CISQ

Consortium for Information & Software Quality ™

List of Weaknesses Included in the CISQ Automated Source Code Maintainability Measure

June 2019

Overview of Structural Quality Measurement in Software

Measurement of the structural quality characteristics of software has a long history in software engineering. These characteristics are also referred to as the structural, internal, technical, or engineering characteristics of software source code. Software quality characteristics are increasingly incorporated into development and outsourcing contracts as the equivalent of service level agreements. That is, target thresholds based on structural quality measures are being written into contracts as acceptance criteria for delivered software. This specification provides automated measures for four structural quality characteristics listed in the ISO/IEC 25010 software quality model that can be calculated from source code—Reliability, Security, Performance Efficiency, and Maintainability.

Recent advances in measuring the structural quality of software involve detecting violations of good architectural and coding practice from statically analyzing source code. Good architectural and coding practices can be stated as rules for engineering software products. Violations of these rules will be called weaknesses to be consistent with terms used in the Common Weakness Enumeration which lists the weaknesses used in these measures.

The four Automated Source Code Quality Measures are calculated from counts of what industry experts have determined to be most severe weaknesses. Consequently, they provide strong indicators of the quality of a software system and the probability of operational or cost problems related to each measure's domain.

The weaknesses comprising each CISQ Automated Source Code Quality Measure are grouped by measure in a table. This document lists the weaknesses in the Maintainability measure. The Common Weakness Enumeration repository (an ITU standard) has recently been expanded to include weaknesses from quality characteristics beyond security. All weaknesses included in these measures are identified by their CWE number from the repository. The title and description of CWEs is taken from information in the online CWE repository (cwe.mitre.org). Each weakness will be described as a 'quality measure element' to remain consistent with the structure of software quality measures enumerated in ISO/IEC 25020.

Some weaknesses drawn from the CWE repository (parent weaknesses) have related weaknesses listed as 'contributing weaknesses' ('child weaknesses' in the CWE). Contributing weaknesses represent variants of how the parent weakness can be instantiated in software. In the following table the cells containing CWE IDs for parents are presented in a darker blue than the cells containing contributing weaknesses. Based on their severity, not all children were included in this standard. Compliance to the CISQ measures is assessed at the level of the parent weakness. A technology must be able to detect at least one of the contributing weaknesses to be assessed compliant on the parent weakness.

Automated Source Code Maintainability Measure Element Descriptions

The quality measure elements (weaknesses violating software quality rules) that compose the CISQ Automated Source Code Maintainability Measure are presented in the table. This measure contains 29 parent weaknesses and no contributing weaknesses.

Table: Quality Measure Elements for Automated Source Code Maintainability Measure

CWE #	Descriptor	Weakness Description
CWE-407	**Algorithmic Complexity**	An algorithm in a product has an inefficient worst-case computational complexity that may be detrimental to system performance and can be triggered by an attacker, typically using crafted manipulations that ensure that the worst case is being reached.
CWE-478	**Missing Default Case in Switch Statement**	The code does not have a default case in a switch statement, which might lead to complex logical errors and resultant weaknesses.
CWE-480	**Use of Incorrect Operator**	The programmer accidentally uses the wrong operator, which changes the application logic in security-relevant ways.
CWE-484	**Omitted Break Statement in Switch**	The program omits a break statement within a switch or similar construct, causing code associated with multiple conditions to execute. This can cause problems when the programmer only intended to execute code associated with one condition.
CWE-561	**Dead code**	The software contains dead code that can never be executed. (Thresholds are set at 5% logically dead code or 0% for code that is structurally dead. Code that exists in the source but not in the object does not count.)
CWE-570	**Expression is Always False**	The software contains an expression that will always evaluate to false.
CWE-571	**Expression is Always True**	The software contains an expression that will always evaluate to true.
CWE-783	**Operator Precedence Logic Error**	The program uses an expression in which operator precedence causes incorrect logic to be used.
CWE-1041	**Use of Redundant Code (Copy-Paste)**	The software has multiple functions, methods, procedures, macros, etc. that contain the same code. (The default threshold for each instance of copy-pasted code sets the maximum number of allowable copy-pasted instructions at 10% of the total instructions in the instance, *alternate thresholds can be set prior to analysis*).
CWE-1045	**Parent Class with a Virtual Destructor and a Child Class without a Virtual Destructor**	A parent class has a virtual destructor method, but the parent has a child class that does not have a virtual destructor.

