



BSI Standards Publication

Systems and software engineering — Life cycle processes — Requirements engineering

National foreword

This British Standard is the UK implementation of [ISO/IEC/IEEE 29148:2018](#). It supersedes [BS ISO/IEC/IEEE 29148:2011](#), which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/15, Software and systems engineering.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2018
Published by BSI Standards Limited 2018

ISBN 978 0 580 97809 8

ICS 35.080

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2018.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

INTERNATIONAL
STANDARD

**ISO/IEC/
IEEE
29148**

Second edition
2018-11-30

Systems and software engineering — Life cycle processes — Requirements engineering

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



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

Documents
© ISO/IEC 2018
© IEEE 2018



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2018, Published in Switzerland

© IEEE 2018

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from ISO, IEC or IEEE at the respective address below.

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

Institute of Electrical and Electronics Engineers, Inc
3 Park Avenue, New York
NY 10016-5997, USA

stds.ipr@ieee.org
www.ieee.org

Contents

Page

Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms, definitions and abbreviated terms	1
3.1 Terms and definitions	1
3.2 Abbreviated terms	7
4 Conformance	8
4.1 Intended usage	8
4.2 Full conformance	8
4.3 Conformance to processes	8
4.4 Conformance to information item content	8
4.5 Tailored conformance	9
4.5.1 Processes	9
4.5.2 Information items	9
5 Concepts	9
5.1 General	9
5.2 Requirements fundamentals	9
5.2.1 General	9
5.2.2 Stakeholders	10
5.2.3 Transformation of needs into requirements	10
5.2.4 Requirements construct	10
5.2.5 Characteristics of individual requirements	12
5.2.6 Characteristics of a set of requirements	13
5.2.7 Requirement language criteria	14
5.2.8 Requirements attributes	14
5.3 Practical considerations	16
5.3.1 Application of iteration and recursion	16
5.3.2 Iteration and recursion in requirements engineering	17
5.4 Requirement information items	18
6 Processes	20
6.1 Requirement processes	20
6.1.1 Guidelines for processes	20
6.2 Business or mission analysis process	21
6.2.1 Purpose	21
6.2.2 Outcomes	21
6.2.3 Activities and tasks	22
6.3 Stakeholder needs and requirements definition process	25
6.3.1 Purpose	25
6.3.2 Outcomes	26
6.3.3 Activities and tasks	26
6.4 System [System/Software] Requirements definition process	34
6.4.1 Purpose	34
6.4.2 Outcomes	34
6.4.3 Activities and tasks	35
6.5 Requirements engineering activities in other technical processes	41
6.5.1 Requirements activities in architecture definition	41
6.5.2 Requirements activities in verification	42
6.5.3 Requirements activities in validation	44
6.6 Requirements management	45
6.6.1 Management overview	45
6.6.2 Change management	45

6.6.3	Measurement for requirements.....	48
7	Information items.....	49
8	Guidelines for information items.....	50
8.1	Requirements information item outlines.....	50
8.2	Business requirements specification.....	50
8.2.1	General.....	50
8.2.2	BRS example outline.....	51
8.3	Stakeholder requirements specification.....	51
8.3.1	General.....	51
8.3.2	StRS example outline.....	52
8.4	System requirements specification.....	53
8.4.1	General.....	53
8.4.2	SyRS example outline.....	53
8.5	Software requirements specification.....	54
8.5.1	General.....	54
8.5.2	SRS example outline.....	55
9	Information item content.....	56
9.1	General.....	56
9.2	General content.....	56
9.2.1	Identification.....	56
9.2.2	Front matter.....	56
9.2.3	Definitions.....	57
9.2.4	References.....	57
9.2.5	Acronyms and abbreviations.....	57
9.3	Business requirements specification (BRS) content.....	57
9.3.1	BRS overview.....	57
9.3.2	Business purpose.....	57
9.3.3	Business scope.....	57
9.3.4	Business overview.....	58
9.3.5	Major Stakeholders.....	58
9.3.6	Business environment.....	58
9.3.7	Mission, goals and objectives.....	58
9.3.8	Business model.....	58
9.3.9	Information environment.....	58
9.3.10	Business processes.....	58
9.3.11	Business operational policies and rules.....	59
9.3.12	Business operational constraints.....	59
9.3.13	Business operational modes.....	59
9.3.14	Business operational quality.....	59
9.3.15	Business structure.....	59
9.3.16	High-level operational concept.....	59
9.3.17	High-level operational scenarios.....	59
9.3.18	Other high-level life-cycle concepts.....	60
9.3.19	Project constraints.....	60
9.4	Stakeholder requirements specification (StRS) content.....	60
9.4.1	StRS overview.....	60
9.4.2	Stakeholder purpose.....	60
9.4.3	Stakeholder scope.....	60
9.4.4	Overview.....	60
9.4.5	Stakeholders.....	60
9.4.6	Business environment.....	60
9.4.7	Mission, goals and objectives.....	61
9.4.8	Business model.....	61
9.4.9	Information environment.....	61
9.4.10	System processes.....	61
9.4.11	System operational policies and rules.....	61
9.4.12	Operational constraints.....	61

