

BS ISO/IEC/IEEE 29148:2011



BSI Standards Publication

Systems and software engineering — Life cycle processes — Requirements engineering

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

raising standards worldwide™

Full document details: BS ISO/IEC/IEEE 29148:2011



National foreword

This British Standard is the UK implementation of ISO/IEC/IEEE 29148:2011.

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.

© BSI 2011

ISBN 978 0 580 71995 0

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 31 December 2011.

Amendments issued since publication

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

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)

Documents
AVA
© ISO/IEC 2011
© IEEE 2011
Document Sharing Hub

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/IEC 2011
© IEEE 2011

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
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 Varembe
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	vi
Introduction.....	vii
1 Scope	1
2 Conformance	1
2.1 Intended Usage.....	1
2.2 Conformance to processes	2
2.3 Conformance to information item content.....	2
2.4 Full conformance.....	2
2.5 Tailored conformance.....	2
2.5.1 Processes.....	2
2.5.2 Information items	3
3 Normative references	3
4 Terms, definitions and abbreviated terms	3
4.1 Terms and definitions	3
4.2 Abbreviated terms	8
5 Concepts	8
5.1 Introduction.....	8
5.2 Requirements fundamentals	8
5.2.1 General	8
5.2.2 Stakeholders	8
5.2.3 Transformation of needs into requirements.....	9
5.2.4 Requirements construct	9
5.2.5 Characteristics of individual requirements	11
5.2.6 Characteristics of a set of requirements.....	11
5.2.7 Requirement language criteria.....	12
5.2.8 Requirements attributes	12
5.3 Practical considerations	14
5.3.1 Iteration and recursion of processes	14
5.3.2 Iteration and recursion in requirements engineering	16
5.4 Requirement information items	17
6 Processes.....	18
6.1 Requirement processes	18
6.1.1 Guidelines for Processes	19
6.2 Stakeholder requirements definition process	19
6.2.1 Purpose	19
6.2.2 Outcomes	19
6.2.3 Activities and tasks	20
6.3 Requirements analysis process	27
6.3.1 Purpose	27
6.3.2 Outcomes	27
6.3.3 Activities and tasks	27
6.4 Requirements engineering activities in other technical processes.....	33
6.4.1 Requirements in architectural design	33
6.4.2 Requirements in verification	34
6.4.3 Requirements in validation	36
6.5 Requirements management	37
6.5.1 Management Overview.....	37
6.5.2 Change management	37
6.5.3 Measurement for requirements.....	39

7	Information items.....	41
8	Guidelines for information items	42
8.1	Requirements information item outlines.....	42
8.2	Stakeholder requirements specification document.....	42
8.2.1	Introduction	42
8.2.2	StRS example outline	42
8.3	System requirements specification document.....	43
8.3.1	Introduction	43
8.3.2	SyRS example outline	44
8.4	Software requirements specification document	45
8.4.1	Introduction	45
8.4.2	SRS example outline	45
9	Information item content.....	46
9.1	Introduction	46
9.2	General content.....	46
9.2.1	Identification.....	46
9.2.2	Front matter.....	47
9.2.3	Definitions	47
9.2.4	References.....	47
9.2.5	Acronyms and abbreviations	47
9.3	Stakeholder requirements specification (StRS) document.....	47
9.3.1	Business purpose.....	47
9.3.2	Business scope.....	48
9.3.3	Business overview	48
9.3.4	Stakeholders	48
9.3.5	Business environment	48
9.3.6	Goal and Objective	48
9.3.7	Business model	48
9.3.8	Information environment	48
9.3.9	Business processes.....	49
9.3.10	Business operational policies and rules.....	49
9.3.11	Business operational constraints.....	49
9.3.12	Business operation modes.....	49
9.3.13	Business operational quality.....	49
9.3.14	Business structure	49
9.3.15	User requirements	49
9.3.16	Operational concept.....	50
9.3.17	Operational scenarios	50
9.3.18	Project constraints	50
9.4	System requirements specification (SyRS) document.....	50
9.4.1	System purpose.....	50
9.4.2	System scope.....	50
9.4.3	System overview.....	51
9.4.4	Functional requirements.....	51
9.4.5	Usability requirements	51
9.4.6	Performance requirements	51
9.4.7	System interfaces	51
9.4.8	System Operations	52
9.4.9	System modes and states.....	52
9.4.10	Physical characteristics.....	52
9.4.11	Environmental conditions.....	53
9.4.12	System security	53
9.4.13	Information management.....	53
9.4.14	Policies and regulations	53
9.4.15	System life cycle sustainment.....	53
9.4.16	Packaging, handling, shipping and transportation.....	53
9.4.17	Verification	54
9.4.18	Assumptions and dependencies.....	54
9.5	Software requirements specification (SRS) document.....	54

