INTERNATIONAL STANDARD

ISO/IEC/IEEE 15026-1

First edition
2019-03

Systems and software engineering — Systems and software assurance —

Part 1: Concepts and vocabulary

Ingénierie des systèmes et du logiciel — Assurance des systèmes et du logiciel —

Partie 1: Concepts et vocabulaire

Reference number
ISO/IEC/IEEE 15026-1:2019(E)

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

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

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and 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.

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, *Software and systems 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 first edition cancels and replaces ISO/IEC 15026-1:2013, which has been technically revised.

The main changes compared to the previous edition are as follows:

— definitions of terms introduced in ISO/IEC 15026-3:2015 are added;

— definitions of terms whose definitions are modified in ISO/IEC 15026-3:2015 are updated.

A list of all parts in the ISO/IEC/IEEE 15026 series can be found on the ISO website.

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

Software and systems assurance and closely related fields share concepts but have different vocabularies and perspectives. This document provides a unifying set of underlying concepts and an unambiguous use of terminology across these various fields. It provides a basis for elaboration, discussion and recording agreement and rationale regarding concepts and the vocabulary used uniformly across ISO/IEC/IEEE 15026 (all parts).

This document clarifies concepts needed for understanding software and systems assurance and, in particular, those central to the use of ISO/IEC 15026-2, ISO/IEC 15026-3 and ISO/IEC 15026-4. It supports shared concepts, issues and terminology applicable across a range of properties, application domains and technologies.

Systems and software engineering — Systems and software assurance —

Part 1: Concepts and vocabulary

1 Scope

This document defines assurance-related terms and establishes an organized set of concepts and relationships to form a basis for shared understanding across user communities for assurance. It provides information to users of the other parts of ISO/IEC/IEEE 15026 including the combined use of multiple parts. The essential concept introduced by ISO/IEC/IEEE 15026 (all parts) is the statement of claims in an assurance case and the support of those claims through argumentation and evidence. These claims are in the context of assurance for properties of systems and software within life cycle processes for the system or software product.

Assurance for a service being operated and managed on an ongoing basis is not covered in ISO/IEC/IEEE 15026 (all parts).

A variety of potential users of ISO/IEC/IEEE 15026 (all parts) exists including developers and maintainers of assurance cases and those who wish to develop, sustain, evaluate or acquire a system that possesses requirements for specific properties in such a way as to be more certain of those properties and their requirements. ISO/IEC/IEEE 15026 (all parts) uses concepts and terms consistent with ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 and generally consistent with the ISO/IEC 25000 series, but the potential users of ISO/IEC/IEEE 15026 (all parts) need to understand the differences from concepts and terms to which they may be accustomed. This document attempts to clarify these differences.

The primary purpose of this document is to aid users of the other parts of ISO/IEC/IEEE 15026 by providing context, concepts and explanations for assurance, assurance cases and integrity levels. While essential to assurance practice, details regarding exactly how to measure, demonstrate or analyse particular properties are not covered. These are the subjects of more specialized standards of which a number are referenced and included in the Bibliography.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

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

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

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

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

3.1 Terms related to assurance and properties

3.1.1
assurance
grounds for justified confidence that a *claim* (3.1.4) has been or will be achieved

3.1.2
assurance case
reasoned, auditable artefact created that supports the contention that its top-level *claim* (3.1.4) (or set of claims) is satisfied, including systematic argumentation and its underlying evidence and explicit assumptions that support the claim(s)

Note 1 to entry: An assurance case contains the following and their relationships:

— one or more claims about properties;

— arguments that logically link the evidence and any assumptions to the claim(s);

— a body of evidence and possibly assumptions supporting these arguments for the claim(s); and

— justification of the choice of top-level claim and the method of reasoning.

3.1.3
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* (3.2.3), *process* (3.2.1), or *system* (3.2.4) (e.g., the price of a product, the owner of a product).

[SOURCE: ISO/IEC/IEEE 29148:2018, 3.1.2]

3.1.4
claim
true-false statement about the limitations on the values of an unambiguously defined property — called the claim's property — and limitations on the uncertainty of the property's values falling within these limitations during the claim's duration of applicability under stated *conditions* (3.1.5)

Note 1 to entry: Uncertainties may also be associated with the duration of applicability and the stated conditions.

Note 2 to entry: A claim potentially contains the following:

— property of the system-of-interest;

— limitations on the value of the property associated with the claim (e.g., on its range);

— limitations on the uncertainty of the property value meeting its limitations;

— limitations on duration of claim's applicability;

— duration-related uncertainty;

— limitations on conditions associated with the claim; and

— condition-related uncertainty.

Note 3 to entry: The term "limitations" is used to fit the many situations that can exist. Values can be a single value or multiple single values, a range of values or multiple ranges of values, and can be multi-dimensional. The boundaries of these limitations are sometimes not sharp, e.g., they can involve probability distributions and can be incremental.

3.1.5
condition
measurable qualitative or quantitative *attribute* (3.1.3) that is stipulated for a *requirement* (3.2.5) and that indicates a circumstance or event under which a requirement applies

[SOURCE: ISO/IEC/IEEE 29148:2018, 3.1.6]

3.1.6
constraint
externally imposed limitation on the *system* (3.2.4), its design, or implementation or on the *process* (3.2.1) 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.

[SOURCE: ISO/IEC/IEEE 29148:2018, 3.1.7]

3.1.7
dependability
<of an item> ability to perform as and when required

Note 1 to entry: Dependability includes availability, reliability, recoverability, maintainability, and maintenance support performance, and, in some cases, other characteristics such as durability, safety and security.

Note 2 to entry: Dependability is used as a collective term for the time-related quality characteristics of an item.

[SOURCE: IEC 60050-192:2015, 192-01-22]

3.2 Terms related to product and process

3.2.1
process
set of interrelated or interacting activities that transforms inputs into outputs

Note 1 to entry: The definition for this term can also be found in ISO 9000 and ISO/IEC/IEEE 12207.

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.30, modified — Note 1 to entry has been added.]

3.2.2
process view
description of how a specified purpose and set of outcomes may be achieved by employing the activities and tasks of existing *processes* (3.2.1)

Note 1 to entry: The process view concept is introduced in ISO/IEC/IEEE 15288:2015, Annex E and ISO/IEC/IEEE 12207:2017, Annex E.

3.2.3
product
result of a *process* (3.2.1)

Note 1 to entry: There are four agreed generic product categories: hardware (e.g., engine mechanical part); software (e.g., computer program); services (e.g., transport); and processed materials (e.g., lubricant). Hardware and processed materials are generally tangible products, while software or services are generally intangible.

Note 2 to entry: Results could be components, *systems* (3.2.4), software, services, rules, documents, or many other items.

Note 3 to entry: The "result" could in some cases be many related individual results. However, *claims* (3.1.4) usually relate to specified versions of a product.

Note 4 to entry: The definition for this term can also be found in ISO 9000 and ISO/IEC/IEEE 12207.

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.32, modified—Notes 2 to 4 to entry have been added.]

3.2.4
system
combination of interacting elements organized to achieve one or more stated purposes

Note 1 to entry: A system is sometimes considered as a *product* (3.2.3) or as the services it provides.

Note 2 to entry: In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g., aircraft system. Alternatively, the word "system" is substituted simply by a context-dependent synonym, e.g., aircraft, though this potentially obscures a system principles perspective.

Note 3 to entry: A complete system includes all of the associated equipment, facilities, material, computer programs, firmware, technical documentation, services and personnel required for operations and support to the degree necessary for self-sufficient use in its intended environment.

Note 4 to entry: The definition for this term can also be found in ISO/IEC/IEEE 12207.

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.46, modified — Note 4 to entry has been added.]

3.2.5
requirement
statement which translates or expresses a need and its associated *constraints* (3.1.6) and *conditions* (3.1.5)

Note 1 to entry: Requirements exist at different levels in the *system* (3.2.4) 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.

[SOURCE: ISO/IEC/IEEE 29148:2018, 3.1.19]

3.2.6
system element
member of a set of elements that constitutes a *system* (3.2.4)

EXAMPLE Hardware, software, data, humans, *processes* (3.2.1) (e.g., processes for providing service to users), procedures (e.g., operator instructions), facilities, materials, and naturally occurring entities or any combination.

