ISO, OMG, Common Criteria and the Content Automation Efforts…

Robert A. Martin

DHS's Build Security In and SwA Websites

www.us-cert.gov/swa/

buildsecurityin.us-cert.gov/swa/

Homeland Security

Software Assurance

https://buildsecurityin.us-cert.gov/swa/

Software Assurance

Community Resources and Information Clearinghouse

Sponsored by DHS National Cyber Security Division

HOME | PEOPLE | PROCESS | TECHNOLOGY | ACQUISITION | WORKING GROUPS

- What Is Software Assurance?
- Events
- Resources

Working Groups

- Workforce Education & Training
- Processes & Practices
- Technology, Tools & Product Eval.
- Acquisition & Outsourcing
- Measurement
- Business Case
- Malware
- Join a Working Group

US-CERT Software Assurance

Build Security In

Resources to help you build security into your systems in every phase of development.

What is Software Assurance?

Software assurance (SwA) is the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that the software functions in the intended manner (from CNSS 4009 IA Glossary - see Wikipedia for definitions and descriptions).

As part of the DHS risk mitigation effort, the Software Assurance Program seeks to reduce software vulnerabilities, minimize exploitation, and address ways to improve the routine development of trustworthy software products with predictable execution, and to improve diagnostic capabilities to analyze systems for hidden vulnerabilities.

FOCUS AREAS

The SwA framework encourages the production, evaluation, and acquisition of better quality and more secure software, providing focuses in these four areas:

- People: Education and training for developers and users
- Process: Sound practices, standards, and practical guidelines for the development of secure software
- Technology: Diagnostic tools, cyber security R&D and measurement
- Acquisition: Specifications and guidelines for acquisition and outsourcing

The Software Assurance Forum and several working groups composed of volunteers from government, industry, and academia are helping the Software Assurance Program achieve its objectives in these focus areas.

WHY IS SOFTWARE ASSURANCE CRITICAL?

The nation's critical infrastructure (energy, transportation, telecommunications, etc.), businesses, and services are extensively and increasingly controlled and enabled by software. Vulnerabilities in that software put those resources at risk. The risk is compounded by software size and complexity, the use of software produced by unvetted suppliers, and the interdependence of software systems. Software assurance deals with the root of the problem by improving software security.

HOW IS SOFTWARE ASSURANCE ADVANCING?

The Software Assurance Forum has provided a collaborative venue for stakeholders to share and advance techniques and technologies relevant to software security. Software Security Assurance: A State-of-the-Art Report (SOAR) represents an output of collaborative efforts of organizations and individuals in the SwA Forum and working groups. The SOAR provides an overview of the current state of the environment in which software must operate and surveys current and emerging activities and organizations involved in promoting various aspects of software security assurance. The report also describes the variety of techniques and technologies in use in government, industry, and academia for specifying, acquiring, producing, assessing, and deploying software that can, with a justifiable degree of confidence, be said to be secure. The report also presents observations about noteworthy trends in software security assurance as a discipline. Many other SwA resources are provided by the SwA working groups.

Contact Us Terms of Use Privacy Policy ©2007 Department of Homeland Security

DHS SwA WG Output

https://buildsecurityin.us-cert.gov/daisy/bsi/resources/dhs.html?branch=1&language=1

Build Security In

Setting a Higher Standard for Software Assurance

Sponsored by DHS National Cyber Security Division

All Sites | BSI Home

Search

User: guest Query Search Recent Changes Page Actions Variants (1)

- Home
- Mission
- Articles [by Content Area]
- DHS SwA Web Site
- Secure Coding Sites
- Additional Resources
- **DHS SwA WG Output**
  - IT Security Essential Body of Knowledge
  - Seven Objectives of Software Assurance Sound Practices
  - SOAR on Software Security Assurance
  - Software Assurance (SwA) in Acquisition: Mitigating Risks to the Enterprise
  - Software Assurance CBK/Principles Organization
  - Software Assurance Program Working Groups
  - Software Project Management for Software Assurance: DACS State of the Art Report
- Published Articles & Books
- Web Articles
- Web Sites
- Events
- About Us
- FAQs

DHS SwA WG Output

These are resources that are either part of other DHS-sponsored work or closely related thereto.

Supplementary Department of Homeland Security Resources [Ordered by Date]

- Software Assurance CBK/Principles Organization [10/30/07]
  The Department of Homeland Security (DHS) Software Assurance Program is seeking review and comment on Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software (version 1.2) and the associated SwA CBK-Principles Matrix. The CBK-Principles-SANS SSI Matrix and Principles Organization: Towards an Organization for Software System Security Principles and Guidelines (version .26a) are also available for reference and review. Please use the comment form for them and submit comments by December 4, 2007 to Kristy Mosteller and Samuel Redwine.
- IT Security Essential Body of Knowledge [10/25/07]
  The IT Security Essential Body of Knowledge (EBK) further clarifies key IT security terms and concepts for well-defined competencies, identifies notional security roles, defines four primary functional perspectives, and establishes an IT Security Role, Competency, and Functional Matrix. We seek continued input from the community to ensure that it accurately reflects baseline skill requirements of the nation's IT security workforce.
- Software Assurance (SwA) in Acquisition: Mitigating Risks to the Enterprise [10/2/07]
  This guide provides information on incorporating software assurance throughout the acquisition process from the acquisition planning phase to contracting, implementation and acceptance, and follow-on phases. For each phase, the guide covers software assurance concepts, recommended strategies, and acquisition management tips. The guide also includes recommended Request for Proposals (RFP) and contract language and due diligence questionnaires that may be tailored by acquisition officials to facilitate the contract evaluation process.
- Software Project Management for Software Assurance: DACS State of the Art Report [9/28/07]
  The purpose of this report is to collect and present information on how the need for software assurance affects software project management. The impact of software assurance on the tasks and concerns of a project manager are addressed, first in terms of discipline, and then in terms of life cycle. A second goal of this SOAR is to provide tools and resources for quantifying the effects of software assurance on software development, both in terms of planning (cost estimation and budgeting) and in terms of overall cost-effectiveness and return on investment.
- SOAR on Software Security Assurance [8/3/07]
  The Software Assurance Forum has provided a collaborative venue for stakeholders to share and advance techniques and technologies relevant to software security. The state-of-the-art report (SOAR) published by the Information Assurance Technology Analysis Center (IATAC) at http://iac.dtic.mil/iatac/download/security.pdf represents an output of collaborative efforts of organizations and individuals in the SwA Forum and Working Groups. The SOAR provides an overview of the current state of the environment in which software must operate and surveys current and emerging activities and organizations involved in promoting various aspects of software security

Software Assurance

https://buildsecurityin.us-cert.gov/swa/technology.html

Software Assurance

Community Resources and Information Clearinghouse

Sponsored by DHS National Cyber Security Division

HOME | PEOPLE | PROCESS | TECHNOLOGY | ACQUISITION | WORKING GROUPS

- Join a Working Group
- US-CERT Software Assurance

Build Security In

Resources to help you build security into your systems in every phase of development.

Focus Area: Technology

In this section you'll find resources relating to software security measurement, SwA research and development, and SwA testing and diagnostic tools.