9.5.1	Purpose	54
9.5.2	Scope	54
9.5.3	Product perspective	54
9.5.4	Product functions	56
9.5.5	User characteristics	57
9.5.6	Limitations	57
9.5.7	Assumptions and dependencies	57
9.5.8	Apportioning of requirements	57
9.5.9	Specific requirements	57
9.5.10	External interfaces	58
9.5.11	Functions	58
9.5.12	Usability requirements	59
9.5.13	Performance requirements	59
9.5.14	Logical database requirements	59
9.5.15	Design constraints	60
9.5.16	Standards compliance	60
9.5.17	Software system attributes	60
9.5.18	Verification	61
9.5.19	Supporting information	61
Annex A	(normative) System operational concept	62
A.1	Overview	62
A.2	Operational concept document (OpsCon)	62
A.2.1	Scope	63
A.2.2	Referenced documents	63
A.2.3	Current system or situation	63
A.2.4	Justification for and nature of changes	66
A.2.5	Concepts for the proposed system	68
A.2.6	Operational scenarios	70
A.2.7	Summary of impacts	71
A.2.8	Analysis of the proposed system	72
A.2.9	Appendices	73
A.2.10	Glossary	73
Annex B	(informative) Concept of operations	74
B.1	Overview	74
B.2	Concept of operation document	74
B.2.1	Purpose	74
B.2.2	Scope	74
B.2.3	Strategic plan	74
B.2.4	Effectiveness	74
B.2.5	Overall operation	74
B.2.6	Governance	75
Annex C	(informative) Process Mapping from ISO/IEC 15288 and ISO/IEC 12207	76
C.1	Stakeholder requirements definition process	76
C.2	Requirements analysis Process	77
C.3	Other technical requirements-related processes	78
Annex D	(normative) Tailoring policies	80
D.1	Introduction	80
D.2	Information item tailoring process	80
D.2.1	Purpose	80
D.2.2	Outcomes	80
D.2.3	Activities and tasks	80
	Bibliography	82

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 role 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.

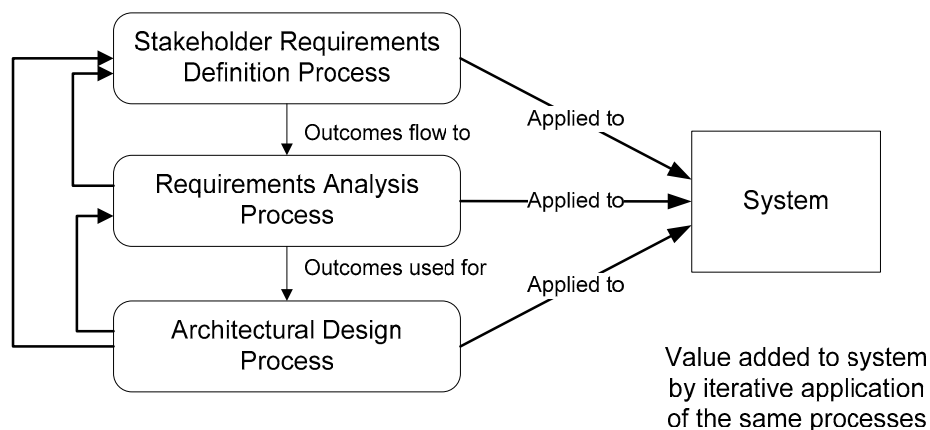


Figure 2 — Iterative application of processes

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.

