|  |
| --- |
| sAR: Security Assessment Report |
| Monday, xx July 2012  My Application Name  Version: Version Number |
| PURPOSE OF DOCUMENT:  Purpose of this report is to provide a Security assessment of the quality of the deliverables. The primary audience for this report is IT executives/Security Experts who are responsible for My Application Name application. |

|  |
| --- |
| UNCLASSIFIED |

Table of Contents

[1. Executive Summary 3](#_Toc345670318)

[1.1. Application Characteristics 3](#_Toc345670319)

[1.2. Summary of Quality Indicators 3](#_Toc345670320)

[2. Security Assessment Overview 4](#_Toc345670321)

[3. Software Security Standards 5](#_Toc345670322)

[4. Mapping CAST Rules to CWE Most Dangerous Software Errors 6](#_Toc345670323)

[4.1. Insecure Interaction Between Components 6](#_Toc345670324)

[4.2. Risky Resource Management 6](#_Toc345670325)

[4.3. Porous Defenses 7](#_Toc345670326)

[4.4. CAST detects following security vulnerabilities identified by OWASP & CWE (not part of top-25) 8](#_Toc345670327)

[5. Security Weaknesses Spotted 10](#_Toc345670328)

[6. Appendix - Assessment Approach Overview 15](#_Toc345670329)

[7. Appendix: Understanding Quality Indicators, Quality Rules 16](#_Toc345670330)

# Executive Summary

The Application Assessment evaluates the overall quality of the My Application Name application.

My Application Name is a Small/Medium/Large/ExtraLarge application and has a VeryLow/Low/Medium/Good/VeryGood quality with a ***Total Quality Indicator (TQI) of 0.00 on a scale of 4.*** Each of the additional health metrics and their scores are identified below.

## Application Characteristics

**Top 5 Technologies**

|  |  |
| --- | --- |
| Name | LOC |
| Techno 1 | 000,000 |
| Techno 2 | 000,000 |
| Techno 3 | 000,000 |
| Techno 4 | 000,000 |
| Techno 5 | 000,000 |

**Technical Size**

|  |  |
| --- | --- |
| Name | Number |
| kLOCs | 000 |
| Files | 0,000 |
| Classes | 0,000 |
| SQL Art. | 00 |
| Tables | 00 |

## Summary of Quality Indicators

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TQI | Robu | Perf | Secu | Trans | Chang |
| Curr. Vers. | **0.00** | **0.00** | **0.00** | **0.00** | **0.00** | **0.00** |
| Prev. Vers. | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Variation | 0.00 % | 0.00 % | 0.00 % | 0.00 % | 0.00 % | 0.00 % |

# Security Assessment Overview

Here we list down all the security vulnerabilities identified by CAST AIP.

**Top technical criteria that most impact the SECURITY 0.00**

|  |  |  |  |
| --- | --- | --- | --- |
| Technical Criterion | Total Violation (#) | Total Check (#) | Grade |
| Criteria1 | 1 | 1 | 2.2 |
| Criteria2 | 2 | 2 | 1.4 |
| Criteria3 | 3 | 3 | 3.23 |
| Criteria4 | 4 | 4 | 3.45 |
| Criteria5 | 5 | 5 | 2.2 |
| Criteria6 | 6 | 6 | 1.4 |
| Criteria7 | 7 | 7 | 3.23 |
| Criteria8 | 8 | 8 | 3.45 |
| Criteria9 | 9 | 9 | 2.3 |
| Criteria10 | 10 | 10 | 3.5 |

Rules list below display the most impacting rules for the current snapshot. Rules are sorted according to the grade and the weight of the rule. In other terms, on top of the list, you will see the rules that have a big impact (low grade \* big weight) and the rules that are difficult to correct (lots of violations to be correct).