NIST SAMATE Reference Dataset: Software development artifacts with known security weaknesses

Common Weakness Enumeration: A dictionary of common software weaknesses

Common Attack Patterns Enumeration and Classification: A taxonomy of attack patterns

Security Measurement: A white paper published by Practical Software and Systems Measurement

Federal Plan for Cyber Security and Information Assurance Research and Development: Available for download on the National Coordination Office for Networking and Information Technology Research and Development site.

The Data & Analysis Center for Software

RELATED WORKING GROUPS

Efforts of these working groups fall within this focus area:

- Technology, Tools and Product Evaluation Working Group
- Measurement Working Group
- Malware Working Group

Contact Us Terms of Use Privacy Policy ©2007 Department of Homeland Security

Overview

- **ISO/IEC JTC 1/SC 22/WG 23, ISO 24772 Programming Language Vulnerabilities**
- **ISO/IEC JTC 1/SC 27/WG 3, NWP Common Criteria TOE “update”**
- **SC 27 WG 1, ISO 15026**
- **OMG Systems Assurance Task Force**

ISO/IEC JTC 1/SC 22/WG 23, ISO 24772 Programming Language Vulnerabilities

The Problem:

- Any programming language has constructs that are imperfectly defined, implementation dependent or difficult to use correctly.
- As a result, software programs sometimes execute differently than intended by the writer.
- In some cases, these weaknesses can be exploited by hostile parties, or can lead to failure in anticipated environments.
  - **Can compromise safety, security, privacy, dependability or other critical properties.**
  - **A vulnerability in any program can be used as a springboard to make additional attacks on other programs.**

ISO/IEC JTC 1/SC 22/WG 23, ISO 24772 Programming Language Vulnerabilities

Vulnerability Template:

- The major portion of Technical Report describes vulnerabilities in a generic manner, including:
  - **Brief description of application vulnerability**
  - **Cross-reference to enumerations and other classifications, e.g. CWE**
  - **Description of failure mechanism, i.e. how coding problem relates to application vulnerability**
  - **Applicable language characteristics**
  - **Avoiding or mitigating the vulnerability**
  - **Implications for standardization**
- Annexes will provide language-specific treatments of each vulnerability.

Example Description

6.17 Boundary Beginning Violation [XYX]

6.17.1 Description of application vulnerability

A buffer underwrite condition occurs when an array is indexed outside its lower bounds, or pointer arithmetic results in an access to storage that occurs before the beginning of the intended object.

6.17.2 Cross reference

[Cross references to CWE, JSF, MISRA, CERT, etc.]

Continued…

6.17.3 Mechanism of failure

There are several kinds of failures (in some cases an exception may be raised if the accessed location is outside of some permitted range):

- **A read access will return a value that has no relationship to the intended value, e.g., the value of another variable or uninitialized storage.**
- **An out-of-bounds read access may be used to obtain information that is intended to be confidential.**
- **A write access will not result in the intended value being updated and may result in the value of an unrelated object (that happens to exist at the given storage location) being modified.**
- **When the array has been allocated storage on the stack an out-of-bounds write access may modify internal runtime housekeeping information (e.g., a functions return address) which might change a programs control flow.**

Continued…

6.17.4 Applicable language characteristics

This vulnerability description is intended to be applicable to languages with the following characteristics:

- **Languages that do not detect and prevent an array being accessed outside of its declared bounds.**
- **Languages that do not automatically allocate storage when accessing an array element for which storage has not already been allocated.**

Continued…

6.17.5 Avoiding the vulnerability or mitigating its effects

Software developers can avoid the vulnerability or mitigate its ill effects in the following ways:.

- **Use of implementation provided functionality to automatically check array element accesses and prevent out-of-bounds accesses.**
- **Use of static analysis to verify that all array accesses are within the permitted bounds. Such analysis may require that source code contain certain kinds of information, e.g., that the bounds of all declared arrays be explicitly specified, or that pre- and post-conditions be specified.**
- **Sanity checks should be performed on all calculated expressions used as an array index or for pointer arithmetic.**

Some guideline documents recommend only using variables having an unsigned type when indexing an array, on the basis that an unsigned type can never be negative. This recommendation simply converts an indexing underflow to an indexing overflow because the value of the variable will wrap to a large positive value rather than a negative one. Also some language support arrays whose lower bound is greater than zero, so an index can be positive and be less than the lower bound.

In the past the implementation of array bound checking has sometimes incurred what has been considered to be a high runtime overhead (often because unnecessary checks were performed). It is now practical for translators to perform sophisticated analysis that significantly reduces the runtime overhead (because runtime checks are only made when it cannot be shown statically that no bound violations can occur).

Continued…

6.17.6 Implications for standardization

- **Languages that use pointer types should consider specifying a standard for a pointer type that would enable array bounds checking, if such a pointer is not already in the standard.**

6.17.7 Bibliography

[None]

WG 23 Participants

- National Bodies
  - Canada
  - Germany
  - Italy
  - Japan
  - France
  - United Kingdom
  - USA
- Other Groups
  - RT/SC Java
  - MISRA C/C++
  - CERT
- Language Standards Groups
  - SC 22/WG 9
  - SC 22/WG14
  - SC 22/WG 5, INCITS J3 (Fortran)
  - SC 22/WG 4, INCITS J4 (Cobol)
  - MDC (Mumps)
  - ECMA (C#, C++CLI)

Overview

- **ISO/IEC JTC 1/SC 22/WG 23, ISO 24772 Programming Language Vulnerabilities**
- **ISO/IEC JTC 1/SC 27/WG 3, NWP Common Criteria TOE “update”**
- **SC 27 WG 1, ISO 15026**
- **OMG Systems Assurance Task Force**

ISO/IEC JTC 1/SC 27/WG 3, NWP
“Secure software development and evaluation under ISO/IEC 15408 and ISO/IEC 18405”

- The way how the CAPEC and related CWE taxonomies are to be used by the developer, which needs to consider and provide sufficient and effective mitigation to all applicable attacks and weaknesses.

- The way how the CAPEC and related CWE taxonomies are to be used by the evaluator, which needs to consider all the applicable attack patterns and be able to exploit all the related software weaknesses while performing the subsequent AVA_VAN activities.

- How incomplete entries from the CAPEC are to be addressed during an evaluation.

- How to incorporate to the evaluation attacks and weaknesses not included in the CAPEC.

New Work Item Proposal

NP submitting

PROPOSAL FOR A NEW WORK ITEM

Date of presentation of proposal: YYYY-MM-DD	Prop
Secretariat: National Body	**ISO/** ISO/

A proposal for a new work item shall be submitted to the secretari committee concerned with a copy to the ISO Central Secretariat.

Presentation of the proposal

Title Secure software development and evaluation under ISO/IEC 1

Scope

In the case where a target of evaluation (TOE) being evaluated, und 18405, includes specific software portions, the TOE developer may technical rationale for mitigating software common attack patterns a in the latest revision of the Common Attack Pattern Enumeration an from http://capec.mitre.org/. The developer's technical rationale is mitigation techniques, from architectural properties to design feature other means.

