changes are made to the operational concepts and stakeholder, system 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, and may be documented in a Requirements Traceability Matrix (RTM).

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

NOTE Subclause 6.3.5 of both ISO/IEC 15288 and ISO/IEC 12207 has additional information on configuration management.

6.5.2.2 Information management

The purpose of the Information Management Process is to provide relevant, timely, complete, valid and, if required, confidential information to designated parties during and, as appropriate, after the system life cycle.

6.5.2.2.1 Plan information management

This activity consists of the following task:

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

> 1) Define the items of information that will be managed during the system life cycle and, according to organizational policy, agreements, or legislation, maintained for a defined period beyond.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.3.6.3 a) 1)]*

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.

6.5.2.2.2 Perform information management

This activity consists of the following task:

NOTE Tasks 2) through 6) under this activity are not included, as there is no specific guidance related to requirements engineering.

> 1) Obtain the identified items of information.
>
> NOTE This may include generating the information or collecting it from appropriate sources.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 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 Subclause 6.3.6 of both ISO/IEC 15288:2008 (IEEE Std 15288-2008) and ISO/IEC 12207:2008 (IEEE Std 12207-2008) has additional detail on information management.

6.5.3 Measurement for requirements

The purpose of the Measurement Process is to collect, analyze, and report data relating to the products developed and processes implemented within the organization, to support effective management of the processes, and to objectively demonstrate the quality of the products.

6.5.3.1 Plan measurement

This activity consists of the following task:

NOTE Tasks 5) through 7) under this activity are not included, as there is no specific guidance related to requirements engineering.

> 1) Describe the characteristics of the organization that are relevant to measurement.
>
> 2) Identify and prioritize the information needs.
>
> 3) Select and document measures that satisfy the information needs.
>
> 4) Define data collection, analysis, and reporting procedures.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.3.7.3 a) 1) through 4)]*

Requirements engineering as a discipline benefits from measuring requirements in both the process and 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 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
- 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 1 ISO/IEC 14143-1 provides details of FSM concepts and their uses.

NOTE 2 Subclause 6.3.7 of both ISO/IEC 15288:2008 (IEEE Std 15288-2008) and ISO/IEC 12207:2008 (IEEE Std 12207-2008) provides additional information on measurement process, as does ISO/IEC 15939.

6.5.3.2 Perform measurement

This activity consists of the following task:

> 1) Integrate procedures for data generation, collection, analysis and reporting into the relevant processes.
>
> 2) Collect, store, and verify data.
>
> 3) Analyze data and develop information products.
>
> 4) Document and communicate results to the measurement users.
>
> *[ISO/IEC 15288:2008 (IEEE Std 15288-2008), 6.3.7.3 b)]*

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:

- Stakeholder requirements specification document (StRS)

- System requirements specification document (SyRS)

- Software requirements specification document (SRS), if adhering to ISO/IEC 12207.

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

NOTE 1 Multiple specification documents for each of the three document types may be produced in the project as discussed in subclause 5.4. For example the SyRS may be produced for systems and system elements.

NOTE 2 The three specification documents, StRS, SyRS and SRS may contain similar information items that could be considered as different views for the same product. For ease of use, this standard presents typical contents of the three forms of the specification separately.

NOTE 3 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 12207:2008 (IEEE Std 12207-2008) and ISO/IEC 15288:2008 (IEEE Std 15288-2008), and the Documentation Management process of ISO/IEC 12207:2008 (IEEE Std 12207-2008), including the knowledge management activities of ISO/IEC 15288:2008 (IEEE Std 15288-2008), subclause 6.2.4.

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 formats in the form of an outline for the resulting documents.

8.2 Stakeholder requirements specification document

8.2.1 Introduction

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 stakeholder perspective including needs of users/operators/maintainers as derived from the context of use. In a business environment, the StRS 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.

The information items 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 business, organizational, and user (stakeholder) requirements in the system requirements specification. This International Standard includes these requirements in the StRS since the contents should be specified from the stakeholders' perspective. They may 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 International Standard may replace 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 will 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 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 a StRS created in an organizational/business context is shown in Figure 6.

1. Introduction

- 1.1 Business purpose
- 1.2 Business scope
- 1.3 Business overview
- 1.4 Definitions
- 1.5 Stakeholders

2. References

3. Business management requirements

- 3.1 Business environment
- 3.2 Goal and objective
- 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. User requirements

6. Concept of proposed system

- 6.1 Operational concept
- 6.2 Operational scenario

7 Project Constraints

8. Appendix

- 8.1 Acronyms and abbreviations

Figure 6 — Example StRS Outline

8.3 System requirements specification document

8.3.1 Introduction

The System Requirements Specification (SyRS) identifies the technical specifications 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. A 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 International Standard 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 ensure 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.