Note 1 to entry: A system element is a discrete part of a system that can be implemented to fulfill specified *requirements* (3.2.5).

Note 2 to entry: The definition for this term can also be found in ISO/IEC/IEEE 12207.

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.47, modified — Note 2 to entry has been added.]

3.3 Terms related to integrity level

3.3.1
integrity level
degree of confidence that the *system-of-interest* (3.3.12) meets the associated *integrity level claim* (3.3.4)

Note 1 to entry: While a definition of "integrity level" is given, existing definitions and the relevant communities do not agree on a definition of "integrity" consistent with its use in "integrity level". Hence, no separate definition of "integrity" is included in this document. For the definition of "integrity" used in ISO/IEC JTC 1/SC 7, see ISO/IEC 25010:2011, 4.1.6.2.

Note 2 to entry: An integrity level is different from the *likelihood* (3.3.6) that the integrity level claim is met but they are closely related.

Note 3 to entry: The word "confidence" implies that the definition of integrity levels can be a subjective concept.

Note 4 to entry: In this document, integrity levels are defined in terms of *risk* (3.4.2) and hence cover safety, security, economic and any other dimension of risk that is relevant to the system-of-interest.

3.3.2
integrity level requirements
set of *requirements* (3.2.5) that, when met, will provide a level of confidence in the associated *integrity level claim* (3.3.4) commensurate with the associated *integrity level* (3.3.1)

Note 1 to entry: An integrity level requirement is different from any requirement in ISO/IEC/IEEE 15288 or ISO/IEC/IEEE 12207.

3.3.3
initial risk
estimated *risk* (3.4.2) before applying *risk reduction measures* (3.3.9)

3.3.4
integrity level claim
proposition representing a *requirement* (3.2.5) on a *risk reduction measure* (3.3.9) identified in the *risk treatment* (3.3.11) *process* (3.2.1) of the *system-of-interest* (3.3.12)

Note 1 to entry: In general, it is described in terms of requirements to avoid, control or mitigate the *consequences* (3.4.1) of *dangerous conditions* (3.4.11), so as to provide a *tolerable risk* (3.3.15) if it is met.

Note 2 to entry: The proposition that can be regarded as an integrity level claim in IEC 61508 is that an E/E/PE safety-related *system* (3.2.4) satisfactorily performing the specified safety functions under all the stated conditions.

3.3.5
level of risk
magnitude of a *risk* (3.4.2) or combination of risks, expressed in terms of the combination of *consequences* (3.4.1) and their *likelihood* (3.3.6)

[SOURCE: ISO Guide 73:2009, 3.6.1.8]

3.3.6
likelihood
chance of something happening

[SOURCE: ISO Guide 73:2009, 3.6.1.1, modified — NOTEs 1 and 2 have been removed.]

3.3.7
residual risk
risk (3.4.2) remaining after *risk treatment* (3.3.11)

[SOURCE: ISO Guide 73:2009, 3.8.1.6, modified — NOTEs 1 and 2 have been removed.]

3.3.8
risk criteria
terms of reference against which the significance of a *risk* (3.4.2) is evaluated

[SOURCE: ISO Guide 73:2009, 3.3.1.3, modified — NOTEs 1 and 2 have been removed.]

3.3.9
risk reduction measure
measure to reduce or mitigate *risk* (3.4.2)

Note 1 to entry: A typical risk reduction measure is a safety-related *system* (3.2.4) in the IEC 61508 series.

3.3.10
risk source
element which alone or in combination has the intrinsic potential to give rise to *risk* (3.4.2)

Note 1 to entry: A hazard in ISO Guide 73 is an instance of a risk source.

Note 2 to entry: A *fault* (3.4.6), an *error* (3.4.5) or a *failure* (3.4.9) in the context of reliability can be a risk source. The definitions of those terms can be found in IEC 61508-4.

Note 3 to entry: A threat in the context of security and a *threat agent* (3.3.14) and an adverse action defined in ISO/IEC 15408-1 can be a risk source.

[SOURCE: ISO Guide 73:2009, 3.5.1.2, modified — NOTE has been removed: Notes 1, 2 and 3 to entry have been added.]

3.3.11
risk treatment
process (3.2.1) to eliminate *risk* (3.4.2) or reduce it to a tolerable level

[SOURCE: ISO Guide 73:2009, 3.8.1, modified — The words "modify risk" have been replaced with "eliminate risk or reduce it to a tolerable level"; NOTEs 1, 2 and 3 have been removed.]

3.3.12
system-of-interest
system whose life cycle is under consideration

Note 1 to entry: The definition for this term can also be found in ISO/IEC/IEEE 15288.

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.63, modified — The abbreviated term "SOI" has been removed; Note 1 to entry has been added.]

3.3.13
target risk
risk (3.4.2) that is intended to be reached

[SOURCE: IEC 61508-4:2010, 3.1.10, modified — Restriction of the hazard has been removed.]

3.3.14
threat agent
entity that can adversely act on *property-of-interest* (3.4.12)

[SOURCE: ISO/IEC 15408-1:2008, 3.1.71, modified — The word "assets" has been replaced with "property-of-interest".]

3.3.15
tolerable risk
level of risk (3.3.5) which is accepted in a given context based on the current values of society

Note 1 to entry: A tolerable risk is sometimes called an acceptable risk, e.g., see ISO/IEC/IEEE 16085, and ISO 14971. The general risk management standards ISO Guide 73 and ISO 31000 use both phrases without explicit definitions.

[SOURCE: ISO/IEC Guide 51:2014, 3.7, modified — Note 1 to entry has been added.]

3.4 Terms related to conditions and consequences

3.4.1
consequence
outcome of an event affecting objectives

[SOURCE: ISO Guide 73:2009, 3.6.1.3, modified — NOTEs 1, 2 and 3 have been removed.]

3.4.2
risk
effect of uncertainty on objectives

Note 1 to entry: An effect is a deviation from the expected — positive and/or negative. In this document the focus is on negative deviations leading to *adverse consequences* (3.4.3).

Note 2 to entry: Risk is often characterized by reference to potential events and *consequences* (3.4.1), or a combination of these.

Note 3 to entry: Risk is often expressed in terms of a combination of the consequences of an event (including changes in circumstances) and the associated *likelihood* (3.3.6) of occurrence. In this document risk is characterized as the combination of the severity of the adverse consequence and the likelihood of an adverse consequence occurring.

Note 4 to entry: Objectives can have different aspects, such as financial, health and safety, and environmental goals and can apply at different levels, such as strategic, *organization* (3.5.1) -wide, project, *product* (3.2.3) and *process* (3.2.1).

Note 5 to entry: Uncertainty is the state, even partial, of deficiency of information related to, understanding or knowledge of, an event, its consequence, or likelihood.

[SOURCE: ISO Guide 73:2009, 1.1, modified — NOTEs have been reordered; information specific to this document has been added in Notes 1 and 3 to entry.]

3.4.3
adverse consequence
consequence (3.4.1) that results in a specified level of loss

Note 1 to entry: An adverse consequence results from the *system-of-interest* (3.3.12) being in a *dangerous condition* (3.4.11) combined with the environment of the *system* (3.2.4) being in its worst case state.

Note 2 to entry: Harm in ISO Guide 51 is an instance of an adverse consequence. The concept of adverse consequences is introduced in order to cover not only harms in the safety context but also loss of assets in the security context and any other losses.

3.4.4
desirable consequence
positive consequence
consequence (3.4.1) associated with a benefit or gain or avoiding an *adverse consequence* (3.4.3)

3.4.5
error
discrepancy between a computed, observed or measured value or *condition* (3.1.5), and the true, specified or theoretically correct value or condition

[SOURCE: IEC 60050-192:2015, 192-03-02, modified — Notes 1 and 2 to entry have been removed.]

3.4.6
fault
defect in a *system* (3.2.4) or a representation of a system that if executed/activated can potentially result in an *error* (3.4.5)

Note 1 to entry: Faults can occur in specifications when they are not correct.

