Glossary

[A] [B] [C] [D] [E] [F] [G] [H] [I] [K][L] [M] [N] [O] [P][Q] [R] [S] [T] [U] [V]

Summary: This glossary defines terminology used within CAST AIP. Note that it does not include terms which are associated with external concepts, or terms generally used globally. Such information should be obtained from related external documentation help system.

A

AEP (Automated Enhancement Points)

AEP is used by default in CAST AIP 8.2.x and above to estimate the size of an application revision. See also CAST OMG-compliant Automated Function Points.

AEFP (Automated Enhancement Function Points)

Automated Enhancement Functional Points (AEFP) assess the changes made to the functional part of the application. AEFP reflects the changes made to both Transactional Functions and Data Functions. Therefore, AEFP is calculated by adding the AEFP value of all Transactional Functions and Data Functions in the application. The AEFP of each individual Transactional Function / Data Function is calculated by multiplying its FP value by its Complexity Factor (AEFP = FP x CF). The Complexity Factor of a certain Transactional Function / Data Function is an adjustment factor (defined by an OMG-specification) which is calculated based on its status (added / modified / deleted) and the complexity of the objects inside the Transactional Function / Data Function. The value reported for AEFP includes only added, modified and deleted Functions. All unchanged Functions are automatically excluded from this value (their Complexity Factor is considered as 0, and thus their AEFP value will also be 0). See also CAST OMG-compliant Automated Function Points an d CAST Automated Enhancement Points Estimation - AEP.

AETP data is not available when the EFP measure has been used to calculate the snapshot.

AETP (Automated Enhancement Technical Points)

As defined in the OMG AEP specification, Automated Enhancement Technical Points (AETP) assess the changes made to the technical part of the application. Therefore, AETP is the count of added / modified / deleted technical points, which are calculated based on the added / modified / deleted technical objects (i.e. objects which are not part of any transaction). In other words, AETP summarizes the evolution performed in the application, but outside the functional scope (these objects are not taken into account in the functional points, so the AEP measure introduces the concept of "technical points" to assess the enhancement done in these types of objects). See also CAST OMG-compliant Automated Function Points and CAST Automated Enhancement Points Estimation - AEP.

AETP data is not available when the EFP measure has been used to calculate the snapshot.

AFP (Automated Function Points)

Automated Function Points (AFP) is an automatic function points counting solution based on the rules defined by the International Function Point User Group (IFPUG) and the Consortium for IT Software Quality (CISQ). CAST automates the manual counting process by using the structural information retrieved by source code analysis, database structure and transactions. See also CAST OMG-compliant Automated Function Points.

(CAST) AIP

AIP (Application Intelligence Platform) is an enterprise-grade software measurement and quality analysis solution designed to analyze multi-tiered, multi-technology applications for technical vulnerabilities and adherence to architectural and coding standards and then provide business relevant information to the IT organization through various dashboards and products built with end users in mind.

(CAST) AIP Console

AIP Console is a web application for managing multiple CAST AIP instances from one location.

Analysis Unit

A set of source code files to analyze. For Java, this can be an Eclipse Project, or a directory containing Web Server resources (JSP files).

Analysis Service schema

An Analysis Service schema stores all analysis results: components, diagnosis findings and violations. Assessment results for all Technical Modules are also stored in an Analysis Service:

- for historical reasons, all assessment results are calculated including Business Criteria
- however the majority of Sizing Results are only calculated in the Dashboard Service database (Technical Debt for example)

Alternative legacy/deprecated names: Local, Local Site, Knowledge Base

Application

A union of Analysis Units that defines the scope of source code for analysis.

APR (Action Plan Recommendation)

The **Action Plan Recommendation** is a feature designed to automatically build an **Action Plan** to **improve the grade/score of a chosen Health Factor** (Business Criteria).

(CAST) Architecture Checker

CAST Architecture Checker is a client/server application installed as part of CAST AIP. Its primary usage is to:

- Create an Architecture Model that reflects a business application or set of applications via the use of Layers and Dependencies between those Layers
- Define the contents of Layers in terms of objects resulting from a CAST AIP analysis

Once the Architecture Model is defined, it can then be used in the CAST Management Studio. When the application or set of applications is then analyzed in the CAST Management Studio and a Snapshot is subsequently generated, the Architecture Model will be taken into account.