This International Standard also makes a clear distinction between the system requirements (what the system shall do) contained in the SyRS and process requirements (how to construct the system) that should be contained in contract documents such as a Statement of Work.

The SyRS presents the results of the definition of need, the system operational concept, 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.

8.3.2 SyRS example outline

The specific requirements section of a 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 a 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 System interface
- 3.5 System operations
- 3.6 System modes and states
- 3.7 Physical characteristics
- 3.8 Environmental conditions
- 3.9 System security
- 3.10 Information management
- 3.11 Policies and regulations
- 3.12 System life cycle sustainment
- 3.13Packaging, handling, shipping and transportation

4. Verification
- (parallel to subsections in Section 3)

5. Appendices
- Assumptions and dependencies
- Acronyms and abbreviations

Figure 7 — Example SyRS Outline

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

8.4 Software requirements specification document

8.4.1 Introduction

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 will be a requirement specification that will state the interfaces between the system and its software portion, and will place 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

8.4.2 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. Specific requirements
- 3.1 External interfaces
- 3.2 Functions
- 3.3 Usability Requirements
- 3.4 Performance 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 Verification, Section 4 in Figure 8, is not included in the original IEEE Std. 830.

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, 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 paychecks, 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 may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.

9 Information item content

9.1 Introduction

This clause states the normative content of the required information items. The content for the software requirements specification document, as stated in subclause 9.5, is only applicable if adhering to System Requirements Analysis Process of ISO/IEC 12207.

NOTE Information content in subclause 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

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 subclauses 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

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;

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 may be provided by reference to one or more appendixes in the documents or by reference to other documents.

9.3 Stakeholder requirements specification (StRS) document

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

9.3.1 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.

9.3.2 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 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.

9.3.3 Business overview

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

9.3.4 Stakeholders

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

9.3.5 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.

9.3.6 Goal and Objective

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

9.3.7 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.

9.3.8 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.

9.3.9 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 a hierarchical structure with decomposition and classification. Each business process should be uniquely named and numbered in the hierarchy. The description of the individual business process should be represented as a diagram representing a sequence of activities.

9.3.10 Business operational policies and rules

Describe logical propositions applied in conducting the business processes. The propositions will 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.

9.3.11 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'.

9.3.12 Business operation 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.

9.3.13 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.

9.3.14 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 alight the system functions to these structures, and to support future structural changes.

9.3.15 User requirements

User requirements (within the context of the StRS) include required inputs/selections/information observations which users/operators/maintainers need to perform through the use of the system; any outputs they require from the system in order to perform these tasks; and any applicable conditions or constraints governing their interaction with (i.e., usability of) the system. These requirements are then used to describe the operational scenarios which 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 will 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 See ISO 9241-11 for more information about the context of use and a sample report. See ISO 20282-1:2006 for more information about context of use for everyday products.

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

9.3.16 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

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 Operational scenarios

Describe examples of how users/operators/maintainers will interact with the system (context 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 subclause 9.3.9.

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

9.3.18 Project constraints

Describe constraints to performing the project within cost and schedule.

9.4 System requirements specification (SyRS) document

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 document. Organization of the content in the document such as the order and section structure may be selected in accordance with the project's documentation policies.

9.4.1 System purpose

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

9.4.2 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.

9.4.3 System overview

9.4.3.1 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.

9.4.3.2 System functions

Describe major system capabilities, conditions and constraints.

9.4.3.3 User characteristics

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

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

9.4.4 Functional requirements

Define functional requirements applicable to system operation.

9.4.5 Usability requirements

Define usability (quality in use) requirements. Usability requirements and objectives for the system include measurable effectiveness, efficiency, and satisfaction criteria in specific contexts of use.

9.4.6 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.

9.4.7 System interfaces

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.

9.4.8 System Operations

9.4.8.1 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 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 18529, *Ergonomics — Ergonomics of human-system interaction — Human-centred lifecycle process descriptions*, contains a formalized model that can be used in the specification, assessment and improvement of the human-centered processes in system development and operation.

9.4.8.2 Maintainability

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

9.4.8.3 Reliability

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.

9.4.9 System modes and states

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

NOTE The mode of a system refers to a collection of states.

9.4.10 Physical characteristics

9.4.10.1 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.

9.4.10.2 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.

9.4.11 Environmental conditions

Include environmental conditions to be encountered by the system. The following areas should be addressed: natural environment (e.g., wind, rain, temperature, flora, fauna, fungus, mold, sand, salt spray, dust, radiation, chemical, 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 environment.

9.4.12 System security

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.

9.4.13 Information management

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.

9.4.14 Policies and regulations

Detail any relevant organizational policies that will affect the operation or performance of the system as well as any relevant external regulatory requirements, or constraints imposed by normal business practices. Examples of requirements include multilingual support, labour policies, protection of personnel information, and reports to a regulatory agency.

Specify 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.

9.4.15 System life cycle sustainment

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.