**Top rules that most impact the SECURITY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rule | Current violation | Previous violation | Evol. | Grade | Evol. |
| Rule1 | 1 | 1 | 1 | 2.2 | 1 |
| Rule2 | 2 | 2 | 2 | 1.4 | 2 |
| Rule3 | 3 | 3 | 3 | 3.23 | 3 |
| Rule4 | 4 | 4 | 4 | 3.45 | 4 |
| Rule5 | 5 | 5 | 5 | 2.2 | 5 |
| Rule6 | 6 | 6 | 6 | 1.4 | 6 |
| Rule7 | 7 | 7 | 7 | 3.23 | 7 |
| Rule8 | 8 | 8 | 8 | 3.45 | 8 |
| Rule9 | 9 | 9 | 9 | 2.3 | 9 |
| Rule10 | 10 | 10 | 10 | 3.5 | 10 |

# Software Security Standards

CAST AIP is aligned to capture the Security requirements listed by CWE (http://cwe.mitre.org/), and CERT as top Security weaknesses.

<Please Note: This table is not automatically populated today. Please replace the XX by the data/values available in Section 4 & 5>

**Summary of Security Violations from CAST AIP**

|  |  |
| --- | --- |
| High Level Area | Number of violations |
| Insecure Interaction Between Component | XX |
| Risky Resource Management | XX |
| Porous Defenses | XX |
| CAST detects following security vulnerabilities identified by OWASP & CWE (not part of top-25) | XX |

# Mapping CAST Rules to CWE Most Dangerous Software Errors

The rules categorized into four high-level areas listed below:

* Insecure Interaction Between Components
* Risky Resource Management
* Porous Defenses
* CAST detects following security vulnerabilities identified by OWASP & CWE (not part of top-25)

## Insecure Interaction Between Components

These weaknesses are related to insecure ways in which data is sent and received between separate components, modules, programs, processes, threads, or systems.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rank | CWE ID | Name | Recommendation/Mitigation/Comments | Corresponding CAST Rule |
| [1] | CWE-89 | Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') | Checking for SQL Injection | Avoid SQL injection vulnerabilities |
| [2] | CWE-78 | Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') | Checking for OS Command Injection | Avoid OS command injection vulnerabilities |
| [4] | CWE-79 | Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') | Checking for Cross-site scripting | Avoid cross-site scripting vulnerabilities |
| [9] | CWE-434 | Unrestricted Upload of File with Dangerous Type | Input Validation | Extend existing rules - Avoid non standard file extensions, Avoid file path manipulation vulnerabilities, Avoid XPath injection vulnerabilities |
| [12] | CWE-352 | Cross-Site Request Forgery (CSRF) | Ensure that application is free of cross-site scripting issues (CWE-79), because most CSRF defenses can be bypassed using attacker-controlled script. | Avoid cross-site scripting vulnerabilities |
| [22] | CWE-601 | URL Redirection to Untrusted Site ('Open Redirect') | Checking for Cross-site scripting | Avoid cross-site scripting vulnerabilities |

