IEEE STANDARDS ASSOCIATION

IEEE Trial-Use Standard— Adoption of ISO/IEC TR 15026-1:2010

Systems and Software Engineering— Systems and Software Assurance— Part 1: Concepts and Vocabulary

IEEE Computer Society

Sponsored by the
Software & Systems Engineering Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

3 June 2011

IEEE Std 15026-1™-2011

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

IEEE Trial-Use Standard— Adoption of ISO/IEC TR 15026-1:2010

Systems and Software Engineering— Systems and Software Assurance— Part 1: Concepts and Vocabulary

Sponsor

Software & Systems Engineering Standards Committee
of the
IEEE Computer Society

Approved 31 March 2011

IEEE-SA Standards Board

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Abstract: This trial-use standard adopts ISO/IEC TR 15026-1:2010, which defines terms and establishes an extensive and organized set of concepts and their relationships for software and systems assurance, thereby establishing a basis for shared understanding of the concepts and principles central to ISO/IEC 15026 across its user communities. It provides information to users of the subsequent parts of ISO/IEC 15026, including the use of each part and the combined use of multiple parts. Coverage of assurance for a service being operated and managed on an ongoing basis is not covered in ISO/IEC 15026.

Keywords: argument, assurance case, claim, computers, computer software, data processing, data security, definitions, IEEE 15026-1, integrity level, life cycle, process, product design, quality assurance, quality control, reliability, safety, security, software assurance, software engineering, software engineering techniques, system, systems assurance, systems engineering, uncertainty, vocabulary

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

Copyright © 2011 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 3 June 2011. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-0-7381-6601-8 STD97092
Print: ISBN 978-0-7381-6630-8 STDPD97092

IEEE prohibits discrimination, harassment and bullying. For more information, visit http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html.
No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

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 or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Introduction

This introduction is not part of IEEE Std 15026-1-2011, IEEE Trial-Use Standard—Adoption of ISO/IEC 15026-1:2010, Systems and Software Engineering—Systems and Software Assurance—Part 1: Concepts and Vocabulary.

The IEEE Software and Systems Engineering Standards Committee (S2ESC) has undertaken a long-term program to harmonize its standards with those of ISO/IEC JTC 1/SC 7, the international standards committee for software and systems engineering. In areas of overlap, one organization sometimes adopts the relevant standard from the other organization, or the two organizations cooperate to produce a single joint standard. In this case, S2ESC has chosen to adopt a relevant document from SC 7.

SC 7 is currently preparing a four-part standard concerning software and systems assurance. Part 1 provides the concepts and vocabulary used in the other parts of the standard. The other three parts are at varying stages of completion; IEEE will consider them for adoption as they are completed.

The plan being followed by SC 7 calls for revising Part 1 following the completion of the other three parts to ensure that Part 1 continues to provide a good foundation of concepts and vocabulary for the other parts. Accordingly, the current ISO publication of 15026-1 is a Type 2 Technical Report, which is defined as a document still under technical development. The revision of Part 1 will be an International Standard.

An IEEE “Trial Use” standard corresponds well to an ISO type 2 technical report. IEEE Trial Use standards are effective for only two years and provide the opportunity for users of the standard to return comments for consideration during revision. Such comments would be useful for SC 7 in performing their revision of Part 1, and, hopefully, would result in an improved revision for subsequent adoption after the trial use period expires.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Copyright © 2011 IEEE. All rights reserved.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association web site at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA web site at http://standards.ieee.org.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of this trial-use standard may require use of subject matter covered by patent rights. By publication of this trial-use standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this trial-use standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Trial use

Publication of this trial-use standard for comment and criticism has been approved by the Institute of Electrical and Electronics Engineers. Trial-use standards are effective for 24 months from the date of publication. Comments for revision will be accepted for 18 months after publication. Suggestions for revision should be directed to the Secretary, IEEE-SA Standards Board, 445 Hoes Lane, Piscataway, NJ 08854-4141, and should be received no later than 3 December 2012. It is expected that following the 24-month period, this trial-use standard, revised as necessary, shall be submitted to the IEEE-SA Standards Board for approval as a full-use standard.

Copyright © 2011 IEEE. All rights reserved.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Participants

At the time this trial-use standard was submitted to the IEEE-SA Standards Board for approval, the Working Group for Life Cycle Processes had the following membership:

James W. Moore, *IEEE Computer Society Liaison to ISO/IEC JTC 1/SC 7*

The following members of the individual balloting committee voted on this trial-use standard. Balloters may have voted for approval, disapproval, or abstention.

Ed Addario
Edward Addy
Johann Amsenga
H. Stephen Berger
Pieter Botman
Lyle Bullock
Sue Carroll
Keith Chow
Paul Croll
Geoffrey Darnton
David Deighton
Sourav Dutta
Andrew Fieldsend
David Friscia
David Fuschi
Ignacio Marin Garcia
Lewis Gray
Ron Greenthaler
Randall Groves
John Harauz
Richard Hilliard
Werner Hoelzl
Glenn Hoffman
Robert Holibaugh
Bernard Homes
Peter Hung
Atsushi Ito
Cheryl Jones
Piotr Karocki
Dwayne Knirk
Thomas Kurihara
George Kyle
Susan Land
David Leciston
Daniel Lindberg
Greg Luri
Wayne W. Manges
Edward McCall
Kathryn Moland
James Moore
Michael S. Newman
Mirko Palazzo
William Petit
Ulrich Pohl
Iulian Profir
Annette Reilly
Robert Robinson
Randall Safier
Bartien Sayogo
Robert Schaaf
Maud Schlich
Richard Schrenker
Stephen Schwarm
Gil Shultz
Carl Singer
James Sivak
Luca Spotorno
Friedrich Stallinger
Thomas Starai
Walter Struppler
Marcy Stutzman
Anusheel Tandon
David Walden
Paul Work
Oren Yuen
Janusz Zalewski

When the IEEE-SA Standards Board approved this trial-use standard on 31 March 2011, it had the following membership:

Richard H. Hulett, *Chair*

John Kulick, *Vice Chair*

Robert M. Grow, *Past Chair*

Judith Gorman, *Secretary*

Masayuki Ariyoshi
William Bartley
Ted Burse
Clint Chaplin
Wael Diab
Jean-Philippe Faure
Alexander Gelman
Paul Houzé
Jim Hughes
Joseph L. Koepfinger*
David J. Law
Thomas Lee
Hung Ling
Oleg Logvinov
Ted Olsen
Gary Robinson
Jon Walter Rosdahl
Sam Sciacca
Mike Seavey
Curtis Siller
Phil Winston
Howard L. Wolfman
Don Wright

*Member Emeritus

Copyright © 2011 IEEE. All rights reserved.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Also included are the following non-voting IEEE-SA Standards Board liaisons:

Satish Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Lisa Perry
IEEE Standards Program Manager, Document Development

Malia Zaman
IEEE Standards Program Manager, Technical Program Development

Jodi Haasz
IEEE Senior Program Manager, International Standards Programs

Copyright © 2011 IEEE. All rights reserved.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Contents of IEEE adoption of ISO/IEC TR 15026-1:2010(E)

ISO/IEC TR 15026-1:2010..1

Copyright © 2011 IEEE. All rights reserved.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

IEEE Trial-Use Standard— Adoption of ISO/IEC TR 15026-1:2010

Systems and Software Engineering— Systems and Software Assurance— Part 1: Concepts and Vocabulary

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

Copyright © 2011 IEEE. All rights reserved.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

TECHNICAL
REPORT

ISO/IEC TR 15026-1

First edition
2010-08-15

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 TR 15026-1:2010(E)

© ISO/IEC 2010

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

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 accepts no 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. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2010

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 either ISO at the address below or ISO's member body in the country of the requester.

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

Published in Switzerland

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Contents

Page

Foreword ... v
Introduction ... vi
1 Scope ... 1
2 Terms and definitions ... 1
3 Document purpose and audience ... 4
4 Organization of report ... 4
5 Basic concepts ... 4
5.1 Introduction ... 4
5.2 Stakeholders ... 4
5.3 System and Product ... 6
5.4 Uncertainty ... 6
5.5 Assurance ... 6
6 How to use multiple parts of ISO/IEC 15026 ... 7
6.1 Introduction ... 7
6.2 Initial usage concerns ... 7
6.3 Internal structure of parts ... 8
6.4 Relationships among parts of ISO/IEC 15026 ... 9
6.5 Authorities ... 9
6.6 Mitigation of ambiguity ... 9
7 Assurance Case ... 10
7.1 Introduction ... 10
7.2 Claims ... 13
7.3 Arguments ... 23
7.4 Evidence ... 34
7.5 Management and life cycle of assurance case ... 39
7.6 Decision making using the assurance case ... 40
8 ISO/IEC 15026 and integrity levels ... 42
8.1 Introduction ... 42
8.2 Defining integrity levels ... 43
8.3 Establishing integrity levels ... 44
8.4 Planning and performing ... 45
8.5 Conditions and their initiating or transitioning events ... 46
8.6 Issues ... 46
8.7 Outcomes ... 48
8.8 Summary ... 48
9 ISO/IEC 15026 and life cycle processes: 15288/12207 ... 49
9.1 Introduction ... 49
9.2 Technical processes ... 50
9.3 Transition, Operation, Maintenance and Disposal ... 55
9.4 Organization processes ... 56
10 Summary ... 57
Annex A (informative) Frequently asked questions ... 58
Annex B (informative) Difficulties with terms and concepts ... 59
Annex C (informative) ISO/IEC 15026 relationships to standards ... 61
Annex D (informative) Phenomena ... 64

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Annex E (informative) **Security** **68**
Annex F (informative) **Selected Related Standards** **79**
Bibliography **85**

Tables

Table 1 — Examples of Stakeholders 5
Table 2 — Some time- and resource-related properties 21
Table 3 — Example ways of showing something is true 24
Table 4 — Communities with different viewpoints and approaches to reasoning 25
Table 5 — Relationship aspects that are possible bases for or relevant to arguments 30
Table D-1 — Some kinds and sources of phenomena 64

List of Figures

Figure 1 — Fragment of Structure 11
Figure 2 — Claim 16
Figure 3 — Argument Context 23
Figure 4 — Simple State Model 28
Figure 5 — Simplified "cause and effect" chains 28
Figure 6 — System and Environment 42
Figure 7 — Two actors cause transitions 47
Figure 8 — Life cycle process groups 49
Figure C-1 — Some relationships among standards 63

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

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.

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

The main task of the joint technical committee 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.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;
- type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard (“state of the art”, for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

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.

ISO/IEC TR 15026-1, which is a Technical Report of type 2, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

ISO/IEC 15026 consists of the following parts, under the general title *Systems and software engineering — Systems and software assurance*:

- *Part 1: Concepts and vocabulary*
- *Part 2: Assurance case*

System integrity levels and assurance in the life cycle will form the subjects of future parts.

ISO/IEC 15026:1998, IEEE Std 1228-1994 and IEEE Standard for Safety Plan were used as base documents in the development of ISO/IEC TR 15026-1.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Introduction

Within software and systems assurance and closely related fields, many specialties and subspecialties share concepts but have differing vocabularies and perspectives. This part of ISO/IEC 15026 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 all parts of ISO/IEC 15026.

This part of ISO/IEC 15026 clarifies concepts needed for understanding software and systems assurance and, in particular, those central to the use of subsequent parts of ISO/IEC 15026. This part of ISO/IEC 15026 supports intellectual mastery of software and systems assurance primarily at the level of shared concepts, issues and terminology applicable across a range of properties, application domains, and technologies.

The appreciation of the contents of this part of ISO/IEC 15026 might undergo change as work proceeds on the other parts of ISO/IEC 15026. A revision of this part of ISO/IEC 15026 reflecting any such changes is expected to be later published as an International Standard.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Systems and software engineering — Systems and software assurance —

Part 1: Concepts and vocabulary

1 Scope

This part of ISO/IEC 15026 defines terms and establishes an extensive and organized set of concepts and their relationships, thereby establishing a basis for shared understanding of the concepts and principles central to ISO/IEC 15026 across its user communities. It provides information to users of the subsequent parts of ISO/IEC 15026, including the use of each part and the combined use of multiple parts.

Coverage of assurance for a service being operated and managed on an ongoing basis is not covered in ISO/IEC 15026.

2 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

2.1
assurance
grounds for justified confidence that a claim has been or will be achieved

2.2
assurance case
representation of a claim or claims, and the support for these claims

NOTE An assurance case is reasoned, auditable artefact created to support the contention its claim or claims are satisfied. It 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).

2.3
approval authority
entity with the authority to decide that the assurance case and the extent of assurance it provides are satisfactory

NOTE 1 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 In two-party situations, 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, e.g. the purchase of off-the-shelf products developed by a single-party, the independence of the approval authority can be a relevant issue to the acquirer.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

2.4
claim
statement of something to be true including associated conditions and limitations

NOTE 1 The statement of a claim does not mean that the only possible intent or desire is to show it is true. Sometimes claims are made for the purpose of evaluating whether they are true or false or undertaking an effort to establish what is true.

NOTE 2 In its entirety, a claim conforming to ISO/IEC 15026-2 is an unambiguous declaration of an assertion with any associated conditionality giving explicit details including limitations on values and uncertainty. It could be about the future, present, or past.

2.5
design authority
person or organization that is responsible for the design of the product

2.6
failure
termination of the ability of an item to perform a required function or its inability to perform within previously specified limits

2.7
fault isolation
ability of a subsystem to prevent a fault within the subsystem from causing consequential faults in other subsystems

2.8
integrity assurance authority
independent person or organization responsible for assessment of compliance with the integrity-level-related requirements

NOTE Adapted from ISO/IEC 15026:1998, in which the definition is "The independent person or organization responsible for assessment of compliance with the integrity requirements."

2.9
integrity level
denotation of a range of values of a property

NOTE 1 Generally, the intention is that meeting these values related to the relevant items will result in maintaining system risks within limits.

NOTE 2 Adapted from ISO/IEC 15026:1998.

2.10
organization
person or a group of people and facilities with an arrangement of responsibilities, authorities and relationships

[ISO/IEC 15288:2008]

NOTE 1 This definition and notes are taken from ISO/IEC 15288:2008. The definition in ISO/IEC 15288:2008 was adapted from ISO 9000:2005.

NOTE 2 A body of persons organized for some specific purpose, such as a club, union, corporation, or society, is an organization.

NOTE 3 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.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

2.11
process

set of interrelated or interacting activities which transforms inputs into outputs

[ISO/IEC 15288:2008 and ISO/IEC 12207:2008]

NOTE This definition does not preclude the existence of a null process, activity or transformation, or of null inputs or outputs.

2.12
process view

description of how a specified purpose and set of outcomes can be achieved by employing the activities and tasks of existing processes

NOTE This definition is adapted from the description of the process view concept in ISO/IEC 15288:2008, D.3.

2.13
product

result of a process

[ISO/IEC 15288:2008 and ISO 9000:2005]

NOTE 1 Results could be components, systems, software, services, rules, documents, or many other items.

NOTE 2 “The result” could in some cases be many related individual results. However, claims usually relate to specified versions of a product.

2.14
system

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

[ISO/IEC 15288:2008]

NOTE 1 A system may be considered as a product or as the services it provides.

NOTE 2 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” may be substituted simply by a context-dependent synonym, e.g. aircraft, though this may then obscure a system principles perspective.

NOTE 3 Notes 1 and 2 are also taken from ISO/IEC 15288:2008.

2.15
system element

member of a set of elements that constitutes a system

[ISO/IEC 15288:2008]

NOTE 1 A system element is a discrete part of a system that can be implemented to fulfil specified requirements. A system element can be hardware, software, data, humans, processes (e.g. processes for providing service to users), procedures (e.g. operator instructions), facilities, materials, and naturally occurring entities (e.g. water, organisms, minerals), or any combination.

NOTE 2 Note 1 is also taken from ISO/IEC 15288:2008.

2.16
systematic failure

failure related in a deterministic way to a certain cause, which can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

3 Document purpose and audience

The primary purpose of this part of ISO/IEC 15026 is to aid users of the other parts of ISO/IEC 15026. For each topic, it first briefly covers what might be needed by engineers and technical mangers new to the topic of assurance cases or integrity levels. Lists of aspects or examples are provided for concreteness and as reminders or checklists. 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.

If a decision is made to use any parts of ISO/IEC 15026, then understanding certain concepts and terms is essential. This part of ISO/IEC 15026 provides context, concepts, and explanations to aid users in doing this as well as aiding in the usage of the other parts.

A variety of potential users of ISO/IEC 15026 exists including developers and maintainers of assurance cases and those who wish to develop, sustain, evaluate, or acquire a system that possesses specific properties of interest in such a way as to be surer of those properties. Users of this International Standard can benefit from knowing the included terms, concepts, and principles. For example, while ISO/IEC 15026 uses terms consistent with ISO/IEC 12207 and ISO/IEC 15288 and generally consistent with the ISO/IEC 25000 series, the users of ISO/IEC 15026 need to know any differences from that to which they are accustomed. The remainder of this part of ISO/IEC 15026 attempts to clarify issues of the concepts of interest to users of ISO/IEC 15026.

4 Organization of report

Clause 5 of this part of ISO/IEC 15026 covers basic concepts such as stakeholders, product, assurance, and uncertainty. Clause 6 covers some issues of which users of the future ISO/IEC 15026-2, ISO/IEC 15026-3, and ISO/IEC 15026-4 need to be initially aware. Clauses 7, 8, and 9 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 oriented to other parts. Clause 8 is for users of ISO 15026:1998, as well as of the future ISO/IEC 15026-3.

Those who have curiosity or initial questions about ISO/IEC 15026 could find it useful to take an early look at Annex A on page 58, the Frequently asked questions annex. Other annexes cover pitfalls with terminology (Annex B), ISO/IEC 15026's relationships to several other standards (Annex C), phenomena (Annex D) as a way of helping ISO/IEC 15026 users to think about possibilities, security (Annex E), and some related standards (Annex F). Annex E gives special attention to security because it is an area expected to be relatively new to many initial users of ISO/IEC 15026. However, ISO/IEC 15026 can be used for both positive concerns, such as high performance, as well as negative concerns, such as security. A bibliography is included at the end.

5 Basic concepts

5.1 Introduction

This clause covers the terms and concepts fundamental to ISO/IEC 15026: stakeholders, systems and products, uncertainty, and assurance.

5.2 Stakeholders

5.2.1 Introduction

Through their life cycle systems and software have multiple stakeholders who affect or are affected by the system and system-related activities. Stakeholders might benefit from, incur losses from, impose constraints on, or otherwise have a “stake” in the system.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

5.2.2 Kinds of stakeholders

A given system will typically have stakeholders from several of the categories in Table 1.

Table 1 — Examples of Stakeholders

Product's larger environment
Regulators
Standards bodies
Specific communities (such as government or the banking industry)
National (possibly multi-national) and international laws, regulations, treaties, and agreements
Enforcement personnel and organizations
Competitors
Entities about whom the product contains information (e.g. customers and suppliers)
Evaluators, regulators, certifiers, accreditors, and auditors
Attackers
The general public
Organizational
Sources of relevant policies (e.g. safety, security, personnel, procurement, and marketing policies)
Decision makers regarding acquisition and usage (including request for proposal writers and issuers as well as makers of decisions to acquire or use)
Authorized units within an organization
Directly related to product
Product developers and maintainers
Integrators of the system or software into a larger product (e.g. OEMs or enterprise-wide application developers)
Those involved in product transition (e.g. trainers and installers)
Product operators and administrators
End users
Others involved throughout the product's systems life cycle (e.g. sustainers and disposers)
System into which product is incorporated
Other systems interacting with the product or using the product’s services
Suppliers of services or consumables to product
Product owners and custodians
Project management
Owners and custodians of elements in the system (e.g. data)

In addition, stakeholders can include non-users whose performance, results, or interests 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, as in, “Both we and the enemy have a stake in keeping within the laws of war.” However, some in the security community and elsewhere use the term “stakeholders” in such a way as to exclude attackers. Attackers can be of many kinds and have a variety of motivations and capabilities. The issue of how hostile or malicious in intention or detrimental in action an entity would need to be to qualify as an attacker is unclear.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

A given system or project might involve more or less of the stakeholders in Table 1. Stakeholder roles and relative importance can be difficult to establish, for example, who—system funders, customers, beneficiaries, attackers, benefit gainers or loss sufferers—is more important or should have more influence on which decisions, including the importance to assurance-related decisions and importance as users of assurance-related artefacts. The existence and characteristics of potential or actual attackers can strongly influence decisions.

5.2.3 Stakeholder interests and assets

Stakeholder interests include any benefit, loss, or advantage, e.g., one says, "In the national interests" or "not in the interest of the organization" or "not in my interest." Interests include the wealth and reputations of persons about whom information is kept. Assets may also be of many kinds, including real estate, facilities, equipment, people, wealth, information or data, an executing process, or anything else that is of value to stakeholders.[1] Assets within the system and its immediate environment do not necessarily include everything that might be relevant. Examples of those assets about which the contents of the system could facilitate positive or adverse actions of any kind include shareholder value, facilities, infrastructure, spies, soldiers, and other valued objects, processes, or conditions. The relevant stakeholders whose interests are of concern usually include the system's owners and users, but developers and operators need to identify relevant stakeholder interests and assets and their value or relative importance to the development and operation of the system.

5.3 System and Product

To be consistent with ISO/IEC 15288 and 12207, ISO/IEC 15026, Systems and software assurance, uses the term "system" throughout. Users of this standard 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 is not a restriction on its use. This standard can be used in reducing uncertainty about a system's dependence on natural phenomena.

5.4 Uncertainty

Uncertainty is used in ISO/IEC 15026 as an inclusive term. It covers lack of certainty whether the uncertainty can be modelled probabilistically or not. This definition allows the term "uncertainty" to be applied to anything. 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, ISO/IEC 15026 users span many communities.

5.5 Assurance

While ISO/IEC 15026 uses a specific definition for the term as being grounds for justified confidence, for clarity ISO/IEC 15026 seldom uses the term "assurance" alone.

Generally, one needs 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 for the relevant claims for 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.

[1] The set of stakeholders whose interests are to be preserved or increased excludes adversaries and possibly others whose interests one might desire to limit, hinder, endanger, or harm. Note, however, that there may be overriding legal requirements to protect such excluded stakeholders, such as trespassers, thieves and enemy soldiers.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Nevertheless, decision makers need to obtain sufficient confidence that is adequately justified. Professionals that use this International Standard need to supply adequate grounds for such confidence and have its adequacy correctly judged by decision makers.

NOTE This need can sometimes lead to including the kinds of evidence that the relevant decision makers find most convincing.

Assurance is a term whose usage varies, but 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 may provide an improved basis for justified confidence. Even if the estimate of a parameter'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.

The term "assurance" may relate to different scopes – from the consequences in the world at large to system elements and their constituents as well as their interactions – and to any property of a system. Kinds and examples of properties are covered in 7.2.7.

Assurance may relate to (1) would the system or software as specified meet real-world needs and expectations, to (2) would or does the as-built and operated system meet the specifications, or to both (1) and (2). Specifications may be representations of static and/or dynamic aspects of the product. One may speak of an external specification, a specification related to the product-environment boundary, or a top-level specification that may contain some internal design. Specifications often include descriptions of capability, functionality, behaviour, structure, service, and responsibility including time- and resource-related aspects as well as limitations on frequency or seriousness of deviations by the product and related uncertainties. ISO/IEC 15288 and ISO/IEC 12207 as well as the IEEE standards on requirements divide these concerns into "functional" and "non-functional" ones.

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

6 How to use multiple parts of ISO/IEC 15026

6.1 Introduction

This clause covers issues regarding use of this International Standard. The topics covered are Initial usage concerns, Internal structure of parts of ISO/IEC 15026, Relationships among parts of ISO/IEC 15026, Authorities, and Mitigation of ambiguity.

6.2 Initial usage concerns

The decision to use one or more parts of ISO/IEC 15026 involves understanding their purpose, scope, and requirements and considering their fit with the user's organizations, policies, processes, practices, personnel, standards and other governing documents. The decision to use ISO/IEC 15026 can be the result of risk assessments, needs for information for decision making (e.g., decisions to launch or acquire a product), customer direction, organizational practices, or regulatory requirements.

When conformance is not required, the decision regarding use might include deciding to conform but not claim conformance, to use the standard as guidance, or to conform to or use only portions as guidance.

Decisions concerning their voluntary use need to analyse the feasibility of doing so, including existing organizational readiness (e.g., need and relevant competencies), riskiness of the situation, cost/benefit (including the amount of value affected by decisions it would support), the advantages of taking a more systematic approach to system-related engineering and management activities and decisions, and the alternative approaches available. On one hand, assurance cases are simply aids for good risk management,

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

but on the other hand, they can involve a significant change in thinking and can influence every system-related activity.

The properties and/or claims covered when using ISO/IEC 15026 are entirely up to the users of the standard who are responding to their own needs and outside requirements. Any property or claim may be selected, regardless of its importance or related risk. However, ISO/IEC 15026-2 is intended to be used for high assurance situations and not low assurance ones, and the other parts are expected to also find their primary use among higher assurance situations.

ISO/IEC 15026 or its parts can be used alone or with other standards or guidance. They can be mapped to most life cycle standards, and can use any set of well-defined qualities or properties. Annex C begins to address these issues.

NOTE Many more or less process-oriented standards exist that are useful for their specificity in the detail and methods they contain. Many of these are usable in conjunction with parts of ISO/IEC 15026.

While ISO/IEC 15026-3 will be generally backwards compatible with ISO/IEC 15026:1998, transitioning to ISO/IEC 15026-3 will require dealing with some differences. ISO/IEC 15026-3 will open 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 will concentrate more on the system itself and its integrity levels rather than on external risk analysis and also includes the creation of integrity levels. Clause 8 discusses integrity levels.

Occasionally user confusion exists concerning "should". Within ISO/IEC 15026, "should" is used "to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) ["should not"] a certain possibility or course of action is deprecated but not prohibited" ([129] page 65). No documented justification is required for doing otherwise.

A final 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.

6.3 Internal structure of parts

The parts of ISO/IEC 15026 are:

- ISO/IEC 15026-1: Concepts and vocabulary: initially a Technical Report and then revised to be an International Standard and possibly a guidance document.
- ISO/IEC 15026-2: Assurance case: will include requirements on the content and structure of the assurance case.
- ISO/IEC 15026-3: System integrity levels: will relate integrity levels to the assurance case and include requirements for their use with and without an assurance case (revision of ISO/IEC 15026:1998).
- ISO/IEC 15026-4: Assurance in the life cycle: addresses concurrent development and maintenance of the system and its assurance case, including project planning for assurance considerations.

The future ISO/IEC 15026-2, ISO/IEC 15026-3 and ISO/IEC 15026-4 have a number of aspects designed to facilitate their use. The main purpose of their structure and layout is to provide separately identifiable individual requirements or small sets of related requirements to facilitate traceability regarding conformance. This structure may make a casual or initial reading less smooth, but eases the repeated readings and references during use.

The parts have limited introductory and explanatory material but are self-contained and intended to be usable by knowledgeable persons as standalone documents.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

6.4 Relationships among parts of ISO/IEC 15026

While each of ISO/IEC 15026-2, ISO/IEC 15026-3 and ISO/IEC 15026-4 will provide a separation of concerns and may be used alone, they may be used together as they form a related set. This part of ISO/IEC 15026 provides background, concepts, and vocabulary that are applicable to all three and particularly relates Clause 7 to ISO/IEC 15026-2, Clause 8 to ISO/IEC 15026-3, and Clause 9 to ISO/IEC 15026-4.

The assurance case is relevant to a greater or lesser extent in all parts. ISO/IEC 15026-2 concentrates on the contents and structure of the assurance case. ISO/IEC 15026-3 relates integrity levels to their role in assurance cases, and ISO/IEC 15026-4 provides details on integrating the assurance case into the system life cycle processes.

While ISO/IEC 15026-3 supports its use with a ISO/IEC 15026-2-conformant assurance case, it also supports use of integrity levels without an assurance case or with an assurance case that is not entirely conformant to ISO/IEC 15026-2. However, users of ISO/IEC 15026-3 require ISO/IEC 15026-2, as parts of it are required related to integrity levels. In addition, ISO/IEC 15026-3 places a subset of the requirements in ISO/IEC 15026-2 on any assurance cases used with integrity levels, and some are also requirements ISO/IEC 15026-3 places on all risk analyses.

ISO/IEC 15026-4 addresses integrating the assurance case into the system life cycle processes and the concurrent development and maintenance of the system and its assurance case. While more extensive, its requirements are consistent with the assurance case life cycle requirements in ISO/IEC 15026-2, although it can be used with an assurance case that is not conformant to ISO/IEC 15026-2. ISO/IEC 15026-3 makes many points about what can be included in an integrity level's imposed requirements on development and maintenance or as evidence within an assurance case. ISO/IEC 15026-4 includes assurance-related concerns across the life cycle and concerns that extend beyond those directly related to the assurance case, including project planning for assurance-related considerations.

6.5 Authorities

Parts of ISO/IEC 15026 involve "authorities" as shown in Clause 3, Terms and definitions. For example, ISO/IEC 15026-3 includes obtaining agreements between the design authority and integrity assurance authority.

NOTE For example, 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 a part of ISO/IEC 15026. To the extent possible claims of conformance to parts 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 parts of ISO/IEC 15026.

Conflict of motivations, competence, diligence, and trustworthiness of any authority are potential issues. Therefore, parts of ISO/IEC 15026 calling for identification of an authority provide descriptions of their degree of independence. This allows decision makers, including potential users of systems, to consider these descriptions in deciding the degree of confidence they should have in any approval.

6.6 Mitigation of ambiguity

Clarity is needed for assurance cases, integrity levels, and defining processes. The requirements for unambiguous language within the documents it requires are explicit in ISO/IEC 15026. For example, each portion of the assurance case needs to be clear and unambiguous to its developers, reviewers, and users. Unambiguous does not necessarily imply precise or deterministic properties or measures, but rather that those properties or measures can be evaluated. Unambiguous also does not imply lack of uncertainty in measurement.

