

Author

CAST

******

**CISQ Compliance Report**

Application Name –

Version –

CAST AIP -

|  |
| --- |
|  |
|  |

Monday, xx July 2012

My Application Name

Version Number

My CAST Version

# Table of Content

Table of Content

1. Introduction

1.1. Application Characteristics

2. CISQ Summary

2.1. CISQ Violations Summary

3. CAST Findings for CISQ Security

4. CAST Findings for CISQ Reliability

5. CAST Findings for CISQ Performance Efficiency

6. CAST Findings for CISQ Maintainability

7. Appendix

7.1. About CAST Software Intelligence

7.2. About CISQ Automated Quality Characteristic Measures

# Introduction

This assessment is an effort to determine the overall quality of the said applications against CISQ rules and measure the overall health of the application. This assessment uses the CAST Application Intelligence Platform (AIP) to automatically scan the implementation of these applications to review the architecture, design, and code against current industry best practices and known design flaws that may impact performance.

CAST AIP applies over 1200 engineering checks based on standards and measurements developed by the Software Engineering Institute (SEI), International Standards Organization (ISO), Consortium for IT Software Quality (CISQ), the Institute of Electrical and Electronics Engineers (IEEE) and the technology provider industry. The resulting analysis identifies specific flaws in the software and aggregates this information into metrics to objectively quantify the structural quality of the application.

## Application Characteristics

This assessment is focused solely on the technical implementation of the said application (user interface to database), with no investigation of the functionality.

|  |  |
| --- | --- |
| Name | Value |
| kLoC | 504 |
| Files | 6,586 |
| Classes | 593 |
| SQL Art. | 0 |
| Tables | 119 |

*Fig 1: Application Technology characteristics Table 1: Application characteristics*

# CISQ Summary

This section provides a summary of the CISQ specified vulnerability identified in the structural quality analysis and measurement by CAST AIP against the CISQ standard. Details about CISQ Standard can be found [here](http://it-cisq.org/standards/automated-quality-characteristic-measures).

## CISQ Violations Summary

Findings summary for CAST under CISQ Standards.

| Rules | Total Violations | Added Violations | Removed Violations |
| --- | --- | --- | --- |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 2: CISQ summary*

# CAST Findings for CISQ Security

Security assesses the degree to which an application protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization (ISO 25010). Security measures the risk of potential security breaches due to poor coding and architectural practices. Security problems have been studied extensively by the Software Assurance community and have been codified in the Common Weakness Enumeration (CWE) at [cwe.mitre.org](http://it-cisq.org/standards/automated-quality-characteristic-measures/security/cwe.mitre.org).

The CISQ Automated Source Code Security Measure draws from the [CWE/SANS Institute Top 25 Most Dangerous Software Errors](http://cwe.mitre.org/top25/#Listing) and identifies the most widespread and frequently exploited security weaknesses in software. Twenty-two of these weaknesses are detectable through analyzing the source code and form the basis of the CISQ measure. These 22 weaknesses constitute the most frequent ways unauthorized parties breach a system. Thus, the CISQ measure is a good predictor of how easily an application can suffer unauthorized penetration that results in stolen information, altered records, or other forms of malicious behavior.

List of CAST violations for CISQ Security.

| CAST Rules | Total Violations | Added Violations | Removed Violations |
| --- | --- | --- | --- |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 3: CISQ Security* *violations*

# CAST Findings for CISQ Reliability

Reliability measures the risk of potential application failures and the stability of an application when confronted with unexpected conditions. According to ISO/IEC/IEEE 24765, Reliability is the degree to which a system, product, or component performs specified functions under specified conditions for a specified period of time. The reason for checking and monitoring Reliability is to prevent or at least reduce application downtime, outages, data corruption, and errors that directly affect users.

The CISQ Automated Source Code Reliability Measure is composed from 29 critical violations of architectural and coding practice that affect the availability, fault tolerance, recoverability, and data integrity of an application. The CISQ Reliability measure produces a quality score based on the count of violations discovered in the software and can be turned into a density measure when divided by the size of the software.

List of CAST violations for CISQ Reliability.

| CAST Rules | Total Violations | Added Violations | Removed Violations |
| --- | --- | --- | --- |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 4: CISQ Reliability* *violations*

# CAST Findings for CISQ Performance Efficiency

Performance Efficiency assesses characteristics that affect an application’s response behavior and use of resources under stated conditions (ISO/IEC 25010). Performance Efficiency affects customer satisfaction, workforce productivity, application scalability, response-time degradation, and inefficient use of processing or storage resources. The Performance Efficiency of an application lies in each individual component ‘s performance, as well as in the effect of each component on the behavior of the chain of components comprising a transaction in which it participates.

The CISQ Automated Source Code Performance Efficiency Measure is composed from 15 critical violations of response time behavior, processor use, and memory use of an application. A quality score is produced based on the count of violations discovered in the source code and can be used as a density metric when divided by software size.

List of CAST violations for CISQ Performance Efficiency.

| CAST Rules | Total Violations | Added Violations | Removed Violations |
| --- | --- | --- | --- |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 5: CISQ Performance Efficiency* *violations*

# CAST Findings for CISQ Maintainability

List of CAST violations for CISQ Maintainability.

| CAST Rules | Total Violations | Added Violations | Removed Violations |
| --- | --- | --- | --- |
| Rule 1 | 0 | 0 | 0 |
| Rule 2 | 0 | 0 | 0 |
| Rule 3 | 0 | 0 | 0 |
| Rule 4 | 0 | 0 | 0 |
| Rule 5 | 0 | 0 | 0 |

*Table 6: CISQ Maintainability* *violations*

# Appendix

## About CAST Software Intelligence

Software Intelligence creates understanding into software architecture, end to end transaction flows, data access patterns and more, helping teams work confidently and faster. Hundreds of companies rely on CAST Software Intelligence to improve end-user satisfaction and time-to-market, prevent business disruption and reduce cost, enabling them to move past today’s obstacles and to tackle the next wave of innovation.

[Click here](https://www.castsoftware.com/software-intelligence) for more information about CAST Software Intelligence.

## About CISQ Automated Quality Characteristic Measures

CISQ has developed Automated Quality Characteristic Measures to measure and manage the structural quality of IT application software. The automated measures for Security, Reliability, Performance Efficiency, and Maintainability are now OMG® approved standards making them global standards for use by IT organizations.

These measures were developed from coding rules covering some of the most serious violations of good architectural and coding practices that should be avoided and can be detected through static code analysis. Each measure counts the number of violations of the architectural and coding rules related to that quality characteristic, and then can be used in creating metrics for defect density, etc.

|  |  |
| --- | --- |
| [Security](http://it-cisq.org/standards/automated-quality-characteristic-measures/security/) | Critical security violations in the source code drawn from the Top 25 security weaknesses in the Common Weakness Enumeration (CWE) repository |
| [Reliability](http://it-cisq.org/standards/automated-quality-characteristic-measures/reliability/) | Critical violations of availability, fault tolerance, and recoverability of software |
| [Performance Efficiency](http://it-cisq.org/standards/automated-quality-characteristic-measures/performance-efficiency/) | Critical violations of response time, as well as processor, memory, and utilization of other resources by the software |
| [Maintainability](http://it-cisq.org/standards/automated-quality-characteristic-measures/maintainability/) | Critical violations of modularity, architectural compliance, reusability, analyzability, and changeability in software |