Nesma on sizing
Part 1: Function Point Analysis (FPA)

In this whitepaper Nesma presents information to enlarge your knowledge about sizing using Function Point Analysis (FPA) and to support your activities in this area.

The information is structured around six subjects:

1. Introduction to Function Point Analysis (FPA)
2. How does FPA work?
3. Additional FPA Guidelines
4. Function Point Analysis methods
5. ISO Standard for Functional Size Measurement
6. Nesma and IFPUG

1. Introduction to Function Point Analysis (FPA)

Function Point Analysis (FPA) is a method to measure the functional size of an information system. FPA measures the functional size by looking at the (functional) transactions and (logical) data files that are relevant to the user in the business. The unit of measurement is “function points”; the functional size of an information system is expressed by a number of function points. Function points are a good measure of the functional size of an information system; the unit of measurement “function points” can be utilized in various ways.
FPA is often used to budget a system development project. The development costs for an information system are related to its size: the bigger the system, the more expensive the development will be. Based on experiences in earlier projects an organization knows, how many hours (on average) one needs to realize one function point: the productivity rate. Size (number of function points) x productivity rate (hours per function point) is a basis for the project budgeting process.

FPA can be applied for development, as well as for enhancement projects.

FPA is a fast method, which does not require knowledge of computers. Assuming suitable documentation, it does not take much time to perform an FPA. It is estimated that for a system which needs one thousand development hours, an FPA can be performed in about one hour.

What does FPA offer?

A “Function point” is the one and only measurement unit that offers the possibility to talk concretely and objectively about the size of an information system to be developed. A statement like “the system has a size of about 314 function points” gives more information than “it is quite a big system”.

Thanks to this, the measurement unit "function points" offers, among other things, the following opportunities:

Better and earlier project cost estimating and budgeting.
Using the functional user requirements, one determines the functional size (the number of function points) of the information system. Using practical experiences in completed projects in the past, one determines the expected productivity rate (hours per fp) for the project. By multiplying size and expected productivity rate, one gets a basis for the project budget for the system development process.

Better controlling projects.
Changes in the functional user requirements can be expressed/sized in function points, so that they are concrete, quantified and controllable.

Better communications about the system development project
If two persons carry out a function point analysis and determine a different number of function points, this is inevitably caused by a different interpretation of the functional user requirements, of the system to be built. Unclear or incomplete functional user requirements become visible when carrying out an FPA.

Measuring productivity.
The number of spent development hours, divided by the number of function points of the constructed and implemented information system, results in the project productivity rate. One may compare this with the standard productivity rate. Differences may be analyzed and may result in concrete control and improvement measures for future projects.

Measuring information system quality.
The number of errors per function point per unit of time is an indicator for the quality of an information system.

Improving the quality of the system development process.
Reducing miscommunications and introducing new control measures as a result of productivity and quality analyses improves the quality of the software development process.

What does FPA not offer?

- FPA is not a project management method
- FPA does not automatically deliver error free project estimates; it does provide important support in the project budgeting process
- FPA is not a project planning method.

In which phases of a project one can perform an FPA?

One can perform an FPA, as soon as the functional user requirements of the information system are known on a high level. Essential are the number of functional user transactions and the conceptual data model.

This may be the case in the Proposal/Feasibility phase, or in the Requirements/Analysis phase, but it is certainly the case in the Functional Design phase. In earlier stages of the project life cycle, one might need to perform an FPA estimate, using indicative or estimated function point counts, because all necessary information to perform a (detailed) FPA might not be available.

Which project phases may be estimated using FPA?

FPA can estimate the development effort for each phase in the system development life cycle. Indeed, based on experiences in projects in the past, one knows for each project phase how many hours per function point on an average were needed in the past to complete the phase.

For the Construction phase, FPA gives very good estimates because the activities in that phase are very concrete and relatively similar between projects.

In the operational phase of an information system, one may use FPA to estimate the operational costs of information systems: a certain number of hours per function point per year.

For what kinds of projects one may use FPA?

One may use FPA for development or for enhancement projects. In enhancement projects, it may happen that implementing functionality demands extra technical effort, because the way the system is technically constructed makes it difficult to construct the enhancement. However, in these cases FPA also indicates how much functionality is actually delivered. The extra needed technical modifications are taken into account by decreasing the project's expected productivity rate (more hours per function point), compared to the standard delivery rate.

2. How does FPA work?

FPA is a method to determine the functional size of an information system or project. The functional size may be used for different purposes, for example budgeting.