This Technical Report (TR) provides guidance for the developer and CAPEC as a technical reference point during the TOE development TOE secure software under ISO/IEC 15408 and 18045, by addressi

a) A refinement of the IS 15408 Attack Potential calculation tab the entries contained in the CAPEC and their characterizatio

b) How the information for mitigating software common attack used in an IS 15408 evaluation, in particular providing guida attack patterns and weaknesses are applicable to the TOE,

   1. the TOE technology;
   2. the TOE security problem definition;
   3. the interfaces the TOE exports that can be used by
   4. the Attack Potential that the TOE needs to provide

c) How the technical rationale provided by the developer for m patterns and related weaknesses is used in the evaluation o development of test cases.

d) How the CAPEC and related Common Weakness Enumerat the evaluator, who needs to consider all the applicable attac specific related software weaknesses while performing the s (AVA_VAN) activities on the TOE.

e) How incomplete entries from the CAPEC are resolved durin

f) How the evaluator's attack and weakness analysis of the TO weaknesses not yet documented in the CAPEC.

The TR also investigates specific elements from the ISO /IEC 15026 the guidelines being developed in the TR within the context of IS 15

Determining attack potential for current CAPEC attacks

Having assigned in the above the corresponding attack potential contribution numeric values wrt all the attack potential factors, summarized as follows, the CEM B.4.2.2 Annex section Table 4 "Rating of vulnerability and TOE resistance" is ready to determine the attack potential for the CAPEC attacks that are associated with specific CWE weakness(es).

Attack potential factor	Value
The CAPEC attack "**elapsed time**" is deemed as "less than one day"	0
The CAPEC schema description "high" for its "Attacker Skill or Knowledge Required" at most is mapped only to the CEM B.4.2.2 Annex section "proficient persons" for its "**specialist expertise**" factor	3
The CAPEC schema description "high" for its "Attacker Skill or Knowledge Required" at most is mapped only to the CEM B.4.2.2 Annex section "restricted information concerning the TOE" for its "**knowledge of the TOE**" factor	3
For those CAPEC attacks having related CWE weakness(es), their "**windows of opportunity**" is deemed as "unnecessary/unlimited access"	0
The CAPEC schema description "Resources Required" at most is mapped only to the CEM B.4.2.2 Annex section "standard equipment" for its "**IT hardware/software or other equipment**" factor	0
Total	6 (which is the sum of the above)

Since the total value due to all the attack potential factors is 6, the CEM B.4.2.2 Annex section Table 4 indicates that the attack potential for the CAPEC attacks that are associated with specific CWE weakness(es) is "basic".

Since an EAL2 TOE must demonstrate resistance to attacks with a "basic" attack potential in accordance with the "Part 3: Security assurance components" of Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 2, any TOE attempting to claim EAL2 or higher must address the CAPEC attacks that are associated with specific CWE weakness(es).

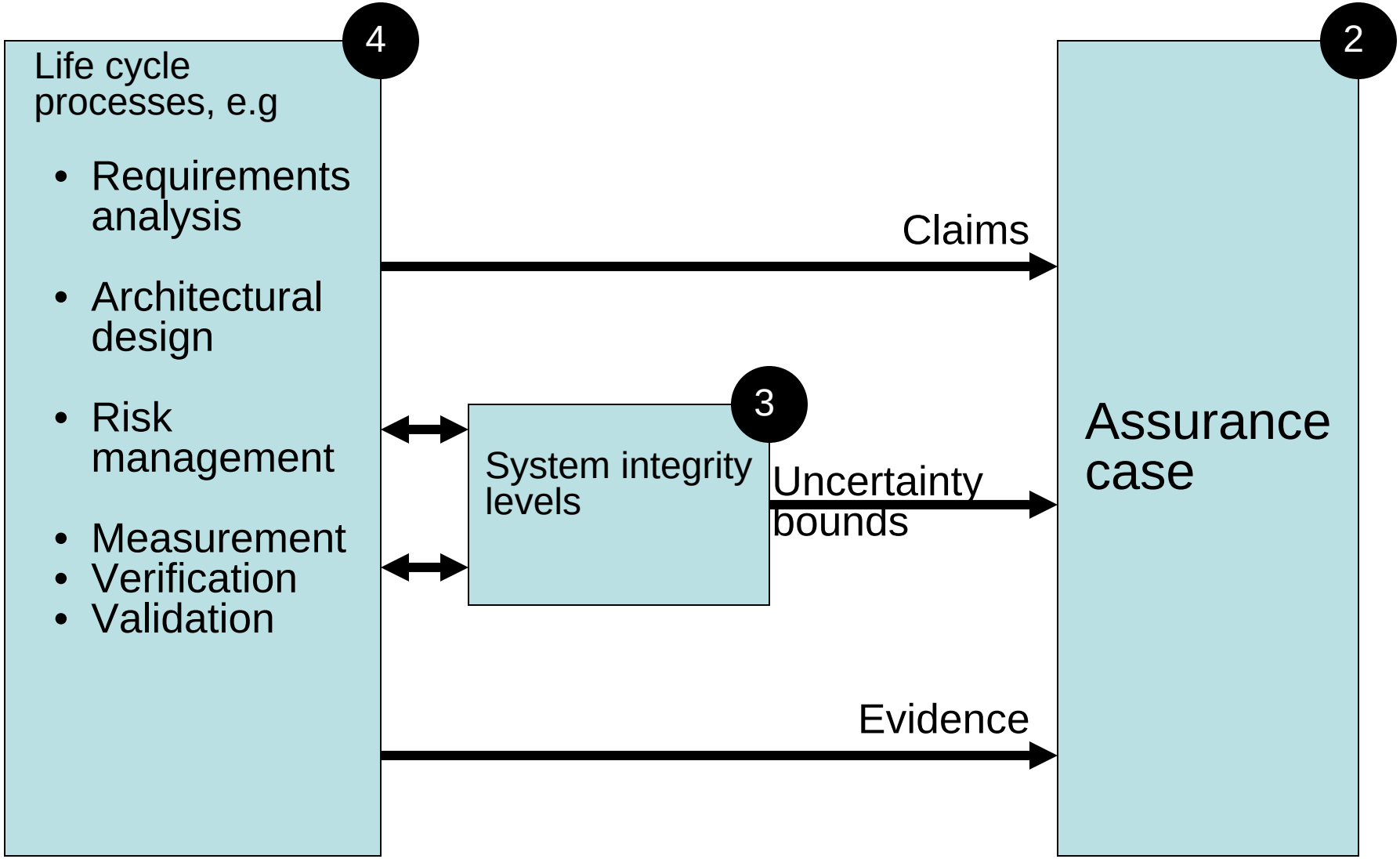
Overview

- **ISO/IEC JTC 1/SC 22/WG 23, ISO 24772 Programming Language Vulnerabilities**
- **ISO/IEC JTC 1/SC 27/WG 3, NWP Common Criteria TOE “update”**
- **SC 27 WG 1, ISO 15026**
- **OMG Systems Assurance Task Force**

