Consortium for Information & Software Quality ™

COUNCIL

Software Certification: Why, How, and Next Steps

Webinar presented December 2, 2020

Speakers

Matthias Haynl
Business Unit Manager - Functional Safety and Cybersecurity for North America, TÜV Rheinland

Dr. Bill Curtis
Executive Director, CISQ

Tracie Berardi
Program Director, CISQ

Karin Athanas
Executive Director, TIC Council

What Is CISQ ?

Carnegie Mellon Software Engineering Institute

OBJECT MANAGEMENT GROUP

Co-founders

Dr. Paul Nielsen, CEO → **CISQ** ← **Dr. Richard Soley, CEO**

↓

Develop specifications for automatable measures of software systems

↓

Gain approval as OMG standards

↓

Fast-track to ISO

CISQ Sponsors

Tech Mahindra, Synopsys, CGI, 7N, SHPI, USC CSSE, CAST, Northrop Grumman

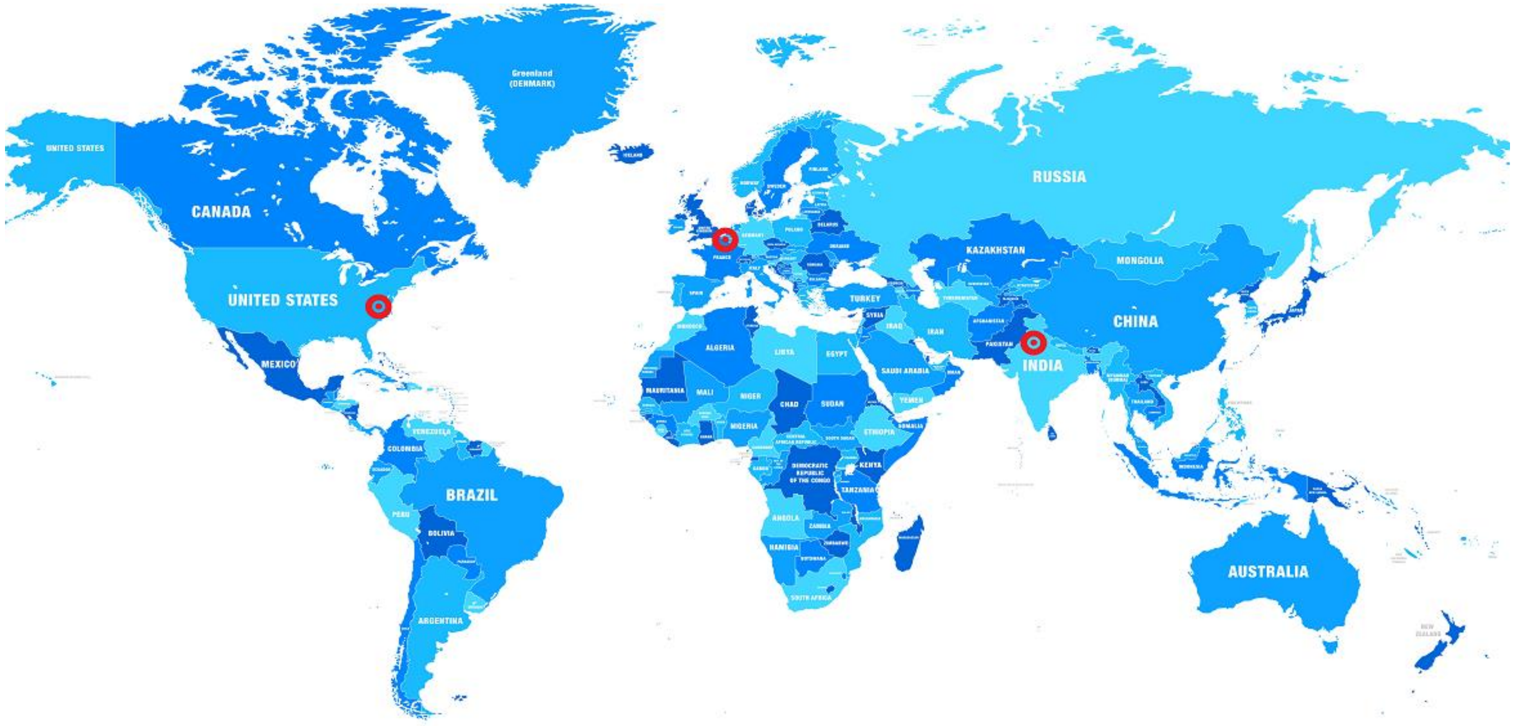
CISQ Partners

Gartner, QuEST Forum, Forrester, IAOP, USCC, TechWell, ITAAC, AFCEA Washington, DC Chapter, ISG, Making Software Better, QA

TIC Council

The Independent Voice of Trust

- Born from the merger of IFIA and CEOC
- ~90-member companies & organizations active in more than 160 countries (HQ mapped)
- TIC Council has its head office in Brussels. It also has an office in Washington and presence in India.