9.4.13	System operational modes and states	61
9.4.14	System operational quality	62
9.4.15	User requirements.....	62
9.4.16	Operational concept.....	62
9.4.17	Operational scenarios.....	62
9.4.18	Other detailed concepts of proposed system.....	62
9.4.19	Project constraints	63
9.5	System requirements specification (SyRS) content.....	63
9.5.1	SyRS overview	63
9.5.2	System purpose.....	63
9.5.3	System scope.....	63
9.5.4	System overview	63
9.5.5	Functional requirements.....	63
9.5.6	Usability requirements.....	63
9.5.7	Performance requirements.....	64
9.5.8	System interface requirements	64
9.5.9	System operations.....	64
9.5.10	System modes and states.....	65
9.5.11	Physical characteristics.....	65
9.5.12	Environmental conditions.....	65
9.5.13	System security requirements.....	65
9.5.14	Information management requirements	65
9.5.15	Policy and regulation requirements.....	66
9.5.16	System life cycle sustainment requirements	66
9.5.17	Packaging, handling, shipping and transportation requirements.....	66
9.5.18	Verification.....	66
9.5.19	Assumptions and dependencies.....	66
9.6	Software requirements specification (SRS) content.....	66
9.6.1	SRS overview.....	66
9.6.2	Purpose.....	66
9.6.3	Scope	66
9.6.4	Product perspective	67
9.6.5	Product functions	68
9.6.6	User characteristics.....	69
9.6.7	Limitations	69
9.6.8	Assumptions and dependencies.....	69
9.6.9	Apportioning of requirements.....	70
9.6.10	Specified requirements	70
9.6.11	External interfaces	70
9.6.12	Functions.....	71
9.6.13	Usability requirements.....	71
9.6.14	Performance requirements.....	71
9.6.15	Logical database requirements	72
9.6.16	Design constraints.....	72
9.6.17	Standards compliance	72
9.6.18	Software system attributes	72
9.6.19	Verification.....	73
9.6.20	Supporting information.....	73
Annex A (normative)	System operational concept.....	74
Annex B (informative)	Concept of operations.....	86
Annex C (normative)	Tailoring policies.....	88
Bibliography		90
IEEE notices and abstract		91

Foreword

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

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

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

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

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

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

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

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

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

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

Introduction

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

This page deliberately left blank

Systems and software engineering — Life cycle processes — Requirements engineering

1 Scope

This document:

- specifies the required processes implemented in the engineering activities that result in requirements for systems and software products (including services) throughout the life cycle;
- provides guidelines for applying the requirements and requirements-related processes described in [ISO/IEC/IEEE 15288](#) and [ISO/IEC/IEEE 12207](#);
- specifies the required information items produced through the implementation of the requirements processes;
- specifies the required contents of the required information items;
- provides guidelines for the format of the required and related information items.