Definitions need to be clear. The variety of definitions that exist among the relevant audiences of the systems and software communities and their specialties and subspecialties and within ISO publications means that

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

multiple-word terms or phrases often need to be used. Definitions of single words are unlikely to be shared across the relevant audiences and communities, and even within a single audience segment the term may be ambiguous or used in varying ways.

Thus, for lack of ambiguity to be achieved, terms need to be adequately defined and authors, reviewers, and users of the assurance cases, integrity levels, and defining processes need to have a shared understanding of the underlying concepts and context.

7 Assurance Case

7.1 Introduction

An assurance case is a means to provide grounds for confidence and to aid decision making. The assurance case has one or more top-level claims in which confidence is needed and has supporting arguments connecting the top-level claims with multiple levels of sub-claims. The sub-claims are in turn supported by evidence and, where appropriate, assumptions.

NOTE This part of ISO/IEC 15026 often refers to a single top-level claim. However, this is not a comprehensive prescription; an assurance case may have multiple top-level claims.

The most common purpose 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. More broadly stated the purpose of an assurance case is to inform stakeholders’ decision-making and to supply grounds for needed stakeholder confidence. Different stakeholders seek to achieve their own goals such as developing a suitably trustworthy system or deciding whether or not to use a system in light of the risks.

NOTE In addition, an assurance case can be created simply to ascertain reality or even what claim is (or possibly was or will be) true.

Usually, an assurance case addresses the reasons to expect and confirm successful production of the system, including concern for the possibilities and risks identified as difficulties or obstacles to developing and sustaining that system. Assurance cases include claims about a system, normally that it satisfies relevant requirements, and supporting arguments for these claims that are in turn supported by evidence or assumptions including their relationships. To convince stakeholders successfully, the possibilities and risks they perceive and their doubts should be addressed whether the developers believe these perceptions to be merited or not.

The assurance case provides a multi-level structure of claims, sub-claims and connecting arguments that are ultimately based on evidence and assumptions that provide a reasoned, auditable argument supporting a claim. Together they show the truth or achievement of the top-level claim(s) or their falsehood or non-achievement. Figure 1 shows the major kinds of components of an assurance case.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

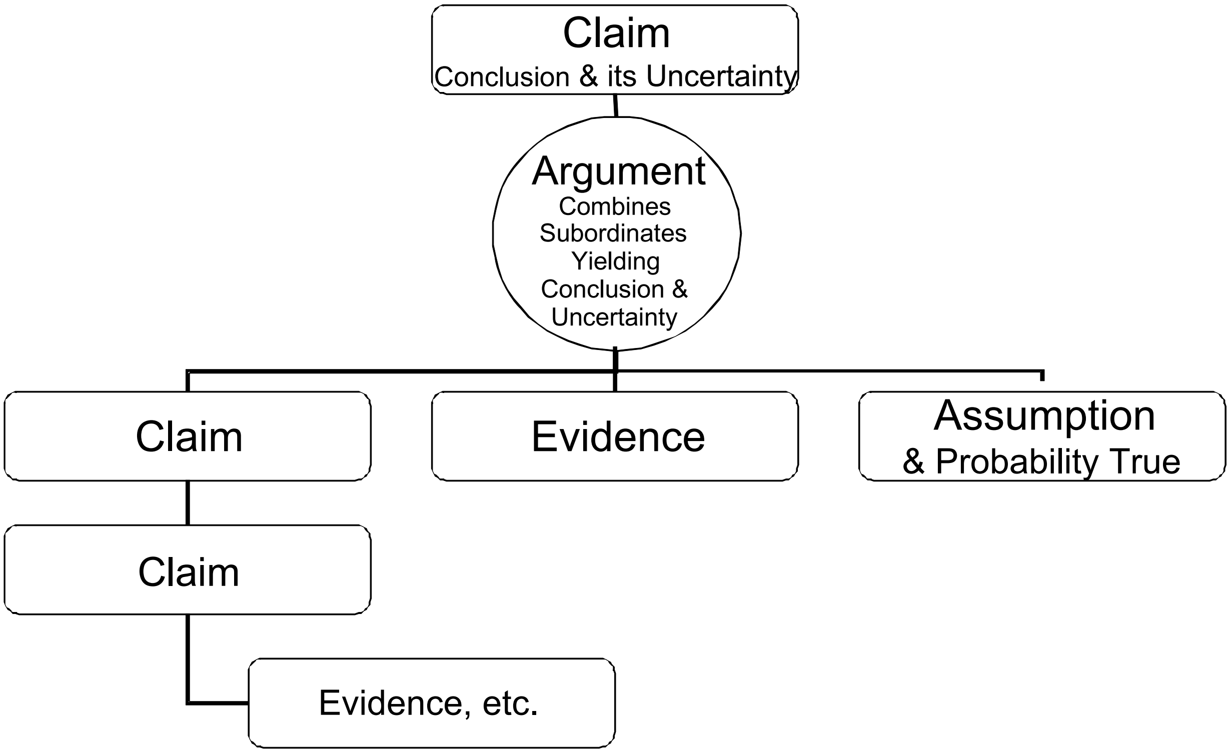


Figure 1 — Fragment of Structure

Best first considered with the initial system concept and requirements, the assurance case subsequently reflects experienced or postulated possibilities and risks; avoidance and mitigation strategies related to its claims; and an assurance argument referring to associated and supporting evidence. This evidence can come from design and construction activities, quality results from reviews, process fidelity records, standards conformance results, personnel qualification records, mathematical proof checkers, analyses, verification and validation activities, tests and trials, and eventually in-service and field data.

Any substantive modifications in the system or the assurance case's top-level claims will necessitate recorded changes to the assurance case. Such changes can also be generated by changes in the environment. 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 [[147], p. 5].

The users of ISO/IEC 15026-2 select the assurance case's purpose and the system and its required properties to be covered by the claim. The assurance case's argument should be supported by evidence and, where appropriate, assumptions.

A combined assurance case for multiple properties may be produced. Thus, the claim's property is possibly composed from multiple properties, and these possibly include consequences.

NOTE 1 An assurance case's claim(s) properties required could perhaps include the system's entire set of requirements for a property of interest. One example might have a top-level claim composed of (1) required limitations on consequences (2) functionality and properties of the system itself (e.g. that this functionality cannot be bypassed). The qualities defined in the ISO/IEC 25000-series include qualities related to functionality and constraints. The Common Criteria v. 3.1 Revision 2 [30] is also interested in both.

NOTE 2 Industry and agency standards and guides that are explicitly about assurance cases are included in the Bibliography. Standards or standards-related entries include [147], [150], [151], [155], [156], [157], [165], [166], [181], [182], [183], [197], [198], and [199].

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

NOTE 3 A safety case is an assurance case that is targeted at establishing safety claims. Another kind of assurance case used regularly is the Reliability, Availability and Maintainability or RAM Case. RAM assurance cases are the topic of [147]. Another approach to RAM cases is documented in the SAE JA1000 Reliability Program Standard [181].

Assurance case content includes relationships, specifications, definitions, justifications, real-world consequences, conditionalities, and uncertainties. Contents may include background information and links providing traceability. Contents need to be sufficient for all relevant stakeholders to be able to comprehend and evaluate the argument or case presented. Depending on the stakeholder and the anticipated evaluation context, contents may need to be scaled up or down accordingly, always conforming to ISO/IEC 15026.

To users of ISO/IEC 15026, the practicality of evaluation is a central concern of assurance cases. Evaluation addresses the issues of establishing estimates of a property's values and their uncertainty by predicting, measuring including testing, and analysing them. Assurance cases address the question of the uncertainty a property does or does not (or did or did not, or will or will not) meet its related claim (or claims).

The assurance case is central to the rational use of systems. Where uncertainty and consequence are serious concerns, an assurance case needs to exist whether written or not—otherwise an essential element needed to provide grounds for confidence (assurance) is missing. These consequences could be either positive or negative (e.g. risk).

Considered as an artefact, an assurance case has quality issues that concern the overall case, claims, arguments, evidence, and assumptions. Among these quality-related aspects are 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 having clearly stated conclusions with explicit degrees of uncertainty. One article [176] covers a substantial list of quality-related characteristics for assurance cases.

The success of an assurance case depends not only on its characteristics as a standalone artefact, but also on the project processes and the more particular assurance case methods, practices, techniques, and tools. Important practices include the assurance case being considered from the earliest stage in an effort; being planned, designed, developed, and maintained concurrently with the system; and being used to influence all activities and systems [148] and Appendix B in [197].

The approach to assurance or the assurance strategy should appear in any feasibility study and be further elaborated to accompany any operational concept document, and a description of the proposed assurance case would normally appear in a proposal document during acquisition.

The assurance case provides an audit trail of the relevant engineering concerns. It provides a justification for why certain activities have been undertaken and how they were judged successful. As a living, top-level control document, its status is continually tracked and typically summarized in Assurance Case Reports at predefined intervals or milestones. The assurance case usually remains with the system throughout its life cycle through disposal.

While certification and regulatory authorities do not always consider everything relevant, 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. It should not only give coherent confidence to developers, sustainers, and acquirers, but also be directly usable by certifiers and accreditors.

Activities called "assurance activities" overlap with other project activities including those directed towards evaluations of both the system and the processes used to develop and sustain it. Activities directly creating, maintaining, and evaluating the assurance case need to be planned and performed and include:

- Create top-level assurance claim from requirements.
- Establish degree of uncertainty needed for information to be used in decision making.
- Establish structure of the argument with sub-claims, including their relationships.
- Create portions of the assurance argument tailored for the desired limitation on uncertainty.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

- Compile portions of the argument and supporting evidence.
- Verify.
- Validate.
- Use as input to certification.

This clause's major subclauses cover Claims, Arguments, Evidence including assumptions, Management and life cycle of assurance case, and Decision making using the assurance case.

7.2 Claims

7.2.1 Introduction

Selecting the top-level claim and the properties it involves are not restricted by ISO/IEC 15026, although their statement is. Top-level claims are often a portion of the total requirements and specification but may be something internal to the system, related to something the system depends upon or not directly related to the primary system of interest. This subclause includes coverage of motivations for claims, their form and scope, and example properties they might involve.

7.2.2 Motivations for a claim

7.2.2.1 Kinds of questions to answer

While the nominal question that the assurance case answers is: (Was, Is, or Will) the claim shown (be) within the required uncertainty limitations? This question might be stated as: Will it be good enough for what we require and are we sure enough about that answer? However, several other kinds of questions might readily be asked:

a) What is the chance that the claim's property will meet its limitations?

b) How lenient would limitations on the property's value need to be to allow us to be sure enough that its values will fall within limitations?

c) What can be shown about the claim's property from the evidence?

In addition, several other more open kinds of questions might be asked:

d) What, if anything, can we be sure (enough) about regarding this situation or system?

e) What if anything can be ascertained about this situation or system?

f) Finally, for each of a) through e), one can also ask, for how long and under what conditions?

These questions represent different motivations or starting places for using an assurance case.

Assurance cases can be used to address verification and validation concerns by answering the following questions:

a) If the assurance case’s top-level claim is met, will this result in meeting real-world intention(s), need(s), and expectation(s)?

b) Will or does the system as designed, implemented, transitioned, and operated meet the top-level claim?

These two questions need to be dealt with by any approach to the life cycle of a system where risk, consequences, or uncertainty are issues. Dealing with these questions aids not only gauging feasibility, suitability, and desirability of development, production, transition, and operations, but also corrective action,

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

learning, and improvement. These questions may be dealt with separately or together, and assurance cases used for one alone, both together in a single assurance case, or both separately as long as the ISO/IEC 15026-2 requirements for such a combination are met.

Risk management combines degrees of uncertainties regarding achievement of questions a) and b) to establish the comprehensive, net or residual potential consequences or risk.

7.2.2.2 Categories of requirements

ISO/IEC 25030 provides a categorization for requirements that, while only for software, has relevance to other aspects of systems as well as to the system as a whole. These categories include functionality, quality (both internal and external), development requirements, and quality in use.

Examples of measures of internal quality are given in ISO/IEC 9126-3 (to be replaced by ISO/IEC 25022). Examples of measures of external quality are given in ISO/IEC 9126-2 (to be replaced by ISO/IEC 25023). Examples of quality in use measures are given in ISO/IEC 25010. Many come from life cycle processes that usually follow development, such as manufacturability, marketability, training, maintainability and test equipment, usability, interoperability, use in extreme environments, legal compliance, cost of operation, and mission accomplishment (not all from ISO/IEC 25010).

Producers and other stakeholders may prioritize properties such as efficiency and reliability and perform trade-off studies between them. Additionally, achieving a quality such as safety might affect speed or other characteristics making them less desirable. Specifying the system's external behaviour is a system design activity and, as such, can be fraught with tradeoffs, including ones among properties or qualities. A number of techniques have been created for addressing these trades, such as those in [25], [70], [131], [169], and [42]. The specifying of a top-level claim is sometimes the result of analyses including trade-off studies.

The need fulfilled by a top-level claim and its assurance case can be small compared to the concerns related to the total system. Will the ground at the site support the planned structure? Will a sub-function on a new airplane be at least as safe as the same sub-function on similar prior airplanes? Is a tool adequately trustworthy? Assurance cases may be used for either large or small requirements or needs.

7.2.3 Form of a claim

A claim takes the form of a true-false statement concerning a property that may be a combination of other properties. The term "property" is used quite generally – a property is a descriptive aspect that may have a claim concerning it, at least in principle, evaluated as true or false. Subclause 7.2.7 explains properties in detail.

This usage of the term "property" derives from, is consistent with, and subsumes the broad use of the term "property" in ISO/IEC CD 25010 where it is used spanning properties including properties that are inherent or not, internal, external, and in use or context. In principle, one could apply ISO/IEC 15026 to claims regarding any property of any importance.

A claim is a true-false statement that concerns 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. Uncertainties also may be associated with the duration of applicability and the stated conditions. Thus, a claim potentially contains the following components:

- Claim's property.
- 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.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

- Limitations on conditions associated with the claim.
- Condition-related uncertainty.

NOTE 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 but rather involve probability distributions, are incremental or have other fuzzy aspects.

Each of these components may have details within it. In particular, the property might include consequences or its worth – how valuable or costly it is or would be. Limitations on a property's values may be fixed (held constant). This might be done for the sake of exploring or analysing what values of other components of the claim (and possibly other properties) would be consistent with these fixed limitations – for example, the property of possessing a certain strength for what duration.

Each of the following can be stated about each of these components within the claim:

- What is required.
- What is planned to be established.
- What is actually shown or established.
- What is possibly or actually contradicted.

NOTE Some subset of them is established within the context of interest at a given moment – e.g., what is required has been specified, what is planned has been determined, or what has actually been shown is known.

The quality of a claim depends on it being fully specified, so the true-false statement may need to be further defined by references to either internal or external material such as definitions of terms or descriptions of context. Aspects needing supplementary definitions may include limitations on the range of property values (e.g., required degree of achievement or tolerances), its duration of applicability, conditions it requires or on which it depends, properties being combined and relations or measures used in the combination, and limitations on uncertainty. While many terms have well-known meanings and even abbreviations, e.g., kilometres per hour, that do not need explicit definition, many do, and all terms need to be explicitly stated, such as units of measure,

Often the conditions under which the claim is said to be true and its duration of applicability are treated as unvarying constants. However, they could be treated as variables and as having uncertainties. This is particularly true when the purpose of the assurance case is to simply establish what is true. For example, uncertainty may exist about the durability of the system or how long it will continue to possess a quality. Figure 2 gives a simplified view of the claim's components and their relationships.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

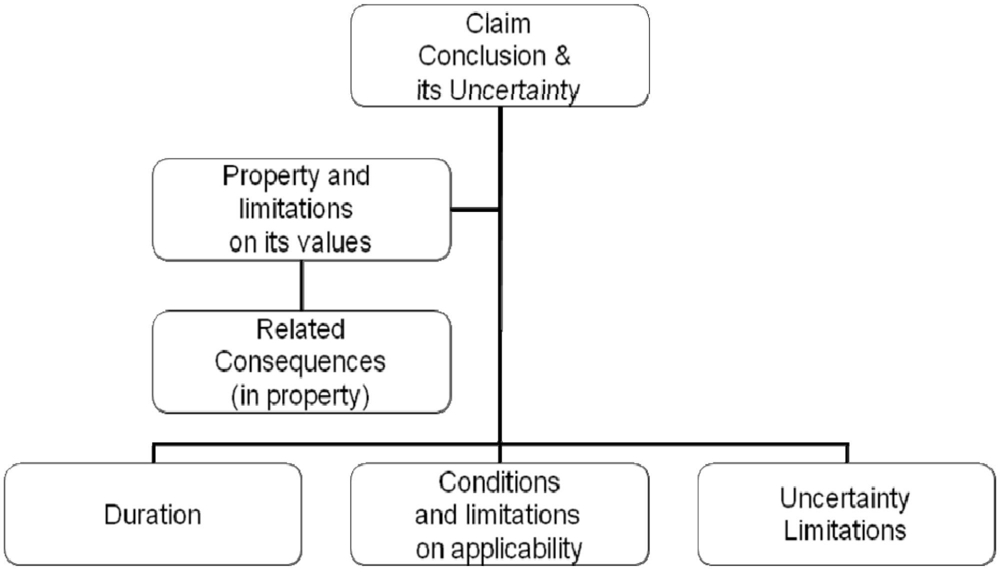


Figure 2 — Claim

In addition to the required limitations on uncertainty, a claim can have several categories of uncertainty associated with it: the uncertainty that is required to be achieved, the uncertainty that is planned to be achieved, and the uncertainty that has already been achieved. The uncertainty about the claim achieved by its supporting argument (and sub-claims, evidence, and assumptions) needs to meet any required limitations on its uncertainty. Finally, uncertainties can exist about uncertainties.

Among the forms in which claims may be stated is in terms of placing limitations on events or the establishment and preservation of conditions. For example, claims might take the forms of:

- For events:
  - Having desired behaviours and events.
  - Limitations on undesirable behaviours and events.
- For conditions:
  - Establishing (and possibly preserving or re-establishing) the preconditions for desired events.
  - Establishing and preserving the conditions that preclude (or limit) undesirable events.

To preclude an event, the relevant condition should imply the negation of the precondition for such an event. Similarly, one speaks of states and state transitions. A state of a system, a possibly relevant condition, can involve many aspects as [[14], p. 13] states, “The total state of a given system is the set of the following states: computation, communication, stored information, interconnection, and physical condition.” If a system contains humans, then the relevant portions of their states are also part of the system state.

7.2.4 Scope of concern

In different situations or activities, concerns for properties can vary in extent and in time. In extent or nature, concerns vary across several echelons of lessening scale from the real world to the individual system element or service, behaviour, or property as well as across their relationships, makeup, contents, and governance. In time, concerns can exist in a context before the conception of a particular system and beyond a system’s life cycle to ultimate consequences and residual obligations. A system-related claim can have a scope of interest that is on one side of or extends across the system environment.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

7.2.4.1 Extent

Recognizing these varying scopes of concern, ISO/IEC 25010 defines three different kinds of quality for a software product:

- Quality in use: the extent to which a software product used by specific users meets their needs to achieve specific goals with effectiveness, productivity, safety and satisfaction in specific contexts of use.
- External software quality: capability of a software product to enable the behaviour of a system to satisfy stated and implied needs when the system is used under specified conditions.
- Internal software quality: Capability of a set of static attributes of a software product to satisfy stated and implied needs when the software product is used under specified conditions.

A somewhat similar sense of expansiveness exists when conceptualizing where the behaviour, events, or conditions of interest occur. These might relate to different scopes of concern, including:

- Real world – concerns regarding funds, lives, real property, natural environment, and other interests of stakeholders, including allies, adversaries, and neutrals or bystanders.
- System-environment interface – user interface, interface with sensor or effector, service offered by the system, service depended upon by the system, intake of consumables, output of system (including by-products), physical support, interaction with test environment/equipment.
- Internal to system:
  - System elements, resources, or assets.
  - Non-computing elements, resources, or assets, possibly containing computing or information sub-elements, resources, or assets.
  - Computing or information elements, resources, or assets – database, stored software, computing hardware.
  - System behaviour – internal operations or operations viewed from an internal perspective.
  - Software behaviour.
  - Enabling functionality – logging, automatic recovery.

Generally, from outside-in (top to bottom of list), the scopes form an ordered layering of extent of concern. These roughly correspond to layers in an assurance case argument that include consideration of consequences and base their argument on behaviour, events, and conditions in the environment as affected by behaviour at the system-environment boundary, which is in turn the result of behaviour internal to the system. Ultimately, concern is usually driven by real-world effects: benefits, costs, and consequences. However, everything down to the internal details of the system may be relevant in building an argument.

7.2.4.2 Duration of applicability

The duration of applicability of a claim might be stated in calendar time, as a time interval or intervals, as during a life cycle process or processes, as during certain activities, as under certain conditions, or in some combinations of ways. The duration of applicability might result from what is required or what is achievable. While the duration of applicability is usually a constant, it may have required, planned, and supported values and should at least be tacitly consistent with the other corresponding values. The duration of applicability also may have associated uncertainty, but this is not a requirement.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

7.2.5 Consequence

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:

- **Consequence:** Any effect (change or non-change), usually associated with an event or condition or with the system and usually allowed, facilitated, caused, prevented, changed, or contributed to by the event, condition, or system. It could yield a benefit, a loss, or neither.
- **Adverse consequence:** Consequence associated with a loss.
- **Desirable (or positive) consequence:** Associated with a gain or avoiding an adverse consequence.

A consequence has value or is desirable or undesirable from a stakeholder's perspective, viewpoint or interests. A consequence may occur anywhere in the system's life cycle or beyond. Certain effects within a system may be treated as consequences such as wear from use and damage from mishaps. For example, "The publishing of the concept for the system induced enquires for both investment and purchase," or, "The system was retired and disposed of long ago, but liability remains and new liability claims continue to occur."

7.2.6 Claim violation

7.2.6.1 Violation-related terms

The following three terms in ISO/IEC 15026 are widely used:

- Fault: A defect in a representation of a system or a system that if followed and/or executed/activated could potentially result in an error. It is incorrect and usually thought of in terms of a static representation or a static instance of the system. Faults can occur in specifications when they are not correct. (See 7.2.7.3.1.)
- Error: An erroneous state of the system.
- Failure: An externally visible deviation from the system's specification.

Under the same conditions, exercising a fault might or might not result in an error. Likewise, an error might or might not result in a failure. At a certain time, a fault, error, or failure can be known or unknown.

Usage is less uniform for the following four terms also used in ISO/IEC 15026:

- External mistake (e.g. human error): External entity's or entities' non-malicious action or inaction, or non-malicious input to or interaction with the system that has the potential to result in a fault or/and error (and thereby possibly in failure) or an adverse consequence either not intended or not intended to be adverse.
- Attack: A malicious action or interaction with the system or its environment that has the potential to result in a fault or an error (and thereby possibly in a failure).
- Adverse consequence, as defined above.
- Violation: A behaviour, act, or event deviating from a system's desired property or claim of interest. Examples might include violation of a performance standard, a speed limit, limitations on tolerances, confidentiality, laws, or a claim of suitability.

Human errors (including organizational ones) can be intended or unintended, planned or unplanned, and in agreement or disagreement with a plan, reflecting a mistaken plan, a cognitive lapse in enacting a plan, or a non-cognitive slip.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

7.2.6.2 Violation-related concepts

NOTE In the area of safety the term “violation” is used to refer to a deliberate human contravention of a procedure or rule.

Threatening entities – also referred to as sources of danger, threat agents, and attackers – can possess capabilities, resources, motivations, and intentions that enable them to initiate and carry out non-malicious (e.g. mistaken) or malicious efforts to violate a claim. Violators use their capabilities to take advantage of system- and/or environment-provided opportunities called vulnerabilities, i.e., “weaknesses…that could be exploited or triggered by a threat source” [161].[2] Non-malicious and malicious entities use specific methods (e.g., agents and kinds of attacks) that often fall within recognizable patterns referred to as patterns of abuse, failure patterns, accident patterns, and attack patterns.

Systems or their environment often employ countermeasures to limit or reduce the opportunities for and ease of violations and limit or reduce adverse consequences such as the extent and intensity of damage that would result from a violation. Generally, attackers make gains only after further effort while system stakeholders make efforts to limit losses. Their respective gains and losses often differ

7.2.7 Properties

7.2.7.1 Introduction

Properties are a means of description, including specification or definition. A property might include a condition, a characteristic, an attribute, a quality, a trait, a measurement, and a consequence. A property might be invariant, or dependent on time, situation, or history.

NOTE In the use of ISO/IEC 15026, a property is usually expected to be relevant directly or indirectly to a system or systems.

Properties may be of interest 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 15026. As this knowledge involves predicting the future, it is often the most difficult and uncertain to attain. Therefore unsurprisingly, a system’s future behaviour and consequences often become principal issues in its assurance.

Many of the properties of interest are qualities of the system. Several standards and reports provide lists and definitions of qualities including ISO/IEC 9126, ISO/IEC 25010 and the related series, ISO/IEC 2382-14, ISO 9241, ISO/TR 18529, and ISO/TS 25238. Several standards or reports mention consequences associated with systems within a specific domain. Examples include ISO 14620 [90], ISO 19706 [104], and ISO/TS 25238 [116]. Risk management standards also address consequences, for example ISO/IEC 16085 [97].

Several general properties have been mentioned and more are listed below. However, a specific system requires specified properties within these general properties. Examples of concern for properties include the integrity of a barrier, the maintainability of a piece of equipment, the availability of a less than three minute response by the fire department, and the early confidentiality of new weapons (e.g. the US F-117 stealth fighter). For information or data, confidentiality may only be relevant to a portion of the system’s data and integrity concerns only relevant to certain operations involving certain data. The limitations on uncertainty for each engineering application of a property may vary with the degree of seriousness of the property and that in turn usually reflects the possible consequences in the real world.

Properties may include (but are not limited to) dependability-related qualities such as reliability, availability, integrity, maintainability, correctness, accuracy, safety, confidentiality, accountability, or potential for human error; time- and resource-related ones such as processing speed, schedulability, throughput, and storage capacity; and human and organizational ones such as those related to human factors, as well as more global ones such as profit or mission achievement.

[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”.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

7.2.7.2 Specifying properties as behaviours

Often the property is specified as a behaviour. For example, a property could be that a certain erroneous state cannot be reached or that a certain sequence of transitions should (or cannot) occur. 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.

These kinds of properties can be stated as conditions or constraints that must be true of the system.[3] 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.

If the system states are adequately known or modelled, this approach can also be taken at the system-environment interface (or software and its environment). One may also wish to model the environment if affects on and within it or changes in its state are important to overall consequences. This is one way requirements related to the environment’s condition (e.g., a certain condition in the environment would be catastrophic) and combined system-environment behaviour can be addressed. This is not an unusual situation as the situation of interest is often larger than the system.

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 [144] including access control mechanisms and policies, and restrictions on information overtly or covertly communicated,

7.2.7.3 Other types of properties

Properties are everywhere. They include anything objectively measurable and many things that are not. Examples include shape, colour, attractiveness, available opportunities, reusability, buoyancy, hardness, and mechanical strength. ISO/IEC 9126-1, ISO/IEC 26702, and ISO/IEC 25030 list many other product qualities that could be the subject of assurance. Other examples are in the following subclauses.

7.2.7.3.1 Correctness

Two major kinds of correctness are relevant:

- Correctness of the specification (or portion thereof) in terms of meeting needs and expectations of stakeholders and for practical purposes such as being feasible.
- Correctness of artefacts and the system in terms of agreement with the specifications – as well as, by extension, agreement during transition, operation, and the rest of the system’s life cycle.

For the latter point, the system might be considered as having two variants of correctness:

- A system is correct at its external boundary if it would always meet its external behaviour specification under the required conditions. That is, it has no failures.
- A system is correct throughout if it contains no faults and therefore is never, under the required conditions, in an error state. In ISO/IEC 25010 terms, this is an internal quality.

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

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Being internally correct (correct throughout) implies being correct at the external boundary, an external quality in ISO/IEC 25010 terms. A system can be externally correct, however, without being internally correct if it can tolerate or recover from internal error states and never display incorrect externally visible behaviour.

7.2.7.3.2 Dependability

Dependability is a qualitative “umbrella” term [[14], p. 13]. ISO/IEC 25010 notes that "dependability characteristics include availability and its inherent or external influencing factors, such as: reliability, fault tolerance, recoverability, integrity, security, maintainability, durability, and maintenance support." Several standards address dependability (e.g. [70], [71], and [75]), and many more address the qualities within it. IEC 50 (191) offers related definitions [69].

Thus, dependability includes reliability, safety, maintainability, integrity, availability, plus related survivability; and when addressing security includes confidentiality, accountability (knowing who or what did something), non-repudiation (their not being able to deny it), authenticity, security compliance, and immunity (the degree to which the product is resistant to attack), In addition, interfacing with humans or usability, particularly error-prone and inconvenient interfaces, can also have a significant effect on dependability.

Assets may be categorized by attributes related to the dependability property of interest. Examples include confidentiality (e.g., Top Secret, Secret, Confidential, or Unclassified); degree of integrity (e.g., accurate and up-to-date versus old and with unknown accuracy); or criticality of availability or acceptable degree of unavailability (e.g. outage length 0-1 minute, 1-5 minute, 5-10 minutes, 10-30 minutes, 30 minutes-2 hours, greater than two hours). Ultimately, such categorizations derive from and are surrogates for the values of the stakeholders’ real-world benefits and losses (and sometimes uncertainties) potentially associated with the property’s preservation and violation.

7.2.7.3.3 Time- and resource-related

Time- and resource-related properties include meeting deadlines, efficiency, and storage capacity. These are not only important alone but also important in combination with dependability-related and other properties.