ISO/IEC 15026: A Four-Part Standard

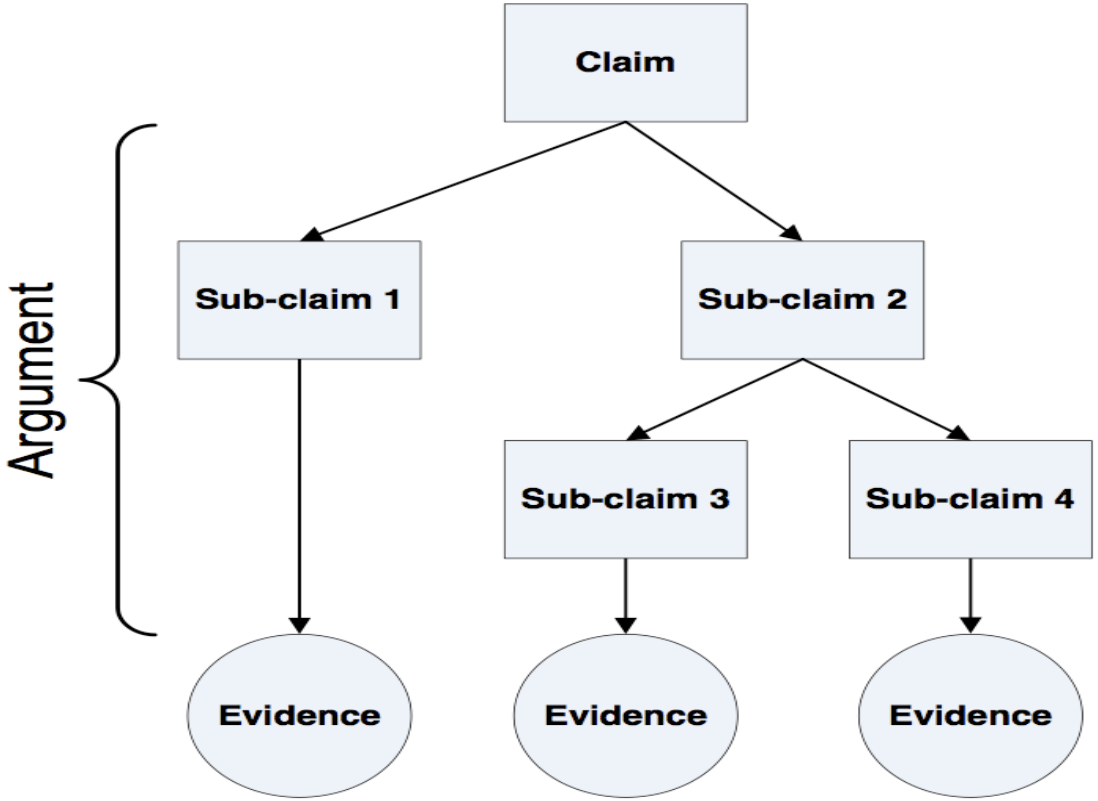
- **Planned parts:**
  - **15026-1: Concepts and vocabulary (initially a TR2 and then revised to be an IS)**
  - **15026-2: Assurance case (including planning for the assurance case itself)**
  - **15026-3: System integrity levels (a revision of the 1998 standard)**
  - **15026-4: Assurance in the life cycle (including project planning for assurance considerations)**
- **Possible additional parts as demand requires and resources permit, e.g.**
  - **Assurance analyses and techniques**
  - **Guidance documents**

ISO/IEC 15026: Examples of relationships among parts



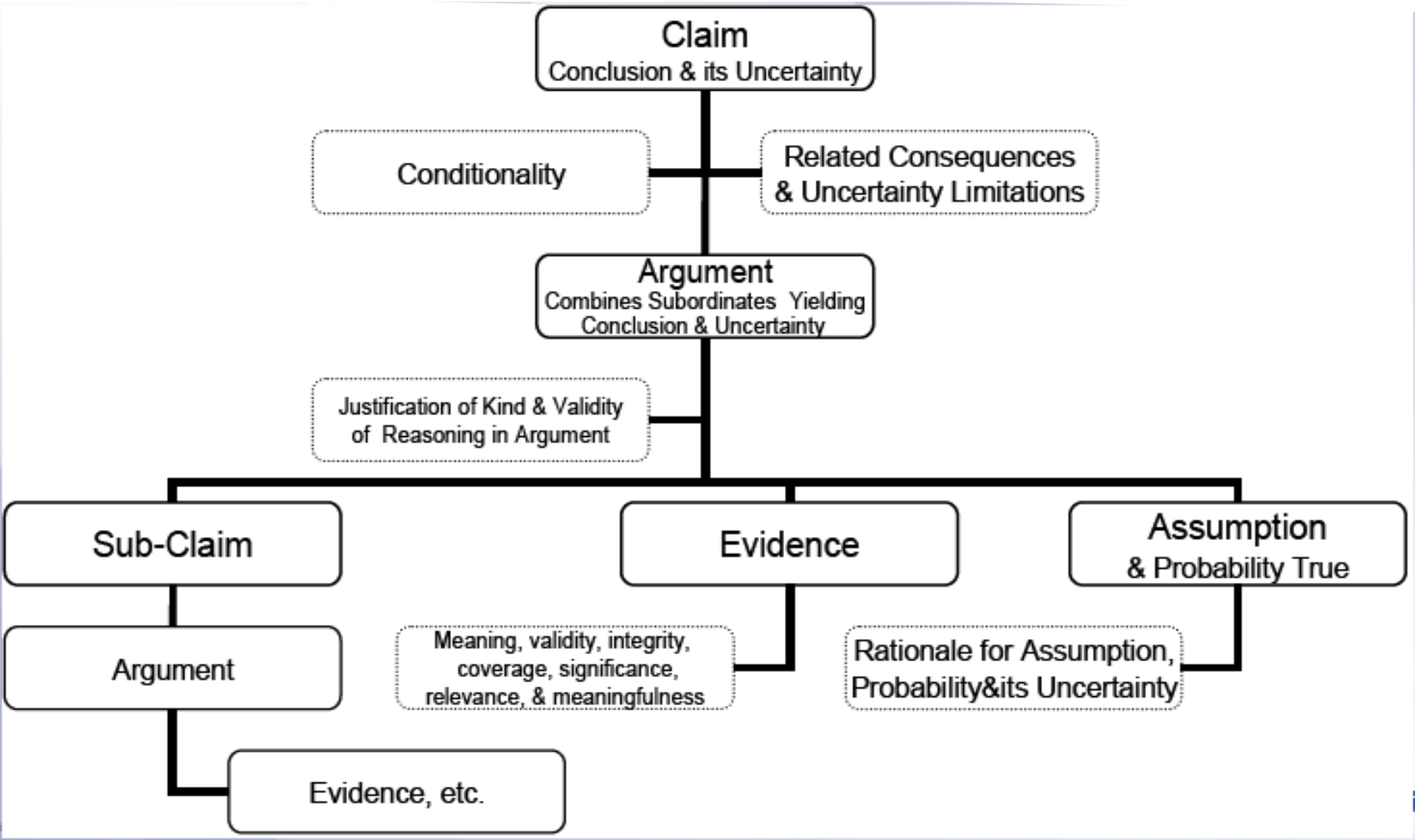
ISO/IEC 15026: Systems & Software Assurance

