

Common Quality Enumeration (CQE)

John R. Marien, MITRE
Robert A. Martin, MITRE

3 December 2015
V 1.01



See Your Total Quality Picture

MITRE

Problem:

- Within the realm of quality issues there is pure black (security focused) on one end and pure white (other quality issues) on the other with varying shades of gray in between
- Somewhere in between these end points is a line drawn in the sand by the Common Weakness Enumeration (CWE) at which point quality issues become issues that can be exploited into vulnerabilities and that is what CWE covers
- Thus the CWE list enumerates quality issues that can be exploited. One, or more, CWEs can create a vulnerability that can be assigned a Common Vulnerability and Exposure (CVE) identifier
- The CWE is the lingua-franca for these security relevant weaknesses
- But there is a large set of quality issues that are not covered by CWE but that can significantly impact the system, drive up testing, maintenance, and in-field failures as well as impact performance

Idea:

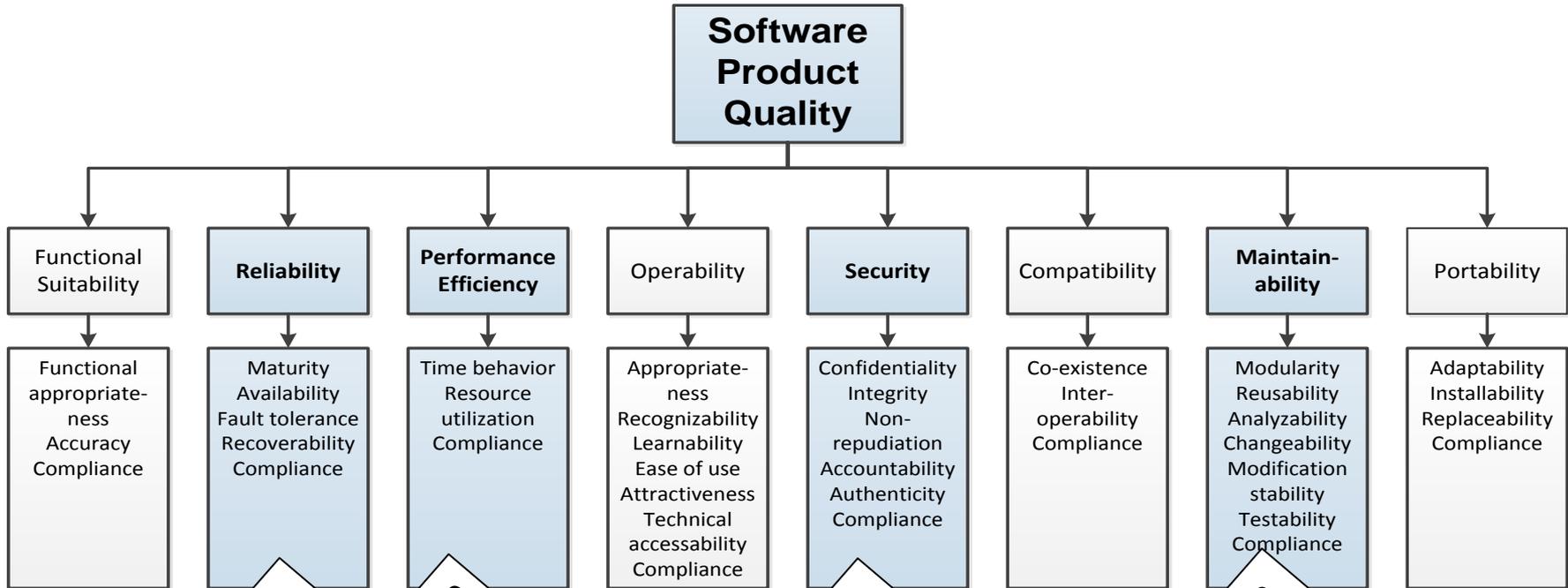
- **Work with static analysis tool vendors and researchers to create the Common Quality Enumeration (CQE) and leverage ISO and OMG work in this area**
 - Create the content for the missing lingua-franca of quality issues that are not security relevant weaknesses
- **With CQE defined, COTS static and dynamic analyzers can then tag their findings with a common identification system.**
- **Use and adoption of CQE will greatly simplify and clarify the correlation and integration of findings for those using multiple tools (a software best practice) and allow for planned coverage of the various quality aspect in your application by being able to combine results into a coherent report of quality issues**