Many relationships and potential tradeoffs can exist among dependability properties and speed, efficiency, or other time- or resource-related properties. An example of a property that is relevant to both types is computational difficulty. Computational difficulty is an issue when one tries to compute something and when one wants to prevent someone else from computing something. The first is of interest in achieving real time performance and the latter in decryption.

Another example spanning the two types is availability and timing-out on whatever the deadline is in the particular measurement of availability. In the end, the issue in availability is not whether the system will eventually respond, but will it respond within a specified (e.g., useful or acceptable) time period.

Table 2 lists several properties related to time and resources. These include rates such as throughput or processor clocking, size, and economic ones. An entry such as “storage capacity” could relate to anything stored from fuel to binary bits. These types of properties are often measures of some aspect of performance and partial measures of merit for a system.

Table 2 — Some time- and resource-related properties

Timing	Throughput
• On time	• Bandwidth
• Meeting Deadline	Speed
• Schedulability	• Rates of Learning and Use
• Delay or Latency	Size
• Response Time	• Storage Capacity
• Sequence	Productivity
• Serializability	Efficiency
• Order Independent	Cost

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

7.2.7.3.4 Human- and organization-related

- An almost unlimited number of properties can be associated with humans and organizations. Among common ones relevant to systems are usability, occupational health and safety, size and reach, physical strength, mission accomplishment, and benefit and loss.
- Normally, existence of a human interface requires concern for human factors within an assurance case because almost every property is affected by it particularly any external quality or a quality in use. Human factors are addressed in several standards, for example [84], [99], and [107].

7.2.7.3.5 Tolerance- and resilience-related

Two common objectives for tolerance and resiliency are:

- Be resilient in response to events or conditions.
- Limit damage or decreases in benefits.

Among the principles sometimes mentioned in connection with resilience are redundancy, diversity, separation, generality, flexibility/adaptability, and restricting dependence. However, a system can have several objectives and activities related to resilience and limiting damage including:

- Forecast events and conditions.
- Maintain readiness.
- Detect events and conditions (desirable and undesirable particularly the latter including precursors, warnings, near misses, and suspicious events).
- Notify and warn.
- Record (e.g. via logs).
- Separation (e.g. by distance, time, barriers, or flow control).
- Continue service, although possibly degraded.
- Damage confinement (including by isolation and risk sharing).
- Diagnosis.
- Repair.
- Put system in a proper state:
- When current state is detected or inferred to be illegitimate (recovery).
- Preventively (e.g. regardless of knowing if needed or if indications exist that might otherwise later experience problem).
- Flexibility and the capability and tactics to successfully adapt and deal with events and conditions.
- Reserves and reserve capacity.
- Characterization, analysis, investigation of root cause or causer.
- Operational "safety" margins.
- Learn and improve.
- Arrangements or agreements with entities in the environment to provide aid (possibly including alliances).

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

The issues also exist concerning the readiness for, response time, speed, capacity, efficiency/cost, and efficacy of these list entries along with doing them when not needed and not doing them when needed. Flexibility and adaptability are often provided, at least in part, by humans.

7.3 Arguments

7.3.1 Introduction

Arguments are the glue that holds the assurance case together by relating its immediate underlying support – sub-claims, evidence, or assumptions – to the claim (or claims) it supports. It yields the combined effect of its evidence, sub-claims, and assumptions into a conclusion.

The term "conclusion" describes the actually shown or established conclusion regarding whether actual property values (did, do, or will) make the true-false statement true or false. The conclusion has its associated uncertainty - that is the uncertainty associated with the claim's property's actual value(s) - meeting it limitations.

The uncertainty regarding the conclusion derives from the uncertainties in the argument's immediate underlying support plus the strength or rigor of the argument and its own effect (plus or minus) on uncertainty. For example, several pieces of evidence that individually would leave much uncertainty about the claim might be combined by an argument into support yielding a claim with low uncertainty.

The resulting uncertainty needs to be within the limitations for uncertainty that were allocated or budgeted to the claim. This required limitation on uncertainty (e.g. it could be in terms of limitation on risk) derives from the uncertainty limitations of claims yet higher in the overall argument structure and ultimately from the limitations associated with the top-level claim. Limitations on the claim in question may be affected by “local” consequences associated with a sub-claim in the assurance case. This might be the situation if the claim relates to a system element whose misbehaviour could have separable consequences of its own.

Figure 3 provides the context for an argument showing how its super-ordinate claim (and sometime claims) is implied by the sub-claims, evidence, and assumptions that lie immediately below it and provide its support.

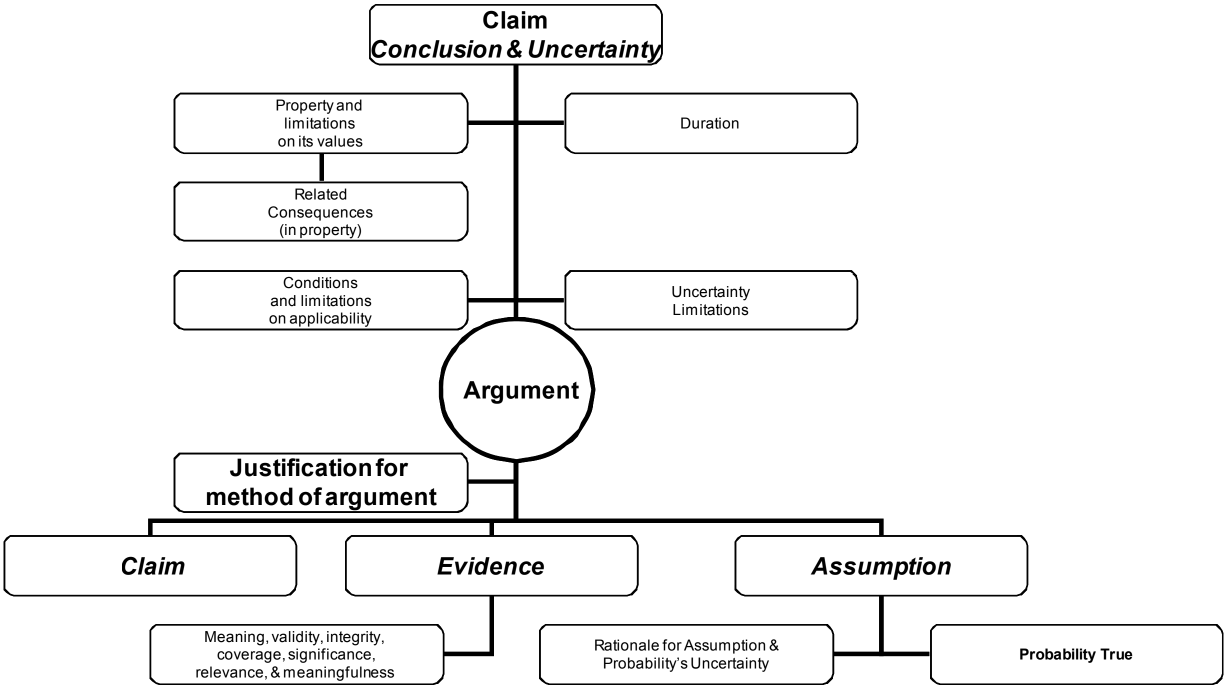

Claim
Conclusion & Uncertainty
Property and limitations on its values
Duration
Related Consequences (in property)
Conditions and limitations on applicability
Uncertainty Limitations
Argument
Justification for method of argument
Claim
Evidence
Assumption
Meaning, validity, integrity, coverage, significance, relevance, & meaningfulness
Rationale for Assumption & Probability's Uncertainty
Probability True

Figure 3 — Argument Context

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

This subclause briefly discusses the variety of reasoning methods that one might see or consider. This is followed by covering the roles arguments might play in an assurance case, and several issues concerning the bases for and the structures of arguments.

7.3.2 Reasoning Methods

7.3.2.1 Introduction

Many methods of reasoning exist: some are more rigorous than others and may produce stronger results, some are quantitative and some not, and different ones may be appropriate for different situations. Human judgement can frequently play a role. For example, using probability related to occurrence of a natural event contrasts with the need for concern for the possibility of an intelligent, malicious action. Regardless of the method used to reason and decide, it can be difficult to gain adequate information, understand, and structure or model the situation in a way that results in very low uncertainty.

Subjects of and approaches to reasoning differ among communities having differing motivations, mindsets, and often multiple methods of reasoning. Methods of reasoning include:

- Quantitative:
  - Deterministic (e.g., formal proofs).
  - Non-deterministic formal systems for reasoning:
    - Probabilistic.
    - Game theoretic (e.g., minimax).
    - 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).

Some examples of methods of reasoning and their ways of showing something is true (possibly with some uncertainty) are listed in Table 3 — Example ways of showing something is true.

Table 3 — Example ways of showing something is true

Logic
Reduction (e.g. infer by laws of logic, definition, substitution, simplification)
Generalize to collection from arbitrary particular member
Contraposition
Contradiction
Induction
Show existence true by an example
(Show false by counterexample)
By Cases
Probability
Inference (e.g. probability theory)
Induction (e.g. statistics, observations, experiments, tests)
Models
Simulation
Analysis
Agreement among
Multiple methods or instances of showing
Suitable humans

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Complex products and situations – and any involving humans – may be beyond the current state of the art to “quantitatively” create precise and accurate predictions. In addition, supplementing quantitative techniques with expert review and judgement is widely used and generally accepted as being a wise necessity.

The legal profession has developed methods that reflect what one might think of as a long history of experiment. [46] Making decisions through conflicts may not be the best way to gain creditable evidence and a good understanding, but it does automatically supply doubt and question the behaviour and possibly the methods being used. This method can lead to categorical thinking, inflexibility ignoring the middle or compromise position, a strong desire for certainty rather than explicit recognition of uncertainty, and living and dealing with the consequences. However, this method has a more moderate parallel in the technique of multiple working hypotheses.

On the other hand, scientific reasoning appears willing to keep the question open and the decision unmade until enough evidence and understanding are achieved. Scientific reasoning tends to use induction, deduction, analogy, experiment, theory formulation, causal reasoning, and problem solving techniques. [44] Generally, theories that are at least theoretically refutable and make differing predictions testable are preferred over competing theories. Lately, experimenting and investigating with “computational” models (e.g., simulations) has gained prominence and a role alongside theory and real-world experimentation and data collection.

Engineering shares with the legal profession the need to make immediate decisions and with the sciences the desire for making decisions on a firm basis. Regarding venturing into uncertain areas and risks, science uses the concept of “informed consent” of stakeholders, and engineering codes of conduct address the problem of working within one’s abilities.

Table 4 — Communities with different viewpoints and approaches to reasoning lists some of the communities and activities having their own – although sometimes overlapping – mindsets and approaches.

Table 4 — Communities with different viewpoints and approaches to reasoning

Mathematical	Safety	Research
Security	Engineering	Correctness
Project Management	Counterintelligence	Risk Management
Crime	Financial	Regulatory
Executive Management	Industrial competitiveness	Subversion
Political or social activism	Litigation/Liability	Espionage
Marketing	Buyer	Terror
User	Diplomacy	Revolution
Intelligence analysis	War fighting	Attacker
Natural disaster		

A variety of bases for argumentation and analysis in the assurance case might be used. Choosing the one (or few) to use can include several factors. While not all listed here, one should consider that some arguments can be more complex or difficult to perform than others. However, choosing a tool because it is easy to use or the engineers are most familiar with it will not always be the best choice. While engineering simplifications can be appropriate, ultimately the bases for arguments and the methods of reasoning need to yield results that adequately reflect and do not contradict reality.

7.3.2.2 Subjective Judgment

While sometimes necessary or advantageous, use of subjective judgement within the assurance case can lead to problems or additional uncertainties, so, generally, (just as with assumptions) the less critical the judgement is the better. Subjective judgements are used in the absence of affordable, suitable, more objective

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

methods and techniques or where needed to supplement or evaluate the results of such techniques. As with other forms of argument, subject judgements take the form of a claim and its support.

Generally, judgements are expected to be of higher quality if they reflect an agreement among multiple persons who cover the full scope of knowledge of the situation and the necessary relevant expertise and experience, and are consistent with known facts. Group decision making has its problems [32], but decisions made in isolation have their own risks. As with other forms of argument, their conclusion needs to be accompanied by an estimate of its uncertainty and be reviewed, recorded, and acceptable to approvers of the assurance case. Finally, if as the result of using human judgement or another form of reasoning a risky amount of uncertainty results, stakeholders relying on the assurance case may need to be warned.

7.3.2.3 Probability *versus* possibility

Historically, the assurance case has often been held together by values, uncertainties, and relationships dealt with using probability-based methods such as statistical confidence, decision theory and Bayesian networks.

The patterns of occurrences of “natural” events and common, non-malicious human behaviours are usually described probabilistically. The probability of a natural event contrasts with the concern for the possibilities open for intelligent, malicious actions whose probability is not determinable or not knowable. This is particularly a concern if the adversary deliberately violates any probability estimates one makes regarding its behaviour – for example to achieve surprise. This distinction is central to the difference in reasoning between safety and security.

Combining probabilities can lead to an expected result. However, combining instances of possibilities is difficult to do in a way that does not simply result in the worse (or best) possible case. Knowing the worst case can have limited usefulness to decision makers who must consider limited resources to overcome it when its occurrence might never or quite rarely happen.

7.3.3 Roles of arguments

Arguments derive roles from their place in the structure of the assurance case. They also may have roles in their use by stakeholders, such as for communication among and use by decision makers. Roles include yielding the combined effect of the evidence, sub-claims, and assumptions that they use, providing a second argument in support of claim, and replicating the same argument with different support, which is usually different evidence.

Issues arising from the assurance case arguments for the property of interest and different properties or qualities can highlight tradeoffs with other properties or functionality.

7.3.3.1 Combining supports for a claim

An argument needs to argue that what supports it is relevant to supporting the claim and that it meaningfully combines what supports it into support for the claim. The argument’s meaning and uncertainty should reflect those of what supports it and the nature of the argument.

When they are used to combine supporting evidence, sub-claims, and assumptions into support for a claim, different methods of reasoning vary in their applicability, power, resulting accuracy and uncertainty, and ease of use. Among rigorous methods, the use of probability-based methods to do this combining has the longest and in many ways the most successful history. As mentioned elsewhere, in some situations its applicability is difficult, questionable, or unsuitable (See 7.3.2.3).

The items supporting the argument have uncertainties associated with them and the argument can increase or reduce uncertainty. A method of argument can be an additional source of uncertainty.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

7.3.3.2 Using multiple arguments

Three scenarios might exist with multiple arguments all supporting the same claim:

- Different arguments with same support (e.g., supported by same or essentially overlapping sets of evidence).
- Same argument with different support (e.g., different evidence).
- Different arguments with different support (e.g., different evidence).

Replication of experiments has an important place in science. Replication by someone or an organization other than the original experimenter can be helpful in confirming a result. Replication can help establish that previous results were not dependent on some unreported aspect and that they were not the result of using an improper method lacking proper interpretation, fidelity, competence or skill, or care in enacting the method; making mistakes in recording, analysing, or communicating (including understanding) descriptions or results; encountering a statistical fluke; or maliciousness. Aspects may be unreported or inadequately or mistakenly reported because they are not noticed, recognized as potentially relevant, accurately observed or measured, accurately and completely recorded, or reported in disagreement with records or in an ambiguous or difficult to understand way.

The more independent and different the conductors of replications are from the original conductor (and each other), the less likely unrecognized or unreported aspects may exist that potentially could affect results, or their meaning or meaningfulness (significance).

Multiple arguments reaching similar conclusions using distinctly different methods or based on different conceptual bases generally add credence to the conclusion.

7.3.3.3 Modification of arguments

Modification of arguments can be needed because of changes or because of their weaknesses, such as unsatisfactory method, execution, conclusions or convincingness (e.g., excessive uncertainty). The structure of the assurance case's argument, particularly modularity and mapping to system design, can make it easier or more difficult to create, understand, and modify [134]. Automated tools to aid in recording, maintaining, and managing assurance cases can help.

7.3.4 Structure of arguments

Overall assurance cases often make arguments falling into one of two patterns (1) nothing significant went or is or will be too wrong and (2) everything necessary went or is or will be right or close enough (through the duration of applicability of the top-level claim) – contrast [182] and [183] with [55].

Both patterns have difficulties. The first requires identifying everything significant that might go or be wrong. This is usually called risk identification and analysis. The second pattern, to be practical, must either argue that only the aspects it covers are significant for the assurance case or true within the portions of the assurance case where the pattern is being used.

Each individual argument within the overall assurance case has the objective of showing its immediate super-ordinate claim or claims from its immediate subordinate supports. This can be achieved either directly from the evidence and assumptions or by breaking the claim into parts that are related to its immediate sub-claims.

In the latter case, generally all of the things needing to be shown to be true in a claim are carried down and allocated to one or more of its sub-claims, and the argument shows how the combination of these sub-claims leads to adequate support for the claim. Sub-claims are generally combined by using specified arrangement(s) often reflecting system structure. Specifying the arrangement(s) is usually necessary to effectively make the argument that combines the sub-claims into the claim. As a side effect, breaking down claims can lead to many of the requirements stated in one claim being repeated in its sub-claims, sometimes verbatim, sometimes slightly enhanced.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

In addition, assurance cases need sub-arguments covering the integrity of the assurance case (e.g., lack of tampering) and possibly the validity of the evidence.

To argue a claim directly, evidence or assumptions should be adequate and a number of things should be included in the argument. For example, SafSec divides the ways of arguing or showing into seven non-exclusive ways and calls these “frameworks.” Compliance with each framework means meeting certain standards, such as the organizational roles being defined, including standard ones. Thereby, SafSec provides one categorization of the rationales for use in direct arguments. For a single claim, many of these "frameworks" must usually be involved. [[183], pp 24-28] Evidence for a modest sub-claim may be as extensive. Subclause 7.4 covers evidence.

While possibly the bulk of significant systems and their environments are somewhat complex and difficult to model, it can be helpful to first consider the other extreme of the simple notional state machine in Figure 4. While clearly idealized, in a limited situation it might provide a basis for thinking about and structuring arguments before adding details.

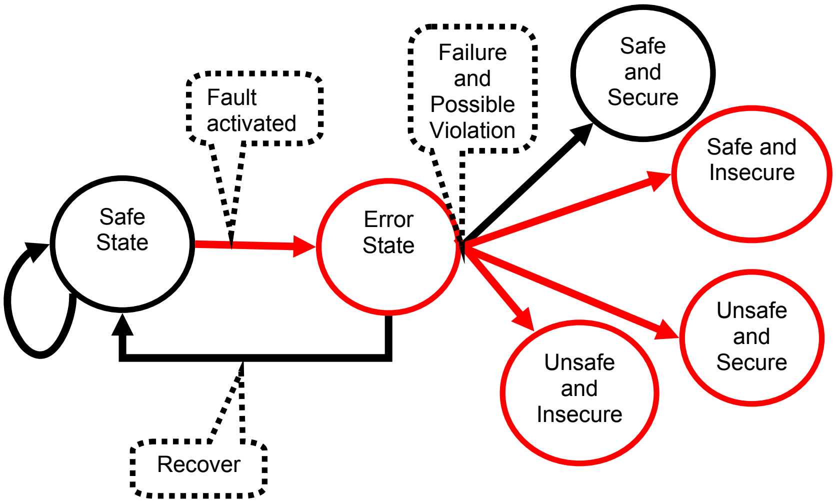
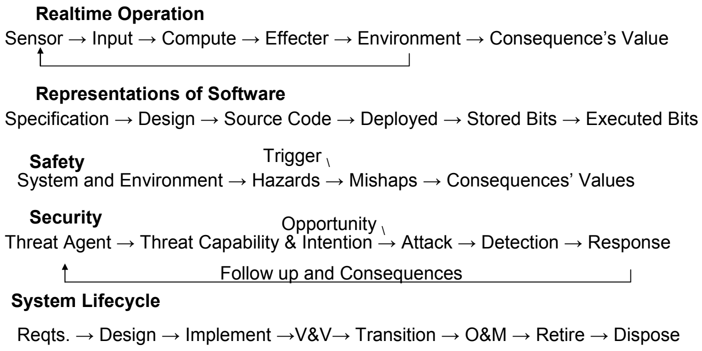


Figure 4— Simple State Model

7.3.4.1 Cause and Effect Relationships

To better understand the problem, consider reasoning using one of the widely used bases, cause-and–effect relationships. Some simplified “cause and effect” chains are shown in Figure 5. Physical processes and operator-system-interaction often use cause-and-effect models to underlie arguments. However, cause and effect is not as simple a concept as it might sound.



Realtime Operation
Sensor → Input → Compute → Effecter → Environment → Consequence’s Value

Representations of Software
Specification → Design → Source Code → Deployed → Stored Bits → Executed Bits

Safety Trigger \
System and Environment → Hazards → Mishaps → Consequences’ Values

Security Opportunity \
Threat Agent → Threat Capability & Intention → Attack → Detection → Response
Follow up and Consequences

System Lifecycle
Reqts. → Design → Implement →V&V→ Transition → O&M → Retire → Dispose

Figure 5 — Simplified "cause and effect" chains

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

While sometimes straightforward, many times cause and effect relationships are difficult to define, complex, subtle or indirect. Table 5 lists some of the ways in which relationships are categorized and characterized, and its length indicates that several subtleties can be involved in using them in arguments. Cause and effect arguments are quite important, but in practice often judgemental [173] and questionable [59] as well as problematic for how humans learn and perceive [23].

Cause-and effect relationships can relate many factors to an effect or a single cause to many effects and can chain together in complex relationships such as cyclic networks. For example, they can be highly sensitive or non-linear, multi-way, cyclic, dynamic, involve feedback, involve humans as well as physical phenomena, and reflect coincidental as well as systematic timing and linkages and may possibly be emergent.

Two link-related aspects of cause and effect that need to be considered are (1) common-cause failure where the common links are not well understood, (2) multiple coincidental events (not all of which are linked) that can together cause an undesirable consequence.

Example Consider an accident caused by three events: (a) the operator was late to work because of a weather-related traffic jam, (b) the river near the plant overflowed, (c) there was an unnoticed crack in the plant foundation. Events (a) and (b) have a common cause - a storm. Event (c) is independent.

The concept of “resonances” has been introduced as one approach to thinking about complex systems, lack of predictability, and self-caused events [59]. While one aspect of the approach fits well phenomena such as rogue ocean waves, for many situations it is an analogy. A related term is "normal accident." Nevertheless, whether complexity is desirable or not, relationships are often complex.

In part, the difficulties sometimes experienced with reasoning about cause-and-effect and some other common modelling techniques result from the system's behaviour and combined effects from interacting with its environment often resulting in emergent phenomena – in the case of both normal performance and many failures. This means the relationships between individual components or aspects and the overall result can be subtle and complex and, therefore unsurprisingly, what will happen is difficult to model or predict.

NOTE Describing the resulting situation in the context of the safety of modern complex systems one expert stated in a 2007 presentation, "Explanations of accidents cannot be limited to “component” failures and malfunctions — either alone or in combination. In complex socio-technical systems, accidents often arise from normal performance variability that interacts in unintended and unanticipated ways. The target for safety management should not be to reduce risks, but to increase the intrinsic ability on all levels of a system to adjust its functioning in the face of changes and disturbances (resilience)." [58]

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Table 5 — Relationship aspects that are possible bases for or relevant to arguments

1. Existent or non-existent	2. Extensive or limited
3. Dependent or independent	4. Strong or weak
5. Established or postulated	6. Sensitive or insensitive
7. Known or unknown	8. Acyclic or cyclic
9. Credible or not credible	10. Positive or negative feedback loop
11. Plausible	12. Stable or unstable
13. Deniable (plausibly or convincingly)	14. Intentional or unintentional
15. Reputable or non-reputable (non-repudiation)	16. Purposeful
17. Long-time or new	18. Non-malicious or malicious
19. Permanent or temporary	20. Trustworthy or untrustworthy
21. Well-established or tentative	22. Trusting or untrusting
23. Common or uncommon	24. Private or confidential, or public or exposed
25. Frequently occurring or unique	26. Goal-directed
27. Ubiquitous or local	28. Multiple-objectives
29. Invariance	30. Cooperative or uncooperative
31. Correlation	32. Competitive or non-competitive
33. Cause and effect	34. Giving, taking, or sharing
35. One-to-one, one-to-many, many-to-many	36. Supplier-consumer
37. One-way or two-way	38. Request and receive
39. Static or dynamic	40. Centralized or decentralized
41. Direct or indirect	42. Peer-to-peer or controller-slave
43. Straight-line or roundabout	44. Use shared resource
45. Simple or complex	46. Support shared dependent
47. Positive or negative	48. Shared variables
49. Direct or inverse	50. Message passing
51. Re-enforcing or detractive	52. Decision-making
53. Affect increasing or decreasing (e.g. force multiplier)	54. Informational, physical, or social
55. Supportive or unsupportive	56. Phenomenological or -ological (e.g. geological, sociological, chemical, electromagnetic, quantum-mechanical)
57. Enabling or hindering	

7.3.4.2 Conditions and Events

The assurance case needs to cover all the conditions and events that could have a significant negative effect on the conclusion of the top-level claim including its uncertainty. The potentially relevant universe of conditions and events can be hard to initially identify [2], and ascertaining which ones might have a significant effect can be difficult without at least initially including them in the assurance case. In some cases, the conditions and limitations associated with the top-level claim provide a limited universe, and it can readily be covered. However when this happens, it can be an indication that the assurance case is taking an unrealistic approach.

NOTE One set of conditions and events that should be avoided is composed of those requiring functionality or features in the system that are not part of its specification or not needed. However, if these extras are unavoidable then the resulting possible events and conditions need to be covered.

For any system interacting with humans or possibly just near humans, human factors will normally raise concerns that should be covered. Among the items that need to be considered are time (absolute and duration), activities and tasks (use, administration, maintenance, transport, storage, installation, retirement,

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

disposal), possible conditions in the environment (e.g. weather, vibration, stacking, and surfaces), input and interactions, human characteristics or behaviour, transfers of control (e.g. ownership, custody, lease, theft, capture, seizure), or modes of operation or system events or conditions. Others include opportunity or danger and their sources of uncertainties or phenomenological causes.

System dependences are an important source of concern, and might possibly result in consequences that affect stakeholder interests. History, analyses, and system characteristics and qualities can give clues to what might happen. Many lists devoted to particular domains, industries, kinds of systems, locations, environments, or qualities exist and can be consulted. Finally, to aid in addressing this identification, Annex D provides substantial but high-level lists and a number of references to online lists and other relevant material.

Historically, some kinds of conditions or events have received more attention than others. Perhaps, the problem aspect that has received the most attention is system failure. A substantial volume of checklists, practice, and literature exists concerning system failure (e.g. [2], [77] Annexes A and B, [17] Chapter 18 pages 475-524, and Annex D particularly after D.3). 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.

An assurance case will be more likely to approach completeness if it includes consideration for possibilities that are:

- Known items with relevant information about them known (obtainable) – ensuring none are overlooked.
- Known kinds of items with the relevant instance's existence, characteristics, or values unknown (known unknowns).

For completeness, one also considers the possibilities of:

- Known kinds of items whose existence, characteristics, or values are known but their relevance is unrecognized (unknown knowns).
- Items not known to be relevant or to exist and nobody knows their characteristics or values (unknown unknowns).

7.3.4.3 Subdivision of Arguments

Recognizing that the reality being argued about is often complex, the overall assurance argument needs to be broken down into layers of claims and sub-claim(s) where the objective is for the sub-claim(s) to be easier to show or closer to the evidence than the claim above them. A claim can be transformed into a sub-claim that implies it, or it can be broken into parts that together imply it. The task of the breaking a claim into sub-claims that are connected to it by an argument can be difficult. However, several bases can be used for doing this.

This connection of layers and the arguments that connect them are typically developed using both top-down and bottom-up approaches. The top-down approach breaks the claim down so smaller, more manageable arguments can be used in justifying the claim. The limitations on required property values and associated uncertainties within a top-level claim are first established deriving from analyses and the purposes, uses, expectations, and intentions regarding the assurance case. The sub-claims derive from what is required for the claims and arguments above them. What is required should be met by the established "actual" values and uncertainties derived from the claim’s supporting arguments, evidence, sub-claims, and assumptions. “Support” can include contrary as well as supportive aspects.

The bottom-up approach identifies potential sub-claims, evidence, and assumptions and either: tries to show the desired claim from them or simply asks what can be shown from them (almost always done but particularly done when assurance case is not being built to show a predefined top-level claim). A gap analysis (comparing the top-down and bottom-up results) may be used to identify what additional argumentation or argument support (sub-claims, evidence, or assumptions) are needed to justify the claim.

One basis for division into sub-claims is an argument "by cases". That is an argument that argues that (1) the claim is true for each of a set of conditions (e.g. night and day, temperature ranges, range of sizes) and (2) the set of conditions together subsume the complete condition under which the claim needs to be true. The same

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

does not follow from showing something is true of each (and every) instance. The statement, "It is true of each part, so it is true of the whole," is often false.

An argument is only relevant for the condition (often including context) it presumes or applies to. The same is true for sub-claims, evidence and assumptions. They should apply to the relevant condition and this fact can be used as a criterion to determine which evidence to create or use, what an assumption should cover, and which of the claims already shown to be true might be used to support an argument. This does not mean that evidence from similar but not identical conditions or situations is irrelevant. On the contrary, evidence concerning prior versions of the system or the same system used elsewhere can have substantial relevance even though not as much relevance as evidence from the condition or situation to which the claim directly applies.

Examples of ways of subdividing arguments include argumentation over subsystems, life cycle or usage phases, modes of operation, kinds of use, conditions of environment, phenomenological aspects, levels of abstraction, terms used in a claim, based on combining multiple measurements or multiple other properties, over development activities, over test results and other evaluative results, over history possibly over of a collection of histories of product instances, over risks, over causes or partial solutions, over kinds of consequences, or argumentation using some other existing analysis or structuring method (e.g. HAZOP or Ishikawa categories where appropriate).

