

# Technical Debt - calculation and modification

- Calculation
  - Total Technical Debt per Application
  - Technical Debt Added in Current Release per Application
  - Technical Debt Removed in Current Release per Application
- Calculation - variable description
- Display of Technical Debt values
- Modifying Technical Debt calculation variables
- Known limitations
  - Technical Debt values at functional module level when objects belong to multiple technologies

**i** **Summary:** The sections below explain how the Technical Debt value is calculated and how the default values can be modified if necessary. Technical Debt configuration is undertaken in the **CAST Management Studio**, via the **Assessment Models editor**. Note that an extension exists to calculate **OMG Technical Debt** in addition to Technical Debt described below.

## Calculation

Technical Debt is calculated on an Application basis, as follows:

- i** Note that **low, medium and high severity violations** refer to the **weight** of a violated Quality Rule as defined in the Assessment Model:
- Low severity = Weight of 1, 2 or 3
  - Medium severity = Weight of 4, 5 or 6
  - High severity = Weight of 7, 8 or 9

## Total Technical Debt per Application

```
( (% of low severity violations to be fixed X # of low severity violations in Application) X  
(Weighted time, in hours, for fixing low severity violations) +  
(% of medium severity violations to be fixed X # of medium severity violations in Application) X  
(Weighted time, in hours, for fixing medium severity violations) +  
(% of high severity violations to be fixed X # of high severity violations in Application) X  
(Weighted time, in hours, for fixing high severity violations) ) X  
Cost per staff hour to fix violations
```

## Technical Debt Added in Current Release per Application

```
( (% of low severity violations to be fixed X # of low severity violations added in current release of  
Application) X  
(Weighted time, in hours, for fixing low severity violations) +  
(% of medium severity violations to be fixed X # of medium severity violations added in current release of  
Application) X  
(Weighted time, in hours, for fixing medium severity violations) +  
(% of high severity violations to be fixed X # of high severity violations added in current release of  
Application) X  
(Weighted time, in hours, for fixing high severity violations) ) X  
Cost per staff hour to fix violations
```



### About Added violations

Added Violations with low/medium/high severity are violations that:

1. involve added and updated objects
2. are not present in the Application in the previous Snapshot
3. involve a Quality Rule with the appropriate Weight

This means that:

- some configuration changes can cause the number of Violations of a given Severity to change without "Adding Violations". E.g.: Changing the Weight of a Quality Rule between Snapshots can cause a change in the number of low/medium/high severity violations without any specific "Added Violation"
- some Objects with Violations but without a relevant checksum (i.e. a value of 0) do not show in the Technical Debt Added (note that checksum values are calculated by CAST AIP for objects resulting from an analysis and are used to determine whether an object has changed between successive snapshots, however, CAST AIP is not able to determine the checksum for certain objects and these objects always have a checksum value of 0).

**In these cases, Technical Debt amount in new Snapshot is different from Technical Debt amount in previous Snapshot + Technical Debt Added amount - Technical Debt Removed amount**

## Technical Debt Removed in Current Release per Application

```
( (% of low severity violations to be fixed X # of low severity violations removed in current release of
Application) X
(Weighted time, in hours, for fixing low severity violations) +
(% of medium severity violations to be fixed X # of medium severity violations removed in current release of
Application) X
(Weighted time, in hours, for fixing medium severity violations) +
(% of high severity violations to be fixed X # of high severity violations removed in current release of
Application) X
(Weighted time, in hours, for fixing high severity violations) ) X
Cost per staff hour to fix violations
```



### About Removed violations

Removed Violations with low/medium/high severity are violations that:

1. involve removed and updated Objects
2. are not present anymore in the Application in the current Snapshot
3. involve a Quality Rule with the appropriate Weight

This means that

- some configuration change can cause the number of Violations of a given Severity to change without "Removing Violations". E.g.: Changing the Weight of a Quality Rule between Snapshots can cause a change in the number of low/medium/high severity violations without any "Removed Violation"
- some Objects with Violations but without a relevant checksum (i.e. a value of 0) do not show in the Technical Debt Removed (note that checksum values are calculated by CAST AIP for objects resulting from an analysis and are used to determine whether an object has changed between successive snapshots, however, CAST AIP is not able to determine the checksum for certain objects and these objects always have a checksum value of 0).
- **Excluded** Violations do not show in the Technical Debt Removed

**In these cases, Technical Debt amount in new Snapshot is different from Technical Debt amount in previous Snapshot + Technical Debt Added amount - Technical Debt Removed amount**

## Calculation - variable description

The variables listed in the calculation above are described below.