3.4.7
attack
malicious action or interaction with the *system* (3.2.4) or its environment that has the potential to result in a *fault* (3.4.6) or an *error* (3.4.5), and thereby possibly in a *failure* (3.4.9), or an *adverse consequence* (3.4.3)

3.4.8
violation
behaviour, act or event deviating from a *system's* (3.2.4) desired property or *claim* (3.1.4) of interest

Note 1 to entry: In the area of safety, the term "violation" is used to refer to a deliberate human contravention of a procedure or rule.

3.4.9
failure
termination of the ability of a *system* (3.2.4) to perform a required function or its inability to perform within previously specified limits; an externally visible deviation from the system's specification

3.4.10
systematic failure
failure (3.4.9) related in a deterministic way to a certain cause that can only be eliminated by a modification of the design or of the manufacturing *process* (3.2.1), operational procedures, documentation or other relevant factors

3.4.11
dangerous condition
state of a *system* (3.2.4) that, in combination with some states of the environment, will result in an *adverse consequence* (3.4.3)

Note 1 to entry: A hazardous situation in ISO/IEC Guide 51 and IEC 61508-4 can be a dangerous condition. A threat in the ISO/IEC 15026 series is also an example of a dangerous condition. A concept of dangerous conditions is introduced in order to cover not only hazardous situations in the safety context but also *errors* (3.4.5) in the reliability, integrity, confidentiality or *dependability* (3.1.7) contexts and other states of a system which can lead to adverse consequences.

Note 2 to entry: Occurrences of *failures* (3.4.9) in the context of reliability or of a definition in IEC 61508-4 often lead to dangerous conditions but not always do.

Note 3 to entry: A dangerous condition therefore has at least the following *attributes* (3.1.3):

a) the associated adverse consequences,

b) the trigger events that lead to the dangerous condition, and

c) the trigger events that lead to the adverse consequences from the dangerous condition.

3.4.12
property-of-interest
object whose loss is considered as a negative effect

Note 1 to entry: The concept of property-of-interest is introduced in order to characterize negative effects of *consequences* (3.4.1).

Note 2 to entry: In the safety context, human lives and health can be property-of-interests.

Note 3 to entry: Assets in the security context, e.g., defined in ISO/IEC 15408-1, can be a property-of-interest.

3.5 Terms related to organization

3.5.1
organization
group of people and facilities with an arrangement of responsibilities, authorities and relationships

EXAMPLE Company, corporation, firm, enterprise, institution, charity, sole trader, association, or parts or combination thereof.

Note 1 to entry: An identified part of an organization (even as small as a single individual) or an identified group of organizations can be regarded as an organization if it has responsibilities, authorities and relationships. A body of persons organized for some specific purpose, such as a club, union, corporation, or society, is an organization.

Note 2 to entry: The definition for this term can also be found in ISO/IEC/IEEE 12207.

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.27]

3.5.2
approval authority
person (or persons) and/or *organization* (3.5.1) (or organizations) responsible for approving activities, artefacts and other aspects of the *system* (3.2.4) during its life cycle

Note 1 to entry: The approval authority may include multiple entities, e.g., individuals or organizations. These can include different entitles with different levels of approval and/or different areas of interest.

Note 2 to entry: In two-party situations, the approval authority often rests with the acquirer. In regulatory situations, the approval authority may be a third party such as a governmental organization or its agent. In other situations, for example, the purchase of off-the-shelf *products* (3.2.3) developed by a single-party, the independence of the approval authority can be a relevant issue to the acquirer.

3.5.3
design authority
person or *organization* (3.5.1) that is responsible for the design of the *product* (3.2.3)

3.5.4
integrity assurance authority
independent person or *organization* (3.5.1) responsible for certifying compliance with the integrity level *requirements* (3.3.2)

4 Organization of this document

Clause 5 covers basic concepts such as assurance, stakeholders, systems and products, uncertainty and consequence. Clause 6 covers some issues of which users of ISO/IEC 15026-2, ISO/IEC 15026-3 and ISO/IEC 15026-4 need to be initially aware of. Clause 6, Clause 7 and Clause 8 cover terms, concepts and topics particularly relevant to users of ISO/IEC 15026-2, ISO/IEC 15026-3 and ISO/IEC 15026-4, respectively, although users of one part can also benefit from some of the information in the clauses for other parts. A Bibliography is included at the end. References to numbered items in the Bibliography are shown in brackets throughout.

5 Basic concepts

5.1 General

This clause covers the concepts and vocabulary fundamental to ISO/IEC/IEEE 15026 (all parts).

5.2 Assurance

ISO/IEC/IEEE 15026 (all parts) uses a specific definition for assurance as being grounds for justified confidence. Generally, stakeholders need grounds for justifiable confidence prior to depending on a system, especially a system involving complexity, novelty or technology with a history of problems (e.g., software). The greater the degree of dependence, the greater the need for strong grounds for confidence. The appropriate valid arguments and evidence to establish a rational basis for justified confidence in the relevant claims about the system's properties need to be made. These properties may include such aspects as future costs, behaviour and consequences. Throughout the life cycle, adequate grounds need to exist for justifying decisions related to ensuring the design and production of an adequate system and to be able to place reliance on that system.

Assurance is a term whose usage varies among the communities who use the term. However, all usage relates to placing limitations on or reducing uncertainty in such things as measurements, observations, estimations, predictions, information, inferences or effects of unknowns with the ultimate objective of achieving and/or showing a claim. Such a reduction in uncertainty can provide an improved basis for justified confidence. Even if the estimate of a property's value remains unchanged, the effort spent in reducing uncertainty about its value can often be cost-effective since the resulting reduced uncertainty improves the basis for decision-making.

Assurance may relate to:

a) would the system or software as specified meet real-world needs and expectations,

b) would or does the as-built and operated system meet the specifications, or

c) both a) and b).

Specifications may be representations of static and/or dynamic aspects of the system. Specifications often include descriptions of capability, functionality, behaviour, structure, service and responsibility including time-related and resource-related aspects as well as limitations on frequency or seriousness of deviations by the product and related uncertainties.

Specifications may be prescriptions and/or constraints (e.g., for and on product behaviours) and may as well include measures of merit and directions regarding trade-offs. Generally, specifications place some limitations on the environment and its conditions (e.g., temperature) and possibly the conditions of the product (e.g., age or amount of wear).

5.3 Stakeholders

Through their life cycle, systems and software have multiple stakeholders who affect or are affected by the system and the system life cycle processes. Stakeholders might benefit from, incur losses from, impose constraints on, or otherwise have a "stake" in the system, and therefore are those that provide the requirements for the system. Stakeholders can include non-users whose performance, results or other requirements might be affected, e.g., entities whose software is executing on the same or networked computers.

A different but important kind of stakeholder is an attacker, who certainly imposes constraints or has interests involved with the system. This document includes the attacker as a stakeholder; however, some in the security community and elsewhere exclude attackers from their use of the term "stakeholders."

The relevant stakeholders whose requirements are of concern include not only the system's owners and users, but also developers and operators who need to identify requirements for the development and operation of the system. Depending on conditions and consequences, the various stakeholders require grounds for justified confidence in properties of the system for which they identified requirements.

5.4 System and product

To be consistent with ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207, ISO/IEC/IEEE 15026 (all parts) uses the term "system" throughout. Users of this document who are more familiar with using the term "product" should note that "system" includes products and services that are the results of processes as well as software and system or software elements or components. While primarily motivated by concern for systems produced (at least in part) by human-controlled or artificial processes, this document can be used in reducing uncertainty about a system's dependence on natural phenomena as well.

5.5 Property

5.5.1 General

ISO/IEC/IEEE 15026 (all parts) relates assurance to requirements of a property of a system or software product. Properties are entities that can be predicated of things or, in other words, attributed to them[176]. Therefore, a property might be a condition, a characteristic, an attribute, a quality, a trait, a measurement or a consequence. A property might be invariant or dependent on time, situation or history. In ISO/IEC/IEEE 15026 (all parts), a property is expected to be relevant directly or indirectly to a system or systems and thus have related requirements.

Properties may have requirements for what they were in the past, what they are presently or what they will be in the future. Generally, the last is the most important in ISO/IEC/IEEE 15026 (all parts). As this knowledge involves predicting the future, it is often the most difficult and uncertain to attain; therefore, a system's future behaviour and consequences (see 5.8) often become principal issues in its assurance.