Artifact

An artifact is a component and is used in the context of CAST AIP metrics to indicate the low-level programming elements used to measure application size and complexity.

Assessment Model

A specification of metrics, quality rules, calculation rules and quality criteria to assess source code quality and risks.

Back to top

B

Backfired Function Point

Back-Fired Function Points (BFP) estimate the number of function points of an application. This code-derived metric is based on the lines of code, weighted by an abacus for a given technology.

Background Fact

A Background Fact is external data that will enrich the content of the CAST Application Analytics Dashboard and also the legacy CAST Engineering Dashboard. Background Facts can therefore provide additional information in a single interface that is not based on source code analysis but that can be advantageously cross-referenced with quality and quantity information. These metrics are numerical values and they are attached to Modules only. To determine the Background Fact value of an Application, the dashboards will display the sum total of all contained Modules.

Best Practices

Best Practices are business-oriented strategic quality indicators (i.e. Business Criteria). They rely on the measure of compliance with a set of Technical Criteria that assess the impact on the application development business. CAST uses three main Best Practices to grade an application's source code:

- Architectural Design
- Documentation
- Programming Practices

Bookmark

A Diagnosis Finding that locates text in a component.

Business Criteria

Business Criteria are strategic quality indicators, either business oriented, or development oriented. They rely on the measurement of compliance with a set of specific Technical Criteria that assess the impact on the application development business - as Health Factors - or the compliance to development Best Practices - as Rule Compliance. Their grade is based upon the weights of contributing Technical Criteria grades. The following are Business Criteria:

Health Factor Business Criteria:

- Changeability
- Efficiency
- Robustness
- Security
- Transferability
- Total Quality Index (TQI)

Best Practices Business Criteria:

- Architectural Design
- Documentation
- Programming Practices

Back to top

C

Central or Central Site

See Dashboard Service database.

Changeability (Health Factor)

Changeability is a software characteristic that measures how flexible and adaptable the application is when it is getting enhanced. If an application has low Changeability, that probably means it has a lot of spaghetti code, it's not very well structured, it's not well documented and it's overly complex. The primary reason why IT organizations are slow in responding to business needs is that most systems of record and differentiation, which need to support new business rules, have low Changeability. The Changeability Health Factor is expressed as an index from 1 to 4, with 4 indicating the most flexible application. The grade is calculated based on the average of a list of Technical Criteria linked to a list of specific Quality Rules.

The biggest benefit to improving changeability is increasing future speed of deliver. Other benefits include:

- Faster response to business demands and shorter time to market
- Lower maintenance costs
- Lower sourcing costs
- More predictable project estimates

CISQ

The Consortium for IT Software Quality (CISQ) is an IT industry leadership group comprised of IT executives from the Global 2000,system integrators, outsourced service providers, and software technology vendors committed to introduce a computable metrics standard for measuring software Quality & Size. CISQ is a neutral, open forum in which customers and suppliers of IT application software can develop an industry-wide agenda of actions for improving IT application quality and reduce cost and risk. www.it-cisq.org. CAST implements CISQ standards in its quality rules - see:

- CISQ-Maintainability
- CISQ-Performance-Efficiency
- CISQ-Reliability
- CISQ-Security

Compliance Ratio

For a Quality Rule, this is a ratio of Successful Checks (= Total Checks minus Failed Checks) and Total Checks. A Compliance Ratio is transformed into a Grade/Score with 4 thresholds (each pair of thresholds define a linear function).

Component

A code fragment or a schema fragment. Fragments are specific to a programming language or a schema language, and specific to analyzers.

Alternative name: Object

Critical Violation

See Violation.

CWE

CWE is a community-developed list of software and hardware weakness types. It serves as a common language, a measuring stick for security tools, and as a baseline for weakness identification, mitigation, and prevention efforts. CAST implements CWE standards in its quality rules - see:

- CWE-2011-Top25
- CWE-2019-Top25
- CWE-2020-Top25
- CWE Latest Version

Back to top

D

Dashboard Service schema

The role of the Dashboard Service schema is to store:

- Analysis results (components, violations, diagnosis findings) for each Snapshot
- Assessment results for each Snapshot
- Assessment Model for each Snapshot
- Assessment results at the level of Functional Module and Application
- The majority of Sizing Results

> **Alternative legacy/deprecated names**: Central or Central Site

Defective Component

A defective component is a Component in violation with a Quality Rule.