This text contains a brief description of the FPA method. However, to make a good quality FPA, it is absolutely necessary to reference the FPA Counting Practices Manual. This manual describes the exact definitions, counting rules and guidelines. In a separate document many practical examples and case studies are presented

FPA carries out the following steps to determine the size of an information system or system development project:

- Step 1: Identify the functions of the system that are relevant to the user
- Step 2: Determine the functional complexity of each function
- Step 3: Calculate the unadjusted function point count of the system

Step 1: Identify the functions of the system that are relevant to the user

FPA measures the amount of functionality an information system offers to the users. This functionality comprises both logical transactions and logical data files. It is important, that the user requires and recognizes the functionality. That’s why FPA names these functions as user functions.

FPA distinguishes between five types of user functions:

- Internal Logical File (ILF)
- External Interface File (EIF)
- External Input (EI)
- External Output (EO)
- External Inquiry (EQ)

An Internal Logical file contains permanent data that is relevant to the user. The information system references and maintains the data. In the context of FPA “to maintain” means to add, change or delete data.

An External Interface File also contains permanent data that is relevant to the user. The information system references the data, but the data is maintained by another information system (in that other system, it is an ILF).

An External Input receives information from outside the application boundary and maintains an ILF of the information system.
Examples: updating customer data in a Customer File; processing and saving order transactions into an Order information system; the medium is not relevant: paper, screen, tape cartridge, data communications, and so on.

An External Output presents information of the information system.
Examples: displaying a list of all accounts payable; generating and printing invoices; generating a diskette with payment orders; the medium is not relevant: paper, screen, tape cartridge, data communications, and so on.

An External Inquiry is a special (simple) kind of an external output.
An external inquiry presents information of the information system based on a uniquely identifying search criterion, without applying additional processing (such as calculations). By example: Displaying the information of a Customer with customer number 123456789.

Step 2: Determine the functional complexity of each function

For user functions, FPA distinguishes between three levels of complexity:

- Low
- Average
- High

To determine the complexity of a user function, FPA provides clear criteria too value the amount of information processing by the user function. These criteria may be found in the FPA Counting Practices Manual.

Remark: In early stages of the systems life cycle, the functionality of a user function may not yet be detailed enough to determine the complexity of the user function. In these situations, one can perform an estimated function point count.

In an estimated function point count, one identifies all user functions, but assigns a default complexity level:

- logical files (ILF, EIF): Low
- user transactions (EI, EO, EQ): Average

Step 3: Calculate the unadjusted function point count of the system

When the user function (step 1) and its complexity (step 2) have been determined, one can assign a number of function points for the user function using the following matrix:

	Complexity		
Function type	**Low**	**Average**	**High**
Internal Logical File	7	10	15
External Interface File	5	7	10
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6

The summation of the function points for all identified user functions is called the unadjusted function point count.

3. Additional FPA Guidelines

This paragraph gives more information about additional Nesma guidelines, that fit within the general framework of the IFPUG FPA guidelines (IFPUG CPM 4.2), and tend to clarify them. That's why they may be of great value to every FPA counter, also to those using the IFPUG rules. The counting rules of Nesma and IFPUG are the same (except a few minor differences). Detailed information on these and other additional FPA guidelines you'll find in the Nesma Counting Practices Manual CPM 2.3 (English language).

Methods for (very) early function point counting

The Nesma defines three types of function point counts:

- detailed function point count (the usual one)
- estimated function point count
- indicative function point count

The methods estimated and indicative function point counts have been developed by NESMA to enable function point counting early in the system life cycle. The Nesma indicative function point count is well known in the world and is referred to as “the Dutch method”.
Detailed information about these early function point methods, including examples and statistical information about the accuracy of these methods, you’ll find in further on in this whitepaper.

Dealing with physical media

The IFPUG CPM 4.2 unfortunately does not (yet) give concrete guidelines for this issue. Without guidelines, even certified FPA counters have very different interpretations, and as a result, big differences in the determined number of function points for the application.

For the Nesma, the physical medium does not, in and of itself, add additional functionality. The same input read from different physical media is counted as only one external input, if the input data element types and the logical processing are the same. The same output written to different physical media is counted as only one external output (or external inquiry) when the logical layout and the logical processing are the same. The IFPUG has not made a clear statement about this in its Counting Practices Manual 4.2.