Many of the properties with requirements are qualities of the system. Several standards and reports provide lists and definitions of qualities that could be the subject of assurance including ISO/IEC 25010 and the related series, ISO/IEC 2382, ISO 9241 and ISO/TS 25238.

This use of the term "property" derives from, is consistent with and subsumes the broad use of the term "property" in ISO/IEC 25010 where it is used spanning properties that are inherent or not, internal, external, and in use or context.

Producers and other stakeholders may prioritize properties such as efficiency and reliability and perform trade-off studies between them and their related requirements. A number of techniques have been created for addressing these trades, such as those in References [87], [53], [124], [159] and [102]. The specifying of a top-level claim for a property is sometimes the result of analyses including trade-off studies.

5.5.2 Properties as behaviours

Often the property is specified as behaviour. During performed operations, behaviour-related properties might be formally specified as a combination of:

— restriction on allowed system states (sometimes called the "safety property");

— system states that must be reached; required progress or accomplishment ("liveness property");

— constraints on flows or interactions; requirements for separation constraint.

These kinds of properties can be stated as conditions or constraints that must be true of the system.[1)] In practice, these are non-trivial and modularized, involving time and starting state(s) as well as state transitions related to interaction with the system's or software's environment.

Many kinds of flows such as of gases, fluids, traffic or information are of possible interest as well as constraints on them such as non-interference and separations to be maintained. In addition, flow constraints are often convenient or necessary to specify aspects of information security[137] including access control mechanisms and policies and restrictions on information overtly or covertly communicated.

5.6 Uncertainty and confidence

Uncertainty is used in ISO/IEC/IEEE 15026 (all parts) as an inclusive term. It covers lack of certainty whether the uncertainty can be modelled probabilistically or not. Uncertainty can include vague notions that may be modelled without the use of probability. Certain communities restrict the application of this term to predictions of future events, to physical measurements already made or to unknowns. While these limited usages may be convenient within those communities, users of ISO/IEC/IEEE 15026 (all parts) span many communities.

The degree of confidence that can be or is justifiably engendered based on a specific assurance case may vary by individual or organization and the situation. The less uncertainty about an assurance case's claims, the higher the degree of justified confidence, However, the conversion of an amount of uncertainty into a degree of justified confidence in suitability for certain applications is not straightforward or well understood. For this and other reasons, consequences are sometimes directly included within the assurance case. While this closes a logical gap, it does not remove the decision maker's act of judgement regarding the merited degree of confidence.

5.7 Conditions and initiating events

The assurance case needs to cover all the conditions that could have a significant negative effect on the conclusion and uncertainty of the top-level claim. The potentially relevant universe of conditions and events can be hard to initially identify[64] and ascertaining which ones might have a significant effect can be difficult without at least initially including them in the assurance case.

Historically, the one condition that has received the most attention is system failure. A substantial volume of checklists, practice and literature exists concerning system failure (e.g. IEC 61508-7,

1) If specified formally, this can allow static analysis of conformity of designs and code, potentially adding creditable assurance evidence.

References [64] and [76], Chapter 18, page 475–524). While much of this work has been done in the communities addressing safety, security or human error, system failure can result in less achievement of a positive property or consequence as well as negative properties or losses.

The dangerousness of system behaviours can differ by the conditions of its environment. These behaviours and conditions often need combining during analysis to establish whether adverse consequences will result or not. The actual conditions of its environment might or might not be known within the system depending on its sensors or inputs and their processing.

The designers of the system might or might not be cognizant of all the initiating events for a condition within the environment; however, dangerous conditions may need to be dealt with even though not all of their initiating events are known or recognizable.

5.8 Consequences

Outside the system, much of the reasoning is based on conditions that could lead to adverse consequences and their initiating events or preconditions. Inside the system reasoning is based on conditions that can lead to dangerous system behaviours and the initiating events or preconditions for these conditions.

In practice, claims can extend beyond the boundaries of the system or its behaviours. In particular, claims can place limitations on consequences of a system's behaviour and/or system-related events, activities, and/or conditions — especially on the values of consequences. One may refer to:

— A consequence is desirable or undesirable from a stakeholder's perspective, viewpoint or interests. A consequence may occur anywhere in the system's life cycle.

In complex socio-technical systems, explanations of mishaps or claim violations cannot be limited to "component" failures. Adverse consequences can result from normal behaviour variability and unintended or unanticipated interactions[116][119]. Regardless of how they arise, dangerous conditions and adverse consequences are subjects for mitigation.

Attackers can possess capabilities, resources, motivations and intentions that enable them to initiate and carry malicious efforts to violate a claim. Violators use their capabilities to take advantage of system-provided and/or environment-provided opportunities called vulnerabilities, i.e., "weaknesses… that could be exploited or triggered by a threat source"[152][2)].

A sometimes misunderstood point is that maliciousness and subversion are concerns even when no security-related system property is involved. Malicious developers might have an effect on successful achievement of almost any property.

Several standards or reports mention consequences associated with systems within a specific domain. Examples include ISO 14620, ISO 19706 and ISO/TS 25238. Risk management standards also address consequences, for example ISO/IEC 16085 and ISO 31000.

6 Using multiple parts of ISO/IEC/IEEE 15026

6.1 General

ISO/IEC/IEEE 15026 or its parts can be used alone or with other standards or guidance. The parts of ISO/IEC/IEEE 15026 can be mapped to most life cycle standards and can use any set of well-defined qualities or properties.

2) For many purposes, the meaningfulness and need to separate vulnerabilities from other weaknesses can be low or non-existent. In addition, a question always exists about the current and future contexts that are relevant for "could be exploited or triggered".

6.2 Initial usage guidance

The properties and/or claims covered when using ISO/IEC/IEEE 15026 (all parts) are entirely up to the users of the standard who are responding to the system's stakeholders' needs and requirements. Any property or claim may be selected for an assurance case, regardless of its importance or related risk; however, the parts of ISO/IEC/IEEE 15026 are designed primarily for those properties that one or more primary stakeholders deem critical. ISO/IEC 15026-4 uses the term "critical properties" for these stakeholder priorities and requirements.

Transitioning from ISO/IEC 15026:1998 to ISO/IEC 15026-3 will require dealing with some differences. ISO/IEC 15026-3 opens up new engineering and decision options, because it takes not only a standalone perspective but also one that includes relating integrity levels to an assurance case. ISO/IEC 15026-3 concentrates more on the system itself and its integrity levels rather than on external risk analysis and also includes the creation of integrity levels. Clause 7 discusses integrity levels.

6.3 Relationships among parts of ISO/IEC/IEEE 15026

The ISO/IEC/IEEE 15026 series is made up of the following parts:

— Part 1, *Concepts and vocabulary*, explains concepts and terms as a basis for all parts of this series.

— Part 2, *Assurance case*, includes requirements on the content and structure of the assurance case.

— Part 3, *System integrity levels*, relates integrity levels to the assurance case and includes requirements for their use with and without an assurance case (revises ISO/IEC 15026:1998).

— Part 4, *Assurance in the life cycle*, gives assurance-related guidance and recommendations for specific activities throughout system and software life cycle processes.

While Part 2, Part 3 and Part 4 of this series provide a separation of assurance topics and may be used alone, they may be used together because they form a related set. This document provides background, concepts and vocabulary that are applicable to all three and specific introductions to coverage of Parts 2, 3 and 4 of this series.

The assurance case is relevant to a greater or lesser extent in all parts, although ISO/IEC 15026-4 discusses achieving the claim and showing the achievement of the claim whether or not such "showing" is contained in an artefact specifically called an "assurance case."

ISO/IEC 15026-2 concentrates on the contents and structure of the assurance case. ISO/IEC 15026-3 relates integrity levels and assurance cases by describing how integrity levels and assurance cases can work together, especially in the definition of specifications for integrity levels or by using integrity levels within a portion of an assurance case. This relationship is governed by the degree of risk and dependencies in the system.

If the risks or the risk treatment are not well understood or if the dependency structure of the whole system or the choice of suitable claims is unclear, using an assurance case is a better choice than using integrity levels. This particularly is the case when facing new kinds of risks or using a new kind of risk treatment. In these situations, justifying the choice of the top-level claim for the assurance case is important.

