

Assessment Model adaptation options

On this page

- [Quality Model adaption options](#)
 - [Aggregated quality indicators - Business and Technical Criteria](#)
 - [Constitutive elements](#)
 - [Contribution weights](#)
 - [Critical contribution](#)
 - [Active status](#)
 - [Execution exceptions](#)
 - [Elementary quality indicators - Quality Rules, Distributions, and Measures](#)
 - [Execution exceptions](#)
 - [Parameter values](#)
 - [Status thresholds](#)
 - [Module Weights](#)
- [Sizing Model adaptation options](#)
 - [Function Points configuration](#)
 - [Backfired Function Points](#)
 - [Automated Function Points](#)
 - [Unadjusted Data Function Points](#)
 - [Unadjusted Transactional Function Points](#)

Summary: Assessment Model adaptation designates all the changes that one can perform on the existing configuration. Please refer to the [Assessment Model enrichment](#) chapter to handle the addition of new indicators.

Quality Model adaption options

Aggregated quality indicators - Business and Technical Criteria

Within the CAST Assessment Model, quality indicator aggregations are always performed according to the same principles, regardless of the level in the Assessment Model. The principles are the weighted average of lower-level values, with the optional ability to cap resulting grade with the value of a "critical" contributor's value.

However, for organization and restitution principles, aggregated quality indicators are named differently. They are Business Criteria and Technical Criteria:

- Business Criteria are referred to as 'Health Factors' when they are business-oriented and 'Rule Compliance' when they are development-oriented.
- Each Business Criterion is the aggregation of multiple Technical Criteria.
- Each Technical Criterion is the aggregation of multiple elementary quality indicators, be they Quality Rules, Quality Distributions, or Quality Measures.

The default CAST standard Assessment Model is installed within each new Dashboard and Analysis Service. To start handling it into the Management Studio, please refer to [Assessment Model configuration access](#)

Constitutive elements

A default list of constitutive quality indicators of each aggregated quality indicator are delivered with the standard CAST Assessment Model, however, they can be adapted if some contributions are missing or considered as inappropriate in a given context.

Contribution weights

A default list of contribution weights are delivered with the standard CAST Assessment Model, however, contribution weights can be adapted if some contributions are thought more or less important in a given context.

Critical contribution

The contribution of a quality indicator into an aggregated quality indicator can be tagged with the "critical" option. This option allows to threshold the grade of the aggregated quality indicator with the lowest grade of its "critical" contribution.

This feature is optional yet its usage leads to better results in the post-6.1-release portal pages: a careful selection of diagnostic-based quality indicators has to be done to deliver the best value in these pages.

In case you work with Services that have been migrated over time from pre-6.0 releases, you may find yourself in a position where there is no "critical" option set (as migrations did not change configurations of existing quality indicators). In this case, it is highly recommended to proceed with such a selection.

Active status

A default list of active quality indicators are delivered with the standard CAST Assessment Model, however, active option can be adapted if some indicators are not to be computed in a given context.

Quality indicators that require further configuration of parameters and mutually exclusive quality indicators are delivered as not active so as not to disturb the quality assessment if not properly configured. Once configured, you can set the active option value back to "true".

Execution exceptions

Elementary quality indicators, that is Quality Rules natively handle technology-oriented information: the list of technologies the Quality Rule is applicable for.

However, some quality indicator sets may not be relevant to a given Module, Application or System for historical or functional reasons.

E.g.: a legacy Cobol Module is known to be of poor quality regarding its algorithmic complexity; this Module won't be evolved; there is therefore no need to spend resources on checking its quality; the Module is nevertheless part of the Portfolio Tree for inventory reasons; you can exclude this Module. In this context, a given Module, Application or System must be excluded from the computation so as to reflect the situation.

Elementary quality indicators - Quality Rules, Distributions, and Measures

Within the CAST Assessment Model, the elementary quality indicators are Quality Rules, Quality Distributions, and Quality Measures.

Execution exceptions

Aggregated quality indicators' large-grain execution exception principles are also valid for elementary quality indicators.

E.g.: Reuse by call distribution delivers best values on framework Modules; knowing which Modules are supposed to be framework is external to source code analysis; other Modules may not need to be assessed with this quality indicator; you can exclude these other Modules.

E.g.: Corporate naming policy handles project-specific prefix controls; you may want to run different prefix controls on different applications; you can then create control variations and run them where appropriate.

In this context, a given Module, Application, or System must be excluded from the computation so as to reflect the situation.

In some cases, objects within a Module may not be subject to the development rules for some historical or functional reasons.

E.g.: some objects do violate a rule but this situation is accepted on behalf of its functional specificity.