TIC Council Mission

As the voice of the global independent testing, inspection and certification industry, the TIC Council engages governments and key stakeholders to advocate for effective solutions that protect the public, support innovation and facilitate trade.

The TIC Council works with its members to promote best practices in safety, quality, health, ethics and sustainability

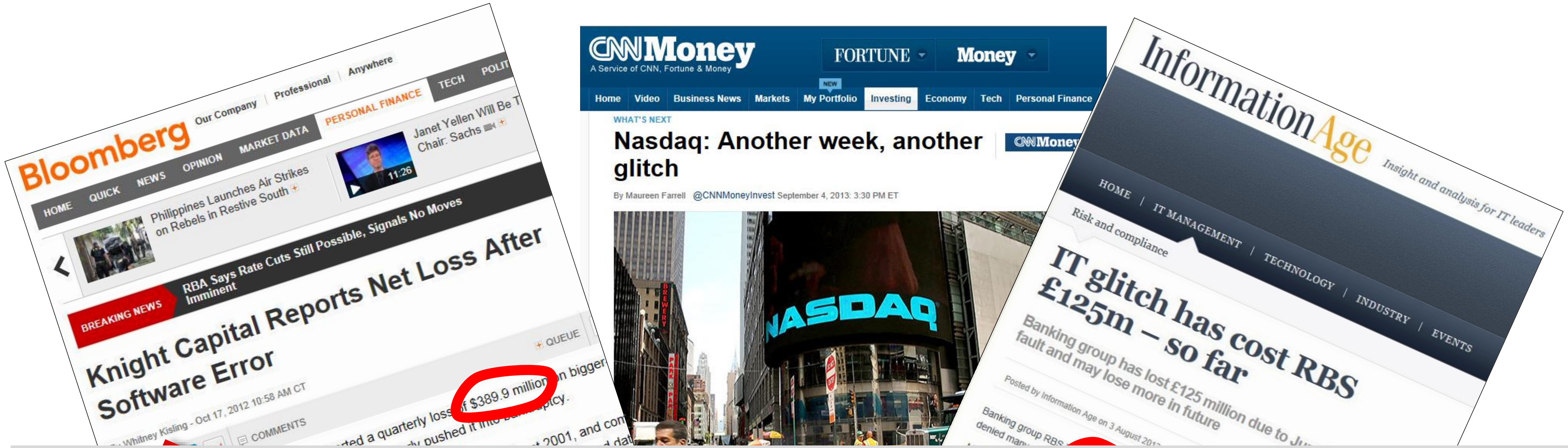
The Compelling Need for Software Certification

Dr. Bill Curtis
Executive Director

CISQ

Consortium for
Information and
Software Quality

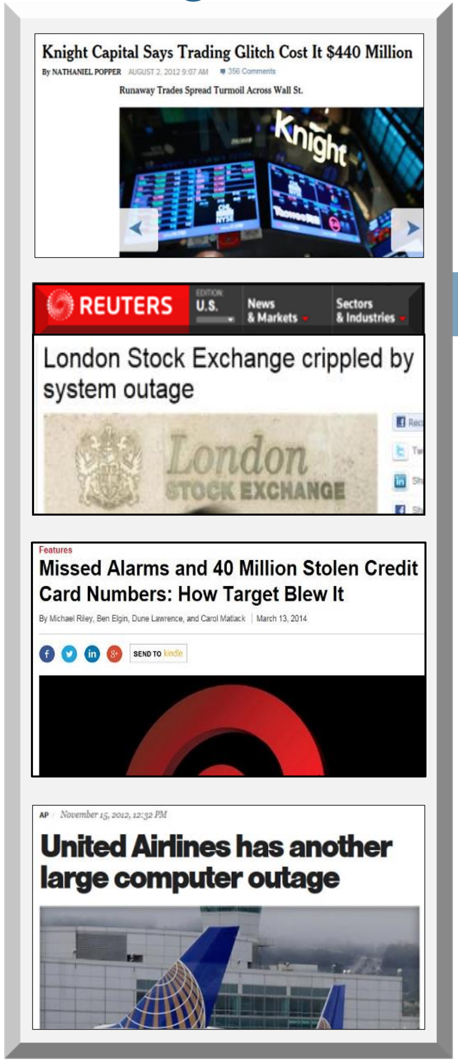
The Era of Nine-Digit Glitches



No person's assets are safe while Wall Street is in session!

Where Is the Accountability?

Nine Digit Glitches



now affect

Board of Directors

CEO, COO, CFO

Business VPs

Corporate Auditors

CIO

accountable for

Governance

Risk management

Business Continuity

Brand protection

Customer experience

Need evidence of governance and action against application risk

Software certification

IEC 61508 (Functional Safety)

IEC 62443 (Cyber Security)

Matthias Haynl

Business Unit Manager – Functional Safety and Cyber Security (OT)

TUV Rheinland of North America, Inc.