Some subdivisions are based on the effects of composition of components within the system or their combination with aspects of the environment. Examples range from the composition of metals in an alloy to the composition of multiple software subprograms in a program. In these cases and for most methods of sub-dividing, predicting the effect of the sub-claims or components when combined requires considering their interaction rather than simply "forming the union" of the components or sub-claims. As mentioned earlier, the statement, "It is true of each part, so it is true of the whole," is frequently false and invalid.

Therefore, prediction regarding combinations of claim components generally requires specifying the arrangement of these components. Often these arrangements reflect the system or environment structure. As the arrangement of elements within a system is intended to produce certain results, the necessity to include this arrangement in the argument to predict some property of the result is unsurprising. Thus, the design rationale can essentially contain the same argument needed in the assurance case as it has the same purpose—to show something the designer was trying to achieve is or will be true. Likewise, the same parallel between rationales can exist for manufacturing and other activities, because the rationale for why they are a certain way applies to the properties of interest in the assurance case. Together, they are trying to achieve the properties and show this achievement.

7.3.4.4 Universality

Some arguments may be of appropriate form, but nevertheless inadequate in practice for a real system. For example, consider the following argument. If the system is in an acceptable (e.g., safe) state, each and every one of the individual actions within the system will result in the system being in an acceptable state, whether done concurrently with other actions or not. Therefore, if started in an acceptable state, the system will always remain in an acceptable state.

The possible practical problems with such an argument include:

- Some actions when performed under certain circumstances actually do take the system from an acceptable state to an unacceptable state.
- The premise concerning concurrency turns out not to be true.
- Conditions or events happen that violate the assumptions of the argument (e.g. conditions outside of those under which the system was designed to operate and behave properly).
- The system was not designed to handle certain situations (possibly the designers never thought of them).
- States thought at design time to be acceptable have unintended or unanticipated consequences that are unacceptable, such as interference among parts or something in the environment.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Underlying many of such practical problems is the problem of achieving universality, e.g., the need for everything to be correct, the need to deal successfully with everything that might happen, or the need for all consequences to be tolerable. Generally, arguments should cover situations where one or more universalities is not previously being achieved or is not true. This, of course, requires that the system or its environment have provisions for these situations either individually or collectively. A simple example of a collective provision might be that if anything not within the acceptable set of events or conditions occurs, the system shuts down and humans are notified.

Some lack of universality means that construction of an argument requires the system or its environment to make provisions for this lack that can be used as bases for argumentation. Many approaches exist under such labels as fault tolerance, safeguards, safety or security controls, safety margins, risk mitigation, and risk sharing (e.g., by acquiring insurance).

NOTE 1 In addition to making such provisions, the problem can be reduced by such actions as increasing generality of the system, designing conservatively or with safety margins, preventing or avoiding problems or their early detection and removal and exploiting the system environment to obtain help while avoiding over-reliance as well as attempting to achieve universality within the areas where universality is feasible (e.g. manufactured bolts are all within tolerances) – and perhaps also when coming close (or even closer) is feasible. Why have unnecessary problems?

NOTE 2 For software, [172] provides coverage of fault tolerance and [17] Chapter 18 addresses handling system failures.

7.3.4.5 Seeking to contradict

Experience in several fields has shown human propensities to perceive things that support their existing opinion or desired outcome while not perceiving or misinterpreting things that do not. The human tendency to conform has also been observed in occurrences of groups reaching agreements among themselves with this very agreement causing a strong loss of objectivity. Several fields have institutionalized a process that systematically introduces contradictory evidence and argument normally by having participants whose explicit role is to do so. Examples include legal trials and systems engineering red teams.

Efforts constructing assurance cases need to be concerned with these tendencies to conform. An argument is to some degree strengthened if an effort to contradict it fails if this effort could reasonably be considered serious enough that contradictory argument or evidence would have been identified if possible. Even when such efforts are not successful in contradicting the assurance case argument, they can identify issues, weaknesses, conditions, events, or other possibilities that need to be considered by the assurance case but had not previously been considered. Such efforts may also contradict only some portions of the assurance case causing them to be reworked.

A decision needs to be made concerning the kinds and amount of resources to devote to an effort at contradiction. Leaving plausible areas for discovering weaknesses or unconsidered errors is unwise. The decision on the correct amount might be made initially or later after observing the ongoing rate of return yielded during performance of the effort.

7.3.5 Summary

Arguments are the glue that holds the assurance case together. At each, from the top-level claim to underlying evidence or assumptions arguments relate the claims supported to support provided by its subordinates at the next lower level – sub-claims, evidence, or assumptions. Often an approach based on identifying all significant risks is used to structure the overall argument in the assurance case.

A variety of methods of reasoning can be used in argumentation. These vary in their applicability, power, resulting accuracy and uncertainty, and ease of use. The sub-claims, evidence, and assumptions supporting an argument have uncertainties associated with them, and the argument can increase or reduce uncertainty.

Arguments need to deal with not only "normal" conditions but also possibilities that parts of the system will not behave as intended and that unforeseen events or conditions can occur including unintended or unforeseen consequences.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Assurance cases often provide rationale for a system being appropriate for a use. When an assurance case fails to show the top-level claim one can change the system or strengthen the assurance case. However, a better choice for systems that already exist or are well along in their development, is to reconsider:

- What claim can be adequately argued?
- What use can be made of the system given this weaker or different claim?

Possibly the system can still be used under limited conditions, in a smaller market, or with added safeguards in its environment, or the decision can still be made not to use or develop the system. If nevertheless used as originally intended, stakeholders should accept added risk and tolerate reduced benefits or other adverse consequences.

NOTE Standards or approaches labelled as being for "evaluation" or "assessment" can sometimes be useful in identifying argumentation methods or methods of combining evidence as well as in identifying relevant information for use as evidence. This is true for "evaluation" or "assessment" of system, process, technology, organizational aspects, and particular qualities.

7.4 Evidence

7.4.1 Introduction

A close connection exists between argument and evidence. The evidence needs to support the argument used, and the argument needs to be such as to effectively use the evidence – evidence either customarily obtained or especially identified or designed (e.g. especially collected or created). Constructing arguments to include the use of evidence that already exists or will be created or collected anyway is efficient and often necessary. However, custom evidence can be needed to fill gaps in this evidence, and it can be designed to provide especially effective support.

Some evidence, such as results from certain types of testing, can be easy to ascertain the meaning of, have a known uncertainty, and can easily have their meaningfulness to the argument established. Others, such as those derived from inadequate or incorrect sampling, can have meanings that are difficult to clearly identify, cannot be readily generalized, or leave huge amounts of uncertainty.

Evidence can be generated as a customary result or artefact or because it is needed for the assurance case or for certification or licensure. All evidence might be useful in the assurance case, and no evidence should be unthinkingly ignored. Subclauses cover General issues, Meaning and meaningfulness, Kinds of evidence, and Assessments, certifications, and accreditations.

NOTE 1 Depending on how the creator of the assurance case conceptualizes it and the amount of inference that has already been done from the evidence outside of the assurance case, evidence can be said to be supporting an argument or directly supporting a claim.

NOTE 2 A distinction is made between direct evidence that reflects relevant properties directly and backing evidence that concerns the nature, quality, characteristics, and history of the evidence.

Evidence not only is better if it is derived from the conditions that the claim applies to, but also is better when together it is relevant to the entire relevant condition. Showing that this is true of the evidence is another argument that brings its supports together to show its claim.

7.4.2 General

For any area or property, many means of obtaining evidence exist. Among these are human experience, history, observations, measurements, tests, evaluative and compliance results, analyses, defects, and inferences. Evidence can already exist, be newly created or collected, or be planned for the future. The evidence should achieve the objectives claimed in the assurance argument [[147] MoD DefStan 00-42 Part 3, section 9.1] and should be obtained both for the argument and against the argument (counter-evidence). The body of evidence can become quite large and may need to be organized by some evidence framework.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Among the areas that evidence can be derived from are:

- The entire lifecycle and across the supply chain.
- The environment.
- Intentions.
- Processes.
- Means and resources (including people and tools).
- Work products.
- Field experience.
- Support.
- Capabilities (possibly not yet exercised).

When judging the credence to be given to a piece of evidence, its relevance, visibility, traceability, and quality are crucial factors. Quality issues concern both the origination and the preservation and handling of the evidence. Origination-related issues include feasibility, conformance to standards and procedures, validity, subjectivity, accuracy, uncertainty (e.g., measurement uncertainty), affordability, and, ultimately, credibility and usefulness. Therefore, one should confirm that the evidence is generated or collected, managed, validated, stored, and used using acceptable practices and controls. SafSec states, "Evidence shall be permanent, traceable and managed in such a way as to provide later readers confidence in its source, contents and validity.” [[183], page 9]. Another guidebook [161] 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 (CM).
- Evidence needs to be accompanied by the metadata needed to properly use it within the assurance case.

Weaknesses in evidence's validity can significantly affect, even destroy, its usefulness.

7.4.3 Meaning and meaningfulness

To properly use evidence in support of an argument or claim, both its meaning and meaningfulness need to be established. The meaning(s) of evidence can be easy or difficult to ascertain, but it generally reflects the:

- Subject of the evidence.
- Properties about which the evidence is relevant.

For example, one might need to ask, "What are we really measuring here?".

The evidence's meaningfulness to each argument or claim that it supports is usually influenced by:

- Its accuracy and the uncertainty associated with its accuracy.
- Its generalizability beyond the instances from which it directly originated.
- The remaining uncertainty about inferences from the evidence (e.g., as reflected in its statistical significance in testing a hypothesis).
- The relevance of its meaning to the argument's or claim's subject and its attributes (e.g., entity, property, conditions, and durations of interest).

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Usefulness of evidence can be reduced if it cannot be stated in terms that can be used by the method of reasoning being used, e.g., not stated probabilistically. The usefulness and meaningfulness of evidence is addressed in several standards or guides. For example, [161] states:

- Evidence should be sufficient for its use in the assurance case arguments, including both its quality and its provenance (history).
- The stated context and criteria apply to each piece of evidence. (e.g., relevance to version, and conditions and duration of applicability)
- Where inputs, states, or conditions can vary, the evidence covers all possibilities or a sufficient sample of them to justify the argument.

The last point addresses generalizability. Many analyses cover all possibilities but few tests do; however, some tests may ensure detection of all faults of a certain kind. In addition, some tests are more generalizable, such as if they use a statistically sound sample.

Documents [197] and [199] include material relevant to evidence in general and from analyses, testing, and field service experience. The following list, loosely adapted and enlarged from these two sources, covers points related to making testing evidence more meaningful and providing backing evidence:

a) Test guidance, procedures, standards and tools defined.

b) Tools used validated and verified.

c) Test procedures validated and verified.

d) Test equipment calibrated and resulting certificates available.

e) Testing covers where possible and practical:

   1) Complete top-level claim.

   2) Needs of arguments (or sub-claims) directly supported by testing.

   3) Relevant properties.

   4) Top-level claim conditions and what is possible during its duration of applicability.

   5) Adequate coverage of the input domain.

   6) All possibilities or can be generalized to them.

   7) Any aspect not adequately covered by tests is covered by another method such that altogether adequate coverage results.

f) Tests reflect needed (low) uncertainty:

   1) Needed uncertainty reflected in documents governing testing.

   2) Needed uncertainty reflected in test specifications.

   3) Complexity of claims and related input analysed and used in selection of test data.

   4) Consequences of failing analysed and used in selection of test data.

   5) Tests adequately thorough.

   6) Needed uncertainty reflected in test criteria and criteria for ending testing.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

g) The test methods and techniques used are appropriate for the properties under consideration.

h) Rationale for item and item recorded, reviewed, and subject to audit for the following:

  1) Test specifications – created independently.
  2) Objective for each test.
  3) Test procedure.
  4) Quality evidence for test procedures.
  5) Test criteria (e.g. for acceptable test results) complete and correctly reflects support needed.
  6) Test results.
  7) An analysis of test results.
  8) Detected and implied faults analysed.

i) Testing versus operation:

  1) Tests' configurations identical to operational.
  2) Differences between the operational and test environments identified and effects assessed.

j) Conduct of testing:

  1) Testing performed independently.
  2) Test guidance, procedures, standards and tools followed:
     i) Procedures or tools used to ensure:
        I) Testing follows test procedure
        II) Results satisfy the test criteria
  3) Test observed independently and reports produced.
  4) Test environment and activities recorded accurately.

k) Tests meet test criteria.

l) Testing results provide required support for arguments (or claims) directly supported.

This last point is simply a restatement of what testing is supposed to achieve related to the assurance case. This list does not include more general evidence regarding testing such as personnel competence, and adequate time, resources, and facilities. Reflecting reality, the list is long and varies from concerns that apply to all testing to those that some might use only when unusually low uncertainty is required. The sources give limited indication of where the items in the lists lie along this dimension.

7.4.4 Kinds of evidence

The introduction (7.4.1) provided a broad list of areas from which evidence might be derived. Further evidence that can contribute to the assurance case includes ensuring that, for each of the areas listed below, the evidence is adequate and that adequate observations and measurements are created or collected and recorded:

1) The quality and history of the people who produced it.

2) The quality and history of the tools used in producing it.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

3) The quality of the environment in which it was produced.

4) The characteristics and history of the processes, activities, tasks, methods, techniques, and technology used to produce it.

5) The quality of the definitions, policy, and procedures governing the processes, activities, tasks, methods, and techniques used in production and the fidelity with which they were followed.

6) Characteristics of the designs including the extent to which they can be practically reasoned about and provide resilience.

7) Quality and results of analysis and simulations.

8) Results of reviews, audits, tests, and other evaluations of the system.

9) Proofs.

10) The execution and operations history of the system.

11) Related experience and consequences.

NOTE IEC 60300-3-2:1993, Dependability management – Part 3: Application guide – Section 2: Collection of dependability data from the field [70] can be relevant – even in some situations not involving dependability properties.

12) Indications of the realism of the assumptions made.

This list does not cover all the evidence that is needed. Consider the lists in [182] and [183] that include a number of areas on which risk-oriented arguments might be built:

- Organizational: the goal is achieved by some organization.
- Procedural: certain actions have been carried out.
- Risk Directed Design: “document a justification for achievement, by the system, of each residual risk; and document a justification that the evidence of achievement of risk reduction is appropriate for the level and importance of the risk reduction.”
- Modular Certification and Technical Modularity: organizational or system interfaces, particularly with external systems, need the “other” side of the interface to justifiably have the assured qualities claimed. In addition, “Module boundaries shall match the organizational boundaries.”
- Evidence: requirements have been established for the recording, handling, and characteristics of evidence to be used.
- Evaluation/Assessment: the project documented a means of demonstrating the achievement, by the system, of each residual risk to a degree of uncertainty appropriate for that risk, obtain agreements on evaluations and assessments among the parties involved, and carry them out successfully (as determined by evaluation/assessment results).

Evidence should be selected based on the need to support arguments and the verification and validation activities. However, if the system is not as claimed, supporting evidence should be harder to obtain and contradictory evidence easier. Thus, similar to the need to achieve claims, the need for evidence drives system development and maintenance decisions. These decisions include evidence selection, generation, and maintenance as well as making this evidence easier to obtain.

7.4.5 Assessments, certifications, and accreditations

A substantial body of relevant experience and practices exists in the assessment, certification, and accreditation communities. Certifications and their related techniques can add to the evidence available for the assurance case and an assurance case can supply evidence needed in certification. However, many

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

regulatory agencies and certification processes do not offer the freedom for system producers to provide what they consider to be the best assurance case and to use it for approval, certification, or proof of compliance.

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 *Information technology. Code of Practice for Information Security Management* combined with ISO/IEC 27001 (formerly with UK standard BS7799-2:2002) 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. [147], [150], [151], [197], and [198]. Many ISO-related standards are listed in the Bibliography.

Standards exist addressing the assessment of software and systems processes. Three examples are ISO/IEC 15504 *Information technology -- Process assessment*, the Capability Maturity Model Integration (CMMI) from the Software Engineering Institute in the U.S., and ISO 21827 - *Systems Security Engineering Capability Maturity Model*.

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 – for example, certifications from the International Information Systems Security Certification Consortium (ISC)^2 and the SANS Institute.

7.5 Management and life cycle of assurance case

Management and life cycle activities include those directly involving the assurance case and the effect that the assurance case has on other activities. Results would be best if the assurance case were considered from the beginning of concept development, used to influence all activities and systems, and became an integral part of the overall engineering process. These activities could all be done only if the system and the assurance case were being developed concurrently. The scope of the set of activities covered by ISO/IEC 15026-2 is different if the assurance case is developed concurrently with a system or developed for an existing system.

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.

The life cycle of the assurance case is not always the same as that of the system. Clause 9 more fully covers the assurance case within the system life cycle processes. The assurance case life cycle is covered normatively in ISO/IEC 15026-2 only insomuch as to adequately ensure that the quality and usefulness of the assurance case would not be clearly endangered by actions within processes, activities, and tasks. The full possibilities for relationships and integration between the system and assurance case life cycles are covered in ISO/IEC 15026-4.

Activities involving the assurance case can extend beyond its duration of applicability to cover areas such as archiving and any obligations and liabilities remaining after its duration of applicability. Process or activity issues include not only the "process" in a limited way but assurance case methods, practices, techniques, tools, and environment as well as the responsibilities, competence, motivations, ethics, independence, and organizational affiliation of all involved. Concern for the effort being well orchestrated can include having a single individual being responsible for the entire assurance case.

Concurrent maintenance is covered during the duration of applicability as well as during any concurrent development with the system. Considering just the assurance case itself several specific activities should be done such as configuration management and approvals.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

One of the principal uses of the assurance case is in risk management. Following ISO/IEC 16085 [97], all risks should be considered concurrently. In practice, this includes risks related to:

- The system and its life cycle.
- The assurance case and its life cycle.
- The project.
- Organizations.
- Individuals.
- Assets inside and outside the system.
- The environment.
- Governments and regulators.
- Society and nations and their interests.

7.6 Decision making using the assurance case

7.6.1 Introduction

The assurance case provides information and reduced uncertainty to decision makers. It provides a basis for confidence that end-users need in the system before they feel comfortable using it and that the producer needs before releasing the system.

While activities such as independent evaluation add to grounds for confidence, the bulk of the wherewithal for the assurance case might be satisfied as part of the processes that produce the system. Without this, how would the producer rationally have and maintain the confidence they need? Moreover, the absence of such wherewithal might be grounds to support a determination of inadequacy.

Most decision makers do not need the full assurance case but need presentations with the relevant content in a form they can understand and use. Care is needed to ensure that such presentations and the full assurance case are consistent and that needed information is not missing from the presentations.

A user of the assurance case might need to answer three questions:

- How confident am I in the accuracy of the assurance case?
- Under what circumstances is the system trustworthy?
- Shall I actually place trust (reliance) in the system?

Related to these are the questions of:

a) What does the claim make as a claim about the system’s (future) behaviour?

b) How good is the agreement of the system’s (future) behaviour with its claim?

c) How uncertain should I be about my answer concerning the system’s agreement with its claim?

d) What will happen in the environment?

e) How uncertain should I be about my predictions concerning the products environment?

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

and additionally:

f) What does the combination of answers to a)-d) mean?

g) How uncertain should I be about my answer to what the combination means?

h) What will happen?

i) How uncertain should I be about my answer to what will happen?

And lastly:

j) What should I decide to do?

For many kinds of systems, such as large systems or pieces of software, answering these questions has always been difficult. Some assurance cases might only answer a) and a non-malicious version of c). This would appear to leave a number of questions unanswered. In theory, an assurance case should answer a) through i), thereby making the answer to j), easy.

7.6.2 Degree of assurance and confidence

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. Arguably, "high-confidence" is not a synonym for "low-uncertainty" or "high-assurance". It is possible to have a high degree of unjustified confidence. This conversion of an amount of uncertainty (e.g., related to the correctness of the as-built system) into a degree of justified confidence in suitability for certain applications is not straightforward or well understood. This situation can be exacerbated when maliciousness is involved.

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.

Historically, the use of assurance has often included the assignment of risk-based levels that were used to match the functionality and uncertainty of the system or system element with its use as well as giving guidance to producers aimed at achieving the appropriate functionality and uncertainty. First, a structure of levels or categories might include terms or methods for answering a) through c). Examples for a) might include automobile kilometres per litre and crash ratings, fire retardant rating, safe or vault rating, and credit rating. For b) an example might be a reliability rating, and for c) the standard deviation yielded by reliability testing. Presumably, the degrees of inadequacy and uncertainty in meeting a claim are related to how the system was produced and to what evidence was collected and how. Production and collection could be guided by the assigned level, as with integrity levels (Clause 8). The nature of this guidance and the fidelity with which it was followed could give input to b) and possibly c). Possible consequences might be indicated by asset values or sensitivity levels, e.g., secret or top secret.

Levels have sometimes been used to give the answer to d) (e.g., level of economic activity or threat level) and the desired answer to f). Levels are also sometimes used to invert the process used to answer f) and solve for a) (required behaviour) as well as give corresponding guidance to ensure this answer to a) will be achieved (or possibly bettered) and yield an acceptable (or tolerable) answer to b). This is done, for example, in doing analysis to go from level of risk to required integrity level.

When this cannot be done for lack of input or an invertible process, the safest answer in the worst case is to give guidance aimed at achieving the best possible answers to a) and b), although this is likely to be overkill. Having done so, the question becomes, "Is the best possible good enough?" In practice, the theoretically best usually exceeds the best feasible and this in turn exceeds what is practicable and affordable.

Despite difficulties, these are fundamental questions that need to be addressed in a practical way. The assurance case with its claims, arguments, and evidence provides one such way.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

8 ISO/IEC 15026 and integrity levels

8.1 Introduction

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. This clause outlines some of the issues and concepts underlying integrity levels and their use particularly for the users and potential users of ISO/IEC 15026-3, *System integrity levels*. Integrity levels have been useful in the past to users of ISO/IEC 15026:1998, and should be even more useful to users of a revised and improved version integrated with the remainder of ISO/IEC 15026.

Figure 6 shows an overview of the mental model underlying much of ISO/IEC 15026-3. Consequences take their values from their affect on the interests (e.g. on their funds, health, equipment, or natural environment) of stakeholders. Such a consequence occurs when its precondition exists, and such preconditions occur as a result of conditions in the system's environment and the system's behaviour. A behaviour of the system occurs when its precondition exists within the system, and conditions in the system occur because of prior conditions, internal behaviours within the system some of which are the result of input from the environment. Likewise conditions in the environment can result from initiating events in the environment.

A tree of events and conditions can lead to the preconditions for consequences at their roots. Sequences of conditions and initiating or transition events are often not inevitable; the transitions can be treated as chance events and consideration given to their timing and possible variations in size or value.

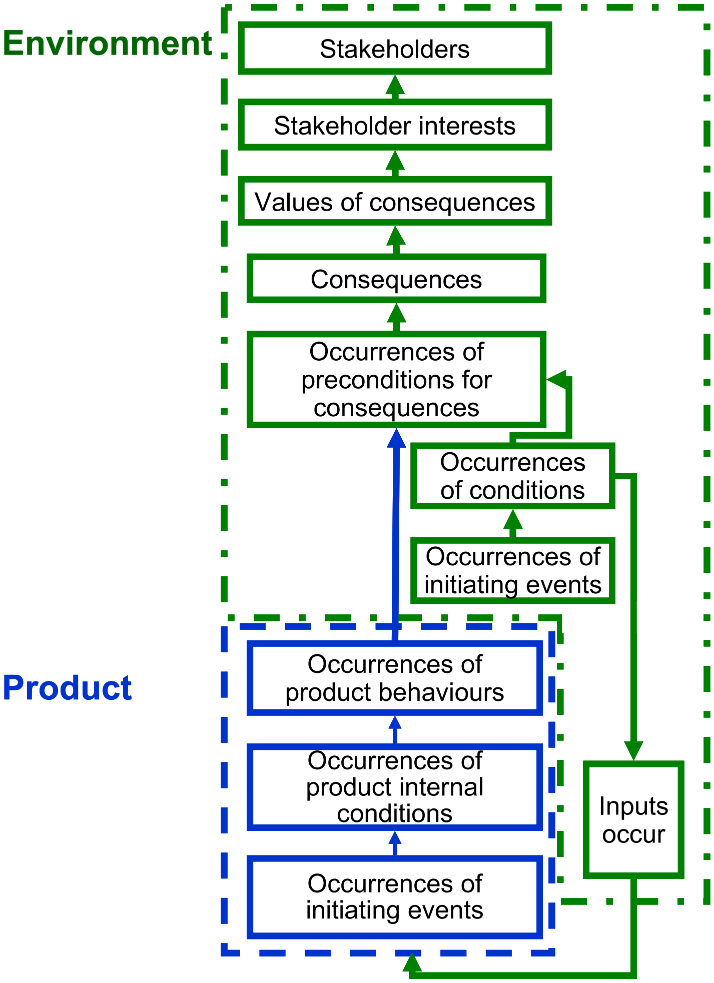

Environment
Stakeholders
Stakeholder interests
Values of consequences
Consequences
Occurrences of preconditions for consequences
Occurrences of conditions
Occurrences of initiating events
Product
Occurrences of product behaviours
Occurrences of product internal conditions
Occurrences of initiating events
Inputs occur

Figure 6 — System and Environment

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

A basic strategy is the prevention of preconditions for adverse consequences, the initiating or transition events leading to them, and the preconditions for those. Another strategy is to limit the possible adverse consequences, e.g. their size, duration, and propagation.

Subclauses cover Defining integrity levels, Establishing integrity levels, Planning and performing using integrity levels, the key issues of Conditions and their initiating or transitioning events, Issues and limitations in using ISO/IEC 15026-3, Outcomes of use, and a brief Summary.

8.2 Defining integrity levels

The specifications associated with an integrity level document two kinds of related requirements called the "integrity level" and "integrity level requirements":

a) **"Integrity level" –** What the integrity level **fulfils or claims:** namely that the system or element meets:

   1) A certain target for a property such as risk, reliability, or occurrences of dangerous failures.

   2) Within specified uncertainty limitations.

   3) Under specified conditions.

b) **"Integrity level requirements" –** What it **imposes** on:

   1) What is done and how, when, etc. – including on organization, processes, activities, tasks, methods, means and resources including personnel and tools, work environment, communication, management or coordination, record keeping, and other aspects of performance.

   2) The system or element – including on associated material, services, and artefacts.

   3) The evidence to be obtained possibly including limitations on its associated uncertainty.

The first, the requirements to fulfil, are called the "integrity level." For clarity and to better specify what is meant, ISO/IEC 15026-3 sometimes uses the term "integrity level's claim" for the requirements the integrity level fulfils.

Ultimately, evidence is central to designing and evaluating the design of the integrity levels. To be acceptably established, the designed levels should be shown to be such that:

1) If the required evidence exists and meets the criteria regarding it, it will be adequate to meet the required limitations on property values and uncertainties implied by the requirements it fulfils from a).

2) Meeting the integrity level requirements imposed on evidence by an integrity level will show the meeting of all the integrity level's requirements.

The requirements related to integrity levels can differ for different circumstances such as aterials (e.g. software versus concrete) or construction or testing techniques (e.g. destructive versus non-destructive).

Whether thinking in terms of individual claims or assurance cases, the needed system integrity levels derive from the limitations on the values and uncertainties regarding the property values of the system itself and/or its elements (e.g. behaviours and contents). These properties of the system itself are under specified conditions that often include aspects of its environment. At least conceptually, these derive in turn from the limitations regarding consequences or from the top-level claim of the assurance case. More particularly for the assurance case, the evidence generated in conforming to an integrity level should adequately support the sub-claims supported by it.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

8.3 Establishing integrity levels

8.3.1 Introduction

Risk analysis is used to establish the needed integrity level for the entire system. This risk analysis may or may not involve an assurance case. Once the integrity level is established for the entire system, integrity levels need to be established for what it depends upon including its internal elements.

8.3.2 Risk analysis

Risk analysis establishes the required integrity level for the entire system. The activities in risk analysis cannot generally succeed by only an outside-in or inside-out approach (likewise top-down or bottom-up), nor can they succeed in a single attempt. The analysis needs to be approached from several directions, and a serious early effort can be useful. Risk analysis is an ongoing and iterative process that should balance what is not yet knowable with what needs to be known and that should be prepared for learning and change.

Therefore all of the activities will normally benefit from the following steps:

a) Involvement of relevant expertise.

b) Examination of the environment of the system with resulting identifications.

c) Review of history relevant to the situation and similar situations.

d) Review of relevant standards and publications.

e) Serious concern for completeness and efforts to evaluate the degree of completeness of results.

f) Build an improving representation and understanding of the situation and keep records of information relevant to this even if it is not immediately needed.

To establish what the required system integrity levels are, one establishes the following:

a) The real-world requirements on consequences.

b) The limitations on values and associated uncertainties of claims regarding consequences.

c) What these consequence-related limitations imply are the required limitations on values and their associated uncertainties regarding claimed properties of the system itself and its elements.

d) The combination of design and properties of the implementation is required within the system to ensure meeting these limitations on values and uncertainties.

These need to be followed by establishing:

a) The integrity level requirements that if met will adequately assure the limitations on property values and their uncertainties required of the implementation of a system and its elements for them to be adequate in combination with the design.

b) What should be done and shown to establish that the realization of a system and its elements meets these integrity level requirements within the limitations on uncertainty.

Consequently, integrity levels most directly derive from the severities of the property values and associated limitations on uncertainties regarding whether the system's implementation adequately meets its verification-related claims about the properties of the system itself. 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

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

be done and shown for various ranges and severities of limitations on property values and their associated uncertainties.

