Some doubts about the objectivity of logical determination of the uniqueness of the elementary process in the Function Point Analysis

Marian Zalcman, Ph.D.

ASSECO Poland, Rzeszów

Table of Contents

1. Introduction

Function Point Analysis (FPA) is one of the best methods for measuring the functional size of the application or project. This high rating was confirmed by the ISO, entering the functional size as a standard in the assessment of software and the Function Point as its unit of measure.

In this paper I have presented some doubts arising from the definition of the uniqueness of the elementary process, which was introduced in the CPM R.4.3/R.4.3.1. The case seems to me very important since it affects the integrity and logical objectivity of the FPA method, which is essential for the proper development of this method (Section 4 of this article).

To set the matters related to the presented topic in the correct proportion, in the right light, and the right place, first you must precisely specify the interdependence of the FPA and FUR (Section 2 of this article), which has an important impact on the way of determining not only the elementary processes, but all elementary functions, and thus the functions of both data and processes of elementary / transactional functions.

I also emphasize and draw the attention to the accuracy and the correct direction of changes introduced in the individual versions of CPM. I demonstrate this in Section 3 of this article. It can be clearly seen that the case, associated with certain doubts in determining the uniqueness of the elementary process (which I present in Section 4 of this article) appeared rather incidentally (perhaps by mistake?)

However, it is in my opinion important enough (as I have already pointed out), it must be clearly articulated and resolved in the near future.

2. The relationship between the FPA and FUR

Function Point Analysis (FPA) is a method for measuring functional size as defined within this International Standard (ISO/IEC 14143-1:2007).
(*CPM R.4.3.1, term 3.36, page 6*)

Function point analysis. The method for measuring functional size as defined within the IFPUG Functional Size Measurement (FSM) Method.
(*CPM R.4.3.1 Part 5 – Appendices and Glossary, page G-4*)

FPA is a very positive and useful method among other methods for the measurement of software, but the condition of its objectiveness and the practical utility consists in the fact that the method is fully based only on the logical design/schema of the system (project or application).

And that is precisely stated in the description of the FPA (*CPM R.4.3.1 Part 1 FSM, Chapter Introduction, page iii*)

Function point analysis measures software by quantifying the tasks and services (i.e., functionality) that the software provides to the user based primarily on logical design. The objectives of function point analysis are to measure:

- functionality implemented in software, that the user requests and receives;
- functionality impacted by software development, enhancement and maintenance independently of technology used for implementation.

Function Points Analysis (FPA) consists of a logical decomposition of the system into logical parts and analyzing them. This decomposition is based on the use of a logical model for each system. According to the assumptions of the analysis, any existing or future system can be reduced to this model.

Logical decomposition of a system is the foundation and the strongest side of the FPA method.

Another important term associated with the measurement of the functional size of the project / application is a **Functional User Requirements (FUR)**:

Functional user requirements - a sub-set of the user requirements specifying what the software shall do in terms of **tasks and services**. (ISO 14143-1:2007)
(*CPM R.4.3.1 Part 5 – Appendices and Glossary page G-4*)

In other words:
FUR is a **strictly specified** subset of the User Requirements that describes (in terms of tasks and services) all the software operations required by the user.

And here is a very important conclusion:

FUR is only an input (input data) for the logical analysis of the FPA.

Therefore it is not the role of FUR to define explicitly elementary functions. These functions are the result of decomposition carried out in the project or application and logical analysis by using the FPA method, for which the input data is the FUR.

Without such an approach, neither the comparison of the functional size of individual projects or applications will not be a reliable, nor we will get repeatable and reliable results in the evaluations performed by different specialists

3. Very suitable direction of changes made in CPM R4.2.1 & R.4.3.1

It must be admitted that the updating and making changes to the various versions of the CPM is carried out in the correct direction and in an organized manner which corresponds to the needs of everyday software reality. These changes assure the FPA method is developing properly, is constantly alive and correctly adapted to the changing conditions of the creation and use of software. To justify this, I briefly mention these changes in the last two versions of CPM.

Also the change (introduction of a separate step to determine the uniqueness of the elementary process) was extremely worthwhile and necessary despite the fact that just some aspects of this operation raises questions/doubts which are the subject of this article. However, these doubts are not related to the advisability of the change (because this is right!), but to some clarifying conditions that have been introduced additionally (see Section 4 below).

Changes introduced in CPM R.4.2.1

I. Introduction of the data distribution (***Business Data, Reference Data and Code Data***), with a particular focus on ***Code Data***

II. Identifying data functions, using Data Modeling Concepts (***Entity (In-) Dependency Method***)

III. A deep and instructive analysis of ***Shared Data***

IV. Additional guidance for identifying and counting functional changes to installed applications (***Enhancement Projects and Maintenance Activities***)

Changes in CPM R.4.3.1 in comparison with CPM R.4.2.1

I. Matching FPA methodology to ISO FSM Standard (very important moment for the future of the FPA method).

II. Changes in the definition of the elementary process.

Elementary process as a basic element of the FPA was in earlier versions the smallest unit of activity, which satisfies all of the following:

- is meaningful to the user,
- is self-contained and
- leaves the business of the application being counted in a consistent state

The new version R.4.3/R.4.3.1 added one very important condition (additional clarification of the definition):

- constitutes a complete transaction.