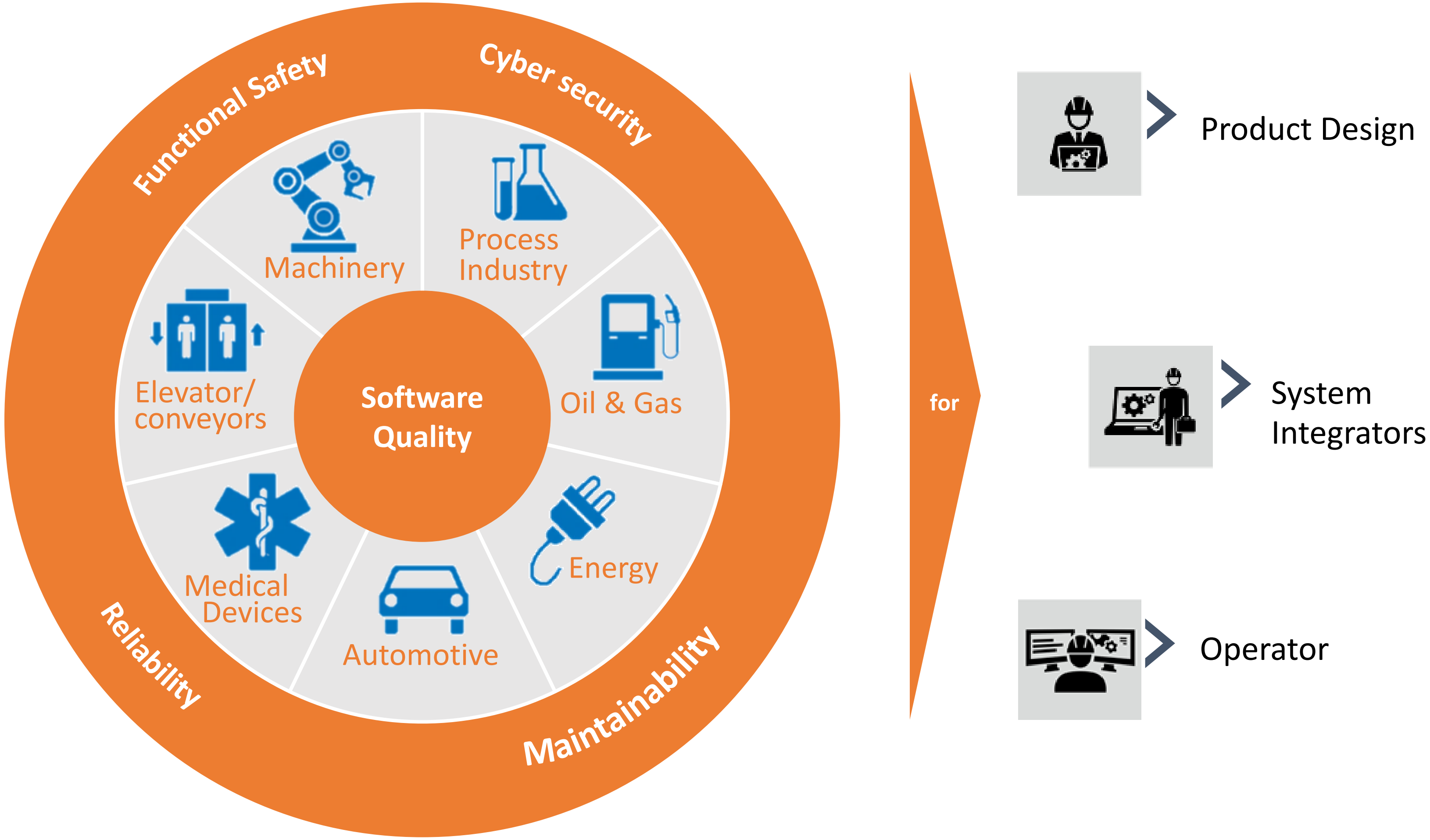
Office: (925) 249-9123 x 2107

Mobile: (925) 353-6047

Content

1. Introduction FS versus CySec objectives
2. Software covered by IEC 61508-3 (e.g. tools, embedded firmware)
3. Examples of techniques and measures to consider during the design
4. Challenges for AI – used in safety context?
5. Level of independency – When is an independent organization needed

Introduction



Software certification evaluates the reliability and safety of software systems or element by an independent organisations

Functional Safety and Cyber Security

Cyber Security

Defence against negligent and wilful actions to protect devices and facilities
IEC 62443.

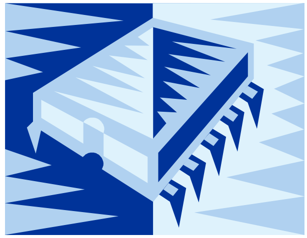
Functional Safety

Defence against random and systematic technical failure to protect life and environment
IEC 61508. Software has only systematic failures.