15026 Part 2: The Assurance Case (Claims-Evidence-Argument)



ISO/IEC 15026: Systems & Software Assurance

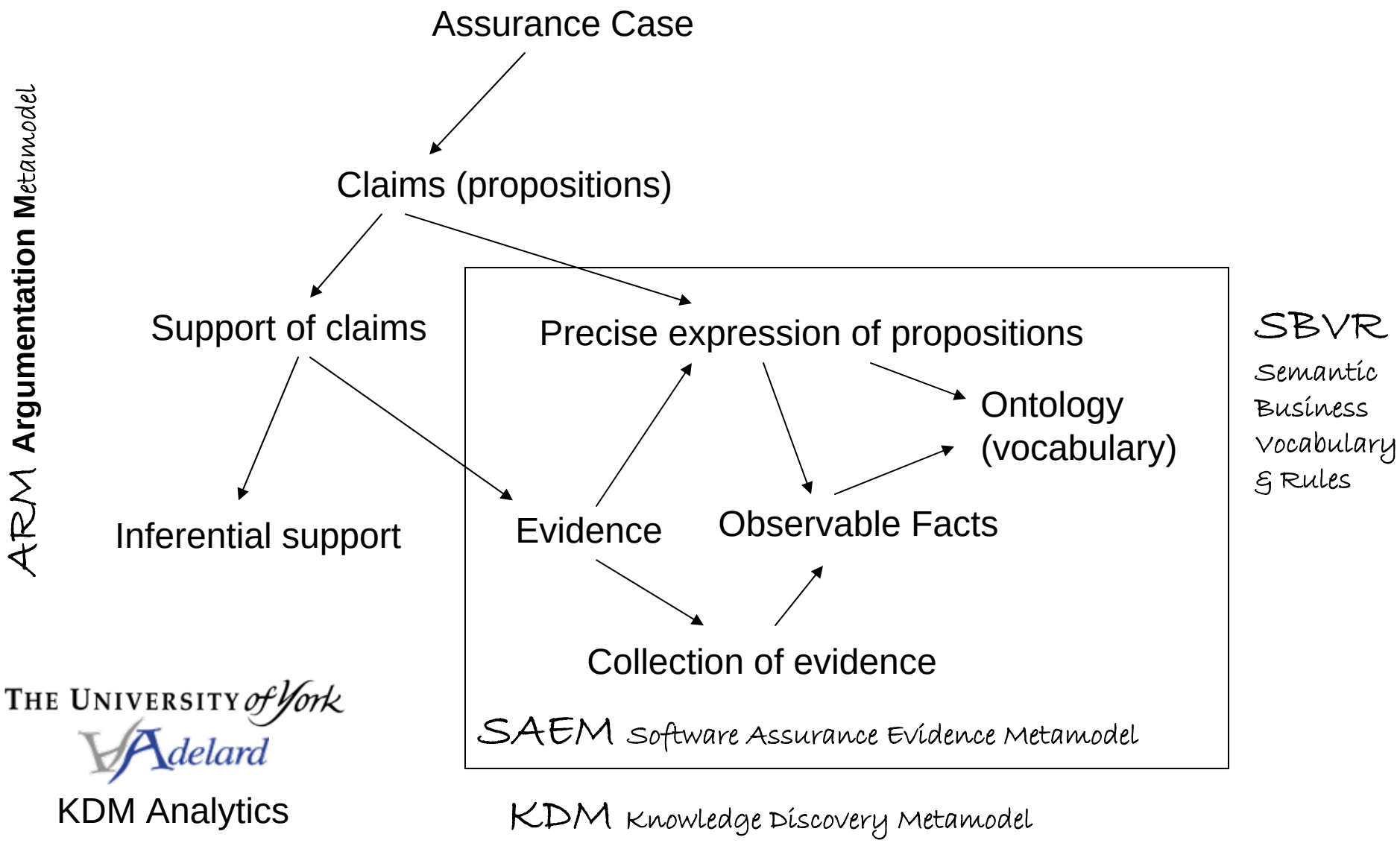
15026 Part 2: The Assurance Case (Claims-Evidence-Argument)



Overview

- **ISO/IEC JTC 1/SC 22/WG 23, ISO 24772 Programming Language Vulnerabilities**
- **ISO/IEC JTC 1/SC 27/WG 3, NWP Common Criteria TOE “update”**
- **SC 27 WG 1, ISO 15026**
- **OMG Systems Assurance Task Force**