## Risky Resource Management

The weaknesses in this category are related to ways in which software does not properly manage the creation, usage, transfer, or destruction of important system resources.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rank | CWE ID | Name | Recommendation/Mitigation/Comments | Corresponding CAST Rule |
| [3] | CWE-120 | Buffer Copy without Checking Size of Input ('Classic Buffer Overflow') | Perform input validation on any numeric input by ensuring that it is within the expected range. Enforce that the input meets both the minimum and maximum requirements for the expected range. | Extend existing rules - Avoid using getopt() function, Never use sprintf() function or vsprintf() function, Never perform C cast between incompatible class pointers, Avoid using static\_cast on class/struct pointers |
| [13] | CWE-22 | Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') | Checking for file path manipulation | Avoid file path manipulation vulnerabilities |
| [14] | CWE-494 | Download of Code Without Integrity Check | Check Download Code Integrity | Extend existing rules - Avoid non standard file extensions, Avoid file path manipulation vulnerabilities, Avoid XPath injection vulnerabilities |
| [16] | CWE-829 | Inclusion of Functionality from Untrusted Control Sphere | When the set of acceptable objects, such as filenames or URLs, is limited or known, create a mapping from a set of fixed input values (such as numeric IDs) to the actual filenames or URLs, and reject all other inputs. | Extend existing rules - Avoid XPath injection vulnerabilities |
| [18] | CWE-676 | Use of Potentially Dangerous Function | Checking for programming best practices | Avoid using snprintf() function, Avoid using realpath() function, Avoid using the scanf() function, etc |
| [20] | CWE-131 | Incorrect Calculation of Buffer Size | Perform input validation on any numeric input by ensuring that it is within the expected range. Enforce that the input meets both the minimum and maximum requirements for the expected range. | Extend existing rules - Never use sprintf() function or vsprintf() function, Never perform C cast between incompatible class pointers, Avoid using static\_cast on class/struct pointers |
| [23] | CWE-134 | Uncontrolled Format String | Whenever possible, use functions that do not support the %n operator in format strings. | Extend existing rules - Never use sprintf() function or vsprintf() function, Avoid using the scanf() function, etc |
| [24] | CWE-190 | Integer Overflow or Wraparound | Perform input validation on any numeric input by ensuring that it is within the expected range. Enforce that the input meets both the minimum and maximum requirements for the expected range. | Extend existing rules - Avoid using getopt() function |

## Porous Defenses

The weaknesses in this category are related to defensive techniques that are often misused, abused, or just plain ignored.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rank | CWE ID | Name | Recommendation/Mitigation/Comments | Corresponding CAST Rule |
| [5] | CWE-306 | Missing Authentication for Critical Function | avoid implementing custom authentication routines and consider using authentication capabilities as provided by the surrounding framework, operating system, or environment. | Extend existing rules - Avoid cross-site scripting vulnerabilities, Avoid LDAP injection vulnerabilities |
| [6] | CWE-862 | Missing Authorization | Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page. | Extend existing rules - Avoid cross-site scripting vulnerabilities, Avoid LDAP injection vulnerabilities |
| [7] | CWE-798 | Use of Hard-coded Credentials | Store passwords, keys, and other credentials outside of the code in a strongly-protected, encrypted configuration file or database that is protected from access by all outsiders, including other local users on the same system. | Extend existing rules - Avoid LDAP injection vulnerabilities |
| [8] | CWE-311 | Missing Encryption of Sensitive Data | Periodically ensure that you aren't using obsolete cryptography. Avoid using old encryption techniques using MD4, MD5, SHA1, DES, and other algorithms that were once regarded as strong. | Extend existing rules - Avoid cross-site scripting vulnerabilities, Avoid LDAP injection vulnerabilities |
| [10] | CWE-807 | Reliance on Untrusted Inputs in a Security Decision | consider getcookies as unsafe | Avoid cross-site scripting vulnerabilities |
| [11] | CWE-250 | Execution with Unnecessary Privileges | Checking for privileges being appropriately implemented based on the scenario/usecase. Perform extensive input validation and canonicalization to minimize the chances of introducing a separate vulnerability. | Extend existing rules - Avoid cross-site scripting vulnerabilities, Avoid LDAP injection vulnerabilities |
| [15] | CWE-863 | Incorrect Authorization | consider getcookies as unsafe | Avoid cross-site scripting vulnerabilities |
| [17] | CWE-732 | Incorrect Permission Assignment for Critical Resource | Path manipulation | Avoid file path manipulation vulnerabilities |
| [21] | CWE-307 | Improper Restriction of Excessive Authentication Attempts | Check login implementation | Extend existing rules - Avoid direct access to database Procedures/Functions, User Interface elements must not use directly the database |
| [25] | CWE-759 | Use of a One-Way Hash without a Salt | Checking for programming best practices | Extend rules - Avoid using Hashtable, Avoid classes overriding only equals() or only hashCode() |