Software covered by IEC 61508-3

Product specific software:

- operating systems
- application software
- firmware
- …



Tools:

- compiler
- design tools
- test tools
- configuration management
- code generators
- requirement management
- libraries
- …

Off-line support tools classes

- IEC 61508-4, 3.2.11

T1 generates no outputs which can directly or indirectly contribute to the executable code (including data) of the safety related system

examples: text editor, configuration control tools

T2 supports the test or verification of the design or executable code, and cannot directly create errors in the executable software

examples: test coverage measurement tool, static analysis tool

T3 generates outputs which can directly or indirectly contribute to the executable code of the safety related system

examples: compiler

Adequate off-line support tools and their classes need to be defined and documented.
Tools certification is possible

Requirements for off-line support tools

- IEC 61508-3, 7.4.4

Requirement	Class		
	T1	T2	T3
Training	X	X	X
Specification / product manual		X	X
Definition of constraints		X	X
Assessment		X	X
Qualification of new version	X	X	X
Assessment against standard			(X)
Conformance to its specification		O	X

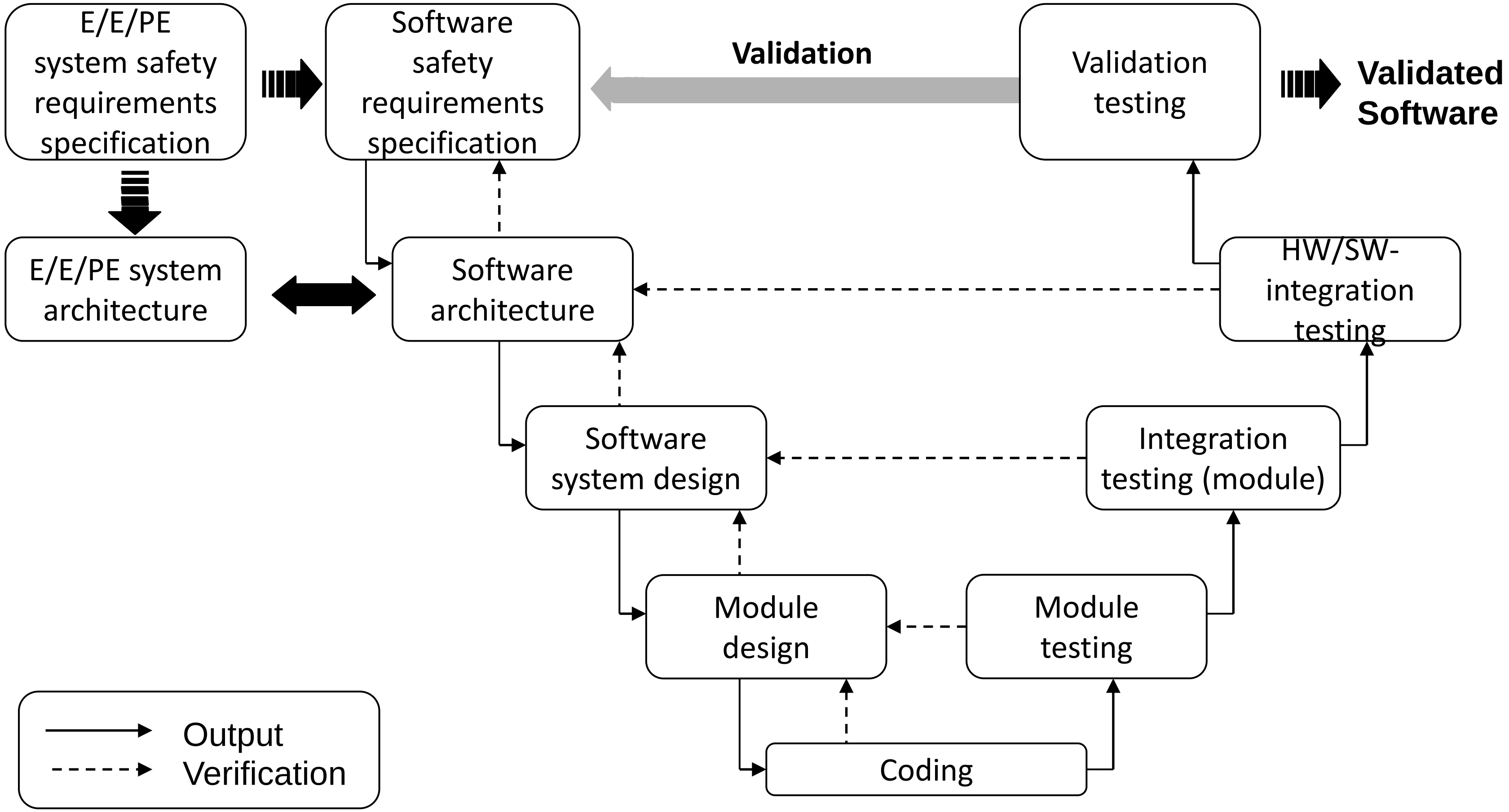
How ?

X : must
O : can
(X) : where appropriate

The requirements for off-line support tools depend on the class.

