

CAST Automated Enhancement Points Estimation - AEP

On this page:

- [Introduction](#)
- [What about Data Functions and Transactional Functions?](#)
 - [Data Functions](#)
 - [Transactional Functions](#)
- [Counting Automated Enhancement Points](#)
 - [Automated Enhancement Functional Points](#)
 - [Transactional Functions](#)
 - [Effort Complexity and Effort Complexity Level](#)
 - [Evolved Effort Complexity](#)
 - [Effort Complexity Variation](#)
 - [Complexity Factor](#)
 - [Data Functions](#)
 - [Complexity Factor](#)
 - [Automated Enhancement Technical Points](#)
 - [Implementation Points](#)
 - [Equivalence Ratio](#)
- [Transitioning from EFP to AEP](#)
 - [Create a new baseline snapshot](#)

Target audience:

CAST AI Administrators

Introduction

In addition to the [Enhancement Function Points \(EFP\) measure](#), CAST AIP supports the new **OMG Automated Enhancement Points (AEP)** specification to estimate the size of an application revision.

After an initial baseline count, CAST AIP catalogs and stores both Data Functions and Transactional Functions structure in the CAST Analysis Service schema. All objects that are part of the Data and Transactional Functions are stored in the Analysis Service along with a checksum that uniquely identifies the current version of each artifact. When a new version of the application source code is analyzed using CAST AIP, the CAST Automated Enhancement Points algorithm uses the computed checksum of each object composing both Data Functions and Transactional Functions to determine:

- Added Functions
- Modified Functions
- Deleted Functions

Then each Added, Modified or Deleted Data Functions and Transactional Functions are weighted using OMG tables.

This chapter presents how the OMG AEP specification is implemented in CAST AIP. For detailed information about the OMG Automated Enhancement Points specification, please refer to the OMG documentation at <http://www.omg.org/spec/AEP/>.



Note that for new Applications on-boarded in this release of CAST AIP this measurement mode is ALWAYS USED by default instead of [EFP mode](#).

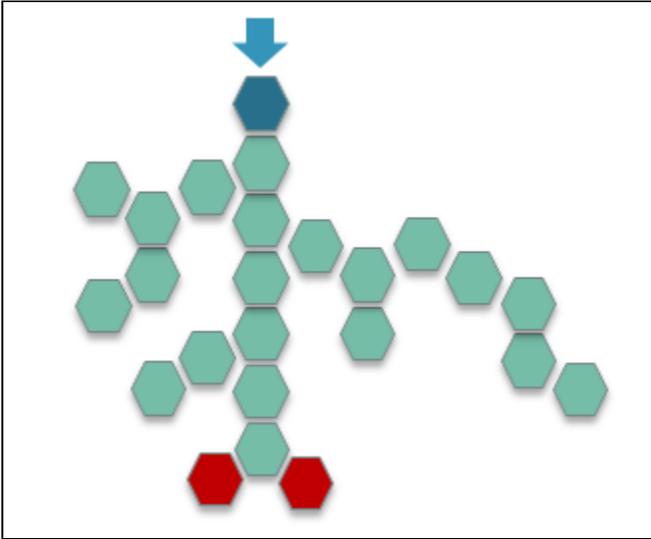
What about Data Functions and Transactional Functions?

Data Functions

Data Functions are defined as specified in the OMG Automated Function Points (AFP) specification. Please see the OMG documentation at <http://www.omg.org/spec/AFP/> for more details.

Transactional Functions

Like those used to count AFP and EFP, Transactional Functions are defined by specifying entry points, data entities, and end points. However, as specified in the OMG AEP specification, Transactional Functions are built by following the links in extended full transaction call graphs (see image below). This means that they contain more objects than the Transactional Functions used to count AFP and EFP and therefore, better reflect changes impacting the Transactional Functions in the application revision.



Full Transaction Call Graph.

Counting Automated Enhancement Points

The OMG AEP specification takes into account both functional and technical parts of applications. It is one of the main differences compared to the [Enhancement Function Points measure](#). As a consequence, the total number of AEP is obtained with the following formula:

$$AEP = AEFP + AETP$$

Where AEFP corresponds to the Enhancement Points for the functional part and AETP corresponds to the Enhancement Points for the technical part of the application revision. Objects are considered as belonging to the technical part if they do not appear in any full transaction call graph of the application.