When the risks and their treatment are well understood, however, developers need not justify the choice of the top-level claim and need only select the proper claims for their context from a known set—an integrity level from a set of integrity levels. In these situations, the generic arguments created by the definers of the integrity level provide the justification that meeting the integrity level requirements will adequately show the meeting of the integrity level. Such a justification (e.g., a generalized assurance case) is usually created one time by a separate organization and used by multiple projects.

ISO/IEC 15026-4 includes assurance-related guidance and recommendations for activities across the life cycle processes including activities that extend beyond those directly related to an assurance case, e.g., project planning for assurance-related considerations.

6.4 Authorities

Parts of ISO/IEC/IEEE 15026 involve "authorities" as defined in Clause 3. For example, ISO/IEC 15026-3 includes obtaining agreements among the integrity level definition authority, the design authority and the integrity level assurance authority. Additionally, a new system needs the approval authorities of acquirers to take charge of analysing the process of creating assurance cases with the design authority and the integrity assurance authority of the suppliers.

However, the "approval authority" for the assurance case is not necessarily the judge of conformance to ISO/IEC/IEEE 15026 (all parts). To the extent possible, claims of conformance to the ISO/IEC/IEEE 15026 series are judged on aspects that are more straightforward and more difficult to dispute than the quality of artefacts and decisions judged in the context of the system or project. In practice, contracts can explicitly call for the acquirer to be the approval authority or the approver of conformance to the series.

7 ISO/IEC/IEEE 15026 (all parts) and the assurance case

7.1 General

ISO/IEC 15026-2 covers the structure and content of an assurance case. It describes the five principal components of an assurance case: claims, arguments, evidence, justifications and assumptions. The purpose of an assurance case is to improve assurance communications by informing stakeholders' decision-making and supplying grounds for needed stakeholder confidence. The most common use of an assurance case is to provide assurance about system properties to parties not closely involved in the system's technical development processes. Such parties may be involved in the system's certification or regulation, acquisition or audit. Usually, an assurance case addresses the reasons to expect and confirm successful production of the system, including the possibilities and risks identified as difficulties or obstacles to developing and sustaining that system.

Unlike logical proofs of the deduction of the claims from the evidence, which covers the absolute truth or Platonic truth aspects, assurance cases deal with the dialectic aspects of the system where the truth is always relative or even subjective. In other words, logical proofs are described under a fixed logical theory, but assurance cases may be rebutted on the basis that the underlying logical theory is inappropriate. The need for assurance case arises when one realizes the properties of the systems in the real world can never be completely formalized in a logical theory, but there is always something which is not covered by any logical formalization.

NOTE 1 When the top-level claim is about safety, security, dependability or RAM (reliability, availability and maintainability), assurance cases associated with these claims are called safety cases, security cases, dependability cases or RAM cases, respectively. See References [141], [144], [145], [148], [157], [169], [60], [84], [85] and [86].

Considered as an artefact, an assurance case has quality-related aspects such as: the nature of content, its form or structure (e.g., method of argumentation or modularity); semantic issues such as completeness; creation and maintenance (including tool support, usability and presentation, integrity, validity, understandability); and the feature of clearly stated conclusions with explicit degrees of uncertainty. One article[166] covers a substantial list of quality-related characteristics for assurance cases. The quality-related aspects of an assurance case are not covered in ISO/IEC/IEEE 15026 (all parts).

Any substantive modifications in the system, changes in the environment or changes in the assurance case's top-level claims will necessitate recorded changes to the assurance case. Thus, an assurance case usually contains a progressively expanding body of evidence built up during development and later life cycle activities that responds as required to all relevant changes (Reference [141], p. 5).

NOTE 2 An assurance case's claim(s) on the values of properties could include the system's entire set of requirements for a property of interest. One example can have a top-level claim composed of:

a) required limitations on consequences;

b) functionality and properties of the system itself (e.g., that this functionality cannot be bypassed).

The qualities defined in the ISO/IEC 25000 family of standards include qualities related to functionality and constraints. The Common Criteria v. 3.1 Revision 2[92] is also interested in both.

7.2 Justification of method of reasoning

An argument has an associated justification for the validity or merit of its method of reasoning. The method of argument can be an additional source of uncertainty.

A variety of bases for argumentation and analysis in the assurance case might be used, and these vary in their applicability, power, resulting accuracy and uncertainty and ease of use. Subjects of and approaches to reasoning differ among communities having differing motivations, mind-sets and often multiple methods of reasoning.

Examples of methods of reasoning include:

— Quantitative:
  — Deterministic (e.g., formal proofs).
  — Non-deterministic formal systems for reasoning:
    — probabilistic,
    — game theoretic (e.g., minimax), or
    — other uncertainty-based formal systems of reasoning (e.g., fuzzy sets).
— Qualitative (e.g., staff performance evaluations, court judgements and qualitative statements of event causality).

Complex products and situations — and any involving humans — are beyond the current state of the art to "quantitatively" create precise and accurate predictions. Subjective judgements are used in the absence of affordable, suitable and more objective methods and techniques or where needed to supplement or evaluate the results of such techniques. Supplementing quantitative techniques with expert review and judgement is widely used and generally accepted. As with other forms of argumentation, subjective judgements take the form of a claim and its support. While sometimes necessary or advantageous, use of subjective judgement within the assurance case can lead to additional uncertainties, so, generally, (just as with assumptions) the less critical the judgement is, the better.

The patterns of occurrences of "natural" events and common, non-malicious human behaviours are usually described probabilistically. However, possibilities for intelligent, malicious actions whose probability is not determinable or not knowable is particularly a concern if the intelligent, malicious adversary deliberately violates any probability estimates one could make regarding their behaviour, e.g., to achieve surprise. This distinction is central to the difference in reasoning between safety and security.

7.3 Means of obtaining and managing evidence

For any property, many means of obtaining evidence exist. Among these are human experience, history, observations, measurements, tests, evaluation and compliance results, analyses, defects and inferences. The evidence should achieve the objectives claimed in the assurance argument (MoD DefStan 00-42 Part 3, section 9.1[141]).

The body of evidence can become quite large and may need to be organized and managed by some framework providing permanence and traceability of the evidence in order to provide users confidence in its source, contents and validity. One guidebook[152] indicates:

— Evidence should be uniquely identified so that arguments can uniquely reference the evidence.

— Evidence should be verifiable and auditable.

— Evidence should be protected and controlled by configuration management.

— Evidence needs to be accompanied by the metadata needed to properly use it within the assurance case.

This last point is simply a restatement of what testing is supposed to achieve related to the assurance case.

7.4 Certifications and accreditations

Every aspect having potential significant consequences for meeting the top-level claim or for the confidence of key stakeholders has a potential place in a full assurance case evidence. It should not only give coherent confidence to stakeholders, but also contain enough information to be used by certifiers and accreditors.

The aviation and nuclear power industries have long histories of standards and certifications, and the security community in ISO/IEC JTC 1/SC 27 has been working on the topic of assurance for many years. Security examples include the Common Criteria, FIPS 140 for cryptology, and ISO/IEC 27002, combined with ISO/IEC 27001 (formerly with UK standard BS 7799-2:2002) form a basis for an Information Security Management System (ISMS) certification of an operational system. The UK Ministry of Defence and Civil Aviation Authority have also produced standards of interest including assurance-case-based standards for reliability, maintainability and safety — e.g. References [141], [144], [145], [84] and [85].

The safety community (e.g., commercial aviation) has used certification (designated agent or licensure) of key personnel as part of its approaches. A number of safety and computer security certifications exist from management-oriented ones to technical ones about specific products, e.g., certifications from the International Information Systems Security Certification Consortium (ISC) and the SANS Institute.

8 ISO/IEC/IEEE 15026 (all parts) and integrity levels

8.1 General

Integrity levels are suitable for use for certain levels of risk or to support an assurance case and impose criteria especially on the project, evidence collected and system. An integrity level can be viewed as a representation of the degree of confidence that is used to reach agreement among stakeholders of a system about risks related to that system.

ISO/IEC 15026-3 first establishes an integrity level framework. The remainder of the standard covers defining integrity levels, using integrity levels, determining system or product integrity levels using risk analyses, assigning system element integrity levels, meeting integrity level requirements using evidence, and agreements and approvals involving authorities (see 5.4).