Development lifecycle (the V-model)

Functional safety Management:
- Safety planning (resources and responsibilities)
- Modification management
- Bug management
- …



Software certification covers a range of formal, semi-formal and informal techniques and measures (e.g. requirement tracking, simulation, testing, code reviews, documentation, etc.).

Techniques for Design and Development

- IEC 61508-3, Tabelle A.3

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Suitable programming language	C.4.5	HR	HR	HR	HR
2	Strongly typed programming language	C.4.1	HR	HR	HR	HR
3	Language subset	C.4.2	---	---	HR	HR
4a	Certified tools and certified translators	C.4.3	R	HR	HR	HR
4b	Tools and translators: increased confidence from use	C.4.4	HR	HR	HR	HR

Software Metrics - reference to IEC 61508

- IEC 61508-3 , Tab. A.4 - Software design and development

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1a	Structured methods **	C.2.1	HR	HR	HR	HR
1b	Semi-formal methods **	Table B.7	R	HR	HR	HR
1c	Formal design and refinement methods **	B.2.2, C.2.4	---	R	R	HR
2	Computer-aided design tools	B.3.5	R	R	HR	HR
3	Defensive programming	C.2.5	---	R	HR	HR
4	Modular approach	Table B.9	HR	HR	HR	HR
5	Design and coding standards	C.2.6	R	HR	HR	HR

Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Software module size limit	C.2.9	HR	HR	HR	HR
2	Software complexity control	C.5.13	R	R	HR	HR
3	Information hiding/encapsulation	C.2.8	R	HR	HR	HR

Challenges for AI – used in safety context

- Runs on complex hardware, designed for massive parallel computing
  - Control of random faults (e.g. IEC 61508)
- Software design and development
  - Avoidance of systematic faults (e.g. IEC 61508)
  - Control of systematic faults (e.g. IEC 61508)
- Software Tools / AI Development Frameworks (e.g. TensorFlow, PyTorch, etc.)
  - Open-source / not qualified for FS / CySec

Defects are in the network. Standard FS techniques and measure are not sufficient (e.g. Data Quality, Neuron Coverage, etc.)

Techniques and measures under IEC 61508 are not sufficient for data driven software design

Status of relevant standards

- AI-Based systems should (in 2020) should not be used for higher safety integrity functions
- ISO/PAS 21448: Safety of the Intended Functionality (SOTIF), published in 2019 as public available specification (PAS) and not as an ISO standard
- ISO/TR 4804 Safety and security for ADS (annex B)
- ISO/SAE 21434 CySec for Automotive
- ISO IEC 29119-11 TR Guidelines on the testing of AI-based systems

Table A.2 – Software design and development – software architecture design

(see 7.4.3)

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
	Architecture and design feature					
5	Artificial intelligence - fault correction	C.3.9	---	NR	NR	NR

What level of independency is needed?

- IEC 61508-1, 8

  - Normative level of independence (Table 5 IEC 61508-1):

Minimum level of Independence	**Safety Integrity Level**			
	1	**2**	**3**	**4**
Independent person	X	X^1	Y	Y
Independent department	--	X^2	X^1	Y
Independent organization	--	--	X^2	X

  - For SIL2 and SIL3 an independent organization generally is involved.
  - Advantage in competition

X: minimum level of independence
X^2 is more appropriate than X^1 due to: lack of experience, higher degree of complexity, greater degree of novelty of design / technology
Y: the level of independence is considered as insufficient.