In this context, an object must be excluded from the computation so as to reflect the situation. This object-level exclusion is handled through the pages of the CAST Engineering Dashboard.

Parameter values

Quality Rules, Quality Distributions, and Quality Measures can feed (when appropriate) on parameters whose values impact the results - grades and statuses - of the Quality indicators.

E.g.:

1. define a defect for a Diagnostic-based Quality indicator as items of a given type with a value of a numerical property greater than the value of a user-defined parameter;
2. define the way to distribute objects of a given type into the Categories of a Quality Distribution according to the value of a numerical property greater than the value of a first user-defined parameter and lower or equal to the value of a second user-defined parameter.

Updating parameter values is part of the normal use of the product to better match the target context.

- Parameters can be used by multiple quality indicators to ensure value consistency.
- **Adding a parameter or changing the nature of the parameter (from a single value to a list of value for instance, or from a default value to technology-dependent values) will require an update in the implementation of the quality indicator to be sure it is handled properly. Running a test snapshot is also highly recommended to ensure the updated behavior is the expected one.**

Status thresholds

The thresholds that are used to turn percentages of rule compliance, split share percentage of distribution categories, and values into quality grades (respectively for Quality Rules, Quality Distributions, and Quality Measures) can also be updated.

Updating the status thresholds should be reserved for experienced users when default behavior does not meet with expectations.



As a general guideline, CAST Quality Rules use one of the following two threshold profile:

- For Quality Rules with a critical contribution into a Technical Criterion, a **"strict"** threshold profile:
 - Threshold to get more than 1.00 grade: 98%
 - Threshold to get more than 2.00 grade: 99%
 - Threshold to get more than 3.00 grade: 99.5%
 - Threshold to get 4.00 grade: 99.99%
- For other Quality Rules, a **"lenient"** threshold profile:
 - Threshold to get more than 1.00 grade: 50%
 - Threshold to get more than 2.00 grade: 90%
 - Threshold to get more than 3.00 grade: 95%
 - Threshold to get 4.00 grade: 99%

Exceptions and differences can occur when needed. However, this is a good practice to align on these two profiles.

Module Weights

One can decide of the weight of a Module with regards to other Modules when it comes to aggregate the result of multiple Modules into an APplication. As a default behavior, each Module had the exact same weight, meaning that every Module was as important as the other ones. One can decide to change this behavior to weight a Module according to one of its Technical Size sizing indicator value, one of its Functional Weight sizing indicator value, one of its uploaded Background Facts value, or its uploaded Business Value. This feature is optional. This is especially true as users can define the Module the way they want. The Modules can be built so as to have the same weight.

When to use it? As soon as the results using the default behavior leads to comments such as "this Module is more important than this one so the Application grade should be impacted by its grade" or "this Module is bigger than this one so the Application grade should be impacted by its grade", this means there is a need to weight Modules a customized way.

The best candidates are:

1. the "Business Value" to have critical components impact the overall quality assessment,
2. the "number of Backfired Function Points" to have the functional weight of components impact the overall quality assessment (valid for all technologies),
3. the "number of Artifacts" to have the technical size impact the overall quality assessment (valid for all technologies),
4. and the "number of LOC" (pro's: anyone can understand, con's: some components have no LOC and if you build Modules with these components alone, the Modules will not impact the overall quality assessment)

Sizing Model adaptation options

Function Points configuration

Backfired Function Points

Starting with Contrex (7.1) release , Backfired Function Points configuration is performed from CAST Management Studio.

The "Backfired Functions Points" quality indicator features parameters to tune the 'FP / LOC' ratio for each technology. Default values are fine but be sure, in case a new language has been added, that there is a valid value for it. Otherwise, the <any type> value will be used.

Automated Function Points

OMG-compliant Automated Function Points estimation configuration requires the use of CAST Transaction Configuration Center and a valid Function Point computation license key.

For more details about Automated FP, please refer to "CAST Automated Function Points Estimation" appendix.

Unadjusted Data Function Points

Unadjusted Data FP computing is based on database Tables and on objects declared as database mapping. If the analysis does not contain any such data structure, the quality indicator will return zero FP.

Unadjusted Transactional Function Points

Unadjusted Transactional FP computing is based on database Tables and on user-facing forms or screens (User Interface elements). In case the language does not feature native form or screen concepts, user-facing components are found using naming conventions or base classes that user-facing components inherit from.

If the analysis does not contain any such data structure, the quality indicator will return zero FP; if the analysis does not contain any language with native form or screen concept and user-facing components selection is not configured, the quality indicator will return zero FP.