This document is applicable to:

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

2 Normative references

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

[ISO/IEC/IEEE 15288:2015](#), *Systems and software engineering — System life cycle processes*

[ISO/IEC/IEEE 12207:2017](#), *Systems and software engineering — Software life cycle processes*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

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

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

- ISO Online browsing platform: available at <http://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>
- IEEE Standards Dictionary Online: available at <http://dictionary.ieee.org>

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

3.1.1

acquirer

stakeholder ([3.1.28](#)) that acquires or procures a product or service from a *supplier* ([3.1.31](#))

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

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

3.1.2

attribute

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

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

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

3.1.3

baseline

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

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

3.1.4

business requirements specification

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

3.1.5

concept of operations

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

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

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

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

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

3.1.6 condition

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

3.1.7 constraint

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

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

3.1.8 context of use

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

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

3.1.9 customer

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

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

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

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

3.1.10 derived requirement

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

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

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

3.1.11 developer

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

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

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

3.1.12 document

uniquely identified unit of information for human use

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

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

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

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

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

3.1.13

human systems integration

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

[SOURCE: INCOSE SEHbk 4:2015]

3.1.14

information item

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

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

3.1.15

level of abstraction

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

3.1.16

operational concept

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

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

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

3.1.17

operational scenario

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

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

3.1.18

operator

individual or organization that performs the operations of a system

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

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

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

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

3.1.19 requirement

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

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

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

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

3.1.20 requirements elicitation

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

3.1.21 requirements engineering

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

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

3.1.22 requirements management

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

3.1.23 requirements traceability

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

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

3.1.24 requirements traceability matrix

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

3.1.25 requirements validation

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

[SOURCE: EIA 632:1999, modified]

3.1.26 requirements verification

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

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

[SOURCE: EIA 632:1999, modified]

3.1.27

software requirements specification

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

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

3.1.28

stakeholder

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

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

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

3.1.29

stakeholder requirements specification

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

3.1.30

state

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

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

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

3.1.31

supplier

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

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

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

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

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

3.1.32

system-of-interest

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

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

3.1.33

system requirements specification

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

3.1.34

trade-off

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

3.1.35

user

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

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

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

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

3.1.36

validation

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

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

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

3.1.37

verification

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

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

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

3.2 Abbreviated terms

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

StRS	Stakeholder Requirements Specification
SyRS	System Requirements Specification
TBD	To Be Determined
TBR	To be Resolved, To be Revised
TBS	To be Supplied, To be Specified.
TPM	Technical Performance Measure

4 Conformance

4.1 Intended usage

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

4.2 Full conformance

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

- to the provisions contained in [5.2.4](#), [5.2.5](#), [5.2.6](#), and [5.2.7](#);
- to the requirements-engineering-related processes of [ISO/IEC/IEEE 15288](#) and [ISO/IEC/IEEE 12207](#) cited in [6.1](#);
- to the information items specified in [Clause 7](#); and
- to the required content of the information items in [Clause 9](#) and [Annex A](#) of this document.

4.3 Conformance to processes

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

The requirements for processes in this document are contained in [6.1](#).

4.4 Conformance to information item content

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

- the required information items stated in this document are produced; and
- it is demonstrated that the information items produced during the requirements engineering activities conform to the content requirements defined in this document.

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

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

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

4.5 Tailored conformance

4.5.1 Processes

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

4.5.2 Information items

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

5 Concepts

5.1 General

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

5.2 Requirements fundamentals

5.2.1 General

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

- being with reference to a defined system, software or service;
- enabling an agreed understanding between stakeholders (e.g., acquirers, users, customers, operators, suppliers);
- having been validated against real-world needs;
- able to be implemented; and
- providing a reference for verifying designs and solutions.

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

5.2.2 Stakeholders

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

5.2.3 Transformation of needs into requirements

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

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

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

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

NOTE See [ISO/IEC 26551](#) for additional guidance on requirements development techniques, including requirements reuse.

5.2.4 Requirements construct

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

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

- it shall be met or possessed by a system to solve a problem, achieve an objective or address a stakeholder concern;
- it is qualified by measurable conditions;
- it is bounded by constraints;
- it defines the performance of the system when used by a specific stakeholder or the corresponding capability of the system, but not a capability of the user, operator or other stakeholder; and
- it can be verified (e.g., the realization of the requirement in the system can be demonstrated).

