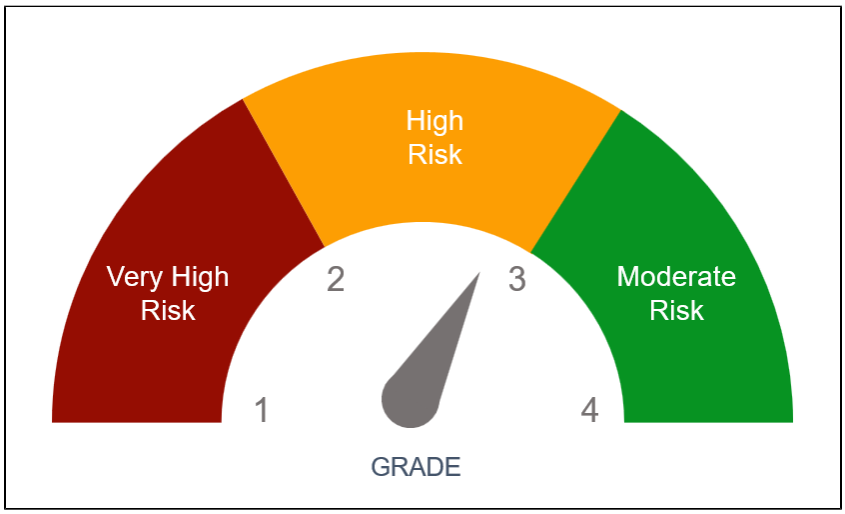
Grade and compliance score calculation

What is a grade?

A grade is a score between 1.00 and 4.00 to assess a risk:



The rule grade is calculated for:

- each functional module
- each functional module split by technology
- application: in this case the application grade comes from the representative functional module which is:
  - either the Full Content Module if it exists
  - or the union content module

What does it mean when a grade equals to zero?

A rule with **zero weight** is a rule that is both enabled (active), but has no impact on any parent technical criterion - this means that the rule can be "previewed" (i.e. violations can be seen) without impacting any grades. Configuring a rule with a weight of 0 is supported when using **AIP Core 8.3.32** - you can do this using the **Assessment Model** interface in **AIP Console**. A zero value for a technical criterion grade means that all child rules contribute to this grade with a zero weight. A zero grade is not an assessment value, it is a result allowing a navigation to the child rules and their violations in "preview" mode. So a zero grade means "no value" for this technical criterion.

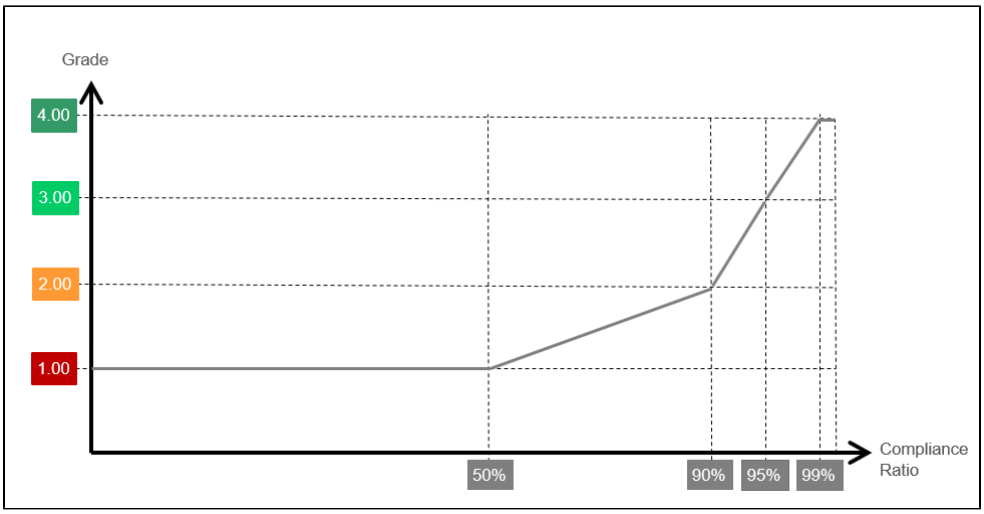
Rules with a **0 weight** or a **0 grade** are never consolidated into the **Measure schema**, so will not appear in the Health Dashboard.

When a rule is disabled (inactive), no result is inserted into the Dashboard schema for this rule, so there is also no impact on the parent technical criterion.