NOTE Property values and their uncertainty values can vary in meaning. The uncertainty of the correctness of a given response might be reasonably thought of as the related reliability of the system. On the other hand, the uncertainty regarding whether the system's reliability is within a range, e.g., greater than a certain value or between two values, is distinctly different than the reliability of the system.

ISO/IEC 15026, including ISO/IEC 15026-3, 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 "Functional safety of electrical/electronic/programmable electronic safety-related systems" [75] and IEC 300-3-9 "Risk Analysis of Technological Systems" provide approaches to risk analysis. As safety-specific terminology is used in IEC 300-3-9, the terms "hazard" and “harm” should be interpreted as "dangerous condition" and “adverse consequence,” respectively. IEC 60300 Dependability management [69] also provides guidance. Annex D can help with identification of dangers.

Other specialized standards include ISO 13849 [89] on machinery, ISO 14620 [90] on space systems, ISO 19706 [104] on fire, ISO/TS 25238 [116] on health informatics, ISO/IEC FDIS 27005 [124] on information security, and UK CAP 760 [199] on air traffic and airports. Also of possible interest are the more general risk management standards ISO/IEC 16085 [97] and ISO/IEC 15939 [96].

8.3.3 Element integrity levels

Once risk analysis has established an integrity level for the entire system, integrity levels need to be established for what the system depends on. This can be complicated by not being free to assign integrity levels for dependences on existing external elements or existing internal elements being reused. Thus, given such constraints and the integrity levels required of the system’s behaviour at its various external interfaces, required integrity levels need to be assigned to elements whose behaviours are depended upon including internal elements.

An element needs to be assigned an integrity level at least as high as that required by any of its uses. The integrity level required by a use depends on the integrity level required of the element using it and its role within the design of this using element.

Redundancy, diversity, and separation or isolation can affect this level. If failure of an element can only result in a dangerous condition in combination with other elements being in a particular state, then possibly one can assign a less stringent limitation to the element's failure occurrences than otherwise assigned. If an element offers one among several opportunities offered by different elements to result in a dangerous condition, then its integrity level might possibly need to be higher than that required of their combined behaviour.

While often the same as that of a using element, the property (possibly including multiple primitive properties) required can be affected by its role within its using element's design.

Thus, a system element is assigned the highest integrity level derived from:

a) Required integrity levels of system interfaces it provides.

b) Integrity levels of using elements and its place in the design of each using element.

8.4 Planning and performing

After establishing and assigning integrity levels with their integrity level requirements, one needs to do the following:

a) Plan to perform what is required to meet and show that the system has (or had or will have) integrity levels.

b) Perform what is required to meet and show that the system has (or had or will have) integrity levels including obtaining evidence to show this.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

And finally, one uses what has been shown:

c) Review, gain approval, and communicate to create needed corrective action, confidence and/or quality of decisions.

Plans derive from real-world realities and are driven in part by integrity-level-requirements-related evidence concerns including obtaining it and ensuring its required values and quality.

8.5 Conditions and their initiating or transitioning events

Outside the system, much of the reasoning is based on concern for 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.

Figure 7 shows how these concerns interact. The dangerousness of system behaviours can differ by the conditions of its environment. As shown by the arrows above the line, 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.

Likewise the system or its designers might or might not be cognizant of all the initiating events for a condition within the environment. Thus, dangerous conditions can need to be dealt with even though not all of their possible initiating events are known or recognizable.

8.6 Issues

For the approach to integrity levels presented herein to be most effective, several issues regarding the system's specification and analysability need to be addressed. For example, one question that needs to be resolved is, "Do system behaviours exist that can lead to adverse consequences but are not forbidden by the specification or can dangerous conditions exist within the system that are not categorized as errors by the design specification?'.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

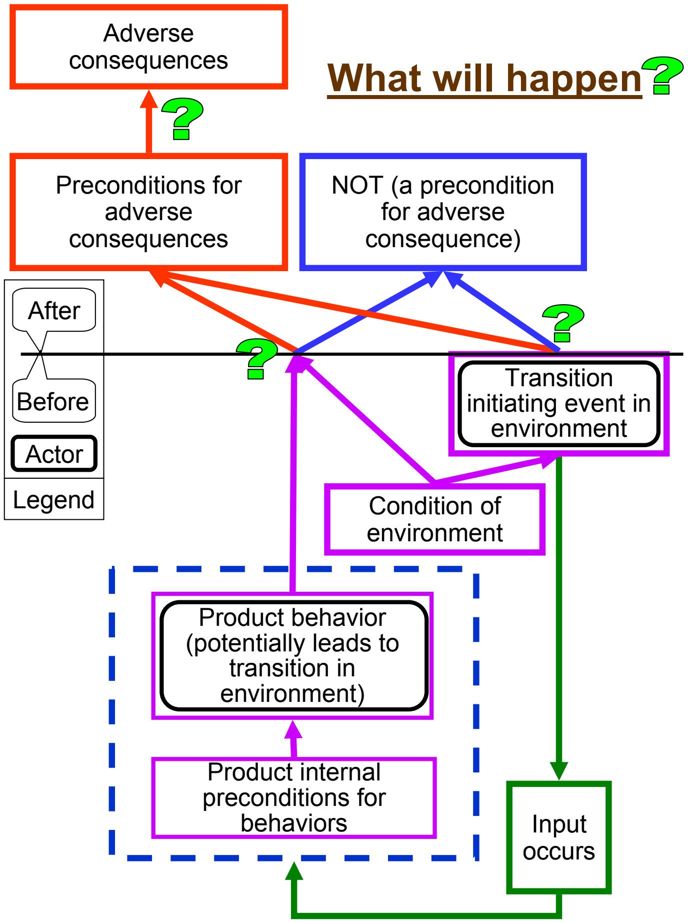

Adverse consequences
What will happen?
?
Preconditions for adverse consequences
NOT (a precondition for adverse consequence)
After
Before
Actor
Legend
?
?
Transition initiating event in environment
Condition of environment
Product behavior (potentially leads to transition in environment)
Product internal preconditions for behaviors
Input occurs

Figure 7 — Two actors cause transitions

Some portions of ISO/IEC 15026-3 presume the analysability of the system and complete knowledge of its relevant relationships with its environment as well as all system behaviours leading to dangerous conditions being failures to meet specifications – unless they are deliberately intended to be allowed. In addition, ISO/IEC 15026-3 often presumes for purposes of analyses that dangerous conditions have identifiable initiating events or causes that can be used during analyses. Users of ISO/IEC 15026-3 need to maintain awareness of the limitations resulting from these presumptions.

The portion of ISO/IEC 15026-3 covering the establishment of integrity levels generally presumes no such behaviours exist but warns they might. Particularly for systems other than analysable ones and situations where the relationship between system behaviours and adverse consequences is not firmly established, possibly substantial uncertainty can exist concerning the existence of such behaviours.

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 [59] [60].

Thus, regardless of how they arise, dangerous conditions and adverse consequences are subjects for mitigation.

Complexity and lack of understanding or predictability of the system or relevant environment can create a situation where the approach provided in the international standard ISO/IEC 15026-3 promises less than in situations with predictability and analysability. Nevertheless, ISO/IEC 15026-3 provides useful improvements over ISO/IEC 15026:1998 and explicitly covers issues only tacitly covered in many integrity-level-related

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

documents. In addition to explicitly recognizing its shortcomings and limitations on applicability, it covers defining individual and sets of integrity levels and their integrity level requirements; customized risk criterions; dealing with dependencies in general and not just internal elements; variations among risks at different system interfaces; and certain kinds of difficult-to-acquire knowledge. Finally, it provides a generic set of requirements, guidance, and recommendations useful to developers of more specialized standards or other governing documents.

8.7 Outcomes

In using integrity levels, the following kinds of outcomes occur:

a) Requirements established for each integrity level because of its role in assurance—particularly in any assurance cases.

b) Specification established for the requirements (criteria) on a system or system element whenever it is assigned a particular integrity level.

c) Decision factors and process established for deciding assignments of integrity levels.

d) Assignment of integrity level to each system or system element (or portion).

e) Showing achievement of assigned integrity levels (that is, meeting the assigned level's criteria).

These outcomes, of course, imply activities exist to:

a) Plan for achieving them.

b) Achieve them.

c) Ensure related activities can be and are done adequately.

d) Document related plans, performance, inputs, and outputs including results, evaluations, and related approvals.

e) Obtain related agreements or approvals.

ISO/IEC 15026-3 emphasizes obtaining agreements between the design authority and integrity assurance authority and possibly obtaining approvals by the integrity assurance authority).

8.8 Summary

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, possibly differing at external interfaces. 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, then 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 including their affects on uncertainties. Argument and evidence (as well as assumption) related uncertainties are a central concern and part of establishing integrity level requirements.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

In practice, obtaining agreements or approvals is an important part of ensuring that integrity-level-related requirements are met. ISO/IEC 15026-3 uses such terms as independent approval authority, design authority, and integrity assurance authority for particular roles.

9 ISO/IEC 15026 and life cycle processes: 15288/12207

9.1 Introduction

An assurance case is seldom created without affecting the system's life cycle. The relationship between the assurance case and the system exists throughout the relevant portions of the life cycle including during the duration of applicability of the top-level claim and in regard to all relevant ISO/IEC 15288 processes.

Figure 8 provides an overview of the life cycle processes of both ISO/IEC 15288 and ISO/IEC 12207. In overall terms, it shows that an organization conducts projects in order to satisfy its goals and these projects deal with systems. Each box depicts a process and those processes are classified as relevant to an organization, its projects, or its systems. The project-enabling processes are executed by an organization to support its projects. The organization also executes agreement processes in order to do business with other organizations. Each project is managed by the project management processes, with appropriate support from the project support processes.

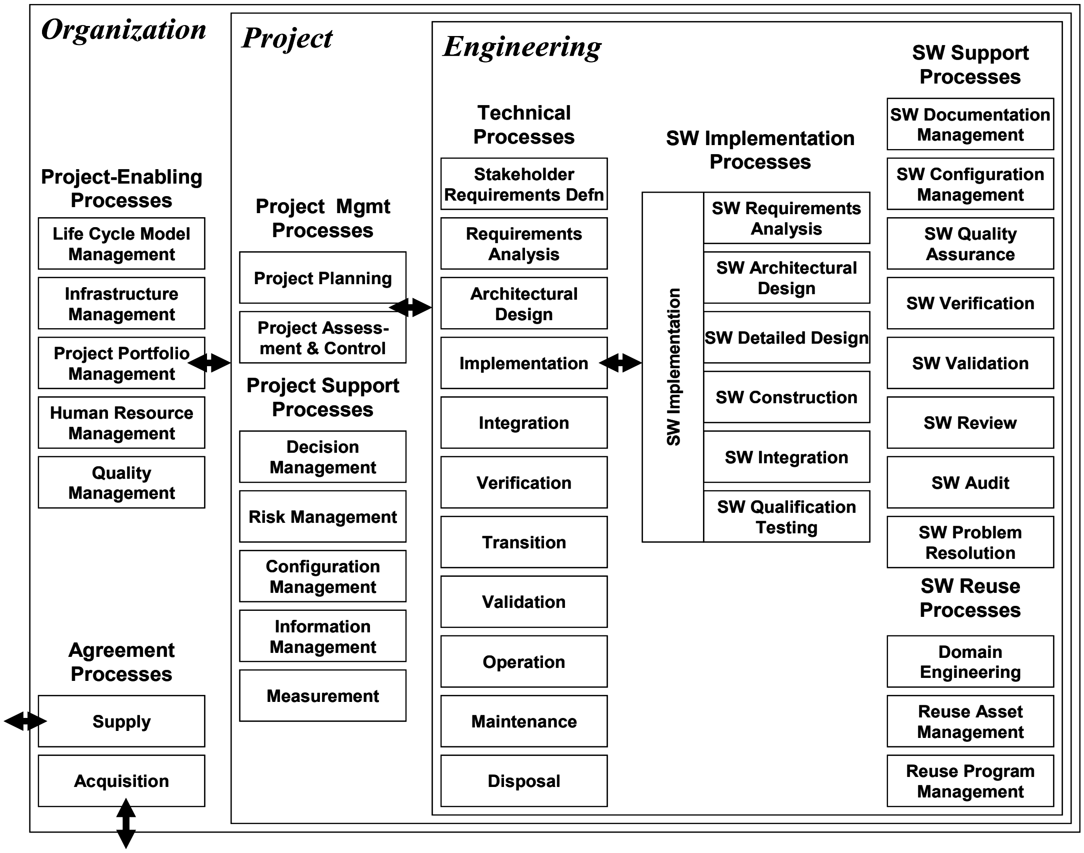


Figure 8 — Life cycle process groups

NOTE To support the assurance case, one commonly needs the execution of a planned and systematic set of activities 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 [153]. "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

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

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 product’s requirements possible. [4]

All of these processes are described in both ISO/IEC 15288 and ISO/IEC 12207 although the processes in ISO/IEC 12207 are specialized to software and, in some cases, have different names reflecting that specialization. ISO/IEC 12207 contains additional processes unique to software. They are depicted as software implementation processes, support processes, and reuse processes. ISO/IEC DTR 24748-1 *Systems and software engineering — Guide for life cycle management* [125] can aid in better understanding life cycles.

While ISO/IEC 15288 and ISO/IEC 12207 are used as a baseline for discussion in this clause and ISO/IEC 15026-4, Assurance in the life cycle, their use is not required for conformance to ISO/IEC 15026-4. Any life cycle and its processes that meet the requirements of ISO/IEC 15026-4 for life cycle processes may be used.

ISO/IEC 15026 is suitable for use as part of an acquisition or supply agreement. The project-enabling processes depicted in the organization part of the diagram can have substantial affects on assurance- and assurance-case-related activities and artefacts as well as on the system, but generally these affects are less direct than project-related ones. .

Major subclauses cover Technical processes, Post-development, Organization processes including acquisition and supply, and a brief Summary.

9.2 Technical processes

9.2.1 Introduction

ISO/IEC 15026 is concerned with engineering from the perspectives of positive methods and avoiding pitfalls. Despite its apparent orientation towards undesirable events, conditions, and affects, it reflects that both realizing an adequate system and being sure that the system is adequate involve these two aspects. Life cycle processes including activities, tasks, and methods need to reflect this as well as show adequate achievement via an assurance case and possibly though meeting integrity level requirements.

This subclause uses the structure of the Technical Processes in ISO/IEC 15288 to mention a number of aspects relevant to users of ISO/IEC 15026. Unless taken into account from the beginning of the system life cycle, many of the requirements and claims of interest are difficult to achieve, since retrofitting them is difficult or impractical. An assurance case is best if developed from the beginning of the system life cycle so that it influences every activity, including planning, conducting, managing, and evaluating the system as well as the assurance case.

9.2.2 Stakeholder requirements definition

Top-level claims and their properties often derive from system requirements. ISO/IEC 15288 in its 6.4.1 Stakeholder Requirements Definition Process lists a number of areas from which requirements can be derived and in 6.4.1.2 Outcomes states one outcome as specification of the required characteristics, attributes, and functional and performance requirements for a system solution. Within Stakeholder Requirements Definition Process in 6.4.1.3, it mentions:

- Consequences of existing agreements, management decisions and technical decisions.
- Activities and sets of activity sequences.

[4] “Do the right thing, do it right, and show it is right.” Sam Redwine

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

- Relevant characteristics of the end users of the system.
- The physical environment, social and organizational influences.
- Interaction between users and the system including the areas of:
- Physical, mental, and learned capabilities.
- Work place, environment and facilities, including other equipment in the context of use.
- Normal, unusual, and emergency conditions.
- Operator and user recruitment, training and culture.
- Critical qualities such as health, safety, security, and environmental damage.

Interactions with stakeholders result in a list of needs and preferences related to the system, its properties of interest and associated limitations on uncertainty; an understanding of the constraints, e.g., relevant laws, regulations, standards, policies, guidelines, and interfaces; and a list of compliance, contractual, evaluation, certification, or approval requirements. Organizations may have existing enterprise architectures, including security elements and frameworks that unify the means of complying to governing directives. Table 6 lists some common types of stakeholder needs.

Stakeholders have needs and requirements to limit adverse consequences and uncertainties. Unfortunately, the normal situation for systems and software includes maliciousness and often an insecure environment. This situation adds to the non-malicious problems and makes concern for dangers and their sources and affects particularly important and relevant. Table 7 lists some of the common kinds of information about sources of danger particularly when safety and/or security are concerns.

Table 6 — Some kinds of stakeholder needs

Decision making	Limitations on uncertainty
Consequences	Limitations on expenses and use of resources and time
Stakeholder interests and asset protection	Interface and environment requirements
Compliance	Trustworthiness and trust management
Usability	Reliability
Availability, tolerance, and survivability	Maintainability, sustainability, and evolvability
Deception	Assurability
Validatability, verifiability, and evaluatability	Certification and accreditation
Market success	

Table 7 — Information about sources of danger

Causes, control, and motivations	Duration or persistence, frequency, and timing
Links and relationships	Capacity to, and causes and chance of change
Propensities and intentions	Limitations and dependencies
Capabilities and resources	Possible uses, conditions, and roles
Violations, damage, and losses	Methods and approaches
Gains	Warnings

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

These concerns for potential real-world consequences associated with the system and its environment throughout its life cycle are usually accompanied by concern for related events, conditions, benefits, losses, and expenses. Limitations on uncertainties may be needed to support a variety of stakeholder decisions including those regarding consequences and risks.

NOTE Natural events can be the source of many dangers. [[35], p. 11] lists kinds of natural disasters occurring in the years 1900-2002.

Initial stakeholder requests and stated needs might or might not remain to become part of the agreed-upon set of requirements. Some of what is involved in the evolutionary progression that arrives at the specification is mentioned in the next subclause.

9.2.3 Requirements analysis

Some of the more relevant kinds of analyses are analysis and resolution of conflicts among stakeholders and their needs, risk analysis, feasibility analysis, and trade-off analysis. Feasibility analysis may consider technical, economic, human wellbeing, legal and regulatory, marketing, organizational, social, political, environmentalism-oriented, and mission-oriented feasibility as well as other aspects of feasibility particularly those of concern to key stakeholders or decision makers. All of these can involve costs and benefits, and reflect possible sources of opportunities, difficulties, dangers and uncertainties as well as positive or adverse consequences.

Particularly when adverse consequences or risks are crucial in decisions needed for system success, the existence of an assurance case to ensure and adequately show results that decision makers find acceptable can be decisive for establishing the feasibility of the system and its success. Both issues—producing an adequate system and showing that the system is adequate—are important.

Key feasibility efforts include creating a concept for production, a concept of operations, and a concept for assurance. The concept for assurance needs to demonstrate feasibility in the context of corresponding concepts for the system life cycle. The requirements for the assurance case need to be established identifying properties of interest.

Requirements analysis and design decisions may exacerbate or ease tradeoffs between safety or security, and efficiency, speed, and usability. For example, innovative user interface design may ease security's impact on usability [34]. Business tradeoffs also exist between effort and time-to-market, and safety or security. Producers want users and other developers to use the features of a system but the likelihood of this is reduced if these are shipped "turned off" because of safety, security, or other concerns over adverse consequences.

The decisions regarding the extent of the assurance case and resulting effort have essentially the same bases as decisions about risk management. ISO TR 15443-3 *Information technology — Security techniques — A framework for IT security assurance – Part 3: Analysis of assurance methods* addresses the issue of how to decide on the extent and nature of the appropriate assurance case at substantial length. It covers the goals, dimensions, elements, and structure of such decisions.

Projects with quality, systematic, and in-depth system-related risk management might require only limited additional artefact(s) to record relationships among parts of mainly existing artefacts to develop an assurance case. The question might not be how much risk management is warranted, but doing what is warranted in a more explicit, systematic, reviewable, auditable, and manageable way. Using an assurance case appropriately might very well not cause unwarranted expense and might result in better decisions on where to put effort for greatest risk reduction.

In practice, the processes for requirements and the architectural design process overlap and sometimes overlap with the implementation process. Thus, the actual work and decisions regarding uncertainty, consequences, and assurance case(s) may involve widespread input from and influence on all three. Among these are decisions regarding the specification of top-level assurance case claim(s) and an approach to assurance.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

9.2.4 Architectural design

The requirements for and decisions about the assurance case are drivers of process- and system-related decisions. The rationale for the system's design needs to justify that it will meet assurance case claim(s) and that it is adequately analysable regarding the property(ies) involved. For example, availability of a computer-based service might include its concurrency interactions being simple enough to allow automated analysis for livelock and deadlock. The design needs to be adequately analysable and reviewable, and dynamically testable. Simulation methods, test facilities for models such as wind tunnels and water tanks, and the use of CAD-CAM make dynamic testing of designs increasing powerful and are all sources of evidence for the assurance case.

The design may be created in a way to "guarantee" the claim's relevant aspects using justified assumptions, such as assumptions regarding sub-claims related to its implementation that presumably are later verified. Reviews of every human-readable artefact normally need to be performed. In addition, review and analysis would be expected at the very least to verify that the architecture design process was correctly followed and no known pitfalls or known kinds of weaknesses resulted.

Usually, the assurance case is easier to create and understand if portions of it are based on the design's structure and rationale. Undue inflexibility regarding a claim (or claims) can result in design problems.

Four sorts of assurance issues that confront the designers of systems with properties to be assured are:

- The system might do something that it should not or when it should not thereby allowing, facilitating, contributing to, or causing undesired events, conditions, or consequences to manifest (i.e., error of commission).

- The system might fail to do something that it should do at the time it should, failing to prevent undesired events, conditions, or consequences from manifesting themselves (i.e., error of omission).

- Whether the system is intended to prevent them or not, the system might include capabilities intended to mitigate or minimize undesired consequences (these are subject to points 1 and 2). Such capabilities can affect predictions concerning consequences.

- The system might be such that adequately low uncertainties and/or consequences cannot be achieved or cannot be shown.

Positive measures to eliminate, prevent, avoid, limit exposure to, or tolerate potentially adverse events, conditions, or consequences have all the advantages of prevention over cure.

Designers need to simultaneously consider (1) what the system will do and (2) adequately showing that the design will result in the system doing what it should and having acceptable, or at least tolerable, consequences. Consideration must be given to the feasibility of creating and transitioning the system so that the system as built and its consequences (1) will be and (2) can be shown to be acceptable (or at least tolerable) over its life cycle. Both of the pairs of items (1) and (2) can have uncertainty that should be kept acceptable (or at least tolerable).

NOTE When safety and security are concerns, a few of the possible measures/countermeasures include limiting the paths that sources of danger might use to affect the system (e.g., reduce attack surface to reduce what information and services an attacker has access to), limiting the portion of the system to be trusted and related dependencies on its environment, limiting the opportunities for and incidence of vulnerabilities and weakness in relevant elements (e.g., reducing the size and complexity of elements, static analysis, peer reviews, and/or random testing), avoiding states or preconditions allowing, facilitating, causing, or contributing to violations of claims or adverse consequences (e.g., keeping temperatures within allowable limits), and limiting the potential consequences of a violation (e.g., through isolation, limited privilege, mutually suspicious components, damage confinement, quick recovery, or an active countermeasure that detects and counters an attack's effects).

At the completion of system design, the design of the assurance case and its planned arguments and evidence should be equally complete encompassing all that is known about the system including the plans for subsequent life cycle processes (e.g., implementation, integration, verification and validation, transition, and maintenance). This knowledge should include sufficient plan details to establish constraints on future decisions needed for the assurance case's feasibility.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

9.2.5 Implementation

Implementation activities and artefacts occur at multiple levels of abstraction of the system. Each level of abstraction needs to be shown to be consistent with the assurance case claim by showing its agreement with the next higher level artefact. Such arguments across levels of abstraction are common and generally necessary. Regardless of the area—computer hardware or software, communications, structures, flows, materials, electromagnetism, medicine, transportation—the implementation of such arguments means the implementation of the design needs to be adequately analysable, reviewable, and appropriately dynamically testable, including intermediate descriptions, rationales, and artefacts.

Automatic generation from specifications or designs of software, instructions for manufacturing (e.g., machine tools), or other implementation artefacts including most tools or aids become a source of uncertainty and become concerns of the assurance case. Any property or aspect affecting the suitability and trustworthiness of these artefacts (e.g., ease of integration and use, throughput, correctness and reliability, accuracy, security, support, and availability) can become an issue.

9.2.6 Integration

The allocation of effort and care in integration and verifying the integration of system elements to become the system are usually risk-based and should reflect the needs of the assurance case. From the perspective of an assurance case, the more serious uncertainties or risks are the ones that could most affect the achievement of the top-level claim and the showing of that achievement.

9.2.7 Verification and validation

Clearly verification and validation have a strong interconnection with the assurance case and the evidence it requires. A key driver of the verification and validation efforts is the assurance case’s specific needs for evidence. These needs depend on what properties and aspects the assurance case’s top-level claim encompasses and the importance or criticality of sub-claims to the argument. Some examples of tasks for verifying or validating sub-claims might be:

- Provide evidence showing that the top-level claim agrees with needs, requirements, or property-oriented policy.
- Develop verification and test plans for the property of interest.
- Develop or acquire the system ensuring the top-level claim and its related properties such as to:
  - Facilitate the creation and structuring of arguments and sub-claims within the assurance case and aid in identifying appropriate and sensible assumptions.
  - Provide needed evidence (adequately covering behaviours, conditions, sources of uncertainties and risks) that the assurance case's arguments and sub-claims require.
- Assure that the design provides for meeting top-level claim and supports the creation of arguments and successful analysis, the conducting of argumentation, and the creation or collection of evidence.
- Assure that the system contains only items called for in the design.
- Assure that implementation is consistent with the design and supports creation of arguments and successful analysis, conduct of argumentation, and creation and/or collection of evidence.
- Assure that the system is free of critical weaknesses or vulnerabilities and corresponds to the design and claims.
- Perform testing of the property and property-related functionality.
- Provide an analysis of insidious possibilities (e.g., rare but catastrophic events, covert channels).

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

- Perform ongoing monitoring of opportunities and dangers, needs, and the environment to verify the continued validity of evidence and assumptions.
- Verify changes performed to maintain conformance to a possibly revised top-level claim while continuing to agree with and show support for the complete assurance case (as revised if needed).
- Verify that the transition process conforms to the requirements of the assurance case.
- Verify that operation, use, disposal, etc. are being performed in a manner that conforms to the requirements of the assurance case.

9.3 Transition, Operation, Maintenance and Disposal

Post-development processes or concerns include training, deployment, monitoring, maintenance, transfers of control (legitimate and illegitimate), operation and use, retirement, disposal and/or other activities involving the system or its environment that are necessary to ensure the system’s assurance case claims after its development including during the duration claimed for its top-level claim.

The processes of operation and maintenance are often the main topic of the assurance case's top-level claim. The transition process includes adequate initial training; however, the following list includes a few transition-related situations where meeting of the requirements imposed by the assurance case can require tailored activities or procedures:

a) Obtaining needed changes in infrastructure or interfaces in the environment particularly if system operation begins before some of them are ready.

b) Beginning operation with a beta version, initial or partial operational capabilities, or with only a portion of the system and switching over to full system operation.

c) Start-up of backup and initial parallel operation with these backups.

d) Parallel operation with replaced system and cross verification.

e) Possible transition to and use of fallback modes, possibly including fallback to the replaced system and including concern for differing dependencies on system’s environment or arrangements for backup.

During the duration of applicability of the top-level claim, the assurance case needs to be maintained along with the system. This need can occur earlier than the start of applicability if the assurance case is completed or approved earlier or if the completed or approved assurance case or corresponding system is inadequate or out of date, e.g., if the assurance case or its evidence are for older versions of the system or if they make assumptions about a future version or the environment that do not become true.

The monitoring and collecting of relevant field data that could strengthen or weaken the assurance case is essential. As many techniques exist for collecting and recording data (e.g. [71]), the crucial decision is what data to collect, record, and analyse. Clearly, any testing or field data used as important evidence in the assurance case needs to be monitored, including data that could show if claims in the assurance case were violated or had near misses. As in field collection of data for any purpose, the usual concerns of expense, resources, automation of collection, and data validity exist.

Transfers of control of the system and the activities that follow can cause problems by negating required assurance case conditions or assumptions. Such transfers of control can be legitimate, such as sale or lease, transfer to a maintenance facility or storage, a change in governance, or seizure by law enforcement or confiscation by a government. Transfers can be illegitimate, such as theft or capture of the system, or takeover of a facility by strikers or activists.

Disposal is not a process that is necessarily beyond the needed duration of applicability. Examples of concerns are safe disposal of hazardous materials or ordinance and the need to remove or destroy sensitive data before the retiring or disposal of the media on which copies of it reside.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

9.4 Organization processes

9.4.1 Introduction

Organizational processes provide governance over projects, supply their environment and enable and support them, including human resource activities, and aid in a project's relationships with outside entities such as in marketing, acquisition, and supply. ISO/IEC 15026 is suitable for use as part of an acquisition or supply agreement process.

9.4.2 Project-enabling processes

The project-enabling processes depicted in the organization part of the diagram in Figure 8 can have substantial affects on assurance, including the assurance case and related activities and artefacts.

An organization can affect a project through its policies and procedures regarding personnel competence, motivation, trustworthiness, communication skills, ability to relate inside and outside the organization, location, recruitment, assignment and retention, compensation, and training, and by its human resources function and the governance, structure and culture of the organization.

Personnel affect the assurance case, and the assurance can place requirements for acceptable personnel. As an example, sub-claims can be made about the behaviour of operational personnel with these generating requirements for training.

The organization's control of the life cycle model and the available infrastructure can have other major effects on the project. Developmental and operational practices and infrastructure can affect what the assurance case can show and on meeting the requirements imposed by the assurance case.

9.4.3 Agreement processes

The most straightforward and comprehensive treatment of the assurance case in the acquisition and supply processes might appear to be the requiring by the acquirer and supplying by the supplier of a full and fully relevant assurance case. However, this requirement sometimes is not feasible or acceptable to one or both parties. Acceptable agreements can be reached without disclosing business secrets or proprietary material that still provide the acquirer with the information needed for decision making.