OMG Systems Assurance Task Force
Claims-Evidence-Arguments Overview



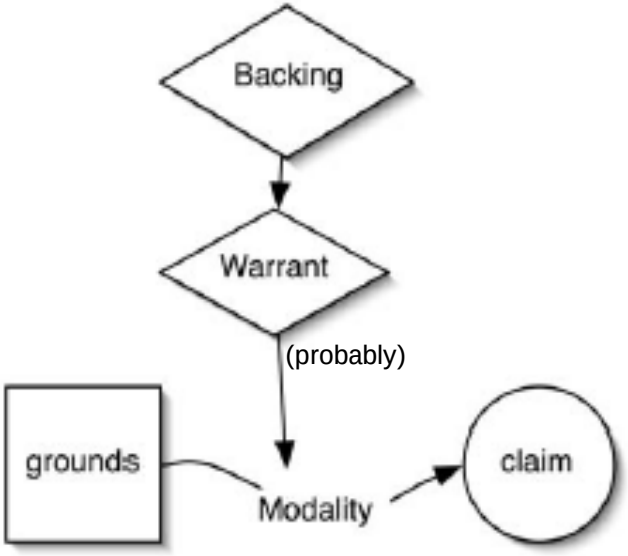
THE UNIVERSITY of York

Adelard

KDM Analytics

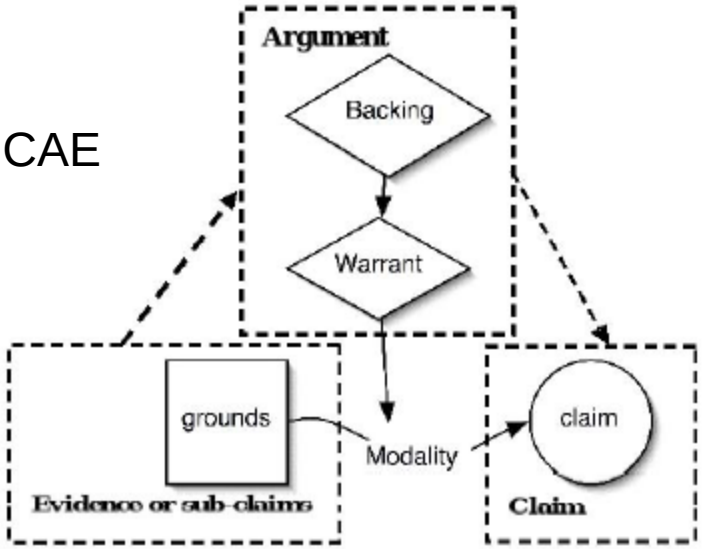
Support by ‘Substantial’ Reasoning

Stephen Toulmin, 1958

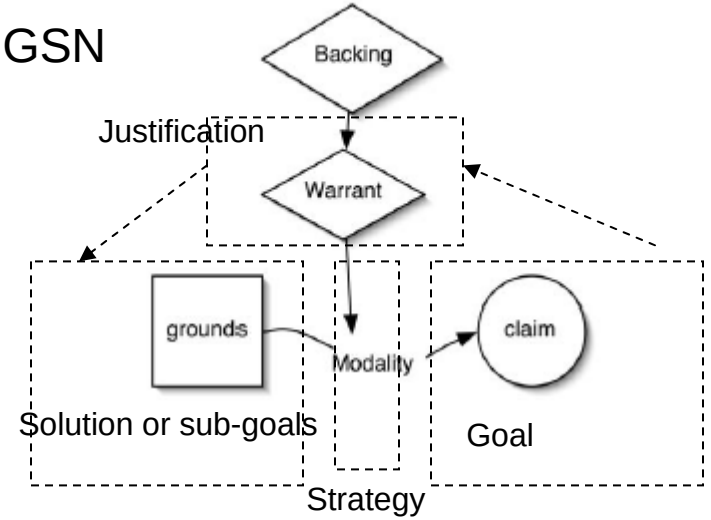


- Claims are assertions put forward for general acceptance
- The justification for claim is based on some grounds, the “specific facts about a precise situation that clarify and make good for a claim”
- The basis of the reasoning from the grounds (the facts) to the claim is articulated. Toulmin coined the term “warrant” for “substantial argument”. These are statements indicating the general ways of argument being applied in a particular case and implicitly relied on and whose trustworthiness is well established”.
- The basis of the warrant might be questioned, so “backing” for the warrant may be introduced. Backing might be the validation of the scientific and engineering laws used