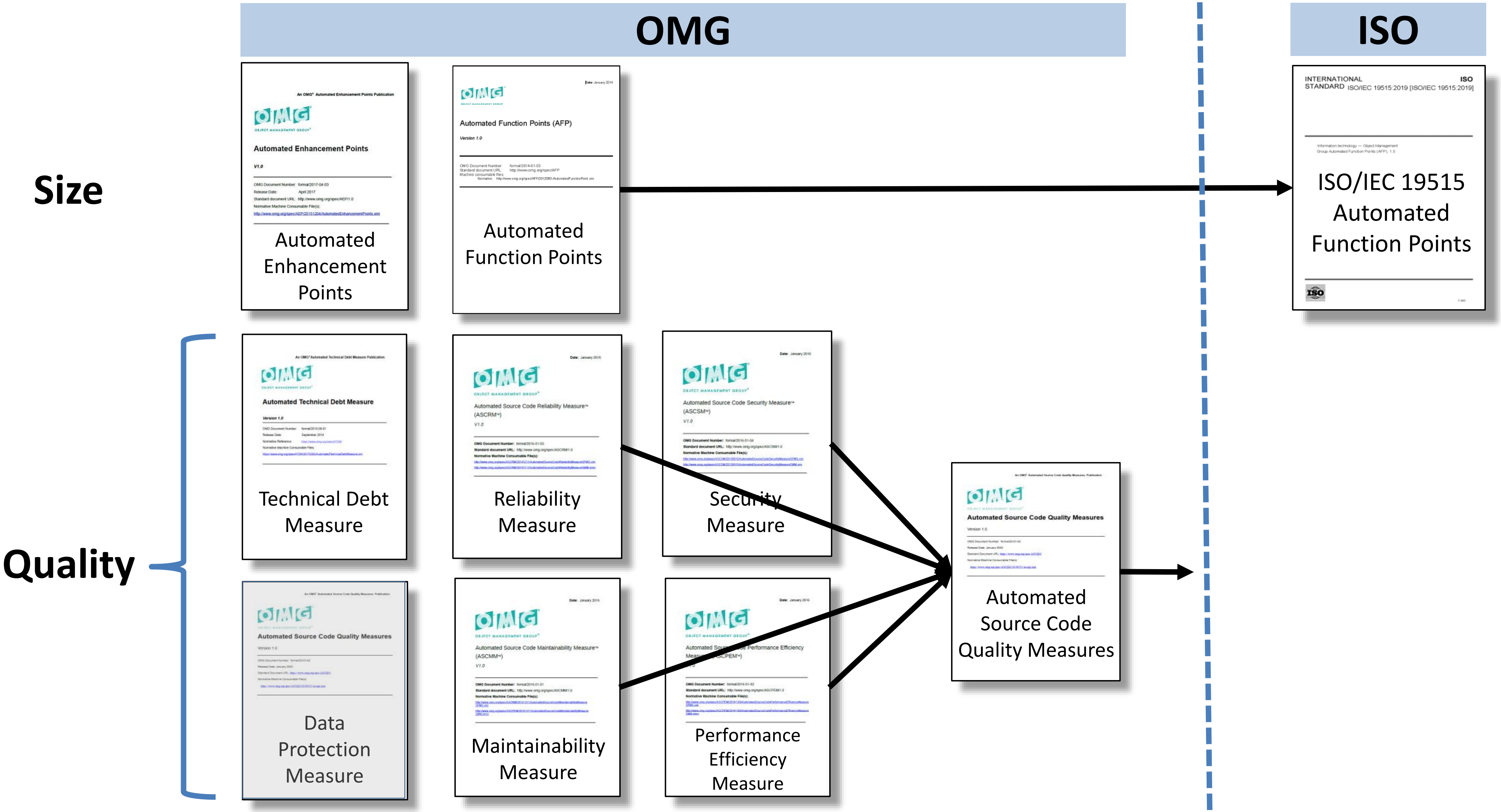
Any Questions?

How Do CISQ Measures Support Certification?

Dr. Bill Curtis
Executive Director

CISQ
Consortium for
Information and
Software Quality

CISQ Measurement Standards



CISQ Measures for Use in Certification

CISQ Structural Quality Measures

Examples of architectural and coding weaknesses included in the CISQ quality measures

CISQ Structural Quality Measures		Examples of architectural and coding weaknesses included in the CISQ quality measures
Security	37 (36) weaknesses	• SQL injection • Cross-site scripting • Buffer overflow
Reliability	36 (38) weaknesses	• Poor error handling • Deadlock • Improper synchronization
Performance Efficiency	16 (3) weaknesses	• Expensive loop operation • Un-indexed data access • Unreleased memory
Maintainability	28 (3) weaknesses	• Excessive coupling • Dead code • Hard-coded literals

CISQ measures calculated from counts of severe weaknesses in software

International team of experts selected CISQ weaknesses based on the severity of their impact on operational risk or cost of ownership.

Only weaknesses considered severe enough that they must be remediated were included

All CISQ weaknesses are included in the Common Weakness Enumeration Repository and have CWE #s.