Please see the section below entitled **Modifying Technical Debt calculation variables** about modifying any of the variables marked as Configurable = Yes.

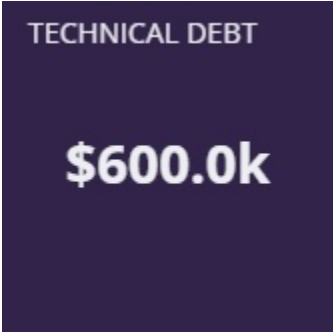
Variable Name	Description	Configurable	Default Value
---------------	-------------	--------------	---------------

<b>% of low severity violations to be fixed</b>	<b>Only a portion of the low severity violations will be fixed</b>	<b>Yes</b>	<b>0%</b>
# of low severity violations	Actual # of low severity (level 1,2,3) violations across all health factors	No (comes directly from analysis)	Not Applicable
# of low severity violations added in current release	Actual # of low severity (level 1,2,3) violations across all health factors added in current release	No (comes directly from analysis)	Not Applicable
# of low severity violations removed in current release	Actual # of low severity (level 1,2,3) violations across all health factors removed in current release	No (comes directly from analysis)	Not Applicable
<b>% of medium severity violations</b>	<b>Only a portion of the medium severity violations will be fixed</b>	<b>Yes</b>	<b>50%</b>
# of medium severity violations	Actual # of medium severity (level 4,5,6) violations across all health factors	No (comes directly from analysis)	Not Applicable
# of medium severity violations added in current release	Actual # of medium severity (level 4,5,6) violations across all health factors added in current release	No (comes directly from analysis)	Not Applicable
# of medium severity violations removed in current release	Actual # of medium severity (level 4,5,6) violations across all health factors added in current release	No (comes directly from analysis)	Not Applicable
<b>% of high severity violations</b>	<b>Only a portion of the high severity violations will be fixed</b>	<b>Yes</b>	<b>100%</b>
# of high severity violations	Actual # of high severity (level 7,8,9) violations across all health factors	No (comes directly from analysis)	Not Applicable
# of high severity violations added in current release	Actual # of high severity (level 7,8,9) violations across all health factors added in current release	No (comes directly from analysis)	Not Applicable
# of high severity violations removed in current release	Actual # of high severity (level 7,8,9) violations across all health factors added in current release	No (comes directly from analysis)	Not Applicable
<b>Weighted time, in hours, for fixing LOW severity violation</b>	<p><b>Not all violations will need the same amount of time, hence we take the weighted time to fix the violations.</b></p> <p>Weighted based on the distribution of level of difficulty to fix violations. Violations will be categorized as follows:</p> <ul style="list-style-type: none"> <li>- Easy</li> <li>- Hard</li> <li>- Very Hard</li> </ul> <p>Wt. time to fix low severity violations can be given a new value using the variables described below:</p> <p>- (Low_%Easy X Low_Time_Easy) + (Low_%Hard+ X Low_Time_Hard) + (Low_%Very_Hard X Low_Time_Very_Hard)</p> <p>For example using the variables below:</p> <p><b>(0.90 X 0.5) + (0.09 x 1) + (0.01 x 8) = 0.62</b></p>	<b>Yes</b>	<b>0.62 hours</b>
	<i>Low_%Easy = % of violations which are "Easy"</i>	<i>N/A (intermediate computing)</i>	<i>90%</i>
	<i>Low_Time_Easy = Time take for fixing "Easy" violations</i>	<i>N/A (intermediate computing)</i>	<i>0.5 hour</i>
	<i>Low_%Hard = % of violations which are "Hard"</i>	<i>N/A (intermediate computing)</i>	<i>9%</i>
	<i>Low_Time_Hard = Time take for fixing "Hard" violations</i>	<i>N/A (intermediate computing)</i>	<i>1 hour</i>
	<i>Low_%Very_Hard = % of violations which are "Very_Hard"</i>	<i>N/A (intermediate computing)</i>	<i>1%</i>
	<i>Low_Time_Very_Hard = Time take for fixing "Very_Hard" violations</i>	<i>N/A (intermediate computing)</i>	<i>8 hours</i>