Approach:

- Working with standards bodies, academia, tool vendors, and MITRE sponsors, we will lay out the framework for the Common Quality Enumeration (CQE), leveraging the lessons from CWE
- By adopting the CWE format and process as a starting point we can focus on covering the rest of the spectrum of quality issues with CQE, complementing the CWE coverage of quality issues that lead to vulnerabilities



Examples from OMG and ISO Software Quality Standards (ISO/IEC 25010)



OMG Automated Source Code Reliability Measure (ASCRM)

OMG Automated Source Code Performance Efficiency Measure (ASCPPEM)

OMG Automated Source Code Security Measure (ASCSSM)

OMG Automated Source Code Maintainability Measure (ASCMM)

Example OMG sources...

Reliability Pattern	Consequence	Objective	Measure Element
ASCRM-RLB-1: Empty Exception Block	Software without consistent and complete handling of errors and exceptions makes it impossible to accurately identify and adequately respond to unusual and unexpected situations.	Avoid improper responses to unusual and unexpected situations	Number of instances where an exception handling block (such as Catch and Finally blocks) of the named callable and method control elements does not contain any other control element

Performance Efficiency Pattern	Consequence	Objective	Measure Element
ASCPem-PRF-1: Static Block Element containing Class Instance Creation Control Element	Software that is coded so as to execute expensive computations repeatedly (such as in loops) requires excessive computational resources when the usage and data volume grow	Avoid upfront initialization of software data elements	Number of instances where a storable data element or member data element is initialized with a value in the 'Write' action and is located in a block of code which is declared as static

Maintainability Pattern	Consequence	Objective	Measure Element
ASCMM-MNT-1: Control Flow Transfer Control Element outside Switch Block	Software that does not follow the principles of structured programming degrades comprehensibility	Avoid the unconditional transfer of control flow outside of switch structures	Number of instances where an unconditional transfer of control is located outside the branching based on the value of a storable element

Solution:

- **Common Quality Enumeration (CQE) will address the quality issues that CWE doesn't cover**
- **The CQE collection will be provided publicly and vendors will subsequently adopt CQE Identifiers as part of their respective tool's findings reports**
- **By having a single community driven "dictionary" of the non-security quality issues (complementing the security issues in CWE) there will now be the ability to correlate findings across and amongst tools – and tools will be better able to describe the issues they cover and planned for filling gaps**
- **NOTE:**
 - **CWE and CQE numbers will be unique to avoid confusion**
 - **CQE numbering starts at 9000 in homage to the original international quality standard, ISO 9000 (now ISO 25000)**

How Can You Help?

- **If you're a tool Vendor**

- Work with us under a Unilateral NDA to flesh out CQE

- **If you're a User**

- Tell us which quality issues are your gravest concerns
- Tell us which tools have helped you assess the quality of your applications (so we can get them to join CQE)

- **If you're a Researcher**

- Tell us your ideas/suggestions on how to organize CQE
- Help review CWE schema for applicability to CQE entries

Questions?



Unilateral NDA Scope:

The proprietary/company-confidential information contemplated hereunder, and the purpose for its disclosure, are described as follows:

Proprietary/company confidential information contained in the underlying Knowledge Repository of the Knowledge Owner's Capability being Researched for sole purpose of establishing a public Common Quality Enumeration (CQE) dictionary that can be used by vendors, customers, and researchers to describe software, design, and architecture related weaknesses that have quality ramifications other than security. The individual contributions from numerous organizations, based on their proprietary/company-confidential information, will be combined into a consolidated collection of weakness descriptions and definitions with the resultant collection being shared publicly. The consolidated collection of knowledge about weaknesses in software, design, and architecture will make no reference to the source of the information used to describe, define, and explain the individual weaknesses.