Automated Enhancement Functional 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.

Transactional Functions

Enhancement Points for Transactional Functions are based on the OMG AFP specification and on a Complexity Factor (CF) used to weight the FP values with regard to status of the Transactional Functions and the complexity of the Artifacts that compose the Transactional Functions. The Complexity Factor for modified Transactional Functions is computed from the Effort Complexity metric attached to each Artifact.

Effort Complexity 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**

Evolved Effort Complexity

The Evolved Effort Complexity estimates the evolution of the Effort Complexity between the current and previous snapshots. It is based on the following formula:

$$\text{Evolved EC} = (\text{EC_add_snap}(n) + \text{EC_mod_snap}(n) + \text{EC_del_snap}(n-1)) / \text{EC_total_snap}(n-1)$$

i Note:

- Excluded items are ignored.
- EC_total for snapshot n = EC added, updated, and unchanged for the Transactional Function in the snapshot n.

Effort Complexity Variation

The Effort Complexity Variation estimates the difference of the Effort Complexity between the current and previous snapshots. It is based on the following formula:

$$\text{EC Variation} = (\text{EC_total_snap}(n) - \text{EC_total_snap}(n-1)) / \text{EC_total_snap}(n-1)$$

i Note:

- Excluded items are ignored.
- EC_total for snapshot n = EC added, updated, and unchanged for the Transactional Function in the snapshot n.

Complexity Factor

As defined in the OMG AEP specification, the Complexity Factor (CF) varies with the status of the Transactional Functions and the changes made to the Artifacts associated to these Transactional Functions. The Complexity Factor is defined as follows:

- Added Transactional Functions: CF = 1
- Deleted Transactional Functions: CF = 0.4
- Modified Transactional Functions:
 - If there is no shared object in the full transaction call graph, then CF is defined by the following matrix:

| Evolved Effort Complexity (ECEvolved) | Effort Complexity Variation (ECvariation) | | | |
|---------------------------------------|---|--------------|--------|--------|
| | ≤ ⅓ (x 100%) | ≤ ⅔ (x 100%) | ≤ 100% | > 100% |
| ≤ ⅓ x 100% | 0.25 | 0.50 | 0.75 | 1.00 |
| ≤ ⅔ x 100% | 0.50 | 0.75 | 1.00 | 1.25 |
| ≤ 100% | 0.75 | 1.00 | 1.25 | 1.50 |
| > 100% | 1.00 | 1.25 | 1.50 | 1.75 |

- If there are shared objects in the full transaction call graph, then CF is defined by the following rules:
 - If 100% of EC is for modified shared objects, then CF = **0.25**
 - If more than 75% of EC is for shared objects, then CF is capped to **0.50**
 - If more than 50% of EC is for shared objects, then CF is capped to **0.75**
 - If less than 50% of EC is for modified shared objects, then CF is **defined by the matrix listed above**, without a cap.

Data Functions

Enhancement Points for Data Functions are based on the OMG AFP specification and on a Complexity Factor used to weight the FP values with regards to the status of the Data Functions and the changes done on the data entities that compose the Data Functions.

Complexity Factor

As defined in the OMG AEP specification, the Complexity Factor (CF) varies with the status of the Data Functions, the last operations that have been performed by automatic calibration algorithm, and the changes done on data entities.

The Complexity Factor is defined as follows:

- Added Data Functions: CF = 1
- Deleted Data Functions: CF = **0.4**
- Merged Data Functions: CF = **0.4**
- Split Data Functions: CF = **0.4**
- Changed type Data Functions: CF = **0.4**
- Modified Data Functions:
 - If modified DET ≤ 33%, then CF = **0.25**

- If modified DET <= 66%, then CF = **0.5**
- If modified DET <= 100%, then CF = **0.75**
- If modified DET > 100%, then CF = **1**

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.