Integrity level requirements reflect what is required to achieve and show that the system or system element has (or had or will have) the properties claimed by its integrity level. A system's integrity level states what would be adequate in terms of properties of the entire system. Thus, showing the properties has a basic role in showing the meeting of larger claims involving the system and its environment including desirable or undesirable consequences. If such larger claims are not made, achieving and showing system element integrity levels supplies a basic part of showing the top-level claim regarding the system itself.

In practice, integrity levels are often discussed in terms emphasizing the evidence needed to meet the integrity level requirements and thereby provide evidence for the arguments supporting claims regarding properties of the system itself. However, the quality of the arguments justifying meeting integrity level requirements as showing the achievement of its related integrity level is also important

because of the effect of that quality on uncertainties. Argument-, evidence- and assumption-related uncertainties are a part of establishing integrity level requirements.

NOTE Integrity levels and standards utilizing them have a significant history especially in safety. Integrity levels in safety-related standards are defined in multi-level sets addressing varying degrees of stringency and/or uncertainty of their achievement with higher levels providing higher stringency and lower uncertainty. One example safety standard is IEC 61508. Elsewhere, similar schemes are used with different labels, e.g., "conformance classes".

8.2 Risk analysis

Risk analysis establishes the required integrity level for the entire system. Risk analysis is an ongoing and iterative process that should balance what is not yet knowable with what needs to be known. The integrity levels resulting from risk analysis are a translation of the values of consequences into the occurrences and timings of conditions or behaviours of the system. This translation is propagated to the integrity levels internal to the system and of its dependences as they are also in terms of occurrences and timings. Thus, integrity levels are a codification of what is needed to be done and shown for various ranges and severities of limitations on property values and their associated uncertainties.

ISO/IEC/IEEE 15026 (all parts) does not cover risk analysis in detail. Many standards and guidance documents exist that offer guidelines for risk analysis and can aid in the identification of potential adverse consequences. IEC 61508 and IEC 31010 provide approaches for risk analysis. As safety-specific terminology is used in IEC 31010, the terms "hazard" and "harm" should be interpreted as "dangerous condition" and "adverse consequence" respectively. IEC 60300 also provides guidance.

Other specialized standards include ISO 13849 on machinery, ISO 14620 on space systems, ISO 19706 on fire, ISO/TS 25238 on health informatics, ISO/IEC 27005 on information security and UK CAP 760 on air traffic and airports. Also of possible interest are the more general risk management standards ISO/IEC 16085 and ISO 31000.

9 ISO/IEC/IEEE 15026 (all parts) and the life cycle

9.1 General

ISO/IEC 15026-4 provides a *process view* for systems and software assurance by providing a statement of purpose and a set of outcomes suitable for systems and software assurance. The concept of a process view is formulated and described in ISO/IEC/IEEE 12207:2017, Annex E and ISO/IEC/IEEE 15288:2015, Annex E. Like a process, the description of a process view includes a statement of purpose and outcomes. Unlike a process, the description of a process view does not include activities and tasks. Instead, the description includes guidance and recommendations explaining how the outcomes can be achieved by employing the activities and tasks of the various processes in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207.

The system life cycle processes are provided in ISO/IEC/IEEE 15288 and software life cycle processes in ISO/IEC/IEEE 12207.

The processes, activities and tasks, as well as guidance and recommendations of process views all have to be performed in the context of a life cycle model. The ISO/IEC/IEEE 24748 series is intended to facilitate the joint usage of the process content of the two life cycle process standards. The ISO/IEC/IEEE 24748 series provides unified and consolidated guidance on the life cycle management of systems and software. Its purpose is to help ensure consistency in system concepts and life cycle concepts, models, stages, processes, process application, iteration and recursion of processes during the life cycle, key points of view, adaptation and use in various domains. ISO/IEC/IEEE 24748-1 illustrates the use of a life cycle model for systems in the context of ISO/IEC/IEEE 15288 and provides a corresponding illustration of the use of a life cycle model for software in the context of ISO/IEC/IEEE 12207.

ISO/IEC 15026-4 gives the user the freedom to choose whether they use a specific artefact called an "assurance case" or document the assurance-related information in other documents. The point is to achieve the top-level claim and then to show the achievement of the claim for the value of a critical

property for a relevant stakeholder. Life cycle processes, activities and tasks need to reflect both realizing an adequate system and being sure that the system is adequate by showing that achievement to the required confidence of the stakeholders.

Users of ISO/IEC 15026-4 may require risk assessment and risk management, measurement and requirements processes that are more fully elaborated than the treatments provided in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207. Three International Standards, ISO/IEC 16085, ISO/IEC/IEEE 15939 and ISO/IEC/IEEE 29148 are designed to be used with ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 to provide more detail for these three processes. Other standards that provide useful requirements and guidance for selected processes are ISO/IEC/IEEE 15289 for documentation resulting from the execution of life cycle processes and ISO/IEC/IEEE 16326 for the project management process.

ISO/IEC/IEEE 15026 (all parts) is intended to be compatible with these life cycle process standards. The goals of assurance, the selection of claims to be assured, assurance-related planning and the construction and maintenance of the assurance case have influences within all life cycle processes.

9.2 Assurance activities in the life cycle

The execution of a planned and systematic set of assurance activities is needed to provide grounds for confidence in system properties. These activities are designed to ensure that both processes and systems conform to their requirements, standards and guidance and defined procedures[147]. "Processes" in this context, include all of the activities involved in the design, development and sustainment of the system. For software, "software products" include the software itself, the data associated with it, its documentation and supporting and reporting paperwork produced as part of the software process (e.g., test results and assurance arguments) as well as whatever else is needed to complete the assurance case. The "requirements" include requirements for the properties that should be exhibited, ultimately based on requirements to limit, reduce or manage property-related costs and losses. The "standards and guidance" may be technical, defining the technologies that can be used in the system or software, or they may be non-technical, defining aspects of the process that are further delineated by the "procedures" that make satisfaction of the system's requirements possible.

Management of life cycle activities includes handling both the activities directly involving the assurance-related information and the effect that the assurance-related information has on other activities. This management is best performed when the top-level claims are considered from the beginning of concept development, used to influence all activities and systems[142] and Appendix B in Reference [84], and became an integral part of the overall engineering process. These activities could all be done only if the system and the body of information showing achievement of those claims were being developed concurrently.

This parallel nature of development rationale and argument is but one of the advantages of concurrent development of the system and its assurance case. The development process and the system can be aimed not only at achieving the claim but doing so in a way that can be shown to be adequate by the assurance case. The assurance case influences the system by causing it to be developed in such a way that an argument is more practical to construct. This often results in a simpler system (at least internally), a system whose system elements can be used in isolation to show certain sub-claims, and an arrangement of system elements such that reasoning about the composition is both within the state of the art and practical. Concurrent processes can include requirements covering more conditions and events as well as adequate resilience, methods being used that produce few faults and validation or verification being targeted to what needs to be shown and showing that adequately.

10 Summary

This document has been written to provide users of ISO/IEC/IEEE 15026 (all parts) an adequate understanding of the concepts and terminology used in ISO/IEC/IEEE 15026 (all parts) that previously may not have been shared across the communities served. The explanations of what is covered in each part of ISO/IEC/IEEE 15026 should provide a basis for selecting and using those parts as well as a rationale behind the organization of ISO/IEC/IEEE 15026 (all parts).