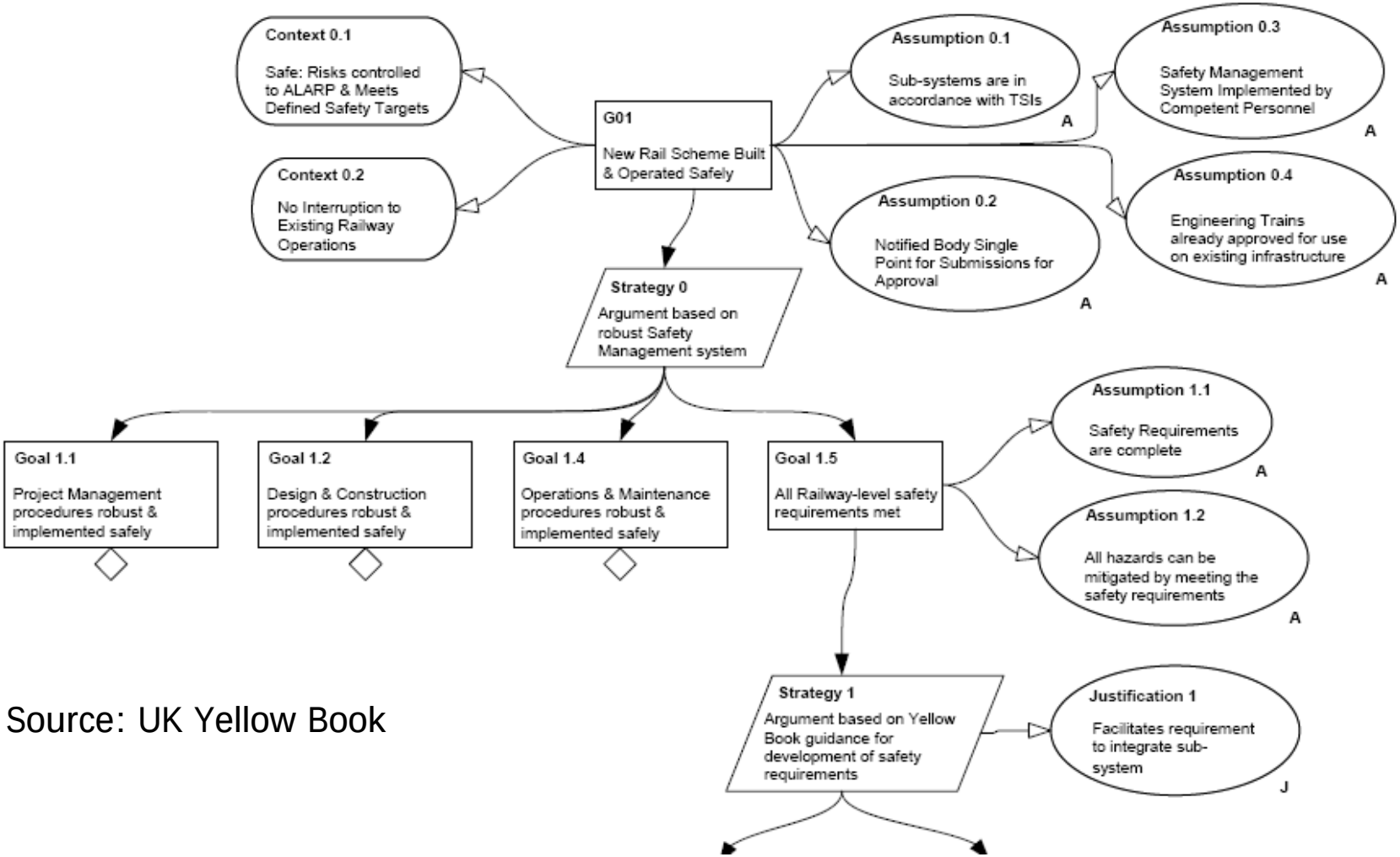
CAE



GSN

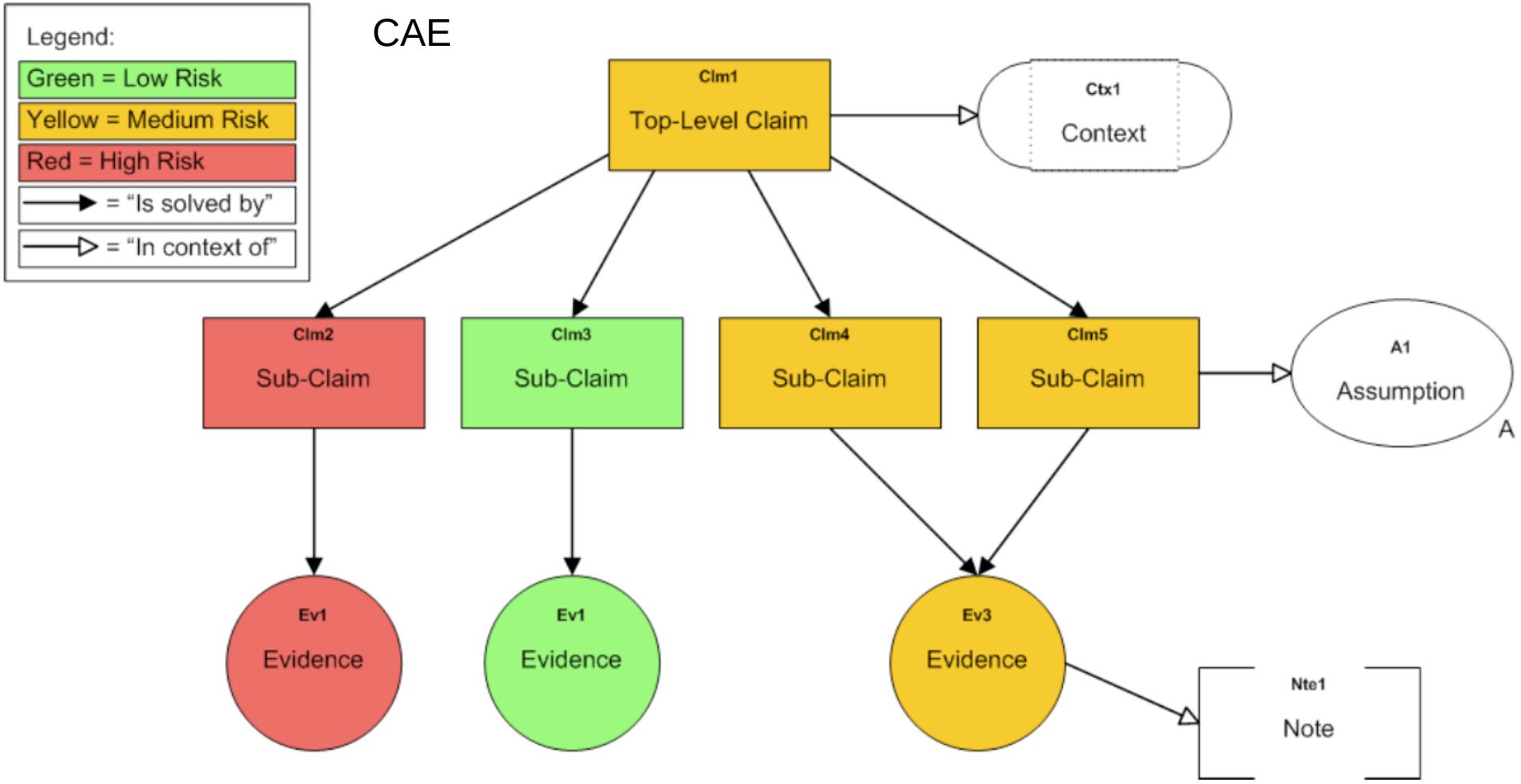


GSN: Safety Case for a Railroad Signalling Scheme

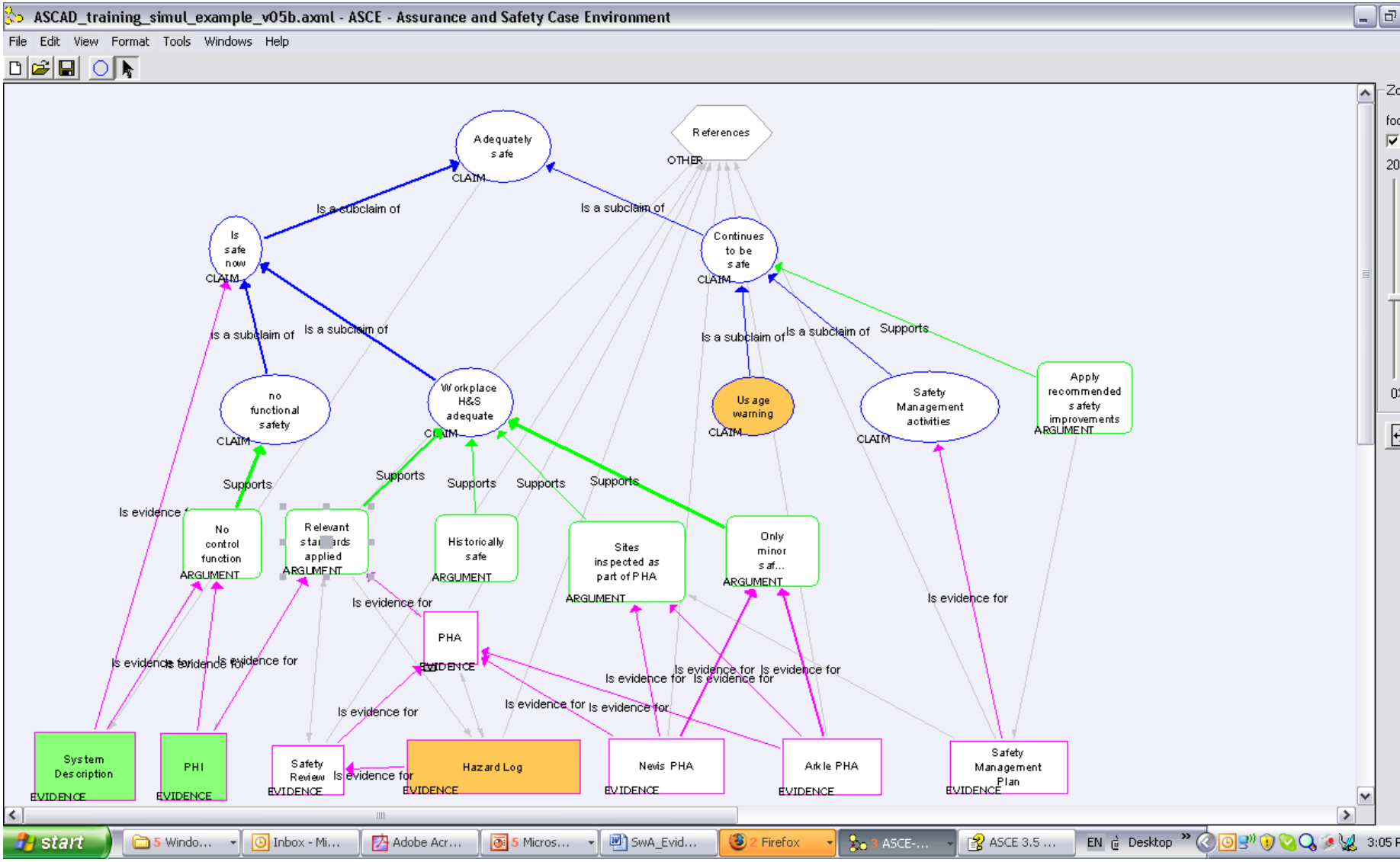


Source: UK Yellow Book

Safety Cases Based on Assurance Cases – Claims-Evidence-Argument in Use for <10 Years



Structured Safety Assurance tools are commercially available

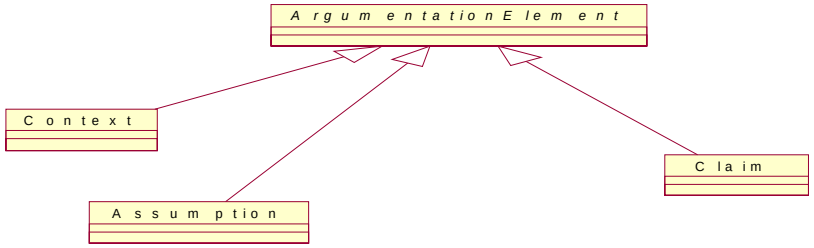


What is Evidence ?