CISQ Supplements ISO/IEC 25000 Standards

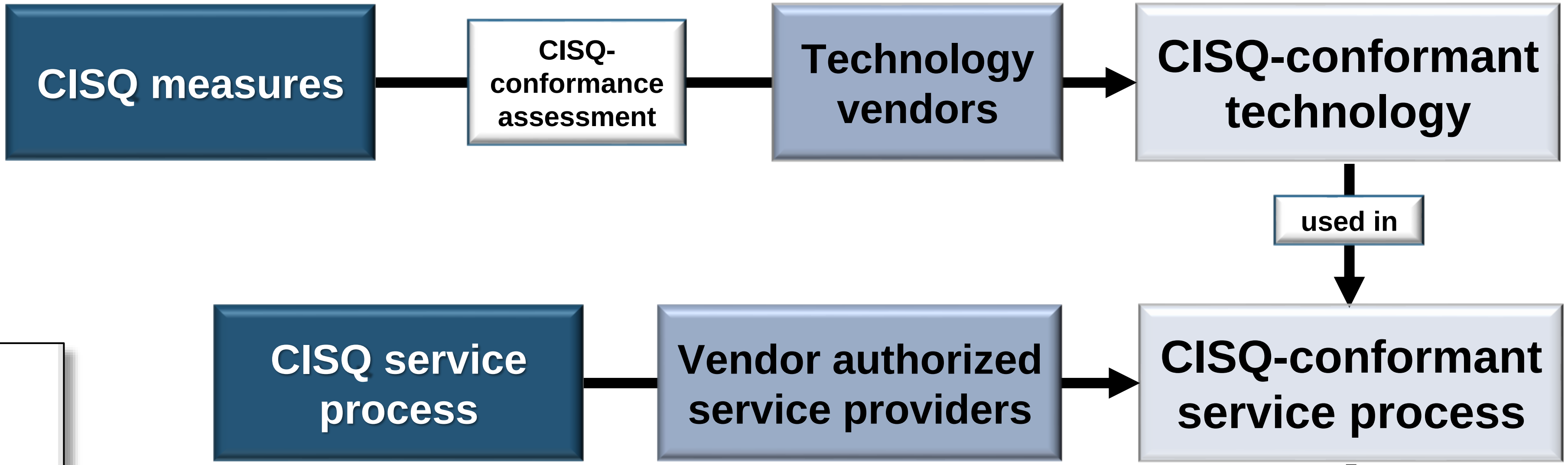
- ISO/IEC 25010 defines a software product quality model of 8 quality characteristics
- CISQ conforms to ISO/IEC 25010 quality characteristic definitions
- ISO/IEC 25023 defines measures, but not automatable or at the source code level
- CISQ supplements ISO/IEC 25023 with automatable source code level measures

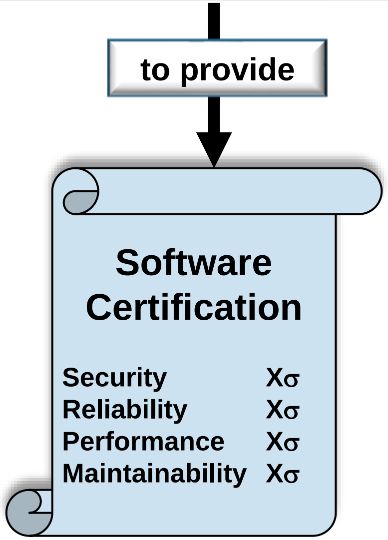
ISO/IEC 25010 — Software Product Quality

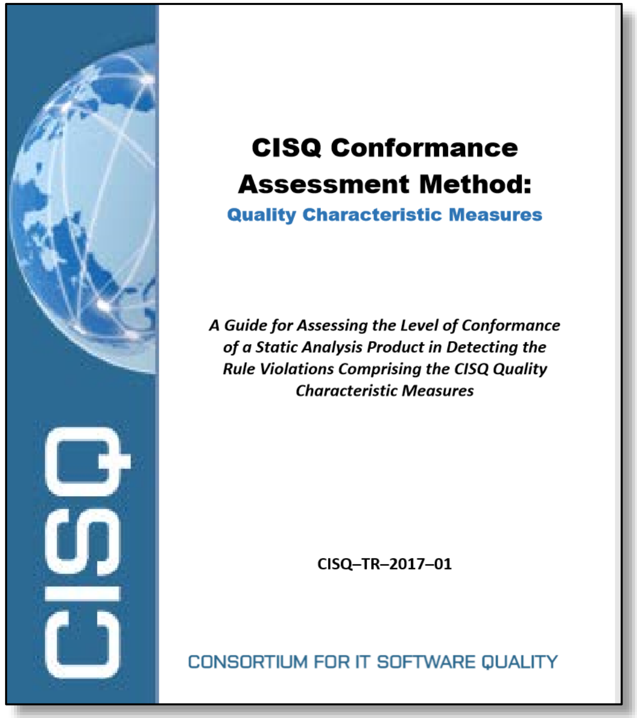
Functional Suitability	Reliability	Performance Efficiency	Operability	Security	Compatibility	Maintainability	Portability
Functional appropriateness	Maturity	Time behavior	Appropriateness	Confidentiality	Co-existence	Modularity	Adaptability
Accuracy	Availability	Resource utilization	Recognizability	Integrity	Interoperability	Reusability	Installability
Compliance	Fault tolerance	Compliance	Learnability	Non-repudiation	Compliance	Analyzability	Replaceability
	Recoverability		Ease of use	Accountability		Changeability	Compliance
	Compliance		Attractiveness	Authenticity		Modification stability	
			Technical Accessibility	Compliance		Testability	
			Compliance			Compliance	

CISQ automated structural quality measures are highlighted in blue

Certification Using CISQ Measures







CISQ Conformance Assessment Method:

Quality Characteristic Measures

A Guide for Assessing the Level of Conformance of a Static Analysis Product in Detecting the Rule Violations Comprising the CISQ Quality Characteristic Measures