(CAST) Delivery Manager Tool

The CAST Delivery Manager Tool is . The CAST Delivery Manager Tool is a standalone application aimed at the person or people responsible for providing source code for analysis (i.e. the Delivery Managers) that entirely manages the discovery, selection, extraction and delivery of source code ready for analysis in the CAST Management Studio.

> **Alternative names**: DMT

Delivery Unit

An organization, such as a contractor or a department, in charge of the delivery of applications. A delivery unit defines the scope of applications for analysis.

Diagnosis Finding

Diagnosis Findings pinpoint statements or properties of the defective component violating a Quality Rule pattern. A Diagnosis Findings can be a Bookmark or a counter. Do not confuse with a Violation.

Diagnosis Procedure

An SQL procedure producing Diagnosis Findings.

Diagnosis Value

A diagnosis value is a specific Diagnosis Finding reported as a value: a counter or a name to reference a source code;

Alternative name: Associated Value

(CAST) Discovery Portal

The CAST Engineering Dashboard is a web application provided as part of CAST AIP (bundled in the same .WAR file as the CAST Engineering Dashboard (legacy)). Its primary usage is to provide a variety of technical "DNA type" information about a company's applications.

- **Alternative names**: CED
- Note that the CAST Discovery Portal is now considered a "legacy" feature. It has been superseded by the CAST Engineering Dashboard.

Back to top

E

Efficiency (Health Factor)

Efficiency is a measure of potential performance and scalability bottlenecks in software. While traditional functional testing can identify some performance issues in applications, most serious efficiency defects manifest in live usage. By using established industry best practices, CAST measures software efficiency by evaluating the complexity of SQL statements, memory management, and use of calls in loops and expensive routines within an application's code. The Efficiency Health Factor is expressed as an index from 1 to 4, with 4 indicating the highest level of efficiency. The grade is calculated based on the average of a list of Technical Criteria linked to a list of specific Quality Rules.

Improving software efficiency can raise business productivity and lower software and operational costs, as well as many other benefits:

- Improve the behavior of customer-facing applications
- Reduce post-production code maintenance costs
- Reduce hardware procurement and maintenance costs

Effort Complexity (EC) and Effort Complexity Level

The Effort Complexity (EC) is the Effort Rate of the Artifact based on its Cost Complexity and its technology. The Artifact EC estimates the complexity of implementing an Artifact or changes to it, based on a composite score of five software metrics that assess the complexity of the software environment in which the Artifact is embedded:

- McCabe Cyclomatic Complexity
- Lines of Code
- Lines of Comment
- Fan-In
- SQL Complexity

Each Artifact EC category gets an EC value, which can be overridden for specific technologies. Defaults are as follows:

- Very High Effort Complexity category: 1.2
- High Effort Complexity category: 0.7
- Moderate Effort Complexity category: 0.2
- Low Effort Complexity category: 0.1

See also CAST Automated Enhancement Points Estimation - AEP.

(CAST) Engineering Dashboard

The CAST Engineering Dashboard is a web application. Its primary usage is for low level, detailed investigation of data stored in the CAST Dashboard Service generated during the analysis/snapshot generation process.

Alternative/depcrecated names: AED, Application Engineering Dashboard

(CAST) Engineering Dashboard (legacy)

The CAST Engineering Dashboard is a web application provided as part of CAST AIP (bundled in the same .WAR file as the CAST Discovery Portal). Its primary usage is for low level, detailed investigation of data stored in the CAST Dashboard Service generated during the analysis/snapshot generation process.

- **Alternative names**: CED
- Note that the CAST Engineering Dashboard (legacy) is now considered a "legacy" feature. It has been superseded by the CAST Engineering Dashboard.