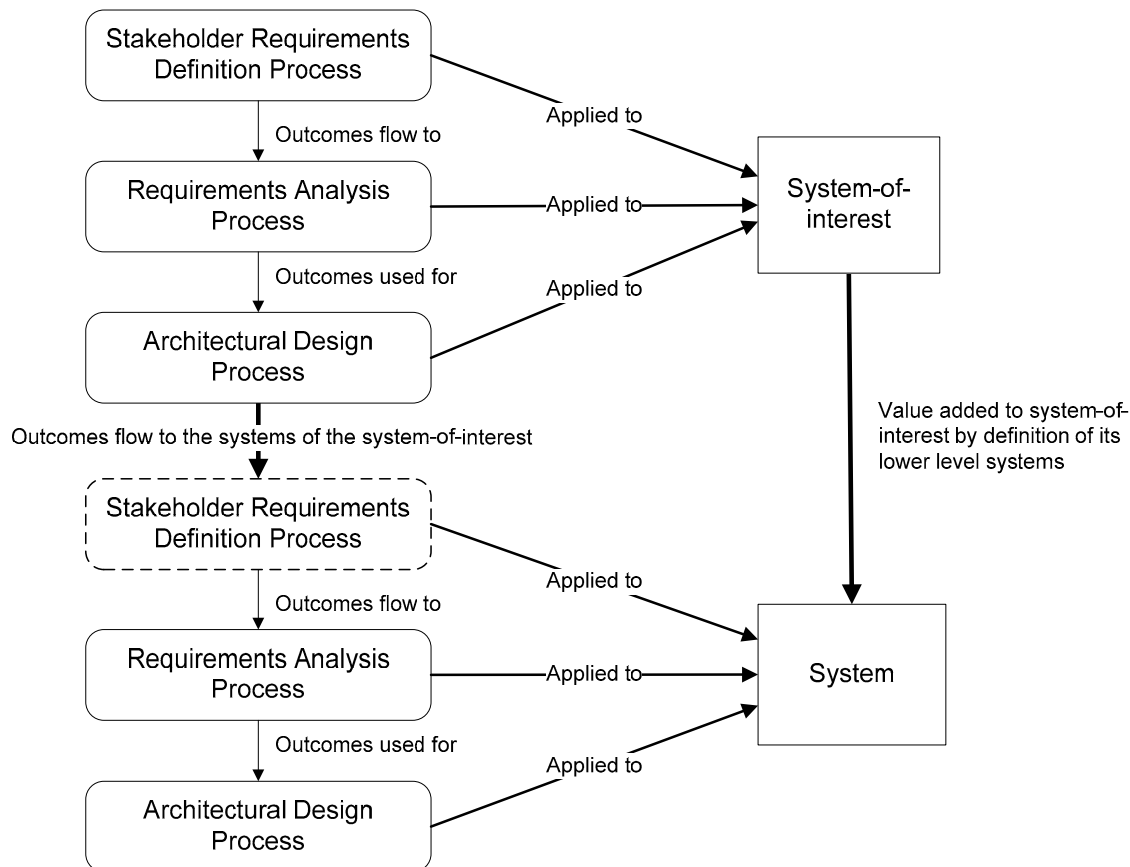


Figure 3 — Recursive application of processes

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.