Bibliography

[1] ISO Guide 51:2014, *Safety aspects — Guidelines for their inclusion in standards*

[2] ISO Guide 73:2009, *Risk management — Vocabulary — Guidelines for use in standards*

[3] ISO/IEC 2382, *Information technology — Vocabulary*

[4] ISO 2394, *General principles on reliability for structures*

[5] ISO 9000, *Quality management systems — Fundamentals and vocabulary*

[6] ISO 9241-400, *Ergonomics of human — system interaction — Part 400: Principles and requirements for physical input devices*

[7] ISO 12100, *Safety of machinery — General principles for design — Risk assessment and risk reduction*

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

[9] ISO 13849 (all parts), *Safety of machinery — Safety-related parts of control systems*

[10] ISO 14620 (all parts), *Space systems — Safety requirements*

[11] ISO 14625, *Space systems — Ground support equipment for use at launch, landing or retrieval sites — General requirements*

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

[13] ISO/IEC/IEEE 15289, *Systems and software engineering — Content of life-cycle information items (documentation)*

[14] ISO/IEC 15408 (all parts), *Information technology — Security techniques — Evaluation criteria for IT security*

[15] ISO/IEC/TR 15443 (all parts), *Information technology — Security techniques — Security assurance framework*

[16] ISO/IEC/TR 15446, *Information technology — Security techniques — Guidance for the production of protection profiles and security targets*

[17] ISO/IEC/IEEE 15939, *Systems and software engineering — Measurement process*

[18] ISO/IEC/IEEE 16085[3)], *Systems and software engineering — Life cycle processes — Risk management*

[19] ISO/IEC/IEEE 16326, *Systems and software engineering — Life cycle processes — Project management*

[20] ISO/TR 16982, *Ergonomics of human-system interaction — Usability methods supporting human-centred design*

[21] ISO/IEC 18014 (all parts), *Information technology — Security techniques — Time-stamping services*

[22] ISO/TR 18529, *Ergonomics — Ergonomics of human-system interaction — Human-centred lifecycle process descriptions*

[23] ISO 19706, *Guidelines for assessing the fire threat to people*

3) Under preparation.

[24] ISO/IEC 19770 (all parts), *Information technology — IT asset management*

[25] ISO/IEC/TR 19791, *Information technology — Security techniques — Security assessment of operational systems*

[26] ISO 20282 (all parts), *Ease of operation of everyday products*

[27] ISO/IEC 21827, *Information technology — Security techniques — Systems Security Engineering — Capability Maturity Model® (SSE-CMM®)*

[28] ISO/IEC/IEEE 24748 (all parts), *Systems and software engineering — Life cycle management*

[29] ISO/IEC 25000, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*

[30] ISO/IEC 25010, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*

[31] ISO/IEC 25012, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model*

[32] ISO/IEC 25020, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Measurement reference model and guide*

[33] ISO/IEC 25030, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Quality requirements*

[34] ISO/IEC 25040, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process*

[35] ISO/IEC 25051, *Software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing*

[36] ISO/TS 25238, *Health informatics — Classification of safety risks from health software*

[37] ISO/IEC 27000, *Information technology — Security techniques — Information security management systems — Overview and vocabulary*

[38] ISO/IEC 27001, *Information technology — Security techniques — Information security management systems — Requirements*

[39] ISO/IEC 27002, *Information technology — Security techniques — Code of practice for information security controls*

[40] ISO/IEC 27004, *Information technology — Security techniques — Information security management — Monitoring, measurement, analysis and evaluation*

[41] ISO/IEC 27005, *Information technology — Security techniques — Information security risk management*

[42] ISO/IEC 27006, *Information technology — Security techniques — Requirements for bodies providing audit and certification of information security management systems*

[43] ISO/IEC 27011, *Information technology — Security techniques — Code of practice for Information security controls based on ISO/IEC 27002 for telecommunications organizations*

[44] ISO/IEC 27033 (all parts), *Information technology — Security techniques — Network security*

[45] ISO/TR 27809, *Health informatics — Measures for ensuring patient safety of health software*

[46] ISO 28003, *Security management systems for the supply chain — Requirements for bodies providing audit and certification of supply chain security management systems*

[47] ISO/IEC/IEEE 29148:2018, *Systems and software engineering — Life cycle processes — Requirements engineering*

[48] ISO 31000, *Risk management — Guidelines*

[49] ISO/IEC/IEEE 42010, *Systems and software engineering — Architecture description*

[50] ISO/IEC 90003, *Software engineering — Guidelines for the application of ISO 9001:2008 to computer software*

[51] IEC 31010, *Risk management — Risk assessment techniques*

[52] IEC 60050-192, *International electrotechnical vocabulary — Part 192: Dependability*

[53] IEC 60300 (all parts), *Dependability management*

[54] IEC 60812, *Analysis techniques for system reliability — Procedure for failure mode and effects analysis (FMEA)*

[55] IEC 61025, *Fault tree analysis (FTA)*

[56] IEC 61078, *Reliability block diagrams*

[57] IEC 61508 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems*

[58] IEC 61511, *Functional safety — Safety instrumented systems for the process industry sector*[several parts]

[59] IEC 61882, *Hazard and operability studies (HAZOP studies) — Application guide*

[60] IEC 62741, *Reliability of systems, equipment, and components. Guide to the demonstration of dependability requirements. The dependability case*

[61] IEEE Std 1228-1994 (Reaff 2010), *IEEE Standard for Software Safety Plans*

[62] IEEE Std 1633-2016*(Revision of IEEE Std 1633-2008), IEEE Recommended Practice on Software Reliability*

[63] Abran A., & Moore J.W. (Executive editors); Pierre Bourque, Robert Dupuis, Leonard Tripp (Editors). Guide to the Software Engineering Body of Knowledge. 2004 Edition. Los Alamitos, California: IEEE Computer Society, Feb. 16, 2004. Available at http://www.swebok.org

[64] Adamski A., & Westrum R. "*Requisite imagination: The fine art of anticipating what might go wrong.*" In: [55], p. 193-220, 2003

[65] Adelard. The Adelard Safety Case Development Manual. Available at http://www.adelard.com/web/hnav/resources/ascad

[66] Alexander I. *Systems Engineering Isn't Just Software.* 2001. Available at

[67] Allen J.H., Barum S., Ellison R.J., McGraw G., Mead N.R. *Software Security Engineering: A Guide for Project Managers.* Addison-Wesley, 2008

[68] Altman W., Ankrum T., Brach W. *Improving Quality and the Assurance of Quality in the Design and Construction of Nuclear Power Plants: A Report to Congress.* U.S. Nuclear Regulatory Commission: Office of Inspection and Enforcement, 1987

[69] Anderson J.P. Computer Security Technology Planning Study Volume I, ESDTR-73-51, Vol. I, Electronic Systems Division, Air Force Systems Command, Hanscom Field, Bedford, MA 01730, Oct. 1972.

[70] Anderson R.J. *Security Engineering: A Guide to Building Dependable Distributed Systems.* John Wiley and Sons, Second Edition, 2008

[71] Ankrum T.S., & Kromholz A.H. "Structured Assurance Cases: Three Common Standards," Ninth IEEE International Symposium on High-Assurance Systems Engineering (HASE'05), pp. 99-108, 2005

[72] Armstrong J.M., & Paynter S.P. The Deconstruction of Safety Arguments through Adversarial Counter-argument. School of Computing Science, Newcastle University CS-TR-832, 2004

[73] Atchison B., Lindsay P., Tombs D. A Case Study in Software Safety Assurance Using Formal Methods. Technical Report No. 99-31. Sept. 1999

[74] Number ATSIN 17 Issued 9. Lapses and Mistakes. Air Traffic Services Information Notice, Safety Regulation Group, ATS Standards Department. UK Civil Aviation Authority, August 2002

[75] Bahill A.T., & Gissing B. Re-evaluating Systems Engineering Concepts Using Systems Thinking. *IEEE Trans. Syst. Man Cybern. C.* 1998 November, **28** (4) pp. 516–527

[76] Berg C.J. *High-Assurance Design: Architecting Secure and Reliable Enterprise Applications.* Addison Wesley, 2006

[77] Bernstein Lawrence, & Yuhas C. M. Trustworthy Systems through Quantitative Software Engineering. Wiley-IEEE Computer Society Press, 2005. About reliability not security

[78] Bishop M., & Engle S. The Software Assurance CBK and University Curricula. Proceedings of the 10th Colloquium for Information Systems Security Education, 2006

[79] Bishop M. *Computer Security: Art and Practice.* Addison-Wesley, 2003

[80] Bishop P., & Bloomfield R. A Methodology for Safety Case Development. Industrial Perspectives of Safety-critical Systems: Proceedings of the Sixth Safety-critical Systems Symposium, Birmingham. 1998

[81] Bishop P., & Bloomfield R. The SHIP Safety Case Approach. SafeComp95, Belgirate, Italy. Oct 1995

[82] Buehner M.J., & Cheng P.W. Causal Learning. In: *The Cambridge Handbook of Thinking and Reasoning*, (Morrison R., & Holyoak K.J. eds.). Cambridge University Press, 2005, pp. 143–68.