CWE-1047	**Modules with Circular Dependencies**	The software contains modules in which one module has references that cycle back to itself, i.e., there are circular dependencies.
CWE-1048	**Invokable Control Element with Large Number of Outward Calls (Excessive Coupling or Fan-out)**	The code contains callable control elements that contain an excessively large number of references to other application objects external to the context of the callable, i.e. a Fan-Out value that is excessively large. (default threshold for the maximum number of references is 5, *alternate threshold can be set prior to analysis*)
CWE-1051	**Initialization with Hard-Coded Network Resource Configuration Data**	The software initializes data using hard-coded values that act as network resource identifiers.
CWE-1052	**Excessive Use of Hard-Coded Literals in Initialization**	The software initializes a data element using a hard-coded literal that is not a simple integer or static constant element.
CWE-1054	**Invocation of a Control Element at an Unnecessarily Deep Horizontal Layer (Layer-skipping Call)**	The code at one architectural layer invokes code that resides at a deeper layer than the adjacent layer, i.e., the invocation skips at least one layer, and the invoked code is not part of a vertical utility layer that can be referenced from any horizontal layer.
CWE-1055	**Multiple Inheritance from Concrete Classes**	The software contains a class with inheritance from more than one concrete class.
CWE-1062	**Parent Class Element with References to Child Class**	The code has a parent class that contains references to a child class, its methods, or its members.
CWE-1064	**Invokable Control Element with Signature Containing an Excessive Number of Parameters**	The software contains a function, subroutine, or method whose signature has an unnecessarily large number of parameters/arguments. (default threshold for the maximum number of parameters is 7, *alternate threshold can be set prior to analysis*).
CWE-1074	**Class with Excessively Deep Inheritance**	A class has an inheritance level that is too high, i.e., it has a large number of parent classes. (default threshold for maximum Inheritance levels is 7, *alternate threshold can be set prior to analysis*).
CWE-1075	**Unconditional Control Flow Transfer outside of Switch Block**	The software performs unconditional control transfer (such as a "goto") in code outside of a branching structure such as a switch block.
CWE-1079	**Parent Class without Virtual Destructor Method**	A parent class contains one or more child classes, but the parent class does not have a virtual destructor method.
CWE-1080	**Source Code File with Excessive Number of Lines of Code**	A source code file has too many lines of code. (default threshold for the maximum lines of code is 1000, *alternate threshold can be set prior to analysis*).

CWE-1084	**Invokable Control Element with Excessive File or Data Access Operations**	A function or method contains too many operations that utilize a data manager or file resource. (default threshold for the maximum number of SQL or file operations is 7, *alternate threshold can be set prior to analysis*).
CWE-1085	**Invokable Control Element with Excessive Volume of Commented-out Code**	A function, method, procedure, etc. contains an excessive amount of code that has been commented out within its body. (default threshold for the maximum percent of commented-out instructions is 2%, *alternate threshold can be set prior to analysis*).
CWE-1086	**Class with Excessive Number of Child Classes**	A class contains an unnecessarily large number of children. (default threshold for the maximum number of children of a class is 10, *alternate threshold can be set prior to analysis*).
CWE-1087	**Class with Virtual Method without a Virtual Destructor**	A class contains a virtual method, but the method does not have an associated virtual destructor.
CWE-1090	**Method Containing Access of a Member Element from Another Class**	A method for a class performs an operation that directly accesses a member element from another class.
CWE-1095	**Loop Condition Value Update within the Loop**	The software uses a loop with a control flow condition based on a value that is updated within the body of the loop.
CWE-1121	**Excessive McCabe Cyclomatic Complexity**	A module, function, method, procedure, etc. contains McCabe cyclomatic complexity that exceeds a desirable maximum. (default threshold for Cyclomatic Complexity is 20, *alternate threshold can be set prior to analysis*).

The cells containing CWE IDs for parents are presented in a dark blue.
The cells containing contributing weaknesses are presented in a light blue.

Note there are no contributing weaknesses in the Maintainability measure.

Master list of quality measure weaknesses: https://www.it-cisq.org/coding-rules/index.htm
Master list PDF: https://www.it-cisq.org/pdf/cisq-weaknesses-in-ascqm.pdf