<b>Weighted time, in hours, for fixing MEDIUM severity violation</b>	<p><b>Not all violations will need the same amount of time, hence we take the weighted time to fix the violations.</b></p> <p>Weighted based on the distribution of level of difficulty to fix violations. Violations will be categorized as follows:</p> <ul style="list-style-type: none"> <li>- Easy</li> <li>- Hard</li> <li>- Very Hard</li> </ul> <p>Wt. time to fix medium severity violations can be given a new value using the variables described below:</p> <p>- (Medium_%Easy X Medium_Time_Easy) + (Medium_%Hard X Medium_Time_Hard) + (Medium_%Very_Hard X Medium_Time_Very_Hard)</p> <p>For example using the variables below:</p> <p><b>(0.90 X 0.5) + (0.09 x 4) + (0.01 x 16) = 0.97</b></p>	<b>Yes</b>	<b>0.97 hours</b>
	<i>Medium_%Easy = % of violations which are "Easy"</i>	<i>N/A (intermediate computing)</i>	<i>90%</i>
	<i>Medium_Time_Easy = Time take for fixing "Easy" violations</i>	<i>N/A (intermediate computing)</i>	<i>0.5 hour</i>
	<i>Medium_%Hard = % of violations which are "Hard"</i>	<i>N/A (intermediate computing)</i>	<i>9%</i>
	<i>Medium_Time_Hard = Time take for fixing "Hard" violations</i>	<i>N/A (intermediate computing)</i>	<i>4 hour</i>
	<i>Medium_%Very_Hard = % of violations which are "Very_Hard"</i>	<i>N/A (intermediate computing)</i>	<i>1%</i>
	<i>Medium_Time_Very_Hard = Time take for fixing "Very_Hard" violations</i>	<i>N/A (intermediate computing)</i>	<i>16 hours</i>
<b>Weighted time, in hours, for fixing HIGH severity violation</b>	<p><b>Not all violations will need the same amount of time, hence we take the weighted time to fix the violations.</b></p> <p>Weighted based on the distribution of level of difficulty to fix violations. Violations will be categorized as follows:</p> <ul style="list-style-type: none"> <li>- Easy</li> <li>- Hard</li> <li>- Very Hard</li> </ul> <p>Wt. time to fix high severity violations can be given a new value using the variables described below:</p> <p>- (High_%Easy X High_Time_Easy) + (High_%Hard X High_Time_Hard) + (High_%Very_Hard X High_Time_Very_Hard)</p> <p><b>(0.80 X 1) + (0.19 x 8) + (0.01 x 24) = 2.56</b></p>	<b>Yes</b>	<b>2.56 hours</b>
	<i>High_%Easy = % of violations which are "Easy"</i>	<i>N/A (intermediate computing)</i>	<i>80%</i>
	<i>High_Time_Easy = Time take for fixing "Easy" violations</i>	<i>N/A (intermediate computing)</i>	<i>1 hour</i>
	<i>High_%Hard = % of violations which are "Hard"</i>	<i>N/A (intermediate computing)</i>	<i>19%</i>
	<i>High_Time_Hard = Time take for fixing "Hard" violations</i>	<i>N/A (intermediate computing)</i>	<i>8 hours</i>
	<i>High_%Very_Hard = % of violations which are "Very_Hard"</i>	<i>N/A (intermediate computing)</i>	<i>1%</i>
	<i>High_Time_Very_Hard = Time take for fixing "Very_Hard" violations</i>	<i>N/A (intermediate computing)</i>	<i>24 hours</i>
<b>Cost per hour of developer time</b>	<b>Blended rate of different people who may work on a violation (architect, lead, developer,QA resource etc.)</b>	<b>Yes</b>	<b>\$75/hr</b>

## Display of Technical Debt values

The Health Dashboard has a default tile that displays the Technical Debt value at multi-application (portfolio) and single-application level:

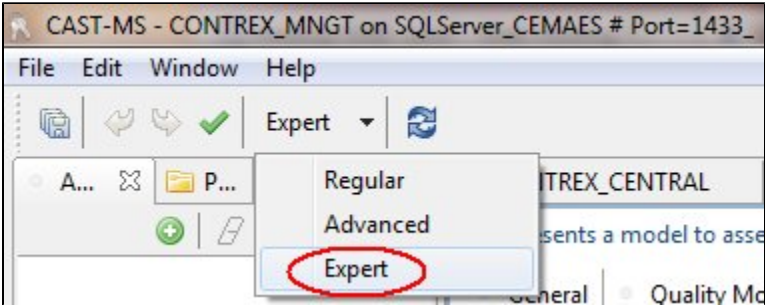


You can find out more about this in [Health Dashboard - Available information - Overview section](#).