All parts of ISO/IEC 15026 can be utilized in acquisition. In preparation for acquisition, an approach to assurance or an assurance strategy should appear in the feasibility study and be further elaborated to accompany any operational concept document. A request for proposal (RFP) can contain requirements for an assurance case including requiring conformance to parts of ISO/IEC 15026 and possibly other domain- or location-dependent standards. These other standards and related guidance documents exist for several sectors or purposes and particularly for government agencies. Examples are [147], [150], [151], [155], [156], [157], [165], [166], [182], [183], [197], [198], and [199].

The RFP could provide information, requirements, and guidance regarding what top-level claim (or claims) is required, including the properties and limitations on their values, durations, conditions, limitations on uncertainties and consequences. Establishing an agreement on the (e.g. acceptable, tolerable, or allowable) limitations on uncertainties in the top-level claim is a vital step. A description of the proposed assurance case should appear in the RFP to allow the acquirer to evaluate the supplier on this point. Conformance to non-assurance-case standards can be required and used as a whole or by its parts as evidence in the assurance case.

Note Legal proceedings have distinctions among burdens of proof – that is, the required degree of uncertainty. For example, in the U.S. levels exist such as preponderance of evidence, clear and convincing evidence, and beyond a reasonable doubt. Common language has these and many others terms regarding uncertainty. While terms such as inconceivable or wildly imaginative are not useful; and an example natural language set ranging from impossible to certain is: impossible, possibly possible, just possible, forlorn hope, surprising, unlikely, doubtful (on one point, a few points, many points), plausible, credible, even-odds, preponderance of evidence, probable, convincing evidence, highly likely, beyond a reasonable doubt, almost certainly certain, certain.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

To form an agreement concerning uncertainty the parties need a shared conceptual understanding of its meaning in the situation ([204] p. 29-30). They must address the possibilities of having uncertainties about the issues of meaning and meaningfulness and the factors that underlie them. Evidence-related issues include accuracy and the associated uncertainty, generalizability, inferences, and relevance to a particular property or argument. As [204] lists more specifically this includes the need to recognize that they can have uncertainties (and therefore possibly disagreements) concerning measurement, sampling, mathematical modelling, cause and effect, inferences, and categorization. Agreeing beforehand on standards of quality for the assurance case including its evidence and incrementally agreeing or approving the assurance case designs, plans, development and maintenance can increase the ease of dealing with uncertainties. However, affordable provisions may need to be made for a resolution process when disagreements persist on substance or on which party provides funding.

EXAMPLE To address possible uncertainty-related disagreements or doubts, the parties might agree upon third parties for mandatory expert consultation. Third parties might also arbitrate such decisions as what will be treated as correct, which party is correct, and is an objection from one party reasonable or unreasonable. Arbitration involves relying on established legal standards for burdens of proof with possibly differing standards for different kinds of issues.

Acquirers need to monitor and evaluate the assurance case progress and risks regarding it during the period of supplier performance and when accepting the system. This evaluation can be more informative and reassuring than acceptance testing, which should always be done in any case and which might supply additional evidence for the next version of the assurance case. The assurance case can also be useful in obtaining required certifications and accreditations.

The obligations and responsibilities regarding maintenance of the assurance case and the remainder of the system should be clear (and enforceable) in the agreement between the parties as well as obligations and responsibilities related to other relevant life cycle processes.

10 Summary

Two basic areas essential to the best possible use of ISO/IEC 15026 have been addressed in some detail in this part of ISO/IEC 15026. First, users need to have an adequate understanding of the concepts and terminology used in ISO/IEC 15026 that previously may not have been shared across the communities served. ISO/IEC 15026 uses concepts and terminology designed to be readily understood by all its users.

Second, this part of ISO/IEC 15026 provides a basis for easier understanding and use of ISO/IEC 15026 as well as for understanding of the rationales behind the international standard itself. This part of ISO/IEC 15026 can aid in learning, instruction and training, discussion, finding relevant references, and gaining intellectual mastery of the issues relevant to system assurance. It also is potentially useful in the development or revision of related standards and guides that intend to be consistent with or elaborate on ISO/IEC 15026.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Annex A
(informative)

Frequently asked questions

A.1 A dozen frequently asked questions and their answers

These are questions that have been frequently asked. They are repeated here with answers and internal references to related clauses, subclauses, and annexes:

1) ISO/IEC 15026-2, ISO/IEC 15026-3 and ISO/IEC 15026-4 were not easy to read the first time. Why is this? Answer: The documents are designed to ease serious, repeated use. See 6.3.

2) Some things that "should" be done do not fit my situation. Why is this? Answer: One needs a clear understanding of the usage of the term "should" that requires justification for not following the guidance. Your situation is actually acceptable. See Clause 2, Terms and definitions, and 6.2.

3) What is an assurance case? Answer: An assurance case makes a claim regarding some property and provides arguments, evidence, and where appropriate assumptions to support it establishing a conclusion regarding it and the conclusions associated uncertainty. It can be used for any property. See clause 6.5.

4) Why is uncertainty not always expressed as a probability? Answer: Probability values may be extremely difficult or impossible to establish where an adversary deliberately goes against the probability estimates one makes. See 7.3.3.2.

5) ISO/IEC 15026-2, ISO/IEC 15026-3 and ISO/IEC 15026-4 mention maliciousness in several places. As I am not concerned with security properties, why do I need to concern myself with maliciousness? Answer: Malicious actions can affect almost any property. See 6.3.

6) I need to create an assurance case related to an existing system. Clearly, I cannot integrate assurance case development with system development. What do ISO/IEC 15026-2, ISO/IEC 15026-3 and ISO/IEC 15026-4 say about this? Answer: ISO/IEC 15026-2 supports assurance case separately from developing the system. ISO/IEC 15026-2 and ISO/IEC 15026-4 can help with the issue of integrating the two during the duration of applicability of the top-level claim. See 7.5 and 9.3.

7) How do the parts of ISO/IEC 15026 relate to each other? Answer: See 6.4.

8) I am about to do an acquisition; what does ISO/IEC 15026 have to help me with this? Answer: ISO/IEC 15026-2, ISO/IEC 15026-3 and ISO/IEC 15026-4 can be used in acquisitions. See 11.4.3 for a limited discussion of this.

9) How does ISO/IEC 15026 relate to the other standards I must or already use? Can I use them together? Answer: ISO/IEC 15026 is consistent with several standards and often still usable with others. See Annex C and 6.4.

10) How can I approach the construction of a supporting argument? Answer: Many approaches are possible. A key issue is the meaningfulness of the argument. See 7.2.7.3.5.

11) Why does ISO/IEC 15026-1 have a substantial annex on security and cover no other property in depth? Answer: For many initial users of ISO/IEC 15026, security has previously not been important in their area of concern until recently and they have only limited familiarity with it and its many aspects. See Annex E.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Annex B
(informative)

Difficulties with terms and concepts

B.1 Introduction

Terms, concepts, and principles that relate to ISO/IEC 15026 span multiple disciplines, activities, roles and technologies. These terms and concepts have a long and varied history. Over time, many terms have come to be used within and across specialist communities such as systems, software, reliability, safety, maintainability, information security, software security, human factors and others well as in different application domains – but often in differing ways.

Debates occur about which meaning is the “right” one among competing dictionaries and ontologies. This is true both in general and among ISO publications. This situation is compounded by some terms having further variations within popular usage or in yet other professions. For example, the professions of psychology, law, and mathematics use terms that are important within ISO/IEC 15026.

Without proper awareness and care, these differences in the use of terms among specialities and communities can cause confusion that hinders productive communication among the users of ISO/IEC 15026. To avoid this confusion, this part of ISO/IEC 15026 states clear, unambiguous terminology even though this may require using two or three words, or even a phrase, where a particular specialty or community may currently use one word. This part of ISO/IEC 15026 also tries to convey a clear understanding of the underlying concepts including subtleties such as what is true in a certain situation, what is known, and how well it is known. Readers also are aided in discerning the assumptions implicit in various concepts and statements.

This part of ISO/IEC 15026 covers many concepts not only useful to users of ISO/IEC 15026 but also useful to the preparers of conformant or compatible standards or other governing documents. These concepts and principles, while not always relevant everywhere, span multiple disciplines, activities, roles, properties, application domains, and technologies. This part of ISO/IEC 15026 particularly emphasizes concepts needed for understanding of software and systems assurance and central to the preparation of, use of, and conformance to ISO/IEC 15026-2, ISO/IEC 15026-3 and ISO/IEC 15026-4.

B.2 Why variation in the use of terms occurs

This subclause covers why some of these variations in terms and concepts discussed in B.1. First, the effective meanings of terms tend to change depending on which of the following they are used in regard to:

- What is needed.
- What is specified.
- What happens or is done.
- What is actually true (e.g. about the software, system, or environment).
- What has been measured, observed, inferred, etc.
- The uncertainty in these measurements, estimates, conclusions, etc.
- The degree of confidence one has.
- The related decisions one makes.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

The fourth, “What is actually true,” may be referred to but is seldom known exactly. Rather, what is known is, “What has been measured, observed, inferred, etc.” with their uncertainties.

Other reasons for variations also exist including (1) what is conceivable, possible, or feasible; (2) what are the entities, events, behaviours, or conditions; and (3) how these allow, facilitate, contribute to, cause, or affect contextual issues.

NOTE 1 For example, outside this and consistent standards, the usage of "assurance" varies as different speakers (or writers) use it to refer to an entity, capability, condition, event, consequence (computing or real-world), physical or mental state, action, activity, or process.

NOTE 2 In ISO/IEC 15026 "uncertainty" is used in a general way to mean lack of certainty. This usage allows the term “uncertainty” to be applied to anything. Different communities restrict the application of this term to limited usage, for example to predictions of future events, to physical measurements already made, or to unknowns. While this usage may be convenient within some communities, ISO/IEC 15026 crosses many communities.

Finally, the effective meaning of a statement (e.g. a specification) can also depend on the situational assumptions underlying it. Thus, the meaning of a statement often derives, at least in part, from the scope to which it is being applied.

B.3 Conclusion

Because of the lack of a common treatment of shared concerns, existing standards addressing different application areas and different topics or properties use different terminology and concepts. This phenomenon increases the costs to users who must concern themselves with more than a single important property. For example, to conform to International Standards on both safety and security, users may find that they need somewhat redundant risk assessment and management processes. On the other hand, ISO/IEC 15026 reflects and benefits from these more specialized standards and has as a purpose to provide a common underlying framework, terminology, concepts, and set of requirements related to assuring properties and to provide a basis for future standards or revisions of standards treating specific properties to benefit from, apply, specialize, and extend on what is presented in ISO/IEC 15026.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Annex C
(informative)

ISO/IEC 15026 relationships to standards

C.1 Relationships

Because ISO/IEC 15026 is intended for application in a variety of contexts, the user may confront differences in terminology and concepts. It is not appropriate for this standard to introduce a broad variety of terms because they may conflict with terminology specific to the context of application. The essential concept introduced by ISO/IEC 15026 is the statement of *claims* in an *assurance case* and the support of those claims. Other terms are less important; nevertheless, it is important that the readers be provided with an understanding of those terms. Therefore, ISO/IEC 15026 uses the terminology and concepts consistent with ISO/IEC 12207:2007, ISO/IEC 15288:2007, and ISO/IEC 15289:2006.

ISO/IEC 15026 does not presume that it is applied in conjunction with ISO/IEC 12207:2007, ISO/IEC 15288:2007, or ISO/IEC 15289:2006. Those who have an alternative basis for their life cycle processes may also use this standard.

ISO/IEC 15026-1 provides background and information that could be useful in understanding and using ISO/IEC 15026-2.

ISO/IEC 15026-4 provides tasks related to the assurance case that must be performed integrated within life cycle processes particularly for concurrent development and maintenance of the system and its assurance case. It provides general requirements and these requirements instantiated in the contexts of ISO/IEC 12207:2008 and ISO/IEC 15288:2008, the International Standards for software and system life cycle processes. Conformance to ISO/IEC 15026-2 and ISO/IEC 15026-4 is separate.

ISO/IEC 15026-4 provides a *process view* for systems and software assurance. It provides a statement of purpose and a set of outcomes and requirements suitable for systems and software assurance.

The concept of a process view was formulated and described in an annex of ISO/IEC 15288:2008. 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 explaining how the outcomes can be achieved by employing the activities and tasks of the various processes in ISO/IEC 12207 and ISO/IEC 15288. ISO/IEC 15026-4 was developed upon the basis of the Specialty Engineering Process View provided as an example in ISO/IEC 15288. It, however, does not presume and does not imply any engineering speciality in assurance and is complete and is self-contained, not necessitating compliance with ISO/IEC 12207 or ISO/IEC 15288. The provisions regarding the assurance case and assurance planning are intended to be compatible with the provisions of ISO/IEC 15289:2006 for information items resulting from life cycle processes. However, the relevant provisions of ISO/IEC 15026 are self-contained; it is not necessary to also show conformance to ISO/IEC 15289.

NOTE This standard benefits from much prior work both outside and inside ISO and IEC and within many fields. Much pioneering work has been done in the safety community. In one example elsewhere, the security community in ISO/IEC JTC 1/SC 27 has been working on the topic of assurance for many years concentrating on systems and software, while security is only one of many areas covered by ISO/IEC 15026, it benefits from SC 27's work. While security is only one of many areas where ISO/IEC 15026-2 can be applied and where mention might be made of standards, two security-related examples to mention are ISO/IEC TR 15443, Information technology—Security techniques—A framework for IT security assurance that discusses the need for arguments and evidence in the context of information technology security [multiple parts], provides a security focus that is not limited to information technology and ISO/IEC 15408:2005, Information technology—Security techniques—Evaluation criteria for IT security [multiple parts], provides a particular form of an assurance case specialized to a specific form of claim. However, no dependency of ISO/IEC 15026 exists upon the use of any of them.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Users of ISO/IEC 15026 may require risk assessment and risk management and measurement processes that are more fully elaborated than the treatments provided in ISO/IEC 12207 and ISO/IEC 15288. Two International Standards, ISO/IEC 16085 and ISO/IEC 15939, are useful in this regard. However, users interested in assurance of some specific properties may decide to apply risk assessment and management and measurement standards that are specifically applicable to the relevant properties and systems.

The provisions of ISO/IEC 15026 are generally consistent with those of the ISO/IEC 25000 series of standards related to system quality, and aim to be generally consistent with the ISO/IEC 27000 series of standards related to information security management systems, the IEC 61508 multi-part standard on functional safety, and various standards of IEC TC 56 related to dependability. However, except as specifically cited, there is no dependency of ISO/IEC 15026 upon the use of those standards.

Many international – as well as industry and national – standards exist addressing the concerns of safety, security, reliability, maintainability, dependability, human factors, and other important topics, but, to date, no common treatment exists of the shared aspects of these concerns. ISO/IEC 15026, including ISO/IEC 15026-2, addresses assurance in a common manner. The motivation is to provide a unified view spanning these many areas across the life cycle. This top-level standard may be applied in conjunction with other standards that address the specific concerns of the properties that are of interest. These other standards, depending upon their source and other factors, may not be completely harmonized with ISO/IEC 15026, so their application may require that the user resolve perceived inconsistencies.

NOTE A discussion of the assurance case and three existing standards not explicitly calling for an assurance case appears in [10].

Several potentially relevant standards are listed in the Bibliography.

NOTE Industry and agency standards and guides explicitly about assurance cases include [147], [150], [151], [155], [156], [157], [165], [166], [182], [183], [197], [198], and [199].

C.2 More on relationships to life cycle process standards

Figure C-1 depicts the relationship of the several standards related to the life cycle processes. At the bottom of the diagram is a foundation of a number of standards that provide common vocabulary, architecture for processes, and a convention for describing those processes. The other depicted standards are built upon this foundation. ISO/IEC 15288 and ISO/IEC 12207 provide life cycle processes for systems and software respectively. They are intended to be interoperable, hence useful for systems with varying content. The two life cycle process standards are supported by four standards that provide additional requirements and guidance on shared issues: ISO/IEC 15289 for documentation resulting from the execution of life cycle processes; ISO/IEC 15939 for the measurement process; ISO/IEC 16085 for the risk management process; and ISO/IEC 16326 for the project management process. In addition there are other standards that provide additional requirements and guidance for selected processes. ISO/IEC 24748 is a guide describing how life cycle processes are organized to manage the entire life cycle of a system or software. ISO/IEC 15026 series, ISO/IEC 15026 is intended to be compatible with these other 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.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

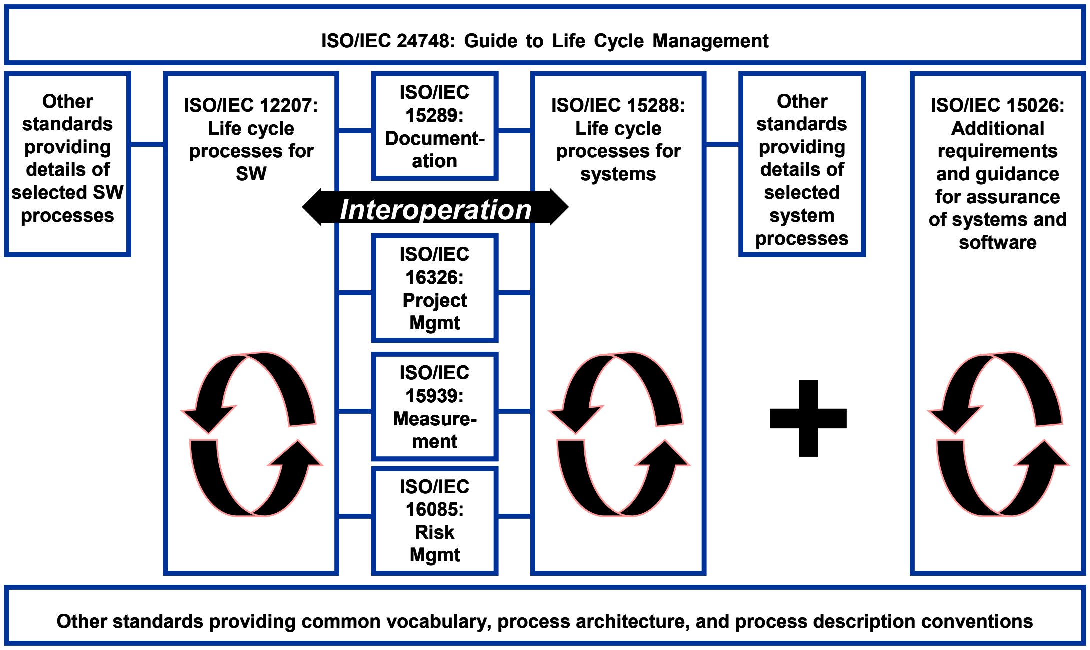

ISO/IEC 24748: Guide to Life Cycle Management
Other standards providing details of selected SW processes
ISO/IEC 12207: Life cycle processes for SW
ISO/IEC 15289: Document-ation
Interoperation
ISO/IEC 16326: Project Mgmt
ISO/IEC 15939: Measure-ment
ISO/IEC 16085: Risk Mgmt
ISO/IEC 15288: Life cycle processes for systems
Other standards providing details of selected system processes
+
ISO/IEC 15026: Additional requirements and guidance for assurance of systems and software
Other standards providing common vocabulary, process architecture, and process description conventions

Figure C-1 — Some relationships among standards

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Annex D
(informative)

Phenomena

D.1 Introduction

ISO/IEC 15026 can be used for positive issues and consequences. For example, some users are interested in readiness, performance, or gain, but others are interested dangers, risks, or losses. To aid them, this annex contains several tables and lists with related sources and kinds of these as well as a list of online sources for information about dangers and lastly lists of the basic types of force and matter.

D.2 Sources and kinds

Phenomena can be associated with fundamental phenomena, such as fundamental forces and states of matter, or in as relation to pragmatic lists of phenomena often compiled from prior experience. This subclause offers tables with lists of both kinds.

D.2.1 Kinds and sources of phenomena and the locations of them and their consequences

Phenomena can have many causes, points of origin, methods of occurrence, kinds and locations, and consequences. Table D-1 is not exhaustive but useful and may be used in combination with standards and guidelines for domain areas.

Table D-1 — Some kinds and sources of phenomena

Altitude (e.g. aircraft, mountain)	Kinetic (e.g. movement, (de)acceleration, explosion, vibration, sound)
Backup and recovery	Logistics, provisioning, and sustainment
Biological (e.g. agricultural (e.g. famine), medical (e.g. health care, epidemic))	Maintenance (e.g. preventive, repair, lack or incorrect)
Boundary crossing (osmosis, intrusion)	Materials (e.g. structural, semiconducting, hazardous)
Capacity limitations (e.g. bandwidth, mental overload)	Nuclear and radiological
Chemical	Observation and perception (range, acuity, surveillance)
Cyber (e.g. computer phenomena or cyber attack)	Physical (physics)
Earthquake	Physical degradation (e.g. ware)
Electrical (e.g. power supply, lighting)	Precipitation (e.g. hail, ice storm, snow, deluge, draught)
Electromagnetic (e.g. electromagnetic pulse)	Readiness (e.g. high state of, inadequate)
Emergency	Sensitivity and tolerance
Equipment (e.g. size, weight, function, safety)	Slide (e.g. landslide, avalanche)
Environment (e.g. systems, work, natural)	
Extraterrestrial sources (e.g. solar flares, meteors)	Strength (e.g. weakness)
Fire (e.g. equipment, structure, wild fire)	Stress
Flexibility (e.g. inflexibility)	Submerge (e.g. submarine, dip, bathe)
Flight (e.g. aircraft, parachute)	Surprise (e.g. lack of anticipation or warning, shock)

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Float (e.g. watercraft, ship, person)	Temperature (e.g. operating range, extreme)
Flood including water damage	Testing (e.g. reliability, load, security, unrealistic testing)
Global warming or cooling	Training and practice
Health (e.g. fitness, injury, epidemic)	Use (e.g. amount, ease, wear)
Human behaviour (performance, error, (in)competence, speed, efficiency/productivity)	Volcano
Information (e.g. amount, accuracy, timeliness, lacking, wrong, corrupt)(See ISO/IEC 25012)	Wave/surge
Insect or rodent (e.g. pollination, infestation)	Wind
Social causes and locations	
Activism (e.g. political, social, religious)	Recreation (e.g. prank)
Civil unrest	Revolution
Crime	Subversion
Diplomacy (e.g. trade agreements)	Terrorism
Espionage	Vandalism
Industrial competitiveness	War
Legal constraint, law enforcement and judiciary	
Possible locations for phenomena occurrence and gain or damage	
Business and trade	Raw material supply
Cyber	Space (e.g. disasters)
Environmental	Persons (physical, psychological, and relationships)
Industrial (e.g. accidents)	Information
	Internal to system
Infrastructure	
Agriculture and food	Search and rescue
Communications	Energy (e.g. electricity, petroleum)
Emergency and disaster response and recovery	External affairs (e.g. national)
Decontamination and cleanup services	Financial services
Emergency health care	Fire fighting
Emergency hazardous materials response	Information technology and services
Emergency management	Internal governance
Emergency and evacuation transportation	Housing and shelter
Emergency water and food supply	Natural resources
Insurance payouts	Hazardous materials services and response
Long-term community recovery and mitigation	Public health and medical services
Mass care, clothing, housing, and human services	Public safety and security
Mortuary services	Public works and engineering
Restoration of electric power	
Transportation	Water supply

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

D.3 Information about dangers and weaknesses

Current information on vulnerabilities, weaknesses (which may be vulnerabilities), and the exploits that target them can be found in a number of sources, including books (in which the information may be better organized as an introduction to the subject, but will be less current), articles, vendors' and independent "alert" services, and databases. For examples, see the following:

- ACM Committee on Computers and Public Policy, The Risks Digest: Forum on risks to the public in computers and related systems, - http://catless.ncl.ac.uk/risks.
- Canada: Transportation Safety Board - www.tsb.gc.ca.
- Common Attack Pattern Enumeration and Classification (CAPEC) http://capec.mitre.org.
- France : Bureau d'Enquêtes et d'Analyses pour la sécurité de l'Aviation Civile - http://www.bea-fr.org/.
- Germany: Bundesstelle für Flugunfalluntersuchung - www.bfu-web.de/.
- Internet Security Systems (ISS) X-Force Database - http://xforce.iss.net/xforce/search.php.
- MITRE Corporation Common Weaknesses Enumeration - http://cwe.mitre.org/.
- MITRE Corporation dictionary of Common Vulnerabilities and Exposures - http://www.cve.mitre.org/.
- NIST National Vulnerability Database - http://nvd.nist.gov/.
- Open Source Vulnerability Database - http://www.osvdb.org/.
- OWASP Top Ten - http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.
- Purdue University Center for Education and Research in Information Assurance and Security (CERIAS) Cooperative Vulnerability Database - https://cirdb.cerias.purdue.edu/coopvdb/public/.
- SANS Top Twenty - http://www.sans.org/top20/.
- Secunia Vulnerability and Virus Information - http://secunia.com/.
- SecurityFocus Vulnerabilities - http://www.securityfocus.com/vulnerabilities.
- UK: Air Accidents Investigation Branch, - www.aaib.dft.gov.uk/.
- US DoD's Joint Task Force-Global Network Operations (JTF-GNO) Information Assurance Vulnerability Alert (IAVA) program - http://www.cert.mil/.
- US: National Transportation Safety Board - www.ntsb.gov/.
- US-CERT Vulnerability Notes Database - http://www.kb.cert.org/vuls/.

The ACM's Risks Digest has over twenty years of information on individual incidents and offers a search capability. For software developers the most useful may be the Common Weaknesses Enumeration (CWE) [31] although the Common Attack Pattern Enumeration and Classification (CAPEC) is a useful companion, and books such as [43], [57], and [208] are available.

What is publicly known, however, may be less than what is known to producers or researchers. Potential attackers may know exploits no one else knows. Even after shipment some software vendors make significant efforts to discover vulnerabilities through internal efforts, and the results are often not publicized.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

D.4 Fundamental forces and states of matter

While Table D-1 lists many phenomena, Tables D-2 and D-3 reflect fundamental starting points for analysis useful in novel situations.

Table D-2 — Fundamental forces

Fundamental forces	Range of force
Strong interaction	10^{-15} meters
Electromagnetic	Infinite
Weak force	10^{-18} meters
Gravitational	Infinite

Table D-3 — States of matter

States of matter	Changes	
Solid	Solid to liquid Melting or fusion	Liquid to solid Freezing
Liquid	Liquid to gas Vaporization, boiling, evaporation	Gas to liquid Condensation
Gas	Gas to solid Solidification	Solid to gas Sublimation
Ionized Plasma		
Quark-gluon plasma		
Bose-Einstein condensate		
Fermionic condensate		
Other		

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Annex E
(informative)

Security

E.1 Introduction

NOTE Material in this Annex is excerpted or adapted from [175] with permission of the editor.

ISO/IEC 25010 and ISO/IEC 15026:1998 define security as, "The protection of system items from accidental or malicious access, use, modification, destruction, or disclosure."

ISO/IEC 25010 associates the following qualities with security: confidentiality, integrity, non-repudiation, accountability, authenticity, and security compliance (immunity, which represents the degree to which the software product is resistant to attack, is covered by integrity), plus related survivability and safety. Another definition of security is, "All aspects related to defining, achieving, and maintaining confidentiality, integrity, availability, accountability, authenticity, and reliability" (ISO/IEC13335-1). In addition, security-related usability can be important to facilitate ease of system operation and use and to avoid unacceptable user inconvenience with this possibly resulting in users deliberately avoiding or bypassing security features.

"Security [of information] often requires the simultaneous existence of 1) availability for authorized actions only, 2) confidentiality, and 3) integrity with no improper meaning 'unauthorized' [changes]."[5] [[14], p. 13] Security is not as simple as this last seems. Neither confidentiality nor integrity can generally be achieved without entities being adequately identified (identity known or established to some level of uncertainty usually desired to be firmly established including being verified – authenticated). In addition, for the identified entity, only actions permitted to it are allowed or possible – usually through use of access control mechanisms, separation, or encryption.

As with many other properties, security properties are not concerns confined to the scope of computing and information resources. For example, real-world lawbreakers and consequences are also certainly relevant as are physical, personnel, operational, and communications security.

When attacks occur, the system may also be required to detect those attacks and alert users, continue service, confine damage, rapidly recover, and aid in diagnosis and repair to prevent future problems.

Sometimes properties including security properties are properties of the whole system. This means that these properties are determinable or observable only[6] in the context of how the multiple elements of the system interact among themselves, and how the system responds to stimuli from external entities (e.g., input from users). Thus, these properties are said to emerge or be "emergent" with system composition.

While necessary for safety or security in the majority of systems, the mere presence of safety or security functionality does not make a system safe or secure. To provide security, individual pieces of security functionality must be impossible to bypass, corrupt, or cause to fail. Given accurate facts about its environment, these inabilities to corrupt, cause to fail, and bypass can emerge from the inherent properties of the system possibly by use of separation and isolation within the design. Software may attempt to achieve this, but dependencies including those on hardware and other system elements meaning these inabilities must ultimately (also) be achieved at the system (or higher) level.

[5] Another definition of security is, "All aspects related to defining, achieving, and maintaining confidentiality, integrity, availability, accountability, authenticity, and reliability." (ISO/IEC 13335-1).

[6] This does not mean that (1) similar or analogous properties may not exist at lower levels or that (2) one might not design a system so a guarantee can be derived from the behaviours of only a portion of a system and a lack of opportunities (to violate or help violate security) for the remainder.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Since for some properties such as security ones, the system is preserving what might be considered systems level properties and may protect many kinds of stakeholder interests and computing resources such as hardware data, software, and running processes; and does so in a systems context; the distinction between a property being software- or system-level matters little. Thus, this annex generally avoids trying to label topics system versus software; rather the all-encompassing term system is used spanning systems, software, services, and possibly other systems as well as their elements or constituents. On the other hand, system-subsystem relationships and differences in levels of abstraction are important and noted where required.