9.4.16 Packaging, handling, shipping and transportation

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

9.4.17 Verification

Provide the verification approaches and methods planned to qualify the system or system element. The information items for verification are recommended to be given in a parallel manner with the information items in subclause 9.4.4 to 9.4.16.

9.4.18 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.

9.5 Software requirements specification (SRS) document

This subclause 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 document. Organization of the information items in the document such as the order and section structure may be selected in accordance with the project's documentation policies.

9.5.1 Purpose

Delineate the purpose of the software to be specified.

9.5.2 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;

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

9.5.3 Product perspective

Define the system's relationship to other related products.

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

A block diagram showing the major elements of the larger system, interconnections, and external interfaces can be helpful.

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.

9.5.3.1 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.

9.5.3.2 User interfaces

Specify the following:

a) *The logical characteristics of each interface between the software product and its users.* This includes those configuration characteristics (e.g., required screen formats, page or window layouts, content of any reports or menus, or availability of programmable function keys) necessary to accomplish the software requirements.

b) *All the aspects of optimizing the interface with the person who uses, maintains, or provides other support to the system.* This may simply comprise a list of do's and don'ts on how the system will appear to the user. One example may be a requirement for the option of long or short error messages. This may also be specified in the Software System Attributes under a section titled Ease of Use.

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

9.5.3.3 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.

9.5.3.4 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;

e) Source.

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.

9.5.3.5 Communications interfaces

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

9.5.3.6 Memory constraints

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

9.5.3.7 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;

d) Backup and recovery operations.

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

9.5.3.8 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.

9.5.4 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.

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.

9.5.5 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 subclause 9.5.9.

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

9.5.6 Limitations

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

a) Regulatory 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

9.5.7 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 then have to change accordingly.

9.5.8 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).

9.5.9 Specific requirements

Specify all of the software requirements to a level of detail sufficient to enable designers to design a software system to satisfy those requirements.

Specify all of the software requirements to a level of detail sufficient to enable testers to test that the software system satisfies those requirements.

At a minimum, 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.

The specific requirements should:

a) Be stated in conformance with all the characteristics described in subclause 5.2 of this International Standard.

b) Be cross-referenced to earlier documents that relate.

c) Be uniquely identifiable.

9.5.10 External interfaces

Define all inputs into and outputs from the software system. The description should complement the interface descriptions in 9.5.3.3.1 through 9.5.3.3.5, 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) Screen formats/organization;

i) Window formats/organization;

j) Data formats;

k) Command formats;

l) Endmessages.

9.5.11 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) Error handling and recovery

d) Effect of parameters

e) Relationship of outputs to inputs, including

   1) Input/output sequences

   2) Formulas for input to output conversion

It may be appropriate to partition the functional requirements into subfunctions or subprocesses. This does not imply that the software design will also be partitioned that way.

9.5.12 Usability requirements

Define usability (quality in use) requirements. Usability requirements and objectives for the software system include measurable effectiveness, efficiency, and satisfaction criteria in specific contexts of use.

9.5.13 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;

c) 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 second.*

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.

9.5.14 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) Data retention requirements.

9.5.15 Design constraints

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

9.5.16 Standards compliance

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

a) Report format;

b) Data naming;

c) Accounting procedures;

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.

9.5.17 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 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 program;

   5) Check data integrity for critical variables;

   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;

  5) Use of a particular operating system.

9.5.18 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 subclause 9.5.10 to 9.5.17.

9.5.19 Supporting information

The SRS should contain additional supporting information including

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;

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 International Standard should consider producing separate System OpsCon and SyRS documents. The OpsCon document can then focus on all necessary requirements engineering tasks 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.

The project shall product the following information items in accordance with the project's policies with respect to the System Operational Concept document. The information items in subclause A.2.5 shall be produced in the course of producing the Stakeholder Requirements Specification document as referred to in subclause 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)

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.

NOTE Extracted from IEEE 1362-1998*: IEEE Guide for Information Technology — System Definition — Concept of Operations (ConOps) Document*, Clause 4. In this International Standard, concept of operation (ConOps) is replaced by operational concept (OpsCon).

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.2.1 Scope

Provide an overview of the OpsCon document and the system to which it applies.

A.2.1.1 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 hierarchical or network manner, the position of this document relative to other OpsCon documents should be described.

A.2.1.2 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.2.1.3 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.2.2 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.2.3 Current system or situation

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 will 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.2.3.1 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.2.3.2 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.2.3.3 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

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, 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 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.2.3.4 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.2.3.5 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.2.3.5.1 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.2.3.5.2 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.2.3.5.3 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.2.3.5.4 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.2.3.6 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.2.4 Justification for and nature of changes

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.2.4.1 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).

   3) If the proposed system implements a new functional capability, explain why this function is necessary.

A.2.4.2 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.2.4.3 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.2.4.4 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