Querying with several selection criteria (“and/or situations”)

The IFPUG CPM 4.2 unfortunately does not (yet) give concrete guidelines for this issue. Without guidelines, even certified FPA counters have very different interpretations, and as a result, big differences in the determined number of function points for the application.

This is often an issue; e.g., a selection screen with state-id and surname. The user can enter one or both of these items in order to select customers (e.g., show me all customers in Washington DC).
Should each and every distinct “and/or situation” be considered a separate EO/EQ or should only one EO/EQ be counted?

The Nesma uses a go between:

“When the user has more options (i.e., an “and/or situation”), count the selections that mutually exclude each other. Each selection or combination of selections that exclude all others is counted separately”.

Other additional Nesma Guidelines

The Nesma has developed counting guidelines on several other issues. Although relevant, the influence of these issues on a function point count is less, than the issues described above. All of these hints and guidelines may be found in the Counting Practices Manual of the Nesma (English language).

This is a summary of the other additional counting guidelines:

- Guidelines for applying FPA in specific situations.
- Guidelines for using FPA in the system life cycle.
- Twenty general points of particular interest and guidelines when applying FPA, e.g.,
- Counting on the basis of traditional design
- Counting application packages
- Counting from screens
- Report generators and query facilities
- Shared use of data
- Guideline for counting combination effects with functions
- Guidelines for determining when an external output must be considered unique
- Guideline for output products in different languages
- Many other additional hints, do’s and don’ts that may be used while counting function types.

4. Function Point Analysis methods

Nesma recognizes three function point analysis methods:

- ***Detailed function point analysis***
- ***Estimated function point analysis***
- ***Indicative function point analysis***

All these three methods are a self-contained Functional Sizing Measurement (FSM) method on their own. The high level FPA method and the indicative FPA method do not require detailed user requirements, while the functional size determined using these methods is very close to the functional size determined using the detailed FPA method. That’s why these two methods are very suited to be applied early in de software development life cycle or in case the functional size needs to be determined fast.

The information in this paragraph is also available as a free document in Dutch, English, Portuguese and Japanese. See the Nesma website.

Detailed FPA

This is the usual function point analysis method and is performed as follows:

- Determine all functions of all function types (ILF, EIF, EI, EO, EQ)
- Rate the complexity of every function (Low, Average, High)
- Calculate the total unadjusted function point count

High-level FPA (a.k.a. estimated FPA)

The high-level function point analysis method is performed as follows:

- Determine all functions of all function types (ILF, EIF, EI, EO, EQ)

- Rate the complexity of every data function (ILF, EIF) as Low and of every transactional function (EI, EO, EQ) as Average
- Calculate the total unadjusted function point count

So, the only difference of this approximation with the detailed function point analysis method is that the complexity is not determined per individual function, but by default.

Indicative FPA

The indicative function point analysis method is performed as follows:

- Determine the number of data functions (ILFs and EIFs);
- Calculate the total unadjusted function point count of the application as follows: indicative size (fp) = 35 x number of ILFs + 15 x number of EIFs

So this approximation is based solely on the logical files (ILFs and EIFs).

The indicative function point analysis is based on the assumption that there will be about three EIs (to add, change, and delete information in the ILF), two EOs, and one EQ on average for every ILF, and about one EO and one EQ for every EIF.

Example of indicative, high-level and detailed FPA

This section illustrates the three FPA methods by a small case study: an application that maintains Customer data and Product data, and references Supplier data. The more accurate functional size one wants, the more detailed user requirements one needs. That's why this case study presents the three methods of function point analysis in the order of increasing accuracy:

- Indicative function point analysis
- High-level (estimated) function point analysis
- (Detailed) function point analysis

Indicative FPA

For an indicative function point analysis just information about the data functions is needed.

User requirements:

- User wants to maintain Customer data and Product data, and to reference Supplier data.

This (rough) specification is enough for an indicative function point count:

- ILF: Customer and Product
- EIF: Supplier

Data function	Function type	Function points (by default)
Customer	ILF	35
Product	ILF	35
Supplier	EIF	15
Indicative functional size		**85 fp**

High-level FPA

To perform a high-level function point analysis we also need information about the transactional functions, so more detailed user requirements are necessary:

- User wants to add, change, delete Customer data, wants to inquire on Customer, and also requires four different reports on Customer with calculated data
- User wants to add, change, delete Product data, wants to inquire on Product, and also requires a report on Product with calculated data
- User wants to inquire on Supplier using supplier number, and also requires a report on Supplier with totaling results

This more detailed specification of the user requirements shows the actual amount of transactional functions, and therefore enables an estimated function point count.