NOTE While exhibiting reliability, safety, and maintainability may not directly result in a security property, the last can contribute to keeping security "up to date." All may make it easier to show that system is secure.[7]

E.2 Kinds of security

Lists of kinds of security or security-related areas can like direct attention to potential areas of concern or motivation for security. On a wide scope, some speck of kinds of security using a variety of terms. These include:

- Operational security.
- Transportation security.
- Financial security.
- Personal security.
- Infrastructure security.
- Environment security.

Emphasizing computing and communications, ISO 7498-2 lists several kinds of security-related areas:

- *Administrative security*, e.g. controlling the importation of software, procedures for investigating security breaches, audit trail analysis.
- *Communications security safeguards*, e.g. authentication, access control, data confidentiality, data integrity, non-repudiation.
- *Computer security safeguards*, e.g. operating system and database system security facilities.
- *Emanations security*, e.g. radio frequency emanation controls (TEMPEST protection).
- *Physical security*, e.g. locks or other physical controls, equipment tamper-proofing.
- *Personnel security*, e.g. employee screening for sensitive posts, security training and awareness.
- *Media security*, e.g. protecting stored data, secure destruction of computer storage media, media scanning for viruses.
- *Life cycle controls*, e.g. trusted system design, implementation, evaluation and certification, programming standards and controls, documentation controls.

Other terms are also used related to computing and communications such as:

- Information and Communication Technology (ICT) security.
- Communications security.

[7] For further information on this topic, see *Security in the Software Lifecycle*, Section 3.1.3, "Other Desirable Properties of Software and Their Impact on Security".

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

- Data security.
- Application security.
- Information security.
- Network security.

Examples of abbreviations in use include COMPSEC, COMSEC, EMSEC, PHYSEC, ICTSEC, INFOSEC, and TRANSEC.

E.3 Security-related properties

E.3.1 Introduction

Beyond what has been covered so far, this section covers additional conceptual material that should be part of the knowledge of everyone involved or interested in security. ISO/IEC 25010 provides definitions for security-related qualities, More in-depth treatments of security properties are available in [20], [175], and [136]; and [14] contains information on characterization and categorization.

NOTE For example, the history of rigorous professional attention to the theory of systems reliability and availability goes back to the 1930's and before, and serious attention to computer and software security goes back at least into the 1960's with a substantial amount of important work occurred in the 1970's and early 1980's. Luckily, a project at the University of California Davis collected many of these seminal computer security works and placed them on the Internet [185]. While their contents may be reflected in later publications, these works are still relevant today – for example, [184], [204] and [7].

E.3.2 Confidentiality

Computing-related confidentiality topics include access control, encryption, hiding, intellectual property rights protection, traffic analysis, covert channels, inference, and anonymity. The last four are discussed here.

E.3.2.1 Traffic analysis

The levels, sources, and destinations of communications traffic can sometimes be revealing even if the content is encrypted. For example, traffic increases in organizations tend to foreshadow major events. The main issues in traffic analysis are ease of detection and analysability. Factors include concealment of origin and destination of communications and the levelling or randomization of traffic volumes and message sizes.

E.3.3 Covert channels

Covert channels are “abnormal” means of communication using such means as timing of overt messages, locations in messages not normally used (e.g. unused bits in packet headers), or (un)availability of resources to convey messages. These may be ignored in low or moderate security situations. While covert channels based on resources can potentially be eliminated, the objectives in high-security systems are usually to identify and minimize covert channels of all kinds. Covert communication channels are measured by the bit rate that they can carry. See [[20], p. 462-469], [154], and [[160], Chapter 8].

E.3.4 Data aggregation inference

Potential can exist to violate confidentiality or privacy by aggregating data whose individual disclosure would not result in harm. Identity theft is often facilitated by the attacker aggregating data.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

E.3.5 Inference

Confidential data may be inferable from other data that is available. One example is inferring individual data by comparing data for different groups – an individual's grade in a course can be calculated from the average grade in the course and the average grade of everyone but the individual.

E.3.6 Anonymity

Anonymity can involve concealing one's identity, activities, attributes, relationships, and possibly existence. Issues include concealing the identity associated with particular data and who is communicating with whom including determining that the same (but unidentified) entity is involved in two communications – linkage. Desired or required privacy[8] is one motivation for anonymity. [23]

E.3.7 Formal security models for confidentiality

A formal security model is a mathematically precise statement of a security policy. Such a model must define a secure state, an initial state, and how the model represents changes in state. The model must be shown to be secure by proving the initial state is secure and all possible subsequent states remain secure. David Bell and Leonard LaPadula of the MITRE Corporation defined the first formal model of confidentiality[9], which stated that if multiple hierarchical levels of confidentiality exist, then one cannot write higher confidentiality data into lower confidential areas and one cannot from a lower confidentiality area read something at a higher level. See [[20], Chapter 5] for an extended exposition also including definitions of "basic" and "simple" security.

A more modern (1980's) model is non-interference. The two concepts are that no one at a lower level of confidentiality should see behaviour that (1) results in any way from any behaviour at a higher level – non-interference [[20], p. 448-50] – or alternately (2) from which any information can be derived about behaviour at a higher level – probabilistic non-interference [52].

E.3.8 Integrity

To maintain system integrity one needs to keep the system in legitimate states or conditions. "Legitimate" must be specified – an integrity security policy could be conditional. For example, it might be allowable for the system to enter otherwise illegitimate states during a transaction, as long as it returns to a legitimate state at the end of the transaction. Early on Biba establish a fundamental integrity property [20] and Clark and Wilson [26] provided in 1987 a discussion of commercially relevant integrity.

Two key sub-problems within integrity are:

- Has something changed?
- Were all of the implemented changes authorized?

Checking that data is unchanged can only have meaning in terms of the question, "Since when?" In practice, this usually means that one must query, "Since in whose possession?" (This possession may or may not be at a specified time.)

Kinds of items where proper privileges and authorization can be of concern include:

- Creating.
- Viewing.

[8] Including protection from cyberstalking

[9] David Elliott Bell and Leonard J. LaPadula, "Secure computer systems: mathematical foundations". MITRE Corporation, 1973 - and - "Secure computer systems: unified exposition and MULTICS interpretation". MITRE Corporation, 1976.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

- Changing.
- Executing.
- Communicating.
- Sharing.
- Encrypting/decrypting.
- Deleting/destroying.

In discussing integrity-related change authorizations, changes commonly concern:

- Credentials (evidence of identity and possibly other attributes).
- Privileges.
- Data.
- Software (possibly considered data).
- The point(s) or paths of execution.
- Time (e.g. resetting the system clock).

Sequence and structure can also be the concern of "integrity" properties. For example, transactional integrity ensures that all parts of a transaction succeed, or none do—it is atomic. Relational integrity (in relational databases) enforces that master-detail relationships are correctly maintained (e.g., if you delete a purchase order, you delete related "detail" records such as purchase order lines enumerating items and quantities ordered.). As mentioned, in 1977, K.J. Biba of the MITRE Corporation defined a mandatory integrity policy model that provided a corollary to the Bell-LaPadula mandatory security model.[10]

E.3.9 Availability

Along with reliability, engineering for availability has a long history in computing. Many traditional approaches and means of prediction exist, but all presume lack of maliciousness. (This is no longer so common in the related area of disaster recovery.) As with all security properties, achieving a specified level of availability is a more difficult problem because one must consider maliciousness. Some of the old approaches and almost all the means of calculation no longer work.

Denial of service attacks from outside – particularly distributed ones originating from many computers simultaneously – can be difficult to successfully overcome. Non-distributed attacks that attempt to take over, exhaust, or destroy resources (e.g. exhaust primary storage) also are a threat. Interestingly, any mechanism designed to deny illegitimate access can tempt attackers to discover a way to use it to deny legitimate access (e.g. locking accounts after a certain number of incorrect passwords tries would allow a malicious person to lock one out of one's account by multiple tries to log in as one with random passwords). Speed of repair or recovery can affect availability.

From a security viewpoint, systems need not only to remain available but preserve their other required security properties, e.g. confidentiality, whether available are not.

[10] K. J. Biba. "Integrity Considerations for Secure Computer Systems" (in MITRE Technical Report TR-3153). The MITRE Corporation, April 1977.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

E.3.10 Accountability

For entities that interact with the system to be held accountable for their actions, those entities must be identified. “Each access to information must be mediated based on who is accessing the information and what classes of information they are authorized to deal with. This identification and authorization information must be securely maintained by the computer system and be associated with every active element that performs some security-relevant action in the system.”[11]

Audit information enables actions affecting security to be traced to the responsible party. The system should be able to record the occurrences of security-relevant events in an audit log or other protected event log. The ability to select the audit events to be recorded is necessary to minimize the expense of auditing and to allow efficient analysis.

Audit data must be protected from modification and unauthorized destruction and, in some environments, their confidentiality must be protected. Because they permit detection and after-the-fact forensic investigations of security violations[12], audit logs can become the targets of attacks that attempt to modify or delete records that could indicate an attacker’s or malicious insider’s actions. In systems that process sensitive data, the audit logs may contain portions of that data, and thus would need to be protected as appropriate for the sensitivity level of that data. In addition, the design of intrusion detection and auditing mechanisms must avoid allowing the exhaustion of log storage space to become a form of attack.

E.3.11 Non-repudiation

Non-repudiation provides proof that any entity that uses a system or acts upon data cannot later deny those actions. Non-repudiation forces users to assume responsibility for their actions so that they cannot disclaim those actions “after the fact” nor deny any event related to themselves—for example, they cannot deny (or repudiate) having been the sender, authorizer, or recipient of a message. Several means of achieving non-repudiation involve cryptographic signatures (more frequently called digital signatures).

ISO/IEC 13888 Information technology – Security techniques – Non-repudiation addresses both symmetric and asymmetric techniques. In symmetric non-repudiation, both the sender and recipient of information are provided with proofs: the sender receives proof that the information was received by the recipient; the recipient receives proof of the identity of the sender. In asymmetric non-repudiation, proof is provided to only one of the parties in a two-party transaction regarding an action of the other party (e.g., sender receives proof of delivery, or recipient receives proof of sender identity, but not both).

E.3.12 Protecting privacy

Privacy needs are one of the key reasons for security. Privacy is a motivation for confidentiality, anonymity, and not retaining data. Avoiding falsehoods that could damage reputations requires data accuracy and integrity. Concern for privacy is widespread, and several relevant laws and regulations exist industry-oriented, sub-national, national, and international.

E.3.13 Safety and security

Not everyone defines the same agreed to boundary between safety and security. However, despite these disagreements many tend toward the position that traditionally they share concerns for adverse consequences and non-malicious but dangerous actions, and security is additionally concerned with malicious and illegal or

[11] Source: DOD 5200.28-STD, Department of Defense Trusted Computer Evaluation Criteria, December 1985.

[12] Other forensic support includes support for identifying suspects and investigating insiders and outsiders. For insiders where the identity of the user may be known, automated recognition of use in an unusual fashion could help support identification of suspects.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

illegitimate actions as well as confidentiality.[13] Safety concerns often centre on lives, health, and environmental damage as well as property damage.

In recent years, the safety community has more examples of successful experience with producing very-low-defect and high-confidence systems and software than does the software security community. The safety community's experience provides valuable lessons for software security practitioners in both producing and assuring software in high-consequence systems (for an introduction and example see [193], [56], and [55]). However, the traditional safety engineering approach differs from the security one in a critical way – it presumes non-existence of maliciousness. Today, security is a concern for most systems as many are directly or indirectly exposed to the Internet or to insider attack as well as to subversion during development, deployment, and updating. While safety-oriented systems so exposed now must also face the security problem, this subclause speaks of traditional safety engineering that does not address maliciousness.

Safety and security are often mentioned together, and some advantage might derive from treating them together. For example, of an approach to an assurance case including security and safety is proposed in [182] and [183][14].

When both are required, a number of areas are candidates for partially combining safety and security engineering concerns including:

- Understanding of the situation.
- Goals.
- Solutions.
- Activities.
- Assurance case[15].
- Claims and particularly subclaims.
- Arguments.
- Evidence.
- Assumptions.
- Evaluations.

In addition, both safety and security have practical concerns for correctness.

E.3.14 Other security-related concerns

In addition to a system's preservation of required security properties within its digital domain, it can contribute to other systems, organizational or societal security goals including:

- Establishing the authenticity of users and data.
- Establishing accountability of users.

[13] Also, security concerns often have more interest in confidentiality than safety concerns do.

[14] The objective of SafSec, developed by Praxis High Integrity Systems is to provide a systems certification and accreditation (C&A) methodology that addresses both safety and security acceptance requirements. SafSec was originally developed for C&A of the United Kingdom Ministry of Defense (UK MOD) Integrated Modular Avionics, and other advanced avionics architectures. SafSec is an example of how approaches from assurance of safety as a required property of software have been applied to the assurance of security properties of software.

[15] The SafSec effort provides guidance on one way to combine assurance cases. [SafSec Standard [183]] [SafSec Guidance [182]]

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

- Providing usability including transparency to users to gain acceptance and resulting security.
- Providing the abilities to:
- Deter and mislead attackers,
- Force attackers to overcome multiple layers of defence,
- Support investigations to identify and convict attackers.
- Limiting real-world damage.
- Aiding physical security, such as in monitoring and entrance control.

Thus, digital systems can help address security concerns at a number of levels.

E.4 Traditional system and software security principles and guidelines

E.4.1 Introduction

Saltzer and Schroeder published their list of principles in 1975, and they remain important [184]. Everyone involved in any way with secure software systems needs to be aware of them. Most of the list below follows the principles proposed by Saltzer and Schroeder and liberally quotes edited selections from that text. These principles have relevance throughout secure software system development and sustainment including requirements; design; construction; and verification, validation, and evaluation.

They cover a number of topics. Several principles help in reducing the number of opportunities for violations. As opportunities or possibilities for violations cannot always be eliminated, steps need to be taken to ensure users properly utilize security and efforts toward security are expended in the best places. To reduce the uncertainties related to the adequacy or correctness of the software system, the portion of the system and mechanisms ensuring security should be as small and simple as practicable and be thoroughly reviewed and analysed.

Defences and protection may not be perfect, and violations will occur. For follow-up, learning, and improvement records of what occurred are needed. In addition, requiring multiple successes by an attacker before substantial damage results can increase time or effort attacker needs to expend and provide some tolerance for vulnerabilities or weaknesses.

E.4.2 Least privilege

Least privilege is a principle whereby each entity (user, process, or device) is granted the most restrictive set of privileges needed for the performance of that entity's authorized tasks. Application of this principle limits the damage that can result from accident, error, or unauthorized use of a system. Least privilege also reduces the number of potential interactions among privileged processes or programs, so that unintentional, unwanted, or improper uses of privilege are less likely to occur.

E.4.3 Complete mediation

Every access to every (security-sensitive) object must be checked for proper authorization; and access denied if it violates authorizations. This principle, when systematically applied, is the primary underpinning of the protection system, and it implies the existence and integrity of methods to (1) identify the source of every request, (2) ensure the request is unchanged since its origination, and (3) check the relevant authorizations as well as ensure request denied if unauthorized and not otherwise (unless by some other mechanism). It also requires that design proposals to allow access by remembering the result of a prior authority check be examined sceptically.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

E.4.4 Fail-Safe defaults

This principle calls for basing access decisions on permission rather than exclusion. Thus, the default situation is lack of access, and the protection scheme identifies conditions under which access is permitted. To be conservative, a design must be based on arguments stating why objects should be accessible, rather than why they should not.

E.4.5 Least common mechanism

Minimize the security mechanisms common to more than one user or depended on by multiple users or levels of sensitivity. Whenever the same executing process services multiple users or handles data from multiple security levels or compartments, this potentially creates an opportunity for illegitimate information flow. Every shared mechanism (as opposed, for example, to non-communicating multiple instances) represents a potential information path between users or across security boundaries and must be designed with great care to ensure against unintentionally compromising security. Virtual machines each with their own copies of the operating system are an example of not sharing the usage of the operating system mechanisms – a single instance is not common across the users of the virtual machines. Thus, one desires the least possible sharing of common mechanisms (instances).

E.4.6 Separation of privilege

A protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of a single key. By requiring two keys, no single accident, deception, or breach of trust is sufficient to compromise the protected information.

Redundancy is also used in the traditional “separation of duties” in human financial processes, e.g. the persons who fill out the check and sign it are two different people.

E.4.7 Psychological acceptability

It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly.[16]

E.4.8 Work factor

The cost of performing to a higher level of quality or of a countermeasure to eliminate or mitigate vulnerability should be commensurate with the cost of a loss if a successful attack were to otherwise occur. Generally, the more valuable the asset targeted by an attacker, the more effort and resources that attacker is willing expend, and therefore the most effort and resources the defender should expend to prevent or thwart the attack.

E.4.9 Economy of mechanism

“Keep the design as simple and small as possible” applies to any aspect of a system, but it deserves emphasis for protection mechanisms, since design and implementation errors that result in unwanted access paths may not be noticed during normal use.

E.4.10 Open design

Security mechanisms should not depend on the ignorance of potential attackers, but rather on assurance of correctness and/or the possession of specific, more easily protected, keys or passwords. This permits the mechanisms to be examined by a number of reviewers without concern that the review may itself compromise the safeguards.

[16] Usable security is a significant issue and is addressed in both the Requirements and Design sections.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

The practice of openly exposing one's design to scrutiny is not universally accepted. The notion that the mechanism should not depend on ignorance is generally accepted, but some would argue that its design should remain secret since a secret design may have the additional advantage of significantly raising the price of penetration. The principle still can be applied, however, restricted to within an organization or a "trusted" group.

E.4.11 Analysability

Systems whose behaviour is analysable from their engineering descriptions such as design specifications and code have a higher chance of performing correctly because relevant aspects of their behaviour can be predicted in advance. In any field, analysis techniques are never available for all structures and substances. To ensure analysability one must restrict structural arrangements and other aspects to those that are analyzable.[17]

E.4.12 Recording of compromises

If a system's defence was not fully successful, trails of evidence should exist to aid understanding, recovery, diagnosis and repair, forensics, and accountability. Likewise, records of suspicious behaviour and "near misses", and records of legitimate behaviours can also have value.

E.4.13 Defence in depth

Defence in depth is a strategy in which human, technological, and operational capabilities are integrated to establish variable protective barriers across multiple layers and dimensions of a system. This principle ensures that an attacker must compromise more than one protection mechanism to successfully exploit a system. Diversity of mechanisms can make the attacker's problem even more difficult. The increased cost of an attack may dissuade an attacker from continuing the attack. Note that multiple less expensive but weak mechanisms do not necessarily make a stronger barrier than fewer more expensive and stronger ones.

E.4.14 Treat as conflict

E.4.14.1 Introduction

Because many security-related issues and activities concern a conflict pitting the system and those aiding in its defence against those who are attacking or may attack in the future, one needs to bring to the situation all that one can of what is known about conflicts. This includes recognizing when attacks in a particular area (e.g. the computer security arena) are part of a wider conflict. Below are some of the key concepts.

E.4.14.2 Intelligent and malicious adversary

The dangers (e.g. threats or hazards) faced may come from a number of sources including intelligent, skilled adversaries. When possession, damage, or denial of assets or interests is highly valuable to someone else, then considerable skill and resources could be brought to bear. When – as is often the case – poor security makes these actions relatively easy and risk free, even things of low value may become attractive.

One cannot simply use a probabilistic approach to one's analyses because, for example, serious, intelligent opponents tend to attack where and when you least expect them – where your estimate of the probability of such an attack is relatively low.[18]

[17] Certainly one would never want a critical structure such as a bridge to be built using a structural arrangement whose behaviour could not be analysed and predicted. Why should this be different for critical software?

[18] In theory, game theory techniques that do not require the probabilities to be known could be applicable, but little progress has been made in practice.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Where judging dangers and consequences is difficult, one possible approach is to make them at least not unacceptable (tolerable) and “as low as reasonably practicable” (ALARP).[19] In employing the ALARP approach, judgments about what to do are based on the cost-benefit of measures or techniques one might apply – not on total budget. However, additional techniques or efforts are not employed once risks have achieved acceptable (not just tolerable) levels. This cost-benefit at the margin (that is, net benefit from a best possible additional step) approach is attractive from an engineering viewpoint, but the amount of benefit cannot always be accurately established.

E.4.14.3 Security is a system, organizational, and societal problem

Security is not merely a specialized concern, and mistaken decisions can result from confining attention to only the system or elements of it. Attempts to break security are often part of a larger conflict such as business competition, crime and law enforcement, social protest, political rivalry, or conflicts among nation states, e.g. espionage. Specifically in secure system development, the social norms and ethics of members of the development team and organization as well as suppliers deserve serious attention. Insiders constitute one of the most dangerous populations of potential attackers, and communicating successfully with them concerning their responsibilities and obligations is important in avoiding subversion or other damage.

E.4.14.4 Measures encounter countermeasures

Despite designers’ positive approach to assuring protection, in practice one also engages in a measure-countermeasure cycle between offence and defence. Currently, reducing the attackers’ relative capabilities and increasing systems’ resilience dominates many approaches. Anonymity of attackers has led to asymmetric situations where defenders must defend everywhere and always,[20] and attackers can choose the time and place of their attacks. Means for reducing anonymity – thereby presumably making deterrence and removal of offenders more effective – could somewhat calm the current riot of illicit behaviour.

E.4.14.5 Learn and adapt

The attack and defence of systems is a normal but not a mundane situation; it is a serious conflict situation with serious adversaries such as criminal organizations, terrorist groups, and nation states as well as competitors committing industrial espionage. Serious analyses of past and current experience can improve tactics and strategy.

One should not forget how to be successful in conflicts. While it is difficult to state the principles of conflict in a brief manner, some principles exist such as exploiting the arenas in which the conflict occurs; using initiative, speed, movement, timing, and surprise; using and trading-off quality and quantity including technology and preparation; carefully defining success and pursuing it universally and persistently but flexibly; and hindering adversaries.

[19] ALARP is a significant concept in UK law, and an excellent engineering-oriented discussion of it appears in Annex B of DEF STAND 00-56 Part 2 [Ministry of Defence 2004b]

[20] Defending everything may not be possible or may waste resources. “He who defends everything defends nothing.” – Frederick II

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Annex F
(informative)

Selected Related Standards

F.1 Introduction

This informative annex lists some of the International Standards related to systems and software and their dependability, safety, security, and other properties. A brief description of many standards is provided. The discursive part of this material derives in part from work by the US National Defense Industry Association and is used with their permission.

F.2 ISO/IEC 2382-14:1997, Information technology — Vocabulary — Part 14: Reliability, maintainability and availability

F.3 ISO 2394:1998, General principles on reliability for structures

F.4 ISO/IEC 90003, ISO/IEC 12207 and ISO/IEC 15288 — SDLC QA standards

Extending the general-purpose Quality Assurance standards in the ISO 9000 series, the following standards cover the application of QA processes to the System or Software Development Life Cycle specifically and have applicability to the requirements of system assurance:

- **ISO/IEC 90003,** *Software engineering — Guidelines for the application of ISO 9001 to computer software* covers software engineering, guiding the application of ISO 9001, the quality assurance standards, to the systems development process:
    - **ISO/IEC 12207** covers software life cycle processes, providing a conceptual framework and terminology.
    - **ISO/IEC 15288** *System Life Cycle Processes* covers systems engineering by defining a set of processes and terminology.

F.5 ISO/IEC 13335, IT security

F.6 ISO 9241-400:2007, Ergonomics of human–system interaction — Part 400: Principles and requirements for physical input devices

F.7 ISO 12100-1:2003, Safety of machinery — Basic concepts, general principles for design — Part 1: Basic terminology, methodology

F.8 ISO 12100-2:2003, Safety of machinery — Basic concepts, general principles for design — Part 2: Technical principles

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

F.9 ISO/IEC 13335, Information technology — Security techniques — Management of information and communications technology security – series

ISO/IEC 13335 (which started life as a Technical Report TR before becoming a full ISO standard) comprises a set of guidelines for the management of IT security, focusing primarily on technical security control measures:

- **ISO/IEC 13335-1**:2004 “Information technology – Security techniques – Management of information and communications technology security – Part 1: Concepts and models for information and communications technology security management”. Explains the concepts and models for information and communications technology security management. (ISO/IEC TR 13335 parts 1 and 2 were combined into the revised ISO/IEC 13335-1: 2004. The original TR13335-2:1997 “Guidelines for the management of IT security - Part 2: Managing and planning IT security” was cancelled.)
- **ISO/IEC 13335-2**, when published, is expected to cancel and replace ISO/IEC TR 13335-3:1998 and ISO/IEC TR 13335-4:2000.
- **ISO/IEC TR 13335-3**:1998 “Information technology – Guidelines for the Management of IT Security – Part 3: Techniques for the management of IT Security”. Covers techniques for the management of IT security. This standard’s revision inserts into ISO 27005
- **ISO/IEC TR 13335-4**:2000 covers the **selection of safeguards** (meaning technical security controls). This standard is also currently under revision and will be inserted into ISO/IEC 27005
- **ISO/IEC TR 13335-5**:2001 provides management guidance on network security. This standard is currently under revision, being merged into ISO/IEC 18028-1. ISO/IEC 18028-1 will eventually cancel and replace ISO/IEC TR 13335-5:2001.

F.10 ISO 13849, Safety of machinery — Safety-related parts of control systems – series

- ISO 13849-1:2006 Safety of machinery -- Safety-related parts of control systems -- Part 1: General principles for design
- ISO 13849-2:2003 Safety of machinery -- Safety-related parts of control systems -- Part 2: Validation

F.11 ISO 14620, Space systems — Safety requirements – series

- ISO 14620-1:2002 Space systems -- Safety requirements -- Part 1: System safety
- ISO 14620-2:2000 Space systems -- Safety requirements -- Part 2: Launch site operations
- ISO 14620-3:2005 Space systems -- Safety requirements -- Part 3: Flight safety systems

F.12 ISO 14625:2007, Space systems — Ground support equipment for use at launch, landing or retrieval sites — General requirements

F.13 ISO/IEC TR 15446:2004, Information technology — Security techniques — Guide for the production of Protection Profiles (PPs) and Security Targets (STs)

ISO/IEC TR 15446 provides guidance relating to the construction of PPs and STs that are intended to be compliant with ISO/IEC 15408 (the "Common Criteria") and gives suggestions on how to develop each section of a PP or ST. This supported by an annex that contains generic examples of each type of PP and ST component, and by other annexes that contain detailed worked examples.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

F.14 ISO 15408:1999

ISO 15408:1999 describes the Common Criteria for Information Technology Security Evaluation. Products that are evaluated against the Common Criteria have a defined level of assurance as to their information security capabilities that is recognized in most of the world. Unfortunately, the evaluation process is quite costly and slow, and is therefore not very widely used apart from the government and defence markets.

F.15 ISO/IEC TR 15543, Information technology — Security techniques — A framework for IT security assurance

- **General**
  - Guides selection of an appropriate assurance method when specifying, selecting, or deploying a security service, product, or environmental factor such as an organization or personnel (known as a deliverable).
  - Facilitates the understanding of the assurance type and effort required to achieve confidence that the deliverable satisfies stated IT security assurance requirements and security policy.
  - Describes fundamentals of security assurance and relation to other security concepts:
    - Clarifies why security assurance is required and dispels misconceptions that increased assurance is gained by increasing the strength of security mechanisms.
    - Includes a categorization of assurance types and a generic life cycle model to identify the appropriate assurance types required for the deliverable:
      - Demonstrates how security assurance must be managed throughout the deliverable's life cycle requiring assurance decisions to be made by several assurance authorities for the life cycle stage relevant to their organization (i.e. developer, standards, and consumer).
      - Accommodates different assurance types and maps into any life cycle approach so as not to dictate any particular design.
  - Includes advanced security assurance concepts, such as combining security assurance methods.
- **The three parts of ISO/IEC TR 15543**
  - Part 1, Overview and Framework provides fundamental concepts and general description of assurance methods:
    - Targets IT security in developing a security assurance program, determining the security assurance of deliverables, entering assurance assessment audits (e.g. ISO 9000, ISO/IEC 21827, ISO/IEC 15408-3), or other assurance activities.
  - Part 2, Assurance Methods describes a variety of assurance methods and approaches and relates them to Part 1 security assurance framework model:
    - Identifies qualitative properties of assurance methods.
    - Aids in understanding how to obtain assurance in a given life cycle stage of deliverable.
  - Part 3, Analysis of Assurance Methods analyses the various methods with respect to their assurance properties and aids Assurance Authorities:
    - In deciding relative value of Assurance Approaches and determining that they will provide the assurance results most appropriate to their needs.
    - To use assurance results to achieve desired confidence of the deliverable.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

F.16 ISO/TR 17944:2002, Banking — Security and other financial services — Framework for security in financial systems

F.17 ISO/IEC 18014-2:2002, Information technology — Security techniques — Time-stamping services – [Three Parts]

F.18 ISO/IEC 18028, Information technology — Security techniques — IT network security

F.19 ISO 19706:2007, Guidelines for assessing the fire threat to people

F.20 ISO 19770, Software asset management

ISO/IEC 19770-1:2006 promotes the implementation of an integrated set of software asset management processes, using good practices for efficient software management. Contents include Scope, terms and definitions, Field of application, Conformance, Intended usage, Agreement compliance, General Software Asset Management processes, Control environment for Software Asset Management, Planning and implementation, Inventory processes, Verification and compliance processes, Operations management processes and interfaces, Life cycle process interfaces.

