

FPA according to Nesma and IFPUG

Nesma FPA and IFPUG FPA are two leading Functional Size Measurement (FSM) methods, compliant with ISO/IEC 14143-1. This paper describes the differences. Sections 1 – 3 deal with the detailed FPA method, section 4 with the high level and indicative FPA methods.

1. History

Before 1990 there were major differences between the Nesma FPA and IFPUG FPA methods. From 1990 on Nesma and IFPUG have worked closely together to define a common standard. Between 1990 and 2004 the IFPUG FPA guidelines have gone through several revisions. Practically, this has resulted in a continually lower function point count for the same functional sample, using a newer version of the IFPUG CPM. In 2004 IFPUG CPM 4.2 was released. Since then the IFPUG FPA guidelines are stable. The Nesma FPA guidelines have been stable since version 1.0 of the manual was published in 1989.

As of IFPUG CPM 4.2 (2004) there are almost no differences between Nesma and IFPUG. The few that still exist are described below. The impact of these differences is relatively small.

Detailed background information about which guideline was changed when in the years before 2004 can be found in the document "[Historical Review Nesma and IFPUG](#)" (in Dutch only).

2. Actual versions of the Nesma and IFPUG CPM

Both Nesma and IFPUG maintain their own FPA Counting Practices Manual (CPM).

- Nesma CPM 2.3 (ISO/IEC 24570:2018) Nesma Counting Practices Manual FPA, release 2.3
- IFPUG CPM 4.3 (ISO/IEC 20926:2010) IFPUG Counting Practices Manual FPA, release 4.3

Nesma CPM 2.3 is the version that was certified by ISO as ISO/IEC 24570.

Both the English and Dutch release of the Counting Practices Manual now have the same version: Nesma CPM 2.3. Previous versions were Nesma CPM 2.1 (English) and Nesma CPM 2.2 (Dutch), these are interchangeable.

IFPUG CPM 4.3 is the version that was certified as ISO/IEC 20926.

3. Comparison of the Nesma and IFPUG FPA guidelines

The Nesma and IFPUG FPA guidelines both go back to the original method of A.J. Albrecht and are almost the same. This section shows the differences.

Nesma and IFPUG use almost the same terminology, have the same criteria for identifying elementary processes and data functions, and differentiate the same five types of user functions: Internal Logical File (ILF), External Logical File (ELF), External Input (EI), External Output (EO), External Inquiry (EQ). In Dutch: ILGV, ELGV, IF, UF, OF.

IFPUG uses the term External Interface File (EIF) for External Logical Files. Since this causes a lot of confusion, Nesma has replaced this term by External Logical File (ELF).

The rules for determining the type and complexity of a function are the same, with a few exceptions:

- a) External Inquiry vs External Output
- b) General System Characteristics
- c) Implied Inquiry
- d) Code data (Code tables)
- e) Physical media

f) Separate External Output next to External Input
In the following paragraph each topic is highlighted.

a) External Inquiry vs External Output

For IFPUG, an External Inquiry is defined as a function that presents data to a user from a logical file (ILF or ELF) without undergoing additional processing (such as calculations, updates to an ILF, etc.). In all other cases it is considered an External Output.

For Nesma, the same rules apply, but in addition, a unique selection key must have been entered and the output must be fixed in scope. In some cases, therefore, IFPUG will count an External Inquiry, while Nesma counts the same function as an External Output (e.g., "Show all customers").

The impact of this difference is *marginal* for the number of function points for a system or project because only the type of function (External Inquiry or External Output) is affected, not the number of counted functions.

b) General System Characteristics

As of Nesma CPM 2.3 the General System Characteristics are removed from the FPA guidelines. In the previous Nesma version they were in an appendix.

For IFPUG the General System Characteristics are part of Appendix-C. That chapter explains the General Systems Characteristics (GSCs) and Value Adjustment Factor (VAF). Note: Applying the GSCs, calculating the VAF, and calculating the adjusted functional size is not included in the IFPUG FSM and is considered optional in the IFPUG CPM.

c) Implied Inquiry

When modifying or deleting data, the data is often first presented to the user for viewing. This is known as an "implied inquiry".

For Nesma, the underlying goal of a function is always the primary objective. Nesma therefore does not consider the implied inquiry as a separate transactional function, but as an integral part of the modify- or delete function. The data element types presented to the user by the implied inquiry are therefore added to those counted in the modify- or delete function. Nesma will only count the External Inquiry if it is specifically identified by the user as a separate transaction for the purpose of querying data.

IFPUG has an example that shows that an implied inquiry must be counted as an External Inquiry (if not already counted elsewhere).

Usually the user will have defined this function as an (explicit) inquiry (and it will thus be counted). The implied inquiry will then not be counted (again), because the same function cannot be counted twice. That's why the impact of this difference is *marginal* for the number of function points for a system or project.

d) Code tables

In general, entities can be seen as being composed of primary data ("business objects") or composed of secondary data ("supportive data") or a third type ("code data").

For the primary and secondary data, both Nesma and IFPUG follow the same guidelines.

Code data usually consist of code tables, also called "FPA-tables" by Nesma. As an example, consider the "translation table": article code, article description.

Nesma will classify all code tables together as one ILF and/or one ELF. The number of record element types will be set equal to the number of identified code tables. Altogether, the FPA-

tables-ILF will also count for one External Input, one External Output and one External Inquiry. For the FPA-tables-ELF no transactional functions are counted, even though External Outputs or External Inquiries may be present.

IFPUG considers code tables to be an implementation of technical or quality requirements for the user, and not part of the functional user requirements. That's why IFPUG states, that in accordance with the ISO FSM standard, code tables and the transactional functions associated with them, are not to be counted using function points.

IFPUG considers code tables with an additional start/end date also as code data where Nesma examples show that when the start/end date is important to the business, these tables should be classified as business data.

The impact of this difference is *marginal* for the number of function points for a system or project. The difference will be *at most* 34 function points for an FPA-tables-ILF, and 10 function points for an FPA-tables-ELF.

e) Physical media

Physical media is ignored in Nesma FPA. Nesma looks at the underlying functionality and explicitly states, that the same elementary process using different physical media, should be counted only once.

If the data element types and the logical processing are the same, input entered through different media will be counted as one External Input by Nesma. The same holds true for External Outputs and External Inquiries. Reports that can be presented on different media (print, screen, etc.) are counted as one External Output or External Inquiry (when the data element types and the logical processing remain the same).

IFPUG does not provide specific guidelines for this situation.

f) Separate External Output next to External Input

Nesma FPA counts a separate External Output next to an External Input while IFPUG sees the External Output as part of the External Input. If e.g. a batch input process performs an update on an ILF and this batch process also produces an output product like a transaction or processing report, for Nesma this report is counted separately as an External Output. For IFPUG this report is not counted. The reason for not counting the report is that the report has no meaning without the batch process and the batch process is not complete without the report.

The impact of this difference is *marginal* in general, although - depending on the number it occurs and the size of the system - it can be *substantial*.

4. Early and Fast Function Point Analysis

To carry out a detailed FPA one needs detailed user requirements, and also detailed FPA guidelines on how to assess these detailed requirements.

Both Nesma and IFPUG recognize and promote the same two FPA methods that can be applied early in the software development life cycle or can be used to do a function point analysis fast:

- High Level FPA method (high accuracy, abstracting from complexity of functions)
- Indicative FPA method (rough order of magnitude of the functional size)

Nesma : [Early Function Point Analysis](#)

IFPUG : [Early Function Point Analysis and Consistent Cost Estimating](#) (uTip 3)