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?

Co-founders

Dr. Paul Nielsen, CEO

Dr. Richard Soley, CEO

CISQ

Develop specifications for automatable measures of software systems

Gain approval as OMG standards

Fast-track to ISO

CISQ Partners

CISQ Sponsors
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

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 accountable for

Board of Directors Governance
CEO, COO, CFO Risk management
Business VPs Business Continuity
Corporate Auditors Brand protection
CIO Customer experience

Need evidence of governance and action against application risk

© 2020 Consortium for Information and Software Quality (CISQ) www.it-cisq.org
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.
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
The requirements for off-line support tools depend on the class.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T1</td>
</tr>
<tr>
<td><strong>Training</strong></td>
<td>X</td>
</tr>
<tr>
<td><strong>Specification / product manual</strong></td>
<td>X</td>
</tr>
<tr>
<td><strong>Definition of constraints</strong></td>
<td>X</td>
</tr>
<tr>
<td><strong>Assessment</strong></td>
<td>X</td>
</tr>
<tr>
<td><strong>Qualification of new version</strong></td>
<td>X</td>
</tr>
<tr>
<td><strong>Assessment against standard</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Conformance to its specification</strong></td>
<td>O</td>
</tr>
</tbody>
</table>

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

**How?**
Development lifecycle (the V-model)

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

<table>
<thead>
<tr>
<th>Technique/Measure *</th>
<th>Ref.</th>
<th>SIL 1</th>
<th>SIL 2</th>
<th>SIL 3</th>
<th>SIL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Suitable programming language</td>
<td>C.4.5</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>2 Strongly typed programming language</td>
<td>C.4.1</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>3 Language subset</td>
<td>C.4.2</td>
<td>---</td>
<td>---</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>4a Certified tools and certified translators</td>
<td>C.4.3</td>
<td>R</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>4b Tools and translators: increased confidence from use</td>
<td>C.4.4</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
</tr>
</tbody>
</table>
### Software Metrics - reference to IEC 61508

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

#### Technique/Measure *

<table>
<thead>
<tr>
<th>Technique/Measure *</th>
<th>Ref.</th>
<th>SIL 1</th>
<th>SIL 2</th>
<th>SIL 3</th>
<th>SIL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a Structured methods **</td>
<td>C.2.1</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>1b Semi-formal methods **</td>
<td>Table B.7</td>
<td>R</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>1c Formal design and refinement methods **</td>
<td>B.2.2, C.2.4</td>
<td>---</td>
<td>R</td>
<td>R</td>
<td>HR</td>
</tr>
<tr>
<td>2 Computer-aided design tools</td>
<td>B.3.5</td>
<td>R</td>
<td>R</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>3 Defensive programming</td>
<td>C.2.5</td>
<td>---</td>
<td>R</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>4 Modular approach</td>
<td>Table B.9</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>5 Design and coding standards</td>
<td>C.2.8</td>
<td>R</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
</tr>
</tbody>
</table>

#### Software Metrics

<table>
<thead>
<tr>
<th>Technique/Measure *</th>
<th>Ref.</th>
<th>SIL 1</th>
<th>SIL 2</th>
<th>SIL 3</th>
<th>SIL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Software module size limit</td>
<td>C.2.9</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>2 Software complexity control</td>
<td>C.5.13</td>
<td>R</td>
<td>R</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>3 Information hiding/encapsulation</td>
<td>C.2.8</td>
<td>R</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
</tr>
</tbody>
</table>
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)

<table>
<thead>
<tr>
<th>Technique/Measure *</th>
<th>Ref.</th>
<th>SIL 1</th>
<th>SIL 2</th>
<th>SIL 3</th>
<th>SIL 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture and design feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Artificial intelligence - fault correction</td>
<td>C.3.9</td>
<td>---</td>
<td>NR</td>
<td>NR</td>
<td>NR</td>
</tr>
</tbody>
</table>
What level of independency is needed?

- IEC 61508-1, 8

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

<table>
<thead>
<tr>
<th>Minimum level of Independence</th>
<th>Safety Integrity Level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Independent person</td>
<td>X</td>
</tr>
<tr>
<td>Independent department</td>
<td>--</td>
</tr>
<tr>
<td>Independent organization</td>
<td>--</td>
</tr>
</tbody>
</table>

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

- Advantage in competition

X: minimum level of independence
X² is more appropriate than X¹ 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

Consortium for Information and Software Quality
CISQ Measurement Standards

OMG

ISO

Size

Automated Enhancement Points

Automated Function Points

Technical Debt Measure

Reliability Measure

Security Measure

Automated Source Code Quality Measures

Quality

Data Protection Measure

Maintainability Measure

Performance Efficiency Measure

ISO/IEC 19515 Automated Function Points
CISQ Measures for Use in Certification

CISQ Structural Quality Measures

<table>
<thead>
<tr>
<th>Measure</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>37 (36)</td>
</tr>
<tr>
<td>Reliability</td>
<td>36 (38)</td>
</tr>
<tr>
<td>Performance</td>
<td>16 (3)</td>
</tr>
<tr>
<td>Maintainability</td>
<td>28 (3)</td>
</tr>
</tbody>
</table>

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

- SQL injection
- Cross-site scripting
- Buffer overflow
- Poor error handling
- Deadlock
- Improper synchronization
- Expensive loop operation
- Un-indexed data access
- Unreleased memory
- 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
  - Functional appropriateness
  - Accuracy
  - Compliance
- Reliability
  - Maturity
  - Availability
  - Fault tolerance
  - Recoverability
  - Compliance
- Performance Efficiency
  - Time behavior
  - Resource utilization
  - Compliance
- Operability
  - Appropriateness
  - Recognizability
  - Learnability
  - Ease of use
  - Technical Accessibility
  - Compliance
- Security
  - Confidentiality
  - Integrity
  - Non-repudiation
  - Accountability
  - Authenticity
  - Compliance
- Compatibility
  - Co-existence
  - Interoperability
  - Compliance
- Maintainability
  - Modularity
  - Reusability
  - Analyzability
  - Changeability
  - Modification
  - Stability
  - Testability
  - Compliance
- Portability
  - Adaptability
  - Installability
  - Replaceability
  - Compliance

CISQ automated structural quality measures are highlighted in blue
Certification Using CISQ Measures

➢ 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:

1. Engineering discipline in product and service development.
2. Quality assurance to risk tolerance.
3. Trustworthy Systems Manifesto

5 principles for senior executives to govern system development and deployment:

WORTH READING

TRUSTWORTHY SYSTEMS MANIFESTO

8th Annual Cyber Resilience Summit on October 13th
Presentations now posted from the virtual event.
Thank you distinguished speaker and co-entrepreneurs, Dr. Bill Curtis and Luke McCormack.

State of the Industry Report on Software Quality Analysis
How is the move to Agile and DevOps changing not only software quality practices, but also developer attitudes and behavior when it comes to code quality?

© 2020 Consortium for Information and Software Quality (CISQ) www.it-cisq.org
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