EFP (Enhancement Function Points)

EFP was used by default in CAST AIP 8.1.x and all previous releases to estimate the size of an application revision. Out of the box in CAST AIP 8.2.x and all later releases the alternative measure AEP is used. The EFP measure is still available for use via a manual update in AIP Console or CAST Management Studio. See also CAST OMG-compliant Automated Function Points.

(CAST) Enlighten

CAST Enlighten is a client/server application installed as part of CAST AIP. Its primary usage is to display (graphically) the objects that have been identified during a source code analysis and to display the links that these objects have to other objects in the Application.

Equivalence Ratio

Implementation Points form part of the AEP measure. The second step to compute AETP allows to align Implementation Points (IP) with Function Points in order to provide consistent values for AETP. An Equivalence Ratio (ER) is then calculated to weight the Implementation Points for Artifacts belonging to the technical part of the application:

- ER = AFP / IP func
- AETP = ER x IP tech

See also CAST Automated Enhancement Points Estimation - AEP.

Extension

It is possible to "extend" CAST AIP with an extension to provide additional analysis and measurement capabilities both for technologies that are not supported "out-of-the-box" and for technologies supported in CAST AIP. Additionally, extensions can simply be any add-on (for example a custom Assessment Model) that has been built by a third-party.

External objects

An external object is created by the CAST analyzer during an analysis (as is the case for standard "objects" resulting from an analysis), however, they are associated to the analyzed project source code in an external way. For example, an external object could be an object that is part of a library and which is called by the analyzed source code, but is not analyzed itself. Often, external objects do not have source code stored in the CAST schemas, the objects are instead simply recorded as existing. An external object often appears as greyed out in the CAST Enlighten Object Browser.

Back to top

F

Failed Checks

Number of defective components for a Quality Rule.

CAST AIP 8.1.x

- For an Application, this number is the sum of Failed Checks of all Functional Modules defined in the Application, regardless of whether a component belongs to multiple Functional Modules. In the case of overlapping of Functional Modules, this number is therefore an approximation.
- For a Functional Module, and non-overlapping Analysis Units, this value is the number of defective components

CAST AIP 8.2.x

- For an Application, Functional Module, and non-overlapping Analysis Units, this value is the number of defective components.

Functional Module

Functional Modules are used in CAST AIP to define a logical break down of Application source code into smaller units. Examples are a user defined module or an automatic module such as a "full content module" or a module generated for an Analysis Unit.

Back to top

G

Generated code

Many technologies supported by CAST AIP for analysis include the ability to produce "auto generated code" from templates or other sources. When this auto generated source code is analyzed by CAST AIP, the following is true:

- Auto generated code is analyzed to help understand the entire code being analyzed
- Objects are created from the code and saved in the CAST Analysis Service database (to help trace transactions for example) and these objects are marked as being "generated"
- Any Quality Rule violations that are caused by these "auto generated" objects are not displayed in the CAST dashboards and they do not contribute to grade calculations
- "Generated" objects are excluded from any aggregated metrics (for example Lines of Code (LOC))

The method used by CAST AIP to determine whether source code is "auto generated" is specific to each technology.

Back to top

H

(CAST) Health Dashboard

The CAST Health Dashboard is a web application. Its primary usage is for high level investigation of aggregated data stored in the CAST Measurement Service generated during the analysis/snapshot generation process.

Alternative/deprecated names: AAD, Application Analytics Dashboard

Health Factor (Business Criteria)

Health Factors are business-oriented strategic quality indicators (i.e. Business Criteria). They rely on the measure of compliance with a set of Technical Criteria that assess the impact on the application development business. CAST uses six main Health Factors to grade an application's source code:

- Changeability
- Efficiency
- Robustness
- Security
- Transferability
- Total Quality Index (TQI)

Back to top

I

(CAST) Imaging

CAST Imaging is a software visualization solution for the IT teams to visualize and navigate through the application architecture layer by layer; something similar to Google Earth. It is a single page application that can be accessed by any developer, architect, business executive to get insight into architecture, technologies, frameworks and other functional layers of the applications.

Implementation Points

Implementation Points form part of the AEP measure. The first step to compute Glossary#AETP is to estimate the Implementation Points (IP) for Artifacts in both the technical and functional parts of the application. Implementation Points are counted as follows:

- IP tech = EC tech (for added, deleted, and modified Artifacts)
- IP func = EC func (for all Artifacts)

See also CAST Automated Enhancement Points Estimation - AEP.

