

# Case Study - Measurement changes after upgrade for selected customer applications



This documentation is no longer maintained and may contain obsolete information.

## On this page:

- [Introduction and methodology](#)
- [Application 1 \(JEE, C#, C++ , Informix, WSDL, Shell\)](#)
  - [Main indicators](#)
  - [Analysis of the variations](#)
- [Application 2 \(PL/SQL, JEE, FLEX, SHELL, TIBCO, WSDL\)](#)
  - [Main indicators](#)
  - [Analysis of the variations](#)
- [Application 3 \(JEE, Oracle, Mainframe, C, Shell\)](#)
  - [Main indicators](#)
  - [Analysis of the variations](#)
- [Application 4 \(JEE, PL/SQL\)](#)
  - [Main indicators](#)
  - [Analysis of the variations](#)
- [Application 5 \(ABAP, SAP SQL\)](#)
  - [Main indicators](#)
  - [Analysis of the variations](#)
- [Application 6 \(C#, PL/SQL\)](#)
  - [Main indicators](#)
  - [Analysis of the variations](#)
- [Application 7 \(VB .NET/ASP.NET, C#, MicrosoftT-SQL\)](#)
  - [Main indicators](#)
  - [Analysis of the variations](#)
- [Application 8 \(C++, PHP\)](#)
  - [Main indicators](#)
  - [Analysis of the variations](#)

## Introduction and methodology

This page provides real-life data about how measurement results are impacted by an upgrade to a newer release of CAST AIP.

The applications selected for reference are real customer applications, anonymized, for which we give only the concerned languages and frameworks. For each, we run a reference analysis in 7.3, 8.0 and 8.1, then upgrade the configuration to 8.1, analyze again and compare the results on a few key indicators. Any significant difference is investigated and fixed in case it is a bug. If the difference is nominal (change in quality model, removed false positive/false negative), the justification is given.

Notes:

- The version used in 7.3/8.0/8.1 is usually the latest service pack available at the time of the test. In case it makes a difference in the analysis of results (because of a fix introduced in a service pack), it will be mentioned in the justification.
- Changes of less than 1% are not systematically analyzed.

## Application 1 (JEE, C#, C++ , Informix, WSDL, Shell)

### Main indicators

AIP version	Size (kLOC)	Critical Violations	Total Quality Index	Data Function Point	Transaction Function Point
7.3	383	Not relevant	Not relevant	499	168
8.0.3	393	729	2.55	499	182
8.1.1	393	729	2.55	499	182
8.2.0	393	770	2.54	499	182
8.3.0					

### Analysis of the variations

## Size

No change since 8.0.

LOC change between 7.3 and 8.0 is caused by a change in the way the 8.0 .NET analyzer is taking the generated code into account.

## Quality

Change in number of Critical Violations in 8.2 caused by an improvement of JEE rule "Avoid using Fields (non static final) from other Classes".

Figures in 7.3 not considered as relevant since the analysis was not done with the default AIP assessment model.

## Function points

No change since 8.0

# Application 2 (PL/SQL, JEE, FLEX, SHELL, TIBCO, WSDL)

## Main indicators

AIP version	Size (kLOC)	Critical Violations	Total Quality Index	Data Function Point	Transaction Function Point
7.3	2193	6066	2.4	1080	4427
8.0	2289	6147	2.14	1080	2065
8.1.1	2193	4893	2.33	1078	1977
8.2.0	2193	8213	2.27	1083	1978
8.3.0					

## Analysis of the variations

### Size

Change in LoC count between 8.0 and 8.1 is caused by a fix in the way external JS libraries are taken into account. In previous versions, they were included in the line counts, when they should not have been.

### Quality

Change in Critical Violations between 8.0 and 8.1.1 is caused by a fix in the rule "Avoid using Fields (non static final) from other Classes". This fix was introduced in version 8.0.2, after the 8.0 measurements for that app were done.

Change in Critical Violations between 8.1.1 and 8.2.0 is caused by an improvement to the rule "Avoid directly instantiating a Class used as a Managed Bean" in order to extend it to other types of managed beans outside of the Spring framework.