## CAST detects following security vulnerabilities identified by OWASP & CWE (not part of top-25)

|  |  |  |  |
| --- | --- | --- | --- |
| CWE ID | Name | Recommendation/Mitigation/Comments | Corresponding CAST Rule |
| CWE-20: | Improper Input Validation | Checking for best programming practices | Avoid SQL injection vulnerabilities, Avoid XPath injection vulnerabilities, Avoid cross-site scripting vulnerabilities |
| CWE-116: | Improper Encoding or Escaping of Output | Checking for best programming practices | Avoid SQL injection vulnerabilities, Avoid OS command injection vulnerabilities, Avoid cross-site scripting vulnerabilities |
| CWE-90 | LDAP Injection | Checking for LDAP injection | Avoid LDAP injection vulnerabilities |
| CWE-91 | XPATH Injection | Checking for XPATH injection | Avoid XPath injection vulnerabilities |
| CWE-73: | External Control of File Name or Path | Checking for file path manipulation | Avoid file path manipulation vulnerabilities |
| CWE-99: | Improper Control of Resource Identifiers ('Resource Injection') | Checking for best programming practices | Avoid file path manipulation vulnerabilities |
| CWE-117: | Improper Output Neutralization for Logs | Checking for log forging | Avoid Log forging vulnerabilities |

# Security Weaknesses Spotted

|  |  |
| --- | --- |
| Avoid SQL injection vulnerabilities | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid OS command injection vulnerabilities | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid cross-site scripting vulnerabilities | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid file path manipulation vulnerabilities | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid XPath injection vulnerabilities | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid LDAP injection vulnerabilities | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid Log forging vulnerabilities | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid direct access to database Procedures/Functions | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| User Interface elements must not use directly the database | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid using Hashtable | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid classes overriding only equals() or only hashCode() | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid non standard file extensions | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid using getopt() function | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Never use sprintf() function or vsprintf() function | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Never perform C cast between incompatible class pointers | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid using static\_cast on class/struct pointers | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid using snprintf() function | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid using realpath() function | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

|  |  |
| --- | --- |
| Avoid using the scanf() function | |
| Description | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed et accumsan felis. Etiam pharetra semper suscipit. Mauris hendrerit placerat lorem sit amet commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. |
| Remediation | Aliquam erat volutpat. Vestibulum quam ante, venenatis at bibendum vitae, viverra eget nulla. Donec pulvinar consequat varius. Morbi eget adipiscing lacus. Sed et libero odio, eget tempus massa. Phasellus venenatis commodo enim eget aliquet. Quisque posuere elit sed nunc aliquam eu ornare elit lacinia. Curabitur luctus, eros id venenatis lacinia, dolor libero tincidunt nibh, eget dapibus orci lectus pellentesque nisl. Ut quis velit est. |

# Appendix - Assessment Approach Overview

This assessment is an effort to determine the overall quality of the application My Application Name and identify any risks that may be inherent in the application. The assessment looks at the implementation of My Application Name to determine whether the application is constructed according to industry best practices, follows best practices for software engineering, and is maintainable.

**Table 1:** Health Factor descriptions and business benefits of measuring them

This assessment is focused solely on the Source code and Database structure with no view to functionality provided by backend services.

The CAST AIP is the industry leading automated code analysis platform, with coverage of all major development tools and languages. CAST AIP automatically scans and analyzes all of the source code and database elements that are part of an Enterprise system. CAST AIP applies over 1000+ metrics based on standards and measurements developed by the Software Engineering Institute (SEI), International Standards Organization (ISO), Consortium for IT Software Quality (CISQ), and Institute of Electrical and Electronics Engineers (IEEE). These metrics objectively measure software for the quality and quantity of work.