Data or transactional function	Function type	Complexity (by default)	Function points (unadjusted)
Customer	ILF	Low	7
Product	ILF	Low	7
Supplier	EIF	Low	5
Add Customer	EI	Average	4
Change Customer	EI	Average	4
Delete Customer	EI	Average	4
Inquire on Customer	EQ	Average	4
Report 1 on Customer	EO	Average	5
Report 2 on Customer	EO	Average	5
Report 3 on Customer	EO	Average	5

Report 4 on Customer	EO	Average	5
Add Product	EI	Average	4
Change Product	EI	Average	4
Delete Product	EI	Average	4
Inquire on Product	EQ	Average	4
Report on Product	EO	Average	5
Inquire on Supplier	EQ	Average	4
Report on Supplier	EO	Average	5
Estimated functional size			**85 fp**

Detailed FPA

To carry out a detailed function point count, one does not only need the number of functions of each function type (EI, EO, EQ, ILF, EIF), but one also needs to determine the functional complexity of each individual function (Low, Average, High). In FPA, the functional complexity of a (data or transactional) function is determined, based on the number of DETs, RETs and File Types Referenced that are relevant to this function. That's why the user requirements (as they were stated above in this example when we discussed the estimated function point count) need to be analyzed in more detail: which data elements (DETs) and logical files (File Types Referenced) are used by a transactional function (EI, EO, EQ), and which logical data groups (RETs) and data elements (DETs) a data function (ILF, EIF) consists of. This detailed analysis of the user requirements could result in the following function point count:

Data or transactional function	Function type	Complexity	Function points (unadjusted)
Customer	ILF	Average	10
Product	ILF	Low	7
Supplier	EIF	Low	5
Add Customer	EI	High	6
Change Customer	EI	Average	4
Delete Customer	EI	Low	3
Inquire on Customer	EQ	Low	3

Report 1 on Customer	EO	Low	4
Report 2 on Customer	EO	Average	5
Report 3 on Customer	EO	Low	4
Report 4 on Customer	EO	High	7
Add Product	EI	Average	4
Change Product	EI	Low	3
Delete Product	EI	Low	3
Inquire on Product	EQ	Average	4
Report on Product	EO	Average	5
Inquire on Supplier	EQ	Low	3
Report on Supplier	EO	Average	5
Functional size			**85 fp**

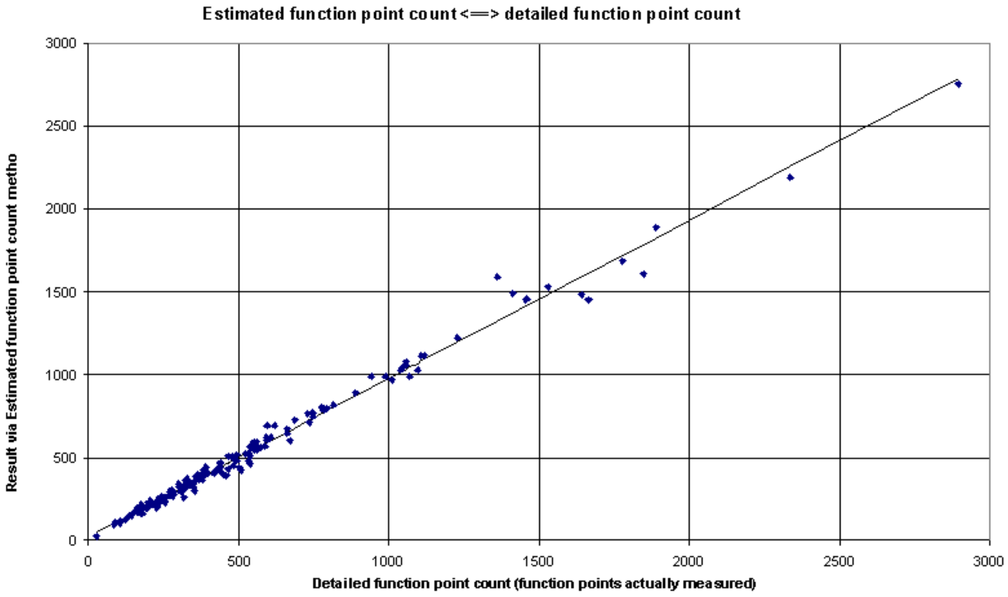
Conclusion

In this particular case study all three methods result in the same functional size of 85 function points. Usually the results are not exactly the same, but still are pretty close to each other. Below the results of research on the accuracy of the estimated and indicative function point analysis methods are shown.