F.21 ISO/IEC TR19791, Information technology — Security techniques — Security assessment of operational systems

ISO/IEC TR 19791:2006 provides guidance and criteria for the security evaluation of operational systems. It provides an extension to the scope of ISO/IEC 15408. The main additions address evaluation of the operational environment surrounding the target of evaluation, and the decomposition of complex operational systems into security domains that can be separately evaluated. It does not define techniques for the identification, assessment and acceptance of operational risk.

F.22 ISO 21827, Systems Security Engineering Capability Maturity Model

Like other Capability Maturity Models (CMMs), the Systems Security Engineering (SSE) CMM defines the essential characteristics of SSE processes, emphasizing those which indicate process maturity. The model covers the entire systems development life cycle from concept definition to decommissioning. It applies to those developing or integrating secure products/systems, and those providing specialist security services such as security engineering. It was published as ISO 21827 in 2002.

F.23 ISO/IEC 25000 series

This series covers software quality.

F.24 ISO/IEC 25020:2007, Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Measurement reference model and guide

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

F.25 ISO/IEC 25051:2006, Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing

F.26 ISO/TS 25238:2007, Health informatics — Classification of safety risks from health software

F.27 ISO/IEC 26702:2007, Systems engineering — Application and management of the systems engineering process

ISO/IEC 26702:2007 defines the interdisciplinary tasks which are required throughout a system's life cycle to transform customer needs, requirements and constraints into a system solution. In addition, it specifies the requirements for the systems engineering process and its application throughout the product life cycle. ISO/IEC 26702:2007 focuses on engineering activities necessary to guide product development, while ensuring that the product is properly designed to make it affordable to produce, own, operate, maintain and eventually dispose of without undue risk to health or the environment.

F.28 ISO/IEC 27000-series

ISO has reserved the ISO/IEC 27000-series numbering for a range of information security management standards. The following ISO 27000-series standards are either published or planned.

F.28.1 ISO 27000

Contains the vocabulary and definitions i.e. terminology for all of these information security management standards

F.28.2 ISO 27001

Is the Information Security Management System requirements standard (specification) against which organizations are formally certified compliant In October 2005, British Standard BS 7799 part 2 was adopted by ISO, re-badged and released as the new international information security standard ISO/IEC 27001:2005. ISO 27001 is the formal standard against which organizations may seek independent certification of their Information Security Management Systems (meaning their frameworks to design, implement, manage, maintain and enforce information security processes and controls systematically and consistently throughout the organizations).

F.28.3 ISO 27002

The standard currently known as ISO 17799 and formerly known as BS 7799 Part 1. This is the code of practice for information security management describing a comprehensive set of information security control objectives and a menu of best-practice security controls

F.28.4 ISO 27003

An implementation guide

F.28.5 ISO 27004

Information security management measurement standard to help measure the effectiveness of information security management system implementations.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

F.28.6 ISO 27005

Will be an information security risk management standard (will replace the recently issued BS 7799 Part 3).

F.28.7 ISO 27006, ISO/IEC 27006:2007, Information technology — Security techniques — Requirements for bodies providing audit and certification of information security management systems

A guide to the certification/registration process for accredited ISMS certification/registration bodies.

F.28.8 ISO/TR 27809:2007, Health informatics — Measures for ensuring patient safety of health software

F.29 ISO 28003:2007, Security management systems for the supply chain — Requirements for bodies providing audit and certification of supply chain security management systems

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Bibliography

[1] Abran, Alain, James W. Moore (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

[2] Adamski, A, and R. Westrum. "Requisite imagination: The fine art of anticipating what might go wrong." In [58], p. 193-220, 2003

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

[4] Alexander, Ian. Systems Engineering Isn't Just Software. 2001. Available at http://easyweb.easynet.co.uk/~iany/consultancy/systems_engineering/se_isnt_just_sw.htm

[5] Allen, Julia H., Sean Barum, Robert J. Ellison, Gary McGraw, and Nancy R. Mead. Software Security Engineering: A Guide for Project Managers. Addison-Wesley, 2008

[6] 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

[7] Anderson, James 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)

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

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

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

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

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

[13] Australian Government: Department of Industry, Tourism and Resources. “National Offshore Petroleum Safety Case Guidelines.” [Online Document] 20 Sept 2006 [cited on: 13 Feb 2007] Available HTTP: http://www.industry.gov.au/content/itrinternet/cmscontent.cfm?objectID=F5BA1B26-65BF-4956-B95174C5AF384515

[14] Avizienis, Algirdas, Jean-Claude Laprie, Brian Randell, and Carl Landwehr, “Basic Concepts and Taxonomy of Dependable and Secure Computing,” IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, pp. 11-33, Jan.-Mar. 2004. Available at http://csdl.computer.org/dl/trans/tq/2004/01/q0011.pdf

[15] AVSAF: Aviation Safety. [Website] 2004 [cited on: 09 Feb 2007] Available HTTP: http://www.avsaf.org/case/

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

[16] Bahill, A.T. and B. Gissing, "Re-evaluating Systems Engineering Concepts Using Systems Thinking". IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews, Vol. 28 No. 4 pp. 516-527, November 1998

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

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

[19] Bishop, Matt, and Sophie Engle. “The Software Assurance CBK and University Curricula.” Proceedings of the 10th Colloquium for Information Systems Security Education, 2006

[20] Bishop, Matt. Computer Security: Art and Practice, Addison-Wesley, 2003

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

[22] Bishop, P. and Bloomfield, R. “The SHIP Safety Case Approach.” SafeComp95, Belgirate, Italy. Oct 1995

[23] Buehner, M. J. & Cheng, P. W. (2005). Causal Learning. In R. Morrison & K. J. Holyoak (Eds.) Handbook of Thinking and Reasoning [62]. Cambridge University Press, pp143-168

[24] Cannon, J. C. Privacy, Addison Wesley, 2005

[25] Chung, Lawrence, et al. Non-Functional Requirements in Software Engineering, Kluwer, 1999

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

[27] CNSS, National Information Assurance Glossary, May 2003. Available at http://www.cnss.gov/full-index.html

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

[29] 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

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

[31] Common Weaknesses Enumeration. MITRE, 2007. Avail;able at http://cwe.mitre.org (200707)

[32] Cooke, Nancy J., Jamie C. Gorman, and Jennifer L. Winner. “Team Cogitation.” p. 239-268 in [45]

[33] Courtois, Pierre-Jacques. Justifying the Dependability of Computer-based Systems: With Applications in Nuclear Engineering. Springer, 2008

[34] Cranor, Lorrie, and Simson Garfinkel. Security and Usability: Designing Secure Systems that People Can Use. O’Reilly, 2005

[35] Dayton-Johnson, Jeff. 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

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

[36] Defence Materiel Organisation, Australian Department of Defence. +SAFE, V1.2: A Safety Extension to CMMI-DEV, V1.2. Software Engineering Institute, CMU/SEI-2007-TN-006, March 2007

[37] Defence Materiel Organisation, Australian Department of Defence. +SAFE, V1.2: A Safety Extension to CMMI-DEV, V1.2. Software Engineering Institute, CMU/SEI-2007-TN-006, March 2007

[38] Department of Defense Instruction 8500.2 (6 February 2003). Information Assurance (IA) Implementation. Washington, DC: US Department of Defense, 2003. Available at http://www.dtic.mil/whs/directives/corres/pdf2/i85002p.pdf

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

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

[41] Dependability Research Group. “Safety Cases.” [Online Document] [cited on: 13 Feb 2007] Available HTTP: http://dependability.cs.virginia.edu/info/Safety_Cases

[42] Despotou, Georgios, and Tim Kelly. “Extending the Safety Case Concept to Address Dependability,” Proceedings of the 22nd International System Safety Conference, 2004

[43] Dowd, Mark, John McDonald, Justin Schuh. The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities. Addison-Wesley, 2006

[44] Dunbar, Kevin, and Jonathan Fugelsang. “Scientific Thinking and Reasoning.” p. 705-727 in [62]

[45] Durso, Francis T. (Editor), Raymond S. Nickerson (Editor), Susan T. Dumais (Editor), Stephan Lewandowsky (Editor), Timothy J. Perfect (Editor). Handbook of Applied Cognition 2 edition. Wiley, 2007

[46] Ellsworth, Phoebe C. “Legal Reasoning.” P. 685-704 In [62]

[47] Ericsson, K. Anders (Editor), Neil Charness (Editor), Paul J. Feltovich (Editor), Robert R. Hoffman (Editor).The Cambridge Handbook of Expertise and Expert Performance. Cambridge University Press, 2006

[48] Eurocontrol. “Safety Case.” [Online Document] 08 Oct 2006 [cited on: 13 Feb 2007] Available HTTP: http://www.eurocontrol.int/cascade/public/standard_page/safety_case.html

[49] Fenton, N., Littlewood, B., Neil, M., Strigini, L., Sutcliffe, A., and Wright, D. “Assessing dependability of safety critical systems using diverse evidence.” IEE Proceedings – Software. 1998

[50] Gasser, M. Building a Secure Computer System. Van Nostrand Reinhold, 1988. Available at http://nucia.ist.unomaha.edu/library/gasser.php

[51] Government of Western Australia: Department of Consumer and Employment Protection. “The Safety Case.” [Online Document] 21 April 2006 [cited on: 13 Feb 2007] Available HTTP: http://www.docep.wa.gov.au/resourcessafety/Sections/Petroleum_Hazard_Facilities/Guidance_material_and_publications/The_safety_case.html

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

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

[54] Greenwell, William S., John C. Knight, and Jacob J. Pease. "A Taxonomy of Fallacies in System Safety Arguments” 24th International System Safety Conference, Albuquerque, NM, August 2006

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

[55] Hall, Anthony and Rodrick Chapman. “Correctness by Construction: Developing a Commercial Secure System,” IEEE Software, vol. 19, no. 1, pp.18-25, Jan/Feb 2002

[56] Herrmann, Debra S. Software Safety and Reliability, IEEE Computer Society Press, 1999

[57] Hoglund, Greg, and Gary McGraw. Exploiting Software: How to break code. Addison-Wesley, 2004

[58] Hollnagel, Erik. (Ed.). Handbook of cognitive task design. Lawrence Erlbaum Associates, 2003

[59] Hollnagel, Erik. Barriers and Accident Prevention, Ashgate, 2004

[60] Hollnagel, Erik (Editor), David D. Woods (Editor), Nancy Leveson (Editor). Resilience Engineering: Concepts and Precepts. Ashgate Pub Co, 2006

[61] Hollnagel, Erik. “Human Error: Trick or Treat?” in [45], p. 219-238, 2007

[62] Hollnagel, Erik. Human Factors: From Liability to Asset. Presentation, 2007. Available at www.vtt.fi/liitetiedostot/muut/Hollnagel.pdf

[63] Holyoak, Keith J. and Robert G. Morrison. The Cambridge Handbook of Thinking and Reasoning. Cambridge University Press, 2005

[64] Howard, Michael, and David C. LeBlanc. Writing Secure Code, 2nd ed., Microsoft Press, 2002

[65] Howard, Michael, and Steve Lipner. The Security Development Lifecycle. Microsoft Press, 2006

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

[67] Ibrahim, Linda, et al, Safety and Security Extensions for Integrated Capability Maturity Models. Washington D.C.: United States Federal Aviation Administration, Sept. 2004. Available at http://www.faa.gov/ipg/pif/evol/index.]

[68] International Council on Systems Engineering INCOSE. Guide to Systems Engineering Body of Knowledge (G2SEBoK). Available at http://www.incose.org/

[69] IEC 50 (191), *International Electrotechnical Vocabulary, Chapter 191: Dependability and Quality of Service*

[70] IEC 60300, *Dependability management [several parts]*

[71] IEC 60300-3-2:1993, *Dependability management – Part 3: Application guide – Section 2: Collection of dependability data from the field*

[72] IEC 60300-3-15, *Dependability management - Part 3-15: Guidance to engineering of system dependability*

[73] IEC 60812:2006, *Analysis techniques for system reliability - Procedure for failure mode and effects analysis (FMEA)*

[74] IEC 61025:2006, *Fault tree analysis (FTA)*

[75] IEC 61078:2006, *Analysis techniques for dependability - Reliability block diagram and boolean methods*

[76] IEC 61508:2005, *Functional safety of electrical/electronic/programmable electronic safety-related systems [several parts]*

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

[77] IEC 61508-7:2000, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 7: Overview of techniques and measures*

[78] IEC 61511:200x, *Functional safety - Safety instrumented systems for the process industry sector [several parts]*

[79] IEC 61882:2001, *Hazard and operability studies (HAZOP studies) - Application guide*

[80] IEEE Std 1228-1994, *IEEE Standard for Safety Plans*

[81] ISO/IEC 2382-14:1997, *Information technology — Vocabulary — Part 14: Reliability, maintainability and availability*

[82] ISO 2394:1998, *General principles on reliability for structures*

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

[84] ISO 9241-400:2007, *Ergonomics of human–system interaction — Part 400: Principles and requirements for physical input devices*

[85] ISO 12100-1:2003, *Safety of machinery — Basic concepts, general principles for design — Part 1: Basic terminology, methodology*

[86] ISO 12100-2:2003, *Safety of machinery — Basic concepts, general principles for design — Part 2: Technical principles*

[87] ISO/IEC 12207:2008, *Systems and software engineering — Software life cycle processes*

[88] ISO/IEC TR 13335 (all parts), *Information technology — Security techniques — Management of information and communications technology security*

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

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

[91] ISO 14625:2007, *Space systems — Ground support equipment for use at launch, landing or retrieval sites — General requirements*

[92] ISO/IEC 15288:2008, *Systems and software engineering — System life cycle processes*

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

[94] ISO/IEC TR 15446:2009, *Information technology — Security techniques — Guide for the production of Protection Profiles and Security Targets*

[95] ISO/IEC TR 15443 (all parts), *Information technology — Security techniques — A framework for IT security assurance*

[96] ISO/IEC 15939:2007, *Systems and software engineering — Measurement process*

[97] ISO/IEC 16085:2006, *Systems and software engineering — Life cycle processes — Risk management*

[98] ISO/IEC 16326:2009, *Systems and software engineering — Life cycle processes — Project management*

[99] ISO/TR 16982:2002, *Ergonomics of human-system interaction — Usability methods supporting human-centred design*

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

[100] ISO/TR 17944:2002, *Banking — Security and other financial services — Framework for security in financial systems*

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

[102] ISO/IEC 18028 (all parts), *Information technology — Security techniques — IT network security*

[103] ISO/TR 18529:2000, *Ergonomics — Ergonomics of human-system interaction — Human-centred lifecycle process descriptions*

[104] ISO 19706:2007, *Guidelines for assessing the fire threat to people*

[105] ISO 19770 (all parts), *Information technology — Software asset management*

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

[107] ISO/TS 20282 (all parts), *Ease of operation of everyday products*

[108] ISO/IEC 21827:2002, *Information technology — Security techniques — Systems Security Engineering — Capability Maturity Model® (SSE-CMM®)*

[109] ISO/IEC 25000, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*

[110] ISO/IEC 25010, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Quality model*

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

[112] ISO/IEC 25020:2007, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Measurement reference model and guide*

[113] ISO/IEC 25030:2007, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Quality requirements*

[114] ISO/IEC 25040, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Evaluation reference model and guide* [21]

[115] ISO/IEC 25051:2006, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing*

[116] ISO/TS 25238:2007, *Health informatics — Classification of safety risks from health software*

[117] ISO/IEC 26702:2007, *Systems engineering — Application and management of the systems engineering process*

[118] ISO/IEC 27000:2009, *Information technology — Security techniques — Information security management systems — Overview and vocabulary*

[119] ISO/IEC 27001:2005, *Information technology — Security techniques — Information security management systems — Requirements*

[21] To be published.

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

[120] ISO/IEC 27002:2005, *Information technology — Security techniques — Code of practice for information security management*

[121] ISO/IEC 27004:2009, *Information technology — Security techniques — Information security management — Measurement*

[122] ISO/IEC 27005:2008, *Information technology — Security techniques — Information security risk management*

[123] ISO/IEC 27006:2007, *Information technology — Security techniques — Requirements for bodies providing audit and certification of information security management systems*

[124] ISO/IEC 27011:2008, *Information technology — Security techniques — Information security management guidelines for telecommunications organizations based on ISO/IEC 27002*

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

[126] ISO/TR 27809:2007, *Health informatics — Measures for ensuring patient safety of health software*

[127] ISO 28003:2007, *Security management systems for the supply chain — Requirements for bodies providing audit and certification of supply chain security management systems*

[128] ISO/IEC 42010:2007, *Systems and software engineering — Recommended practice for architectural description of software-intensive systems*

[129] ISO/IEC Directives, Part 2: *Rules for the structure and drafting of International Standards*, Fifth edition, 2004

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

[131] Kazman R., M. Klein, and P. Clements, 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

[132] Kelly, T. “Arguing Safety – A Systematic Approach to Managing Safety Cases.” Doctorial Thesis – University of York: Department of Computer Science. Sept 1998

[133] 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

[134] Kelly, T. and 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.

[135] Ladkin, P. “The Pre-Implementation Safety Case for RVSM in European Airspace is Flawed.” [Online Document] 29 Aug 2002 [cited on: 09 Feb 2007] Available HTTP: http://www.rvs.uni-bielefeld.de/publications/Reports/SCflawed-paper.html

[136] Landwehr, Carl, “Computer Security,” IJIS vol. 1, pp. 3-13, 2001

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

[138] LeBoeuf, Robyn A., and Eldar B. Shafir. “Decision Making.” p. 243-266 In [62]

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

[139] Leveson, Nancy. “A Systems-Theoretic Approach to Safety in Software-Intensive Systems,” IEEE Transactions on Dependable and Secure Computing 1, 1 (January-March 2004): 66-86, 2004

[140] Lipner, Steve and Michael Howard, The Trustworthy Computing Security Development Lifecycle, Microsoft, 2005. Available: http://msdn.microsoft.com/security/ dufault.aspx ?pull=/library/en-us/dnsecure/html/sdl.asp#sdl2_topic8?_r=1

[141] Maguire, Richard. Safety Cases abd Safety Reports: Meaning, Motivation and Management. Ashgate, 2006

[142] McDermid, J. “Software Safety: Where’s the Evidence?” 6th Australian Workshop on Industrial Experience with Safety Critical Systems and Software (SCS ’01), Brisbane. 2001

[143] McGraw, Gary. Software Security: Building Security In. Addison Wesley, 2006

[144] McLean, J. “Security Models.” Encyclopedia of Software Engineering (J. Marciniak editor). Wiley 1994

[145] Meier, J.D., Alex Mackman, Srinath Vasireddy, Michael Dunner, Ray Escamilla, and Anandha Murukan, Improving Web Application Security: Threats and Countermeasures, Microsoft, 2004. Available at: http://download.microsoft.com/download/d/8/c/d8c02f31-64af-438c-a9f4-e31acb8e3333/ Threats_Countermeasures.pdf

[146] Merkow, Mark S. and Jim Breithaupt, Computer Security Assurance Using the Common Criteria, Thompson Delamr Learning, 2005

[147] Ministry of Defence Standard 00-42 Issue 2, Reliability and Maintainability (R&M) Assurance Guidance. Part 3, R&M Case, 6 June 2003

[148] Ministry Of Defence. Defence Standard 00-55 (PART 1)/Issue 2, Requirements for Safety Related Software in Defence Equipment Part 1: Requirements, 2 1 August 1997

[149] Ministry of Defence. Defence Standard 00-55 (PART 2)/Issue 2, Requirements for Safety Related Software in Defence Equipment Part 2: Guidance, 2 1 August 1997

[150] Ministry of Defence. Interim Defence Standard 00-56, Safety Management Requirements for Defence Systems Part 1: Requirements, 17 December 2004

[151] Ministry of Defence. Interim Defence Standard 00-56, Safety Management Requirements for Defence Systems Part 2: Guidance on Establishing a Means of Complying with Part 1, 17 December 2004

[152] Moore, A., Klinker, E., and Mihelcic, D. “How to Construct Formal Arguments that Persuade Certifiers.” Industrial Strength Formal Methods. Academic Press. 1999

[153] National Aeronautics and Space Administration (NASA) Software Assurance Guidebook (NASA-GB-A201). Available at http://satc.gsfc.nasa.gov/assure/agb.txt

[154] National Computer Security Center. A Guide to Understanding Covert Channel Analysis of Trusted Systems, NCSC-TG-030, NCSC, November 1993. Available at: http://www.radium.ncsc.mil/tpep/process/overview.html

[155] National Offshore Petroleum Safety Authority. “Safety Case Guidelines.” September 2004

[156] National Offshore Petroleum Safety Authority. Guideline: Construction Safety Cases. 30 Aug 2006

[157] National Offshore Petroleum Safety Authority. Safety case approach. [Online Document] [cited on: 13 Feb 2007] Available HTTP: http://www.nopsa.gov.au/safety.asp

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

[158] National Research Council (NRC) Computer Science and Telecommunications Board (CSTB). Cybersecurity Today and Tomorrow: Pay Now or Pay Later. National Academies Press, 2002. Available at http://darwin.nap.edu/books/0309083125/html.

[159] National Security Agency, The Information Systems Security Engineering Process (IATF) v3.1. 2002

[160] Naval Research Laboratory, Handbook for the Computer Security Certification of Trusted Systems, US Naval Research Laboratory, 1995

[161] NDIA System Assurance Committee. Engineering for System Assurance version .90. National Defense Industrial Association (USA), 2008

[162] NIST. Federal Information Processing Standards Publication (FIPS PUB) 200: Minimum Security Requirements for Federal Information and Information Systems. March 2006. Available at http://csrc.nist.gov/publications/fips/fips200/FIPS-200-final-march.pdf.

[163] NIST. NIST SP 800-33, Underlying Technical Models for Information Technology Security, December 2001

[164] NIST. NIST Special Publication 800-27: Engineering Principles for Information Technology Security (A Baseline for Achieving Security). Revision A, June 2004. Available at http://csrc.nist.gov/publications/nistpubs/800-27A/SP800-27-RevA.pdf

[165] OPEN Process Framework. Safety Cases. [Online Document] 5 Nov 2005. [cited on: 13 Feb 2007] Available HTTP: http://www.opfro.org/index.html?Components/WorkProducts/SafetySet/SafetySet.html~Contents

[166] OPSI. The Offshore Installations (Safety Case) Regulations 2005. [Online Document] 9 Nov 2005. [cited on: 13 Feb 2007] Available HTTP: http://www.opsi.gov.uk/si/si2005/20053117.htm

[167] Park, J., Montrose, B., and Froscher, J. “Tools for Information Security Assurance Arguments.” DARPA Information Survivability Conference & Exposition II, 2001. DISCEX ’01. Proceedings. 2001

[168] Petroski, Henry. Design Paradigms. Cambridge University Press, 1994

[169] Prasad, D., Dependable Systems Integration using Measurement Theory and Decision Analysis, PhD Thesis, Department of Computer Science, University of York, UK, 1998

[170] Prieto -Diaz, Ruben, Common Criteria Evaluation Process, Commonwealth Information Security Center Technical Report CISC-TR-2002-003, 2003. http://www.jmu.edu/cisc/research/publications/CCevaluationProcesTR03-5.pdf

[171] PSM Safety & Security TWG. “Security Measurement.” Nov 2004

[172] Pullum, L. L. Software Fault Tolerance, Artech House, 2001

[173] Randell, B., and Koutny, M. Failures: Their Definition, Modelling and Analysis. School of Computing Science, Newcastle University CS-TR NO 994, Dec 2006 Randell, B., and Rushby, J. M. Distributed Secure Systems: Then and Now. CS-TR No 1052 School of Computing Science, Newcastle University, Oct 2007

[174] Rechtin, E. Systems Architecting of Organizations: Why Eagles Can't Swim. Boca Raton, FL: CRC Press, 2000

[175] Redwine, Samuel T., Jr. (Editor). Software Assurance: A Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software Version 1.1. US Department of Homeland Security, September 2006

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

[176] Redwine, Samuel T., Jr. “The Quality of Assurance Cases” Workshop on Assurance Cases for Security: The Metrics Challenge, International Conference on Dependable Systems and Networks, 2007

[177] Redwine, Samuel T., Jr., and Noopur Davis (Editors). Processes for Producing Secure Software: Towards Secure Software. Vols. I and II. Washington, D.C.: National Cyber Security Partnership, 2004. Available at http://www.cigital.com/papers/download/secure_software_process.pdf

[178] Riguidel, Michel, Gwendal Le Grand, Cyril Chiollaz, Sved Naqvi, Mikael Formanek, “D1.2 Assessment of Threats and Vulnerabilities in Networks”, Version 1.0. European Union Security Expert Initiative (SEINIT), 31 August 2004. Available at http://www.seinit.org/documents/Deliverables/SEINIT_D1.2_PU.pdf

[179] Ross, Karol G., Jennifer L. Shafer, and Garry Klein. “Professional Judgements and ‘Naturalistic Decision Making’.” p. 403-420 in [47]

[180] Ross, Ron et al. Recommended Security Controls for Federal Information Systems, NIST Special Publication 800-53, Feb. 2005. Available at http://csrc.nist.gov/publications/PubsSPs.html

[181] SAE JA1000 Reliability Program Standard, SAE International, June 1998. Available at http://www.sae.org

[182] SafSec Project. SafSec Methodology: Guidance Material: Integration of Safety and Security. Available at: http://www.praxis-his.com/safsec/safSecStandards.asp

[183] SafSec Project. SafSec Methodology: Standard: Integration of Safety and Security. Available at: http://www.praxis-his.com/safsec/safSecStandards.asp

[184] Saltzer, J. H. and M. D. Schroeder, “The protection of information in computer systems,” Proceedings of the IEEE, vol. 63, no. 9, pp. 1278-1308, 1975. Available on-line at http://cap-lore.com/CapTheory/ProtInf/

[185] Seminal Papers - History of Computer Security Project, University of California Davis Computer Security Laboratory. Available at: http://seclab.cs.ucdavis.edu/projects/history/seminal.html

[186] Serene. “Safety argument.” [Online Document] [cited on: 13 Feb 2007] Available HTTP: http://www2.dcs.qmul.ac.uk/~norman/SERENE_Help/sereneSafety_argument.htm

[187] Severson, K. “Yucca Mountain Safety Case Focus of NWTRB September Meeting.” United States Nuclear Waste Technical Review Board. Aug 2006

[188] Sieck, Winston R., and Gary Klein. “Decision making.” p. 195-218 in [45]

[189] Software and Systems Engineering Vocabulary (sevocab). Available at www.computer.org/sevocab/

[190] Software Engineering Institute. Assurance Case and Plan Preparation. [Online Document] [cited on: 26 Feb 2008] Available at http://www.sei.cmu.edu/pcs/acprep.html

[191] Sommerville, Ian. Software Engineering, 8th ed., Pearson Education, 2006

[192] Stoneburner, Gary, Hayden, Clark and Feringa, Alexis. Engineering Principles for Information Technology Security (A Baseline for Achieving Security), Revision A, NIST Special Publication 800-27 Rev A, June 2004

[193] Storey, Neil. Safety-Critical Computer Systems. Addison Wesley, 1996

[194] Strunk, E. and Knight, J. “The Essential Synthesis of Problem Frames and Assurance Cases.” IWAAPF’06, Shanghai, China. May 2006

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

[195] Swiderski, F. and W. Snyder, Threat Modeling, Microsoft Press, 2004

[196] U.S. NRC. “Quality Assurance Case Studies at Construction Projects.”

[197] UK CAA. CAP 670 Air Traffic Services Safety Requirements. UK Civil Aviation Authority Safety Regulation Group, 12 June 2003 amended through 30 June 2006

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

[199] UK CAA 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, 13 January 2006

[200] Vanfleet, W. Mark and R. William Beckwith, Dr. Ben Calloni, Jahn A. Luke, Dr. Carol Taylor, and Gordon Uchenick. “MILS: Architecture for High Assurance Embedded Computing,” Crosstalk, August, 2005

[201] Viega, J., The CLASP Application Security Process, Secure Software, 2005. Available at http://www.securesoftware.com

[202] Viega, J., The CLASP Application Security Process, Secure Software, 2005. Available at http://www.securesoftware.com

[203] Viega, John, and Gary McGraw, Building Secure Software: How to Avoid Security Problems the Right Way, Reading, MA: Addison Wesley, 2001

[204] Walker, Vern R. "Risk Regulation and the 'Faces' of Uncertainty," Risk: Health, Safety and Environment. p. 27-38, Winter 1998

[205] Ware, Willis H. Security Controls for Computer Systems (U): Report of Defense Science Board Task Force on Computer Security, The RAND Corporation, Santa Monica, CA (Feb. 1970).

[206] Weaver, R. “The Safety of Software – Constructing and Assuring Arguments.” Doctorial Thesis – University of York: Department of Computer Science. 2003.

[207] Weaver, R., Fenn, J., and Kelly, T. “A Pragmatic Approach to Reasoning about the Assurance of Safety Arguments.” 8th Australian Workshop on Safety Critical Systems and Software (SCS’03), Canberra. 2003

[208] Whittaker, J. A. and H. H. Thompson. How to Break Software Security: Effective Techniques for Security Testing. Pearson Education, 2004

[209] Williams, J. and Schaefer, M. “Pretty Good Assurance.” Proceedings of the New Security Paradigms Workshop. IEEE Computer Society Press. 1995

[210] Williams, Jeffrey R. and George F. Jelen, A Framework for Reasoning about Assurance, Document Number ATR 97043, Arca Systems, Inc., 23 April 1998.

[211] Yates, J. Frank, and Michael D. Tschirhart. “Decision-Making Expertise.” p. 421-438 in [47]

[212] Yee, Ka-Ping , “User interaction design for secure systems,” In Proceedings of the 4th International Conference on Information and Communications Security, Springer-Verlag, LNCS 2513, 2002

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.

ICS 35.080

Price based on 95 pages

© ISO/IEC 2010 – All rights reserved

Authorized licensed use limited to: Politechnika Gdanska. Downloaded on August 31,2011 at 10:16:16 UTC from IEEE Xplore. Restrictions apply.