(*CPM R.4.3.1 Part 2 – page 7-10*)

III. Data Conversion Activity (***Converting Data as an elementary process***)

IV. Modifications in the definition of the uniqueness of the elementary process i.e. the introduction of a separate step for determination of the uniqueness of the elementary process before determining the transactional function type as EI, EO, or EQ. Before this step was a part of the identification of the type of transactional function. This step is very important as it affects the clarity and consistency of the analysis of elementary processes. If we specify a fragment of an activity as an elementary process, it is logical to define its uniqueness, and then deal with assigning the right type: EI, EO or EQ. Unfortunately, the introduction of this step did not take place without creating (in the form of additional Notes) some concerns/doubts related to the very precise determination / definition of the uniqueness of the elementary process. Discussion of this issue is a fundamental essence of this article and is presented in the next section (Section4).

4. Doubts related to the lack of precision in determining the uniqueness of the elementary process

In earlier versions (up to and including R.4.2.1 CPM), as already emphasized, the uniqueness of the elementary process was investigated after determining the type of a process EI, EO or EQ. Therefore, the uniqueness of the process was studied as the uniqueness of qualifying process (EI, EO or EQ) and its determination was a part of the classification conditions of the process (appropriate for its type). It was simply an additional, specific condition for the uniqueness of the elementary process.

This condition had a form:

For the identified process, one of the following three statements must apply:

- Processing logic is unique from the processing logic performed by other external inputs (or other external outputs or external inquiries) for the application.
- The set of data elements identified is different from the sets identified for other external inputs (or other external outputs or external inquiries) for the application.
- The ILFs or EIFs referenced are different from the files referenced by other external inputs (or other external outputs or external inquiries) in the application.

(*CPM R.4.2.1 Part 1 – pp. 7-11 and 7-12*)

In the R.4.3.1 version it is assumed (and rightly!) that the step of determining uniqueness of the elementary process is a separate step of the procedure. It will be executed immediately after the step of identification of the elementary process (and before the step of its classification as EI, EO or EQ) and is described by specific conditions.

Below is given a complete record concerning the uniqueness of the elementary process as it is specified in the manual CPM R.4.3.1 (together with Notes and Examples).

> To determine unique elementary processes, the following activities shall be performed
>
> **Definition**
>
> When compared to an Elementary Process (EP) already identified, count two similar EPs as the same Elementary Process if they:
>
> - Require the same set of DETs and
> - Require the same set of FTRs and
> - Require the same set of processing logic to complete the elementary process
>
> **Note:** One elementary process may include minor variations in DETs or FTRs as well as multiple alternatives, variations or occurrences of processing logic below.
>
> **Note:** When the two elementary processes are compared and it is determined that they contain different DETs, FTRs or Processing Logic, they are identified as separate elementary processes if they are specified as distinct functional requirements by the user.
>
> **Note:** The uniqueness test stated above is intended to be used as a means to compare two EPs that have already been identified and not as justification for splitting a single EP into two EPs as a result of variations. Splitting a single EP into two EPs based on variations would indicate that the rules for identifying an EP were not being satisfied.
>
> For example, when an EP to Add Employee requires additional DETs to account for European as well as US employee addresses (postal code/zip code, country/state, phone number country and city code). The EP is not divided into two EPs to account for the minor differences in the employee's address. The EP is still Add Employee, and there is variation in the processing logic and DETs to account for differences in the address and phone number.
>
> For example, when an EP to Add Employee has been identified, it is not divided into two EPs to account for the fact that an employee may or may not have dependents. The EP is still Add Employee, and there is variation in the processing logic and DETs to account for dependents.
>
> For example, when the functional user requirements specify the need for two similar reports (such as when Report 1 contains Customer Name, Customer Id, and Address and Report 2 contains Customer Name, Customer Id, Address, and Phone Number), the reports are identified as separate EPs since the functional user requirements specify the need for different DETs. The reports are not combined into a single EP simply because they have similar DETs.
>
> - Do not split an elementary process with multiple forms of processing logic into multiple elementary processes. If an elementary process is inappropriately sub-divided, it would no longer meet the criteria (listed above) of an elementary process.
>
> (*CPM R.4.3.1 Part 2 page 7-11 and 12*)

The condition in Definition is the most just and reasonable and everything would be fine, if not the first two Notes, appearing directly under this condition:

Doubts about the two Notes are as follows:

Note 1: What are the **"minor variations"** in DETs and FTRs, and **"multiple alternatives, variations or occurrences"** of processing logic? These all "**small**" differences - it means what? It is extremely inaccurate and imprecise wording (neither mathematical nor logical, rather literary!!), which allows interpretation of the given definition of the uniqueness condition!

Without such a precise definition of the terms introduced, we can assume, bringing the issue to the absurd, that each project / application may <u>always</u> consist of one EI, one EO and one EQ, as other EIs, EOs and EQs may be only treated as "<u>minor differences</u>" (or "<u>minor variations</u>" in DETs or FTRs as well as "<u>multiple alternatives, variations or occurrences</u>" of processing logic) in comparison to the others (if only so requested by the user in the FUR)!

<u>Note 2</u>: The definition of the uniqueness of the elementary process should be the basis for the FPA method to determine the uniqueness of the extracted elementary processes using logical decomposition of the system.

However, according to the Note 2 this all depends on the whim of the user. The user in the functional requirements may decide, whether two processes should be treated as separate (unique) or not. Why the uniqueness of the elementary process is to be determined by FUR?

Where is the place for the FPA and logical decomposition? The conclusions from them should determine each elementary process and its uniqueness.

Doubts are augmented by the Examples that follow immediately after the Notes, especially the first and last (third) Examples. In both of them the