Calculation of a rule grade

A rule grade is calculated from the compliance ratio as a point of the following curve:

Click to enlarge



The curve is defined as four linear segments, bounded with thresholds. The input parameters are:

- The thresholds: four decimal values. If not defined the default values are [10.0, 70.0, 90.0, 99.0].
- The number of failed checks (the number of violations)
- The total number of checks

The grade is calculated as follows:

```
// From AIP 8.3.29
function computeRuleGrade (nbFailedChecks, nbTotalChecks, thresholds)
{
    // Descending thresholds are not taken into account here, as there are no longer used
    if (nbFailedChecks == 0 || nbTotalChecks == 0) return 4.0 ; // wrong rule
    if (nbTotalChecks <= nbFailedChecks) return 1.0; // wrong rule or nbFailedCheks == nbTotalChecks
    let violationRatio = nbFailedChecks / nbTotalChecks;
    let complianceRatio = (100.0 - violationRatio * 100.0);
    if (complianceRatio < thresholds[0]) return 1.0;
    if (complianceRatio < thresholds[1]) return 1.0 + (complianceRatio - thresholds[0]) / (thresholds[1] -
thresholds[0]);
    if (complianceRatio < thresholds[2]) return 2.0 + (complianceRatio - thresholds[1]) / (thresholds[2] -
thresholds[1]);
    if (complianceRatio < thresholds[3]) return 3.0 + (complianceRatio - thresholds[2]) / (thresholds[3] -
thresholds[2]);
    return 4.0;
}
```

Calculation of a Distribution grade

For each distribution, there are four categories:

- Very high
- High
- Moderate
- Low

Each object is distributed into a category. AIP counts the number of items in each category:

- count1
- count2
- count3
- count4

count_total = count1 + count2 + count3 + count4

AIP computes four grades for each category:

- grade1 = computeDistGrade(count1/count_total, thresholds of category 1)
- grade2 = computeDistGrade(count2/count_total, thresholds of category 2)
- grade3 = computeDistGrade(count3/count_total, thresholds of category 3)
- grade4 = computeDistGrade(count4/count_total, thresholds of category 4)

The distribution grade is MIN (grade1, grade2, grade3, grade4).

Calculation of a Technical Criterion grade

The grade of a technical criterion is calculated from all the rule contributors of the technical criterion:

$$TC_k(M) = \min\left(\frac{\sum_{i=1}^{n}\left(w_i\ QI_i(M)\right)}{\sum_{i=1}^{n}(w_i)}, \text{critical}(\ QI_j(M)), \dots\right) \text{ with } QI_i \text{ contributor to } TC_k \text{ with } w_i$$

These quality indicators are:

- Rules
- Distributions
- Measures

The input parameters are:

- An array of contributions; each contribution is defined with:
  - a grade
  - a contribution weigth
  - a contribution critical flag

The grade is calculated as follow:

```
// From AIP 8.3.29
function computeCriterionGrade (contributions)
{
    // Calculate weighted average of all contributions
    // Calculate the minimum critical contribution grade
    // Then return the Min
    let sum = 0.0;
    let totalWeights = 0.0;
    let minCriticalGrade = 4.0;
    for (let i = 0, len = contributions.length; i < len; i++)
    {
        let contribution = contributions[i];
        let weight = contribution.weight;
        let grade = contribution.grade;
        if (weight == 0.0) continue; // do not take into account zero weight
        if (!grade) continue; // do not take into account null/zero grade (when all contributors have a zero
weight)
        if (contribution.critical && grade < minCriticalGrade)
           minCriticalGrade = grade;
        sum += weight * grade;
        totalWeights += weight;
    }
    if (totalWeights == 0.0) return 0.0; // grade is 0.0 when all weights are zero
    let weightedAverage = (sum / totalWeights);
    return Math.min (minCriticalGrade, weightedAverage);
}
```

Calculation of a Business Criterion grade

The grade of a business criterion is calculated from all the Technical Criteria contributors of the business criterion:

$$BC_k(M) = \frac{\sum_{i=1}^{n}\left(w_i\ TC_i(M)\right)}{\sum_{i=1}^{n}(w_i)}, \text{ with } TC_i \text{ contributor to } BC_k \text{ with } w_i$$

The input parameters are:

- An array of contributions; each contribution is defined with
    - a grade
    - a contribution weight
    - a contribution critical flag

The grade is calculated also with the "computeCriterionGrade" algorithm, except that there is no critical contribution to a business criterion.

Calculation of the Compliance Scores

Compliance score for a rule:

```
// Compliance score is introduced since 8.3.29.
// if the compliance score cannot be calculated we return the null value, so that it is not taken into account
for parent quality indicators
function computeRuleComplianceScore (nbFailedChecks, nbTotalChecks)
{
    if (nbFailedChecks == 0 || nbTotalChecks == 0) return null ; // wrong rule
    if (nbTotalChecks <= nbFailedChecks) return null; // wrong rule or nbFailedCheks == nbTotalChecks
    return (1.0 - nbFailedChecks / nbTotalChecks);
}
```

For technical criteria, and business criteria:

```
// From AIP 8.3.29
function computeComplianceScore (contributions)
{
    // Calculate average of all contributions
    let sum = 0.0;
    let totalWeights = 0.0;
    for (let i = 0, len = contributions.length; i < len; i++)
    {
        let contribution = contributions[i];
        let weight = contribution.weight;
        let score = contribution.complianceScore;
        if (weight == 0.0) continue; // do not take into account zero weight
        if (!grade) continue; // do not take into account null/zero grade (when all contributors have a zero
weight)
        sum += score ;
        totalWeights += weight;
    }
    if (totalWeights == 0.0) return null; // score is null when all weights are zero
    return (sum / contributions.length); // simple average
}
```