Back to top

K

Knowledge Base

See Analysis Service database.

Back to top

L

LISA (Large Intermediate Storage Area)

A location (i.e. a folder) on your local hard drive that is designated for use by the CAST Management Studio to store miscellaneous files generated during the analysis process.

- These files will still exist once the analysis is complete.
- This location is used more particularly in the J2EE and .NET technologies to store data generated when the **User Input Data Flow Security Analysis** feature is activated.
- The location must be capable of receiving a large amount of data (several hundred MB).
- This folder is similar in nature to the Internet Explorer cache.

LTSA (Large Temporary Storage Area)

A location (i.e. a folder) on your local hard drive that is designated for use by the CAST Management Studio to store temporary files generated during the analysis process.

- These files will be **removed** once the analysis is complete.
- This folder is similar in nature to the Windows %TEMP% folder.

Local or Local Site

See Analysis Service database.

Back to top

M

Maintainability Index

Determines the cost and difficulty/ease to maintain an application in the future. Increased maintainability index makes applications cheaper to maintain with more predictable results.

Management Service database

A Management Service database stores configuration options for the CAST Management Studio are related resources.

> **Alternative legacy/deprecated names**: Management, MNGT

Measurement Service database

A Measurement Service database stores consolidated results from one or multiple Dashboard Service databases for use with the CAST Application Analytics Dashboard.

> **Alternative legacy/deprecated names**: Measurement base

(CAST) Management Studio

The CAST Management Studio is a client/server application installed as part of CAST AIP. It is used to manage the entire Application analysis and snapshot generation process.

MIPS

MIPS is short for Million Instructions Per Second). CAST measures this using its MIPS Reduction Index extension.

Module

Modules are executable software components or tightly coupled sets of executable software components (one or more), developed and deployed together, that deliver some of the steps needed by an Application to operate. The modules that together make up application code units. CAST scores can be seen as a result of the second unit of analysis within the application.

Back to top

N

NIST

The NIST (National Institute of Standards and Technology) is a physical sciences laboratory and a non-regulatory agency of the United States Department of Commerce. Its mission is to promote innovation and industrial competitiveness. NIST's activities are organized into laboratory programs that include nanoscale science and technology, engineering, information technology, neutron research, material measurement, and physical measurement. CAST implements NIST standards in its quality rules - see:

- NIST-SP-800-53R4-AC
- NIST-SP-800-53R4-AU
- NIST-SP-800-53R4-CA
- NIST-SP-800-53R4-CM
- NIST-SP-800-53R4-IA
- NIST-SP-800-53R4-SA
- NIST-SP-800-53R4-SC
- NIST-SP-800-53R4-SI

O

OMG-ASCQM

OMG (Object Management Group) ASCQM (Automated Source Code Quality Measures) are calculated from detecting and counting violations of good architectural and coding practices in the source code that could result in unacceptable operational risks or excessive costs. Establishing standards for these measures at the source code level is important because they have been used in outsourcing and system development contracts without having international standards to reference. CAST implements OMG-ASCQM standards in its quality rules - see:

- OMG-ASCQM-Maintainability
- OMG-ASCQM-Performance-Efficiency
- OMG-ASCQM-Reliability
- OMG-ASCQM-Security

OWASP

The Open Web Application Security Project (OWASP) is a nonprofit foundation that works to improve the security of software. Through community-led open source software projects, hundreds of local chapters worldwide, tens of thousands of members, and leading educational and training conferences, the OWASP Foundation is the source for developers and technologists to secure the web. CAST implements OWASP standards in its quality rules - see:

- OWASP-2013
- OWASP-2017
- OWASP-API-2019
- OWASP-Mobile-2016

P

PCI DSS

Payment Card Industry Data Security Standard (PCI DSS) is an information security standard for organizations that handle branded credit cards from the major card schemes. CAST implements PCI DSS standards in its quality rules - see:

- PCI-DSS-V3.1-Req-1
- PCI-DSS-V3.1-Req-2
- PCI-DSS-V3.1-Req-3
- PCI-DSS-V3.1-Req-4
- PCI-DSS-V3.1-Req-5
- PCI-DSS-V3.1-Req-6