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

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

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

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

- Requirements are mandatory binding provisions and use 'shall'.
- Non-requirements, such as descriptive text, use verbs such as 'are', 'is', and 'was'. It is best to avoid using the term 'must', due to potential misinterpretation as a requirement.
- Statements of fact, futurity, or a declaration of purpose are non-mandatory, non-binding provisions and use 'will'. 'Will' can also be used to establish context or limitations of use.
- Preferences or goals are desired, non-mandatory, non-binding provisions and use 'should'. They are not requirements.
- Suggestions or allowances are non-mandatory, non-binding provisions and use 'may'.
- Use positive statements and avoid negative requirements such as 'shall not'.
- Use active voice: avoid using passive voice, such as 'it is required that'.
- Avoid using terms such as 'shall be able to'.

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

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

[Condition] [Subject] [Action] [Object] [Constraint of Action] EXAMPLE: When signal x is received [Condition], the system [Subject] shall set [Action] the signal x received bit [Object] within 2 seconds [Constraint of Action]. Or [Condition] [Subject] [Action] [Object] [Constraint of Action] EXAMPLE: At sea state 1 [Condition], the Radar System [Subject] shall detect [Action] targets [Object] at ranges out to 100 nautical miles [Constraint of Action]. Or [Subject] [Action] [Constraint of Action] EXAMPLE: The Invoice System [Subject] shall display pending customer invoices [Action] in ascending order of invoice due date [Constraint of Action].

Figure 1 — Examples of functional requirements syntax

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

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

Examples of constraints giving rise to requirements include:

- interfaces to already existing systems (e.g., format, protocol or content) where the interface cannot be changed;
- physical size limitations (e.g., a controller shall fit within a limited space in an airplane wing);
- laws of a particular country;
- available duration or budget;
- pre-existing technology platform;
- maintenance constraints; or
- user or operator capabilities and limitations.

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

NOTE 3 [6.3.3](#) and [6.4.3](#) detail the process to define stakeholder and system requirements.

5.2.5 Characteristics of individual requirements

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

- Necessary. The requirement defines an essential capability, characteristic, constraint and/or quality factor. If it is not included in the set of requirements, a deficiency in capability or characteristic will exist, which cannot be fulfilled by implementing other requirements. The requirement is currently applicable and has not been made obsolete by the passage of time. Requirements with planned expiration dates or applicability dates are clearly identified.
- Appropriate. The specific intent and amount of detail of the requirement is appropriate to the level of the entity to which it refers (level of abstraction appropriate to the level of entity). This includes avoiding unnecessary constraints on the architecture or design while allowing implementation independence to the extent possible.

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

- Unambiguous. The requirement is stated in such a way so that it can be interpreted in only one way. The requirement is stated simply and is easy to understand.
- Complete. The requirement sufficiently describes the necessary capability, characteristic, constraint or quality factor to meet the entity need without needing other information to understand the requirement.

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

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

- Feasible. The requirement can be realized within system constraints (e.g., cost, schedule, technical) with acceptable risk.
- Verifiable. The requirement is structured and worded such that its realization can be proven (verified) to the customer's satisfaction at the level the requirements exists. Verifiability is enhanced when the requirement is measurable.
- Correct. The requirement is an accurate representation of the entity need from which it was transformed.
- Conforming. The individual items conform to an approved standard template and style for writing requirements, when applicable.

5.2.6 Characteristics of a set of requirements

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

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

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

- i) include all requirements types relevant to the system under consideration;
- ii) account for requirements in all stages of the life cycle; and
- iii) involve all stakeholders in the requirements elicitation, capture, and analysis activity.

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

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

- Consistent. The set of requirements contains individual requirements that are unique, do not conflict with or overlap with other requirements in the set, and the units and measurement systems are homogeneous. The terminology used within the set of requirements is consistent, i.e. the same term is used throughout the set to mean the same thing.
- Feasible. The complete set of requirements can be realized within entity constraints (e.g., cost, schedule, technical) with acceptable risk.

NOTE 4 Feasible includes the concept of 'affordable'.

- Comprehensible. The set of requirements is written such that it is clear as to what is expected by the entity and its relation to the system of which it is a part.
- Able to be validated. It is practicable that satisfaction of the requirement set will lead to the achievement of the entity needs within constraints (e.g., cost, schedule, technical, legal and regulatory compliance).