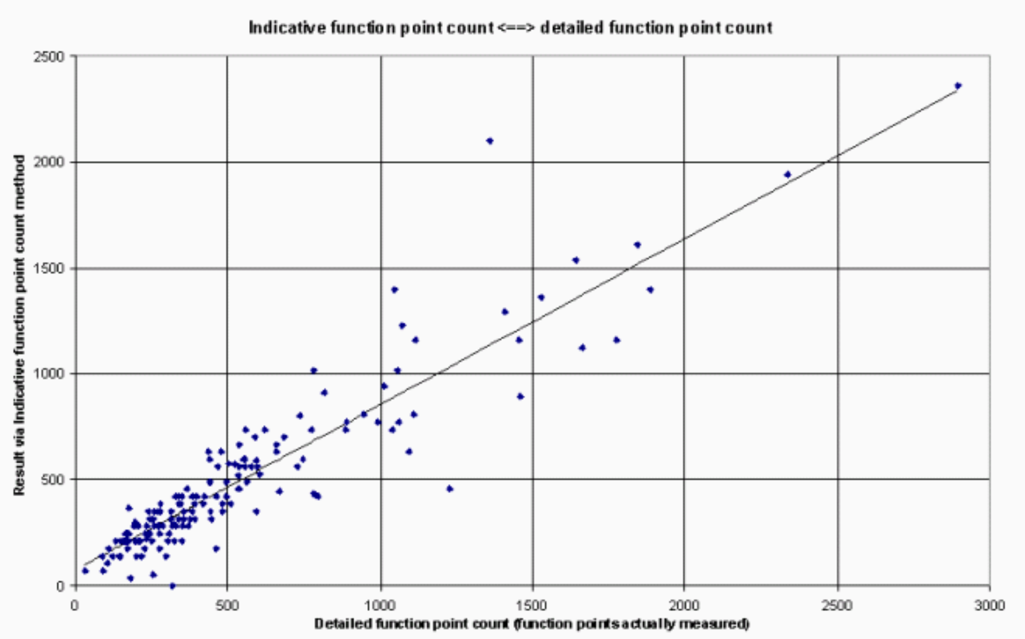
Results of research based on a 100+ projects data base

Using a database of about 100+ developed and implemented applications Nesma did research on the accuracy of the estimated and indicative FPA approximation methods. The implemented applications were simultaneously measured using all three FPA methods. The results are presented in two graphs:

1. The size measured via the high-level function point analysis method versus the size measured via the detailed function point analysis method:



2. The size measured via the indicative function point analysis versus the size measured via the detailed function point analysis method:



There is a good correlation (straight line) in both cases. In the graph of the indicative function point analysis, however, there are considerable deviations (up to about 50%) in some cases. That is why one should be careful using the indicative function point analysis. The strength of this indicative FPA method is that one easily gets a rough estimate of the size of an application in only a very short time.

In an application with more (or less) than a normal amount of inputs or outputs, one might need to change the multipliers of 35 and 15, but the philosophy behind the approach can generally be used.

When to use which method for function point counting

A detailed function point analysis is more accurate than an estimated or an indicative analysis, but it also costs more time and needs more detailed specifications. It's up to the project manager and the phase in the system life cycle as to which function point analysis method is used.

The results of the high-level function point analysis and the detailed function point analysis are very close. There is no statistically significant difference in the outcomes of both FPA methods. That's why many organizations have chosen to use the high-level FPA method by default, instead of the detailed FPA method.

In many applications an indicative function point analysis gives a surprisingly good estimate of the size of the application. It is often relatively easy to carry out an indicative function point analysis, because a data model is available or can be made with little effort. Be careful in using this method, because it provides just a rough indication of the size, and deviations are possible.

5. ISO Standard for Functional Size Measurement

This section provides you with information about the international standard for functional sizing. This standard is more a reference framework than a concrete FPA-like method. That's why ISO calls the methodology FSM: Functional Size Measurement.
Within this reference framework meanwhile FPA methods have been certified by ISO as official ISO standard, among them Nesma, IFPUG and Cosmic.

Below you'll find an article by Francois Collier (chair of ISO SC7) that sketches the path towards the FSM standard and the recognition of FPA methods.