CISQ–TR–2017–01

CISQ

CONSORTIUM FOR IT SOFTWARE QUALITY

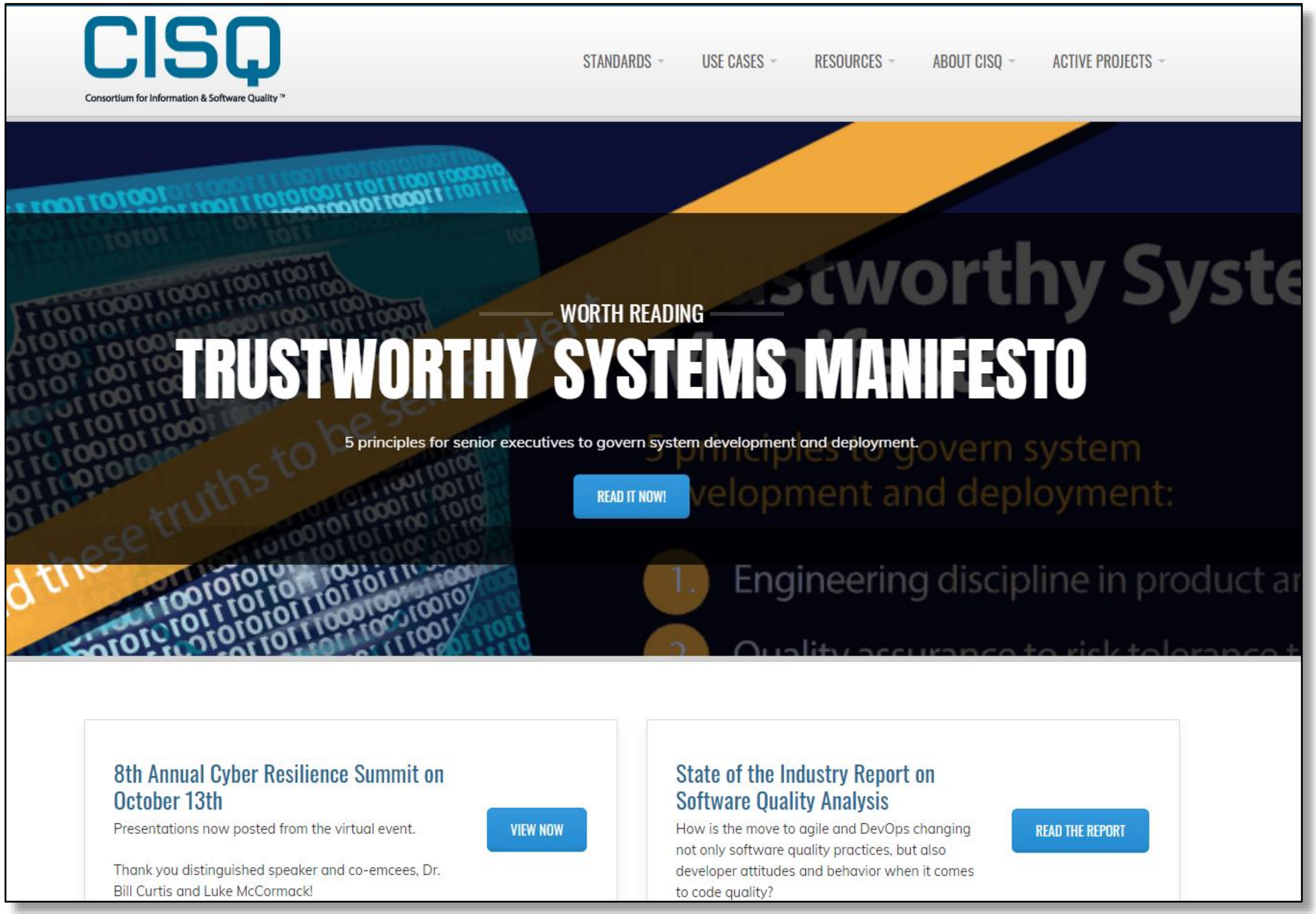
- **CISQ**
  - does not certify software
  - only assesses vendor conformance
  - CISQ endorses vendor technologies
- **Service providers**
  - use CISQ-endorsed technology
  - in a CISQ-conformant service process
  - to provide software certifications

Trustworthy Systems Manifesto

As a greater portion of mission, business, and safety critical functionality is committed to software-intensive systems, these systems become one of, if not the largest source of risk to enterprises and their customers. Since corporate executives are ultimately responsible for managing this risk, we establish the following principles to govern system development and deployment.

1. **Engineering discipline in product and process**
2. **Quality assurance to risk tolerance thresholds**
3. **Traceable properties of system components**
4. **Proactive defense of the system and its data**
5. **Resilient and safe operations**

CISQ Membership Is Free — www.it-cisq.org



Over 3,000 individual members from large software-intensive organizations:

Thank You for Attending!

Contact us

Dr. Bill Curtis, Executive Director, CISQ: bill.curtis@it-cisq.org
Matthias Haynl, TÜV Rheinland and TIC Council: matthias.haynl@us.tuv.com
Karin Athanas, Executive Director, TIC Council: kathanas@tic-council.org
Tracie Berardi, Program Director, CISQ: tracie@omg.org