## Modifying Technical Debt calculation variables

To modify any of the variables listed in the section above, please use the **CAST Management Studio**:

- Open the **Assessment Model** provided by default in the **Assessment Models editor** (note that this Assessment Model is the default Model compatible with the current version of AIP Core - i.e. it does not contain any customization that may have been migrated from previous versions of CAST) and ensure you are working in **Expert Mode**:



**Note** that if you do decide to use this default Assessment Model, you must ensure that your CAST Dashboard Service is configured to use it.

- Click the **Sizing Model** tab and scroll down to the very bottom of the list of **Sizing Measures**:

Name	Active	Detached	Sizing Measure type	Aggregation type
Number of Triggers	<input type="radio"/> true	<input type="radio"/> false	Technical Size	Count objects
Number of Units	<input type="radio"/> true	<input type="radio"/> false	Technical Size	Count objects
Number of Userobjects	<input type="radio"/> true	<input type="radio"/> false	Technical Size	Count objects
Number of Views	<input type="radio"/> true	<input type="radio"/> false	Technical Size	Count objects
Number of WEB Pages	<input type="radio"/> true	<input type="radio"/> false	Technical Size	Count objects
Technical Debt	<input type="radio"/> true	<input type="radio"/> false	Technical Debt Statistics	Sum values
Technical Debt density	<input type="radio"/> true	<input type="radio"/> false	Technical Debt Statistics	Sum values
Technical Debt of added Violations	<input type="radio"/> true	<input type="radio"/> false	Technical Debt Statistics	Sum values
Technical Debt of removed Violations	<input type="radio"/> true	<input type="radio"/> false	Technical Debt Statistics	Sum values

- Select the **Technical Debt** measure and drill down into the **Parameters** tab. This will show you the parameters you can modify:

Name: Technical Debt

Active:

Detached:

Represents an elementary assessment of a component size

Documentation | Computing Settings | **Parameters** | General | Notes

Contextual Parameters

Type	Name
Decimal number	% of high severity violations to be fixed
Decimal number	% of low severity violations to be fixed
Decimal number	% of medium severity violations to be fixed
Decimal number	Cost per hour of developer time
Text	Currency for development costs
Decimal number	Weighted time, in hours, for fixing HIGH severity violation
Decimal number	Weighted time, in hours, for fixing LOW severity violation
Decimal number	Weighted time, in hours, for fixing MEDIUM severity violation

- Note the **Parameter** you would like to modify and then move into the **Contextual Parameters** tab.
- Locate the Parameter in the list and select it - this will display the **Default Value** as highlighted below:

General | Quality Model | Sizing Model | **Contextual Parameters**

Contextual Parameter

Type	Name
Decimal number	% of high severity violations to be fixed
Decimal number	% of low severity violations to be fixed
Decimal number	% of medium severity violations to be fixed
Text list	attribut name to exclude
Text list	attribut name to exclude
Decimal number	Average Class Complexity Threshold
Decimal number	Average Class Fan-In Threshold
Decimal number	Average Class Fan-Out Threshold
Decimal number	Average Complexity Artifact Threshold

General | Used By | Contextual Value Overridings

Name: % of high severity violations to be fixed

Option set when there is a default value to use.

Managed Default Value:

Default Value: 100.0

Represents an input that controls the behavior of a Quality Rule or Sizing

- Modify the values in the **Default Value** field.
- Repeat for each Technical Debt Parameter you want to modify.
- You will then need to generate a new snapshot so that the new values are taken into consideration.

## Known limitations

### Technical Debt values at functional module level when objects belong to multiple technologies

When a physical source file is detected by CAST as belonging to **multiple technologies**, the resulting objects will be counted for **EACH technology** that they belong to and therefore the **Technical Debt value for these objects will be incorrect at module level**. Take for example a **.yml** file which may be analyzed as PHP and also as HTML5: for the purposes of Technical Debt, any resulting objects will be counted once for PHP and once for HTML5, therefore creating incorrect values for Technical Debt.