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

5.2.7 Requirement language criteria

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

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

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

- superlatives (such as 'best', 'most');
- subjective language (such as 'user friendly', 'easy to use', 'cost effective');
- vague pronouns (such as 'it', 'this', 'that');
- ambiguous terms such as adverbs and adjectives (such as 'almost always', 'significant', 'minimal') and ambiguous logical statements (such as 'or', 'and/or');

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

- open-ended, non-verifiable terms (such as 'provide support', 'but not limited to', 'as a minimum');
- comparative phrases (such as 'better than', 'higher quality');
- loopholes (such as 'if possible', 'as appropriate', 'as applicable');
- terms that imply totality (such as 'all', 'always', 'never', and 'every');

NOTE 2 It is very difficult to verify such requirements.

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

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

5.2.8 Requirements attributes

5.2.8.1 General

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

5.2.8.2 Examples of requirements attributes

Important examples of requirements attributes include:

- Identification. Each requirement should be uniquely identified (i.e., number, name tag, mnemonic). Identification can reflect linkages and relationships, if needed, or they can be separate from identification. Unique identifiers aid in requirements tracing. Once assigned, the identification is unique - it is never changed (even if the identified requirement changes) nor is it reused (even if the identified requirement is deleted).
- Version Number (and indication of the version of the requirement). This is to make sure that the correct version of the requirement is being implemented as well as to provide an indication of the volatility of the requirement. A requirement that has a lot of change could indicate a problem or risk to the project.
- Owner. The person or element of the organization that maintains the requirement, who has the right to say something about this requirement, approves changes to the requirement, and reports the status of the requirement.
- Stakeholder Priority. The priority of each requirement should be identified. This may be established through a consensus process among potential stakeholders. As appropriate, a scale such as 1-5 or a simple scheme such as High, Medium or Low, could be used for identifying the priority of each requirement. The priority is not intended to imply that some requirements are not necessary, but it may indicate what requirements are candidates for the trade space when decisions regarding alternatives are necessary. Prioritization needs to consider the stakeholders who need the requirements. This facilitates trading off requirements and balancing the impact of changes among stakeholders.
- Risk. A risk value assigned to each requirement based on risk factors. Requirements that are at risk include requirements that fail to have the set of characteristics that well-formed requirements should have. Failing to have these characteristics can result in the requirement not being implemented (fails system verification) and entity needs not being realized (fails system validation). Risk can also address feasibility/attainability in terms of technology, schedule, cost, politics, etc. If the technology needed to meet the requirement is new with a low maturity the risk is higher than if using a mature technology used in other similar projects. Risk may also be inherited from a parent requirement.

NOTE Additional guidance on risk factors can be found in [ISO/IEC 16085](#).

- Rationale. The rationale for establishing each requirement should be captured. The rationale provides the reason that the requirement is needed and points to any supporting analysis, trade study, modelling, simulation or other substantive objective evidence.
- Difficulty. The assumed difficulty for each requirement should be noted (e.g., Easy/Nominal/Difficult). This provides additional context in terms of requirements breadth and affordability. It also helps with cost modelling.
- Type. Requirements vary in intent and in the kinds of properties they represent. Use of a type attribute aids in identifying relevant requirements and categorizing requirements into groups for analysis and allocation.

5.2.8.3 Examples of the requirements type attribute

Important examples of the requirements type attribute include:

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

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

- **Interface.** Interface requirements are the definition of how the system is required to interact with external systems (external interface), or how system elements within the system, including human elements, interact with each other (internal interface). External interface requirements state characteristics required of the system, software or service at a point or region of connection of the system, software or service to the world outside of the item. They include, as applicable, characteristics such as location, geometry and what the interface is to be able to pass in each direction.
- **Process Requirements.** These are stakeholder, usually acquirer or user, requirements imposed through the contract or statement of work. Process requirements include: compliance with national, state or local laws, including environmental laws, administrative requirements, acquirer/supplier relationship requirements and specific work directives. Process requirements may also be imposed on a program by corporate policy or practice. System or system element implementation process requirements, such as mandating a particular design method, are usually captured in project agreement documentation such as contracts, statements of work and quality plans.
- **Quality (Non-Functional) Requirements.** – Include a number of the 'ilities' in requirements to include, for example, transportability, survivability, flexibility, portability, reusability, reliability, maintainability and security. The kinds of quality requirements (e.g., "ilities") should be identified prior to initiating the requirements activities. This should be tailored to the system(s) being developed. As appropriate, measures for the quality requirements should be included as well.
- **Usability/Quality-in-Use Requirements** (for user performance and satisfaction) – Provide the basis for the design and evaluation of systems to meet the user needs. Usability/Quality-in-Use requirements are developed in conjunction with, and form part of, the overall requirements specification of a system.