CAST AIP provides Application Analysts the ability to examine and drill down on critical application characteristics and attributes. The primary Application Health Factors that are addressed are:

|  |  |  |
| --- | --- | --- |
| Health Factor | Description | Example business benefits |
| Robustness | Attributes that affect the stability of the application and the likelihood of introducing defects when modifying it | * Improves availability of the business function or service * Reduces risk of loss due to operational malfunction * Reduces cost of application ownership by reducing rework |
| Performance | Attributes that affect the performance of an application | * Reduces risk of losing customers from poor service or response * Improves productivity of those who use the application * Increases speed of making decisions and providing information * Improves ability to scale application to support business growth |
| Security | Attributes that affect an application’s ability to prevent unauthorized intrusions | * Improves protection of competitive information-based assets * Reduces risk of loss in customer confidence or financial damages * Improves compliance with security-related standards and mandates |
| Transferability | Attributes that allow new teams or members to quickly understand and work with an application | * Reduces inefficiency in transferring application work between teams * Reduces learning curves * Reduces lock-in to suppliers |
| Changeability | Attributes that make an application easier and quicker to modify | * Improves business agility in responding to markets or customers * Reduces cost of ownership by reducing modification effort |

# Appendix: Understanding Quality Indicators, Quality Rules

CAST AIP has 1000+ quality rules and each rule produces a Grade. Depending on the impact the grades are aggregated into high level Indicators: **Quality indicators** and **Best practices indicators**.

Each aggregation is a weighted average of the contributing metrics grades where certain metric grades are flagged critical, i.e. it is nearly a defect. We talk about **Critical Violations**.

#### Quality Indicators

The structure, classification and terminology are from the ISO 9126‐3 and the subsequent ISO 25000:2005 quality model. The main focus is on internal structural quality. Subcategories have been created to handle specific areas like business application architecture and technical characteristics such as data access and manipulation or the notion of transactions. The dependence tree between software quality characteristics and their measurable attributes is represented in the following diagram, where each of the 5 characteristics that matter for the user or owner of the business system depends on measurable attributes: Application Architecture Practices, Coding Practices, Application Complexity, Documentation, Portability, and Technical & Functional Volume.

|  |  |
| --- | --- |
| Quality Indicator | Description |
| Performance / Efficiency | The source code and software architecture attributes are the elements that ensure high performance once the application is in run‐time mode. Efficiency is especially important for applications in high execution speed environments such as algorithmic or transactional processing where performance and scalability are paramount. An analysis of source code efficiency and scalability provides a clear picture of the latent business risks and the harm they can cause to customer satisfaction due to response‐time degradation. |
| Robustness / Reliability | An attribute of resiliency and structural solidity. Reliability measures the level of risk and the likelihood of potential application failures. It also measures the defects injected due to modifications made to the software (its “stability” as termed by ISO). The goal for checking and monitoring Reliability is to reduce and prevent application downtime, application outages and errors that directly affect users, and enhance the image of IT and its impact on a company’s business performance. |
| Security | A measure of the likelihood of potential security breaches due to poor coding and architectural practices. This quantifies the risk of encountering critical vulnerabilities that damage the business and provides a list of prevention measures. |
|  |  |
| Transferability | The effort necessary to diagnose the cause of a failure or section of code to be modified. It establishes the level of dependency on specific developers |
| Changeability | The effort necessary to modify the source code. It establishes the level of responsiveness to business-driven change requests |
| TQI | Total Quality Index (TQI) is computed on all the measures made by the CAST AIP |

#### Best practices Indicators

|  |  |
| --- | --- |
| Health Factor | Description |
| Programming Practices | Measures the level of compliance of the application to coding best practices. Compliance to best practices reduces risks of failures in production and improves productivity through increased readability and reduced debugging. |
| Architectural Design | Measures the level of compliance of the application to software architecture and design rules. Compliance to architecture rules improves productivity through better use of existing frameworks and code and reduced debugging. |
| Documentation | Measures the level of compliance of the application to code documentation best practices. Compliance to documentation best practices improves productivity through increased readability and faster understanding of source code. |

The risk level of a grade shall be assessed according to the below scale

|  |  |
| --- | --- |
| Scale | Risk Level |
| 4 | Low Risk |
| 3 | Moderate Risk |
| 2 | High Risk |
| 1 | Very High Risk |