The functional size measurement family of standards is an example of the optimal use of the JTC 1 standardization process.

One of the many challenges for software developers products has been to find methods to properly estimate effort and time required for the development of a software product from the requirements. To address this issue, Allan Albrecht of IBM developed in the late 1970 a measurement approach called Function Point Analysis (FPA) that quantifies the functions contained within software in terms that are meaningful to the software users.

With time, as it became more popular, FPA evolved and some variations were developed. To address this issue, ISO/IEC JTC 1/SC7 decided in 1993 to initiate work in functional size measurement (FSM). A working group (WG12) was put together with representation from 12 countries and also from the principal FSM methods developing user groups. These were the International Function Point Users Group (IFPUG), the Common Software Measurement International Consortium (COSMIC), UK Software Metrics Association (UKSMA) and Netherlands Software Metrics Association (Nesma).

The strategy that was followed by the working group was to first develop a series of generic standards and guidelines on FSM, the 14143 series of standards. This set of standards was developed using the normal consensus building ISO process. Once these standards were published, the established methods standards setting organizations were invited to use the JTC1 PAS process to get ISO recognition, an invitation that IFPUG (ISO/IEC 20926), UKSMA (ISO/IEC 20968) and NESMA (ISO/IEC 24570) accepted. For the COSMIC (ISO/IEC 19761) standard, the normal ISO process was followed.

The outcome of this work is that all four major FSM methods now conform to a minimal set of minimal requirements set by ISO/IEC standards, and that they are all now also ISO/IEC standards.

Nesma FPA recognized by ISO

The NESMA-FPA-methodology has been certified by ISO as an official ISO standard. The basis for the standard is the NESMA FPA Counting Practices Manual. The standard is named NESMA ISO/IEC 24570. The complete name of the standard is:

ISO/IEC IS 24570 Software Engineering – NESMA functional size measurement method version 2.1 – Definitions and counting guidelines for the application of Function Point Analysis

This is a success for Nesma, and a clear sign of international recognition. The ISO-certification may be seen as a guarantee for the quality of the FPA-methodology.

6. Nesma and IFPUG

When the Nesma issued the first version of its manual Definitions And Counting Guidelines For The Application Of Function Point Analysis in 1990, it assumed the principles of the IFPUG Counting Practices Manual (IFPUG CPM 2.0) that were valid at the time.

In those days, FPA was particularly applied to measure productivity and, therefore, people counted after the fact; i.e., on the basis of an application already built. Quite naturally, this was the departure point of the counting guidelines for IFPUG CPM 2.0. The Nesma, however, also wanted to use FPA for budgeting purposes and, therefore, wanted to count beforehand on the basis of an application's functionality. In order to do this, it adapted a number of counting guidelines so they could be applied to logical models. This inevitably led to a number of differences in how the Nesma and the IFPUG counted function points in those days.

With the publication of IFPUG CPM 4.0(1994), the IFPUG also began to count function points on the basis of functionality and to assert that FPA counting must be independent of implementation. Owing to this change in philosophy, and to the intensive cooperation between the Nesma and the IFPUG the counting guidelines of the NESMA and the IFPUG continuously came closer and closer.

With the publication of IFPUG CPM 4.1 (1999) the FPA counting guidelines became the same, except a few guidelines. With the publication of IFPUG CPM 4.2 (2004) the last major differences between IFPUG and NESMA disappeared.

Differences between IFPUG FPA and Nesma FPA

Both the Nesma and the IFPUG now use the same philosophy, the same concepts and terms, and the same rules and guidelines within FPA.

For an actual insight in the remaining (minor) differences please review the document: FPA according to Nesma and IFPUG – the present situation. This document is updated on a regular basis.

Additional FPA Guidelines: Counting Issues made more concrete

Although there are no major differences remaining between IFPUG FPA and Nesma FPA, Nesma published some concrete, operational guidelines on complex counting issues for helping counters. So far, the IFPUG has not (yet) provided specific guidelines on these issues, or to a lesser degree. These additional FPA guidelines fit within the general IFPUG guidelines; they just tend to clarify or interpret the IFPUG guidelines. That's why these guidelines are also applicable for those FPA counters using the IFPUG Counting Practices Manual.

These additional FPA guidelines are relevant to every function point analyst. The Nesma CPM covers them all in detail. If you would like to get an impression of the topics covered by the Nesma Counting Practices Manual, you may review the table of contents of Nesma CPM.