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

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

NOTE 3 Additional guidance on human factors requirements can be found in [6.3.3.4](#).

5.3 Practical considerations

5.3.1 Application of iteration and recursion

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

5.3.1.1 Iterative application of processes

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

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

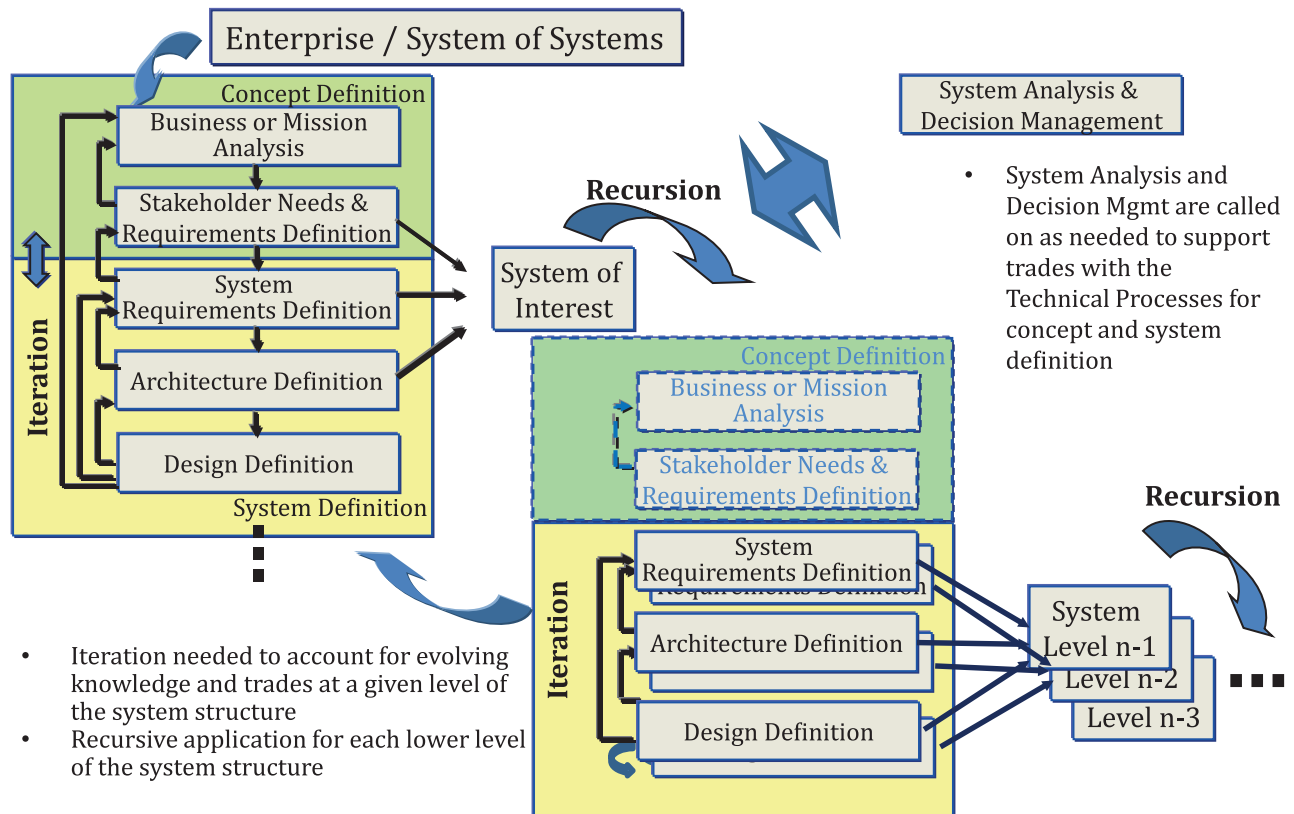


Figure 2 — Iterative and recursive application of processes

5.3.1.2 Recursive application of processes

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

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

5.3.2 Iteration and recursion in requirements engineering

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

The main forms of appropriate iteration within requirements engineering include:

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

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

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

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

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

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

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

5.4 Requirement information items

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

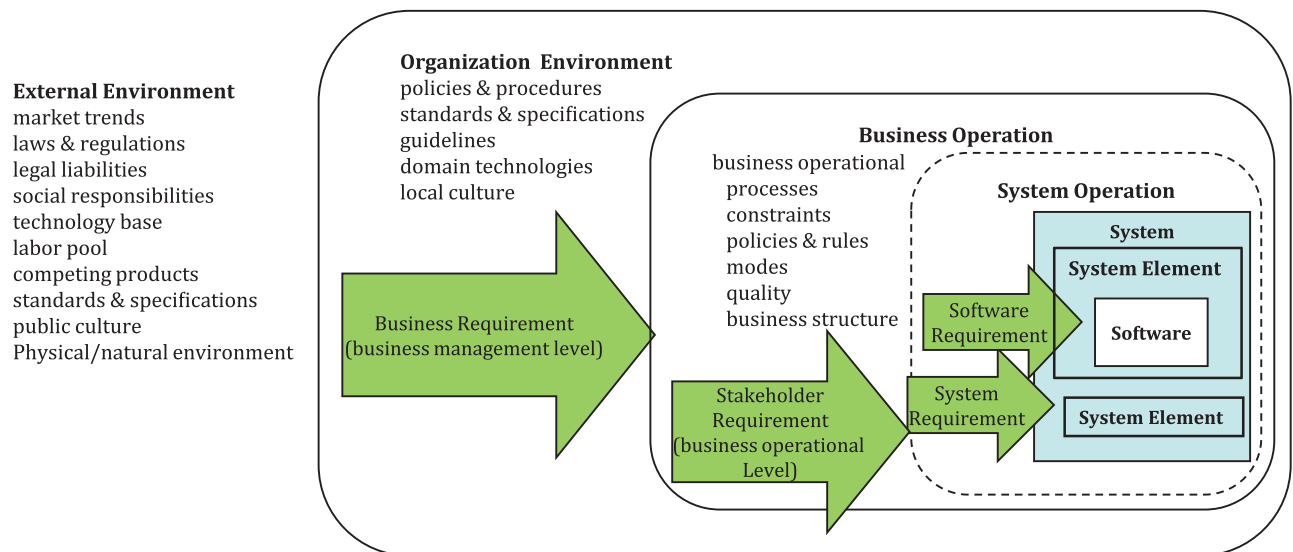


Figure 3 — Example of requirements scope in a business context

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

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

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

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

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

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

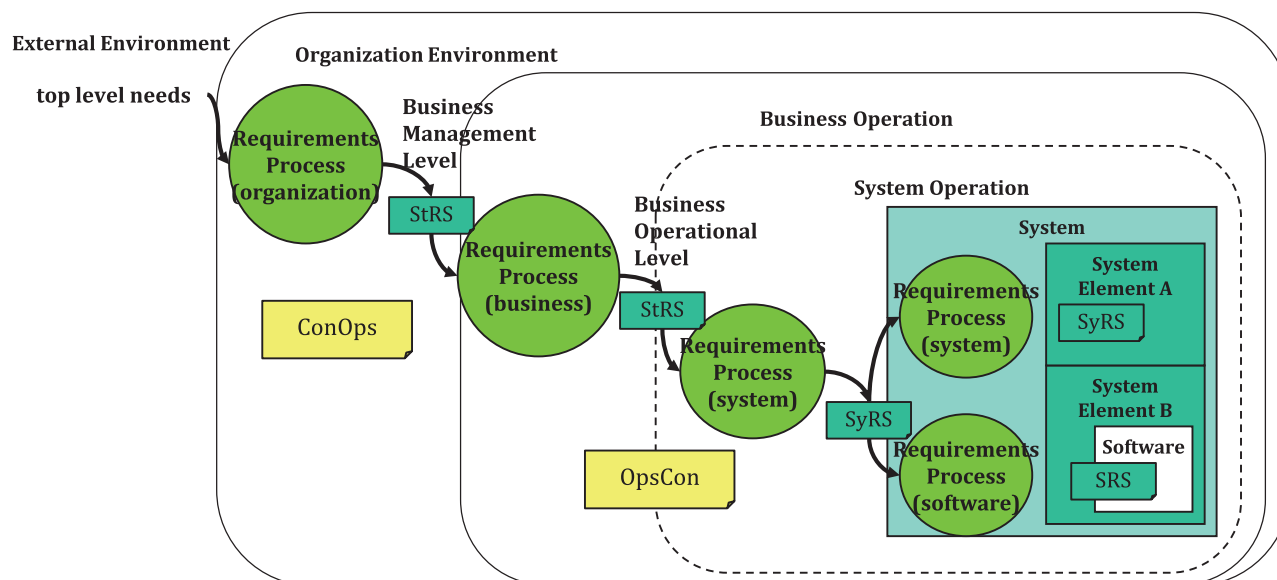


Figure 4 — Example of the relationships between requirements processes and specifications

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

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

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

6 Processes

6.1 Requirement processes

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

- Business or mission analysis process ([ISO/IEC/IEEE 15288:2015](#), 6.4.1 or [ISO/IEC/IEEE 12207:2017](#), 6.4.1).
- Stakeholder needs and requirements definition process ([ISO/IEC/IEEE 15288:2015](#), 6.4.2 or [ISO/IEC/IEEE 12207:2017](#), 6.4.2)
- System requirements definition process ([ISO/IEC/IEEE 15288:2015](#), 6.4.3) or System/software requirements definition process ([ISO/IEC/IEEE 12207:2017](#), 6.4.3)

6.1.1 Guidelines for processes

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

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

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

The principal processes are:

- 1) Business or Mission Analysis process;
- 2) Stakeholder Needs and Requirements Definition process; and