- PCI-DSS-V3.1-Req-7
- PCI-DSS-V3.1-Req-8
- PCI-DSS-V3.1-Req-10

Propagated Risk Index (PRI)

Propagated Risk Index (PRI) is a measurement of the riskiest artifacts or objects of the application along the Health Factors of Robustness, Performance and Security. PRI takes into account the intrinsic risk of the component coupled with the level of use of the given object in the transaction. It systematically helps aggregate risk of the application in a relative manner allowing for identification, prioritization, and ultimately re-mediation of the riskiest objects.*

Back to top

Q

Quality Distribution

A Quality Distribution is an operational quality indicator, designed to assess a component based on the balance of the distribution of an attribute value among objects of the component. It relies on the distribution of tested objects according to one of their properties (e.g.: object size) into four categories, for an Application or a Functional Module.

Quality Measure

A Quality Measure is an operational quality indicator, designed to assess a component based on a single measure value (such as % of copy/pasted code) in order to determine a grade between 1.0 (poor) and 4.0 (good) for an Application or a Functional Module.

Quality Rule

A Quality Rule is an operational quality indicator, designed to assess a component based on the compliance to a coding or architecture practice. A Quality Rule is defined for a single technology or a set of technologies (unified which is the default) and produces a grade between 1.0 (very high risk) and 4.0 (low risk) for an Application or a Functional Module.

Back to top

R

Result

An assessment result of an Application or a Functional Module.

Result Detail

Additional values, indicators, related to a Result:

- intermediate calculation results
- breakdown of a measure
- related quantitative values

Robustness (Health Factor)

Robustness is an indication of the likelihood that an application will incur defects, corrupt data or completely fail in production. Often referred to as "resilience", CAST's Robustness measure is based on industry best practices around algorithmic and control flow complexity, controlled data access at an architectural level, architectural object-oriented design, error and exception handling, and the level of coupling and inter-dependency. The Robustness measure also evaluates the ease with which an application can be tested for defects. CAST expresses the Robustness Health Factor as an index from 1 to 4, with 4 indicating the highest level of robustness. The grade is calculated based on the average of a list of Technical Criteria linked to a list of specific Quality Rules.

While improving the robustness of critical business applications will reduce incidents that impact user satisfaction, there are also several additional tangible benefits:

- Improve customer satisfaction
- Extend business continuity
- Reduce support and defect recovery costs
- Help make the software more secure
- Maximize revenue generation opportunities

Back to top

S

Security (Health Factor)

Security measures the likelihood of potential security breaches linked to coding practices and application source code. CAST expresses the Security Health Factor as an index from 1 to 4, with 4 indicating the highest level of security. The grade is calculated based on the average of a list of Technical Criteria linked to a list of specific Quality Rules.

(CAST) Server Manager

CAST Server Manager is a traditional client/server application installed as part of CAST AIP. It is used to install CAST AIP schemas and CAST AIP extensions.

Sizing Measure

A quantitative measure.

Snapshot

A CAST Snapshot is a capture at one moment in time of the status of a set of executable software components (one or more). The scope of a Snapshot depends on the nature of the information that is captured.

Source Code Delivery Folder

Location for storing successive and compressed versions of applications' source code as packaged by the Delivery Manager(s).

Source Code Deployment Folder

Location of the most recent version of the applications' source code for analysis in uncompressed format.

(CAST) Storage Service

The CAST Storage Service is a RDBMS provided as part of CAST AIP and can be used to host the CAST AIP schemas. The CAST Storage Service is a repackaged PostgreSQL RDBMS.

Alternative names: CSS

STIG

Security Technical Implementation Guide (STIG) is a cybersecurity methodology for standardizing security protocols within networks, servers, computers, and logical designs to enhance overall security. These guides, when implemented, enhance security for software, hardware, physical and logical architectures to further reduce vulnerabilities. CAST implements STIG standards in its quality rules - see:

- STIG-V4R8-CAT1
- STIG-V4R8-CAT2
- STIG-V4R8-CAT3

Back to top

T

Technical Criteria

Technical Criteria are operational quality indicators, designed to assess a technical area. They rely on the measurement of compliance with a set of specific Quality Rules, Distributions, and Measures that assess a specific technical domain or area. Their grade is based upon contributing Quality Rules, Quality Distributions and Quality Measures grades.