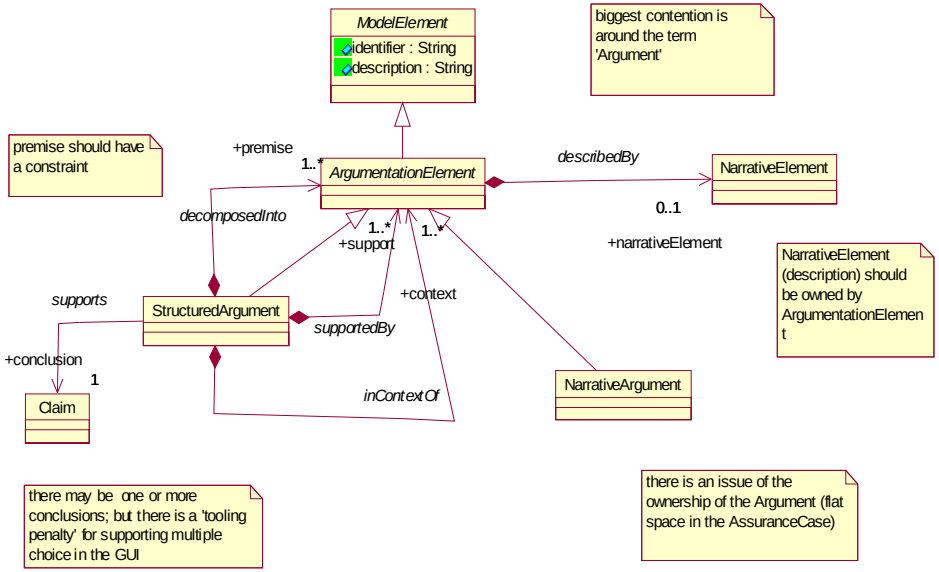
- Evidence is data that supports certain claim
  - Not assumptions, clarification or subclaims
- Evidence can be diverse (various things may be produced as evidence)
  - Documents as evidence
  - Test results as evidence (someone has to make the verdict)
  - Measurement results as evidence
  - Process, product
- Evidence has provenance
  - Source
  - Evidence acquisition involves certain processes (reviews, testing, analysis, etc.)
- Evidence has “quality”
- Evidence is stored in evidence repositories
- Argument structure determines what evidence is acquired
  - Also argument criticality determines evidence “quality”
- Evidence can help partition arguments
  - Evidence may provide context

The Assurance Case/Argument: OMG Evidence and Claims/Arguments Standards

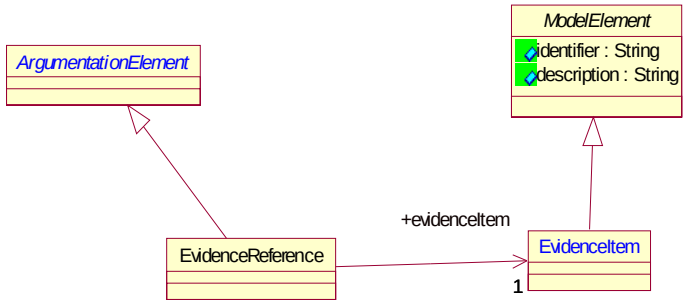
ARM:Claims



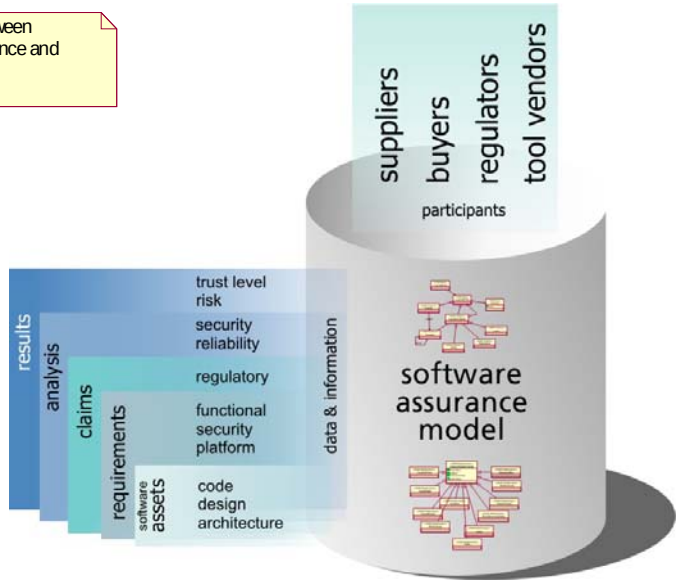
ARM:Arguments



SAEM: Evidence



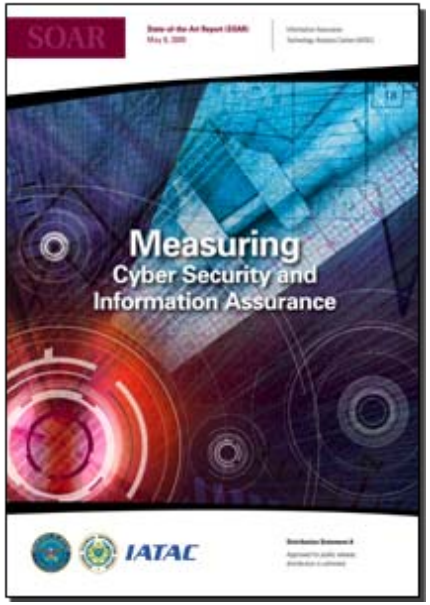
duplication between EvidenceReference and EvidenceItem

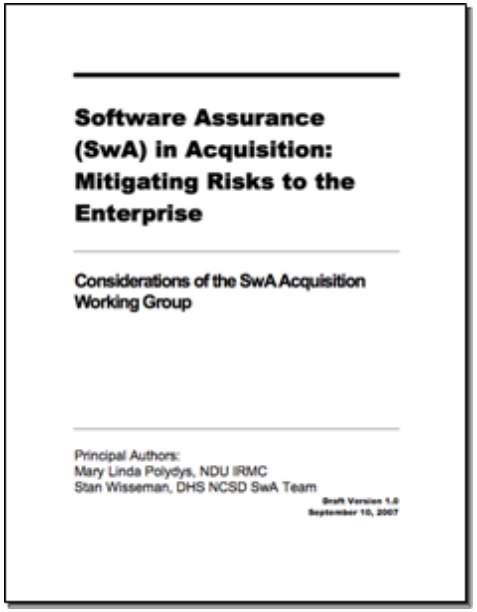


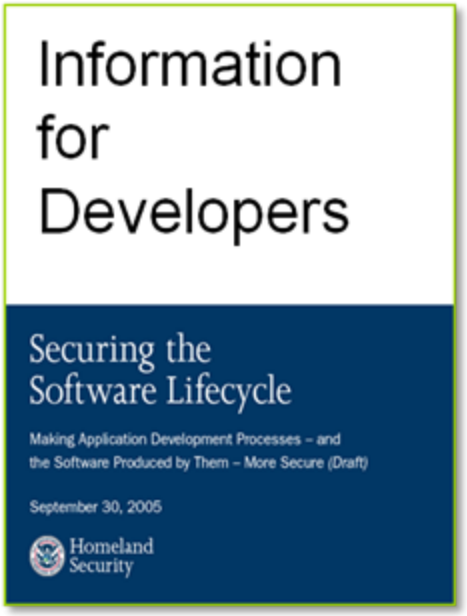
Software Assurance Community: Opportunities, and Items to leverage

CWE™

Build Security In
Setting a Higher Standard for Software Assurance
Sponsored by DHS National Cyber Security Division







CAPEC™

Next SwA Forum:
2-6 Nov 09 – Crystal City Marriott

Questions?