- 3) System [System/Software] Requirements Definition process.

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

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

6.2 Business or mission analysis process

6.2.1 Purpose

The purpose of the Business or Mission Analysis process is to define the business or mission problem or opportunity, characterize the solution space, and determine potential solution class(es) that could address a problem or take advantage of an opportunity.

[[ISO/IEC/IEEE 15288:2015](#), 6.4.1.1]

[[ISO/IEC/IEEE 12207:2017](#), 6.4.1.1]

6.2.2 Outcomes

As a result of the successful implementation of the Business or Mission Analysis process:

- a) The problem or opportunity space is defined.
- b) The solution space is characterized.
- c) Preliminary operational concepts and other concepts in the life cycle stages are defined.
- d) Candidate alternative solution classes are identified and analyzed.
- e) The preferred candidate alternative solution class(es) are selected.
- f) Any enabling systems or services needed for business or mission analysis are available.
- g) Traceability of business or mission problems and opportunities and the preferred alternative solution classes is established.

[[ISO/IEC/IEEE 15288:2015](#), 6.4.1.2]

[[ISO/IEC/IEEE 12207:2017](#), 6.4.1.2]

6.2.3 Activities and tasks

6.2.3.1 General

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

6.2.3.2 Prepare for Business or Mission Analysis

This activity consists of the following tasks.

Review identified problems and opportunities in the organization strategy with respect to desired organization goals or objectives.

[[ISO/IEC/IEEE 15288:2015](#), 6.4.1.3 a) 1)]

[[ISO/IEC/IEEE 12207:2017](#), 6.4.1.3 a) 1)]

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

Define the business or mission analysis strategy.

[[ISO/IEC/IEEE 15288:2015](#), 6.4.1.3 a) 2)]

[[ISO/IEC/IEEE 12207:2017](#), 6.4.1.3 a) 2)]

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

Identify and plan for the necessary enabling systems or services needed to support business or mission analysis.