[83] Cannon J.C. *Privacy.* Addison Wesley, 2005

[84] CAP 670 Air Traffic Services Safety Requirements. UK Civil Aviation Authority Safety Regulation Group, 2012

[85] CAP 730 Safety Management Systems for Air Traffic Management A Guide to Implementation. UK Civil Aviation Authority Safety Regulation Group, 12 September 2002

[86] CAP 760 Guidance on the Conduct of Hazard Identification, Risk Assessment and the Production of Safety Cases For Aerodrome Operators and Air Traffic Service Providers, 10 December 2010

[87] Chung L. *Non-Functional Requirements in Software Engineering.* Kluwer, 1999

[88] Clark D.D., & Wilson D.R. A Comparison of Commercial and Military Computer Security Policies, Proc. of the 1987 IEEE Symposium on Security and Privacy, IEEE, pp. 184-196, 1987

[89] CNSS. National Information Assurance Glossary, CNSS Instruction No. 4009, 26 April 2010. Available at http://www.cnss.gov/full-index.html

[90] Committee on Information Systems Trustworthiness. *Trust in Cyberspace, Computer Science and Telecommunications Board.* National Research Council, 1999

[91] Committee on National Security Systems (CNSS) Instruction 4009: National Information Assurance (IA) Glossary. Revised May 2003. Available at http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf

[92] Common Criteria Recognition Arrangement (CCRA). Common Criteria v3.1 Revision 2. NIAP September 2007. Available at https://www.commoncriteriaportal.org/.

[93] Common Weaknesses Enumeration. MITRE, 2012. Available at http://cwe.mitre.org

[94] Cooke N.J., Gorman J.C., Winner J.L. Team Cogitation. p. 239-268 In: [43]

[95] Courtois P.-J. *Justifying the Dependability of Computer-based Systems: With Applications in Nuclear Engineering.* Springer, 2008

[96] Cranor L., & Garfinkel S. *Security and Usability: Designing Secure Systems that People Can Use.* O'Reilly, 2005

[97] Dayton-Johnson J. Natural disasters and adaptive capacity. OECD Development Centre Research programme on: Market Access, Capacity Building and Competitiveness. Working Paper No. 237 DEV/DOC(2004)06, August 2004

[98] Department of Defense Directive 8500.1 (6 February 2003). Information Assurance (IA), Washington, DC: US Department of Defense, ASD(NII)/DoD CIO, April 23, 2007. Available at http://www.dtic.mil/whs/directives/corres/pdf/850001p.pdf

[99] Department of Defense Strategic Defense Initiative Organization. Trusted Software Development Methodology, SDI-S-SD-91-000007, vol. 1, 17 June 1992

[100] Department of Homeland Security National Cyber Security Division's. "Build Security In" (BSI) web site, 2012, http://buildsecurityin.us-cert.gov

[101] Dependability Research Group. Safety Cases. University of Virginia, Available at: https://web.archive.org/web/20131229153301/http://dependability.cs.virginia.edu/info/Safety_Cases

[102] Despotou G., & Kelly T. Extending the Safety Case Concept to Address Dependability, Proceedings of the 22nd International System Safety Conference, 2004

[103] Dowd M., McDonald J., Schuh J. *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities.* Addison-Wesley, 2006

[104] Dunbar K., & Fugelsang J. Scientific Thinking and Reasoning. In: [59], p. 705–727

[105] Durso F.T., Nickerson R.S., Dumais S.T., Lewandowsky S., Perfect T.J. eds. *Handbook of Applied Cognition* 2nd edition. Wiley, 2007

[106] Ellsworth P.C. Legal Reasoning. In: [59], p. 685–704

[107] Ericsson K.A., Charness N., Feltovich P.J., Hoffman R.R. eds. *The Cambridge Handbook of Expertise and Expert Performance.* Cambridge University Press, 2006

[108] Fenton N., Littlewood B., Neil M., Strigini L., Sutcliffe A., Wright D. Assessing dependability of safety critical systems using diverse evidence. *IEE Proc. Softw.* 1998 **145** (1) pp. 35–39

[109] Gasser M. Building a Secure Computer System. Van Nostrand Reinhold, 1988. Available at: https://web.archive.org/web/20120526151008/http://deke.ruc.edu.cn/wshi/readings/cs02.pdf

[110] Gray J.W. Probabilistic Interference. Proceedings of the IEEE Symposium on Research in Security and Privacy. IEEE, p. 170-179, 1990

[111] Greenwell W., Strunk E., Knight J. Failure Analysis and the Safety-Case Lifecycle. IFIP Working Conference on Human Error, Safety and System Development (HESSD) Toulouse, France. Aug 2004

[112] Greenwell W.S., Knight J.C., Pease J.J. A Taxonomy of Fallacies in System Safety Arguments. 24th International System Safety Conference, Albuquerque, NM, August 2006

[113] Hall A., & Chapman R. Correctness by Construction: Developing a Commercial Secure System. *IEEE Softw.* 2002 Jan/Feb, **19** (1) pp. 18–25

[114] Herrmann D.S. *Software Safety and Reliability.* IEEE Computer Society Press, 1999

[115] Hoglund G., & McGraw G. *Exploiting Software: How to break code.* Addison-Wesley, 2004

[116] Hollnagel E., Woods D.D., Leveson N. eds. *Resilience Engineering: Concepts and Precepts.* Ashgate Pub Co, 2006

[117] Hollnagel E. ed. *Handbook of cognitive task design.* Lawrence Erlbaum Associates, 2003

[118] Hollnagel E. Human Error: Trick or Treat? In: [43], p. 219–238

[119] Hollnagel E. *Barriers and Accident Prevention.* Ashgate, 2004

[120] Holyoak K.J., & Morrison R.G. eds. *The Cambridge Handbook of Thinking and Reasoning.* Cambridge University Press, 2005

[121] Howard M., & LeBlanc D.C. *Writing Secure Code.* Microsoft Press, Second Edition, 2002

[122] Howard M., & Lipner S. *The Security Development Lifecycle.* Microsoft Press, 2006

[123] Howell C. Assurance Cases for Security Workshop (follow-on workshop of the 2004 Symposium on Dependable Systems and Networks), June, 2005

[124] Kazman R., Asundi J., Klein M. *Making Architecture Design Decisions: An Economic Approach*, SEI-2002-TR-035. Software Engineering Institute, Carnegie Mellon University, 2002

[125] Kazman R., Klein M., Clements P. ATAM: Method for Architecture Evaluating the Quality Attributes of a Software Architecture. Technical Report CMU/SEI-200-TR004. Software Engineering Institute, Carnegie Mellon University, 2000

[126] Kelly T. Arguing Safety — A Systematic Approach to Managing Safety Cases. Doctorial Thesis — University of York: Department of Computer Science. Sept 1998

[127] Kelly T. Reviewing Assurance Arguments — A Step-by-Step Approach. Workshop on Assurance Cases for Security: The Metrics Challenge, International Conference on Dependable Systems and Networks, 2007

[128] Kelly T., & Weaver R. The Goal Structuring Notation — A Safety Argument Notation. Workshop on Assurance Cases: Best Practices, Possible Obstacles, and Future Opportunities, Florence, Italy. July 2004

[129] Ladkin P. The Pre-Implementation Safety Case for RVSM in European Airspace is Flawed. 29 Aug 2002. Available at http://www.rvs.uni-bielefeld.de/publications/Reports/SCflawed-paper.html

[130] Landwehr C. Computer Security. IJIS. 2001, **1** pp. 3–13

[131] Lautieri S., Cooper D., Jackson D. SafSec: Commonalities Between Safety and Security Assurance. Proceedings of the Thirteenth Safety Critical Systems Symposium — Southampton, 2005

[132] LeBoeuf R.A., & Shafir E.B. Decision Making. In: [59], p. 243–266

[133] Leveson N. A Systems-Theoretic Approach to Safety in Software-Intensive Systems, *IEEE Trans. Dependable Sec. Comput.* 2004, **1** (1) pp. 66-86

[134] Lipner S., & Howard M. The Trustworthy Computing Security Development Lifecycle, Microsoft, 2005. Available at http://msdn.microsoft.com/en-us/library/ms995349.aspx