### Function points

Change in Data Function Points before between 8.1.1 and 8.2.0 is caused the fix of a problem where in some cases some DB tables could be included in several data functions, which is incorrect according to the OMG specification. This fix will be included in 8.1.2.

# Application 3 (JEE, Oracle, Mainframe, C, Shell)

## Main indicators

AIP version	Size (kLOC)	Critical Violations	Total Quality Index	Data Function Point	Transaction Function Point
8.0	824	4662	2.36	295	2130
8.1.1	825	4662	2.35	291	2170
8.2.0	825	4641	2.37	291	2425
8.3.0					

## Analysis of the variations

### Size

No change.

### Quality

The small change in number of Critical Violations between 8.1 and 8.2 still has to be analyzed.

### Function points

Between 8.0 and 8.1, the reduction in number of data function points is caused by a bug in a DMT extension installed on the test configuration, not by a change in the analysis procedure.

Between 8.1 and 8.2, the increase in number of transaction function points is caused by the fix of a bug which caused Struts objects to be lost, and transactions to be missed.

## Application 4 (JEE, PL/SQL)

### Main indicators

AIP version	Size (kLOC)	Critical Violations	Total Quality Index	Data Function Point	Transaction Function Point
8.1.1	27	516	2.44	189	582
8.2.0	27	513	2.45	189	582
8.3.0					

### Analysis of the variations

#### Size

No change.

#### Quality

No significant change.

#### Function points

No change.

## Application 5 (ABAP, SAP SQL)

### Main indicators

AIP version	Size (kLOC)	Critical Violations	Total Quality Index	Data Function Point	Transaction Function Point
7.3	227	Not relevant	Not relevant	379	281
8.0	227	1838	2.67	379	281
8.1.1	227	1469	2.64	379	281
8.2.0	227	1469	2.67	379	281
8.3.0					

### Analysis of the variations

#### Size

No change.

#### Quality

Figures in 7.3 not considered as relevant since the analysis was not done with the default AIP assessment model.

Changes between 8.0 and 8.1 are linked to following 2 fixes:

- "Avoid using SQL queries inside a loop - 7424" : This rule is no longer enable for SAP analysis. It was redundant with ""Avoid Open SQL queries in loops - 7868". The change was done in the service packs 7.3.7, but was not present in the 7.3/8.0 versions used in this test.
- "Avoid missing WHEN OTHERS in CASE statements - 7518" is not longer critical in 8.1. The change was done to be consistent with similar rules in other languages.

### Function points

No change.

## Application 6 (C#, PL/SQL)

### Main indicators

AIP version	Size (kLOC)	Critical Violations	Total Quality Index	Data Function Point	Transaction Function Point
8.0	88	395	3.21	741	84
8.1.0	88	396	3.21	741	84
8.3.0					

### Analysis of the variations

#### Size

No change.

#### Quality

No significant change.

#### Function points

No change.

## Application 7 (VB .NET/ASP.NET, C#, MicrosoftT-SQL)

### Main indicators

AIP version	Size (kLOC)	Critical Violations	Total Quality Index	Data Function Point	Transaction Function Point
8.1.1	2009	9793	2.31	3872	3808
8.2.0	2009	9793	2.31	3936	3824
8.3.0					

### Analysis of the variations

#### Size

No change.

#### Quality

No change.

#### Function points

Change in Function Points before between 8.1.1 and 8.2.0 is caused the fix of a problem where in some cases some DB tables could be included in several data functions, which is incorrect according to the OMG specification. This fix will be included in 8.1.2.

## Application 8 (C++, PHP)

**Main indicators**

AIP version	Size (kLOC)	Critical Violations	Total Quality Index	Data Function Point	Transaction Function Point
8.1.1	335	594	3.07	241	860
8.2.0	335	594	3.07	241	860
8.3.0					

**Analysis of the variations**

**Size**

No change.

**Quality**

No change.

**Function points**

No change.