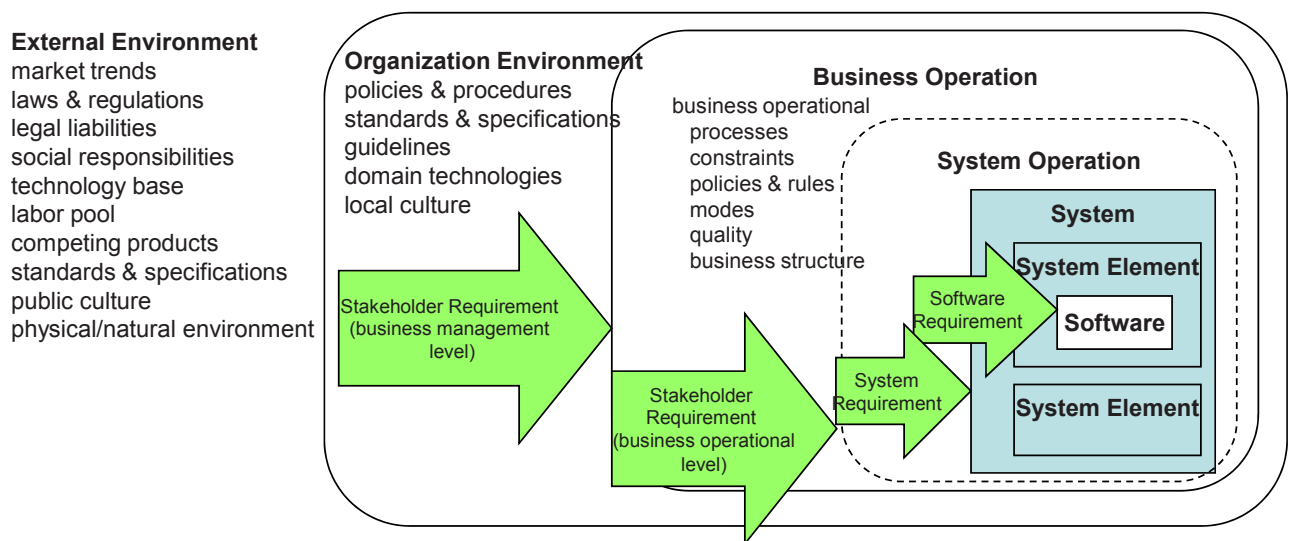


Figure 4 — Example of requirements scope in a business context

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.

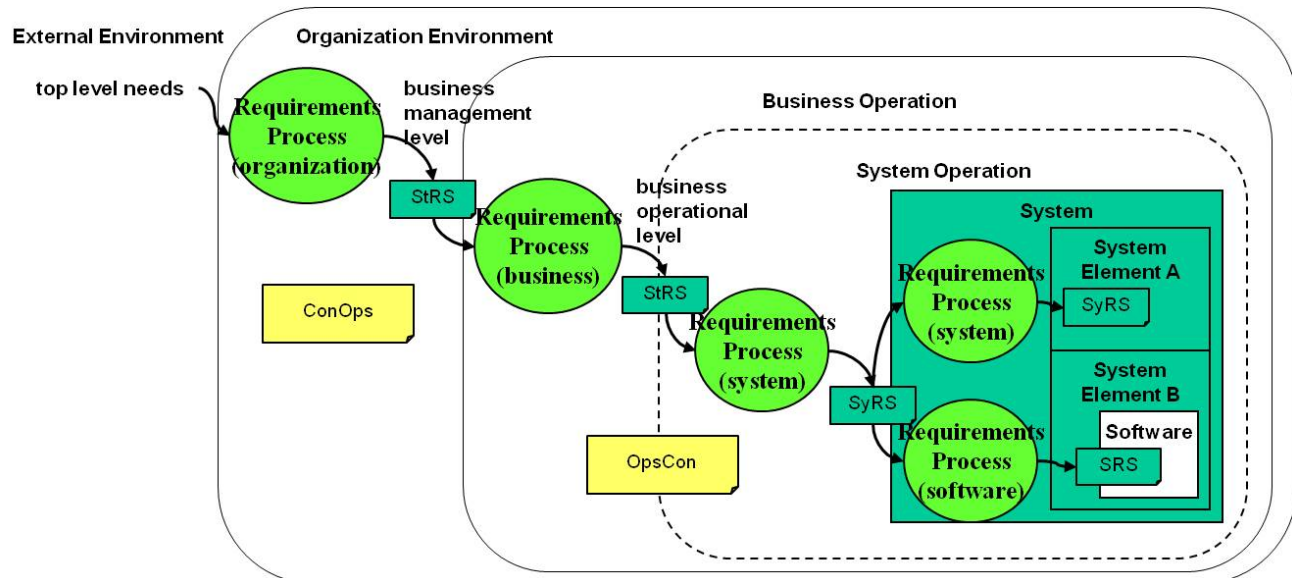


Figure 5 — Example of the sequence of requirements processes and specifications

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.