Implementation Points

The first step to compute 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)

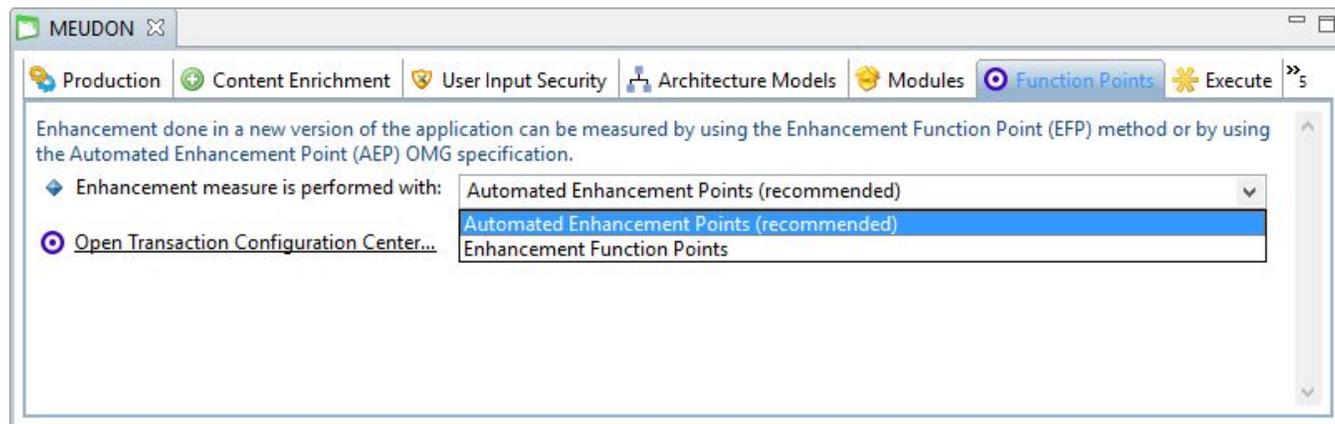
Equivalence Ratio

The second step to compute AETP allows to align Implementation Points 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

Transitioning from EFP to AEP

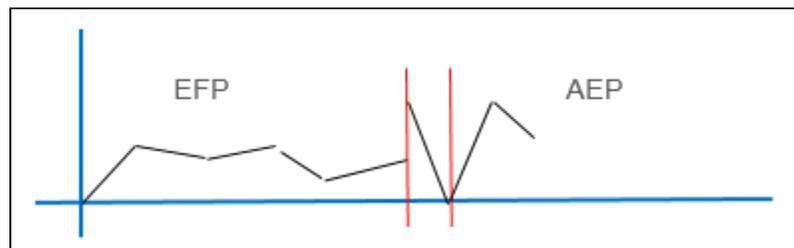
Applications that have been on-boarded and analyzed with a previous release of CAST AIP (8.1.x) are measured with the Enhancement Function Points (EFP) mode and this measure will be used by default for Applications that have been upgraded to CAST AIP 8.2.x (or later releases) to minimize the impact on results. It is, however, possible to change the enhancement measure to Automated Enhancement Points (AEP) for upgraded Applications via the **CAST Management Studio > Application Editor**:



When changing the measurement mode it is important to take into consideration the impact this will have on the analysis/snapshot results and that it will introduce a disruption into the measurement trending. There are important differences between the two measures (AEP and EFP) that must be understood:

- The EFP mode does not manage the technical part of application whereas AEP does.
- When using EFP mode, Transactional Functions are based on the reduced call graph whereas for AEP, Transactional Functions are based on the full call graph.
- The AEP specification defines Complexity Factors based on artifacts' complexity or changes made to data entities.

When moving from EFP to AEP to get more precise values, results will show objects that do not belong to the reduced transaction call graph as "added" and this will impact the measure, even if these objects have not really been added. Points related to the technical part of the Application will also appear in the results. Moreover, taking in to account object complexity to weight the FP values will impact the results as well (see image below):



Changing the measure from EFP to AEP

Moving in the opposite direction from AEP to EFP will result in having objects that do not belong to the full transaction call graph to be considered as "deleted". This will also impact the results, even if those objects have not been really removed. In addition, Complexity Factors will be replaced by Impact Factors that are set to 1 by default and points related to the technical part of the application will disappear (see image below):



Changing the measure from AEP to EFP

Create a new baseline snapshot

If you plan to change the measurement mode, it is necessary to recreate a new baseline snapshot. To do that, after you change the measure you must generate a new snapshot for the same Application version but it must not be considered as consistent for the new measure. The next snapshot you generate for the Application will be consistent and can be used as the first snapshot using the new enhancement measure.