Technical Debt

Also known as Design Debt is the accumulated amount/cost of rework that will be necessary to correct and/or recover from the deviation between the current design of the system, versus that which is minimally complex yet sufficiently complete to ensure correctness & consistency for timely delivery. This effort grows more than linearly over time as a system becomes bigger and more complex.

Total Quality Index (Health Factor)

The Total Quality Index (TQI) measures the general maintainability level of the application. CAST expresses the TQI Health Factor as an index from 1 to 4, with 4 indicating the highest level of maintainability. TQI differs slightly in comparison to other Health Factors - it is an average of ALL the available Technical Criteria linked to all Quality Rules provided by CAST (other Health Factors have specific contributing Technical Criteria and Quality Rules).

Total Checks

Number of components that have been checked for a specific Quality Rule.

(CAST) Transaction Configuration Center

The CAST Transaction Configuration Center is a client/server application installed as part of CAST AIP. Its primary usage is for calibrating the initial Function Point count made by CAST AIP during an analysis. Calibration includes removing technical and temporary objects from the list of Function Points counted, aggregating and splitting several function points into one and changing the type of the Data or Transactional Functions.

Alternative names: TCC

TQI

See Total Quality Index (Health Factor).

Transaction Risk Index (TRI)

TRI is an indicator of the riskiest transactions of the application. The TRI number reflects the cumulative risk of the transaction based on the risk in the individual objects contributing to the transaction. The TRI is calculated as a function of the rules violated, their weight/criticality, and the frequency of the violation across all objects in the path of the transaction. TRI is a powerful metric to identify, prioritize and ultimately remediate riskiest transactions and their objects.

Transferability (Health Factor)

Transferability measures how easily applications can be moved across teams or team members including in-house and outsourced development teams. CAST expresses the Transferability Health Factor as an index from 1 to 4, with 4 indicating the highest level of transferability. The grade is calculated based on the average of a list of Technical Criteria linked to a list of specific Quality Rules.

Back to top

U

Unadjusted Data Functions

Unadjusted Data Functions = sum of (Function Points of all Data Entities). See also CAST OMG-compliant Automated Function Points.

Unadjusted Function Points

Unadjusted Transactional Functions = Sum of (Function Points of all User Forms). See also CAST OMG-compliant Automated Function Points.

Unify

A Quality Rule unifying a set of alternative Quality Rules; each alternative Quality Rule is defined for a single technology. For example "7166 - Avoid Artifacts with High Cyclomatic Complexity" gathers all violations of quality rules (666,1118,1652,2296,2646,3138,3654,4148,4780,5134,5580,6186,6618,7112). Note that these rules (flagged with unify=false) are always disabled.

Back to top

V

Violation

A violation identifies a defective component breaking a Quality Rule pattern.

IMPORTANT: For a given component and a given Quality Rule pattern there is 0 or 1 violations. If a component breaks a rule N times, then each occurrence is detailed into the Diagnosis Findings structure with a value counter equal to N, and/or with N values, and/or with N code bookmarks.

Critical Violations, i.e., violations to critical Quality Rules, identify each single occurrence of situations that can jeopardize the application regarding their Robustness, Performance, Security, Transferability, or Changeability. The consequences are so dire that:

- Each single occurrence are accounted for and made highly visible via specific indicators within CAST dashboards and portals;
- Poor results for a critical quality rule can not be traded against good results regarding for other quality rules, no matter how many and how good these latter are.

CAST AIP delivers pre-configured sets of **critical violations**, but these can be adapted to each organization's context.

Violation Index

Violation Index (VI) assesses the overall quality of an object regarding a health concern (Robustness, Security,…), by a weighted aggregation on the Glossary#violation it carries.

Violation Pattern

The Violation Pattern is the pattern that is searched for in the Analysis Service content (source code, cartography, etc.) to pinpoint Violations. The Violation Pattern should be described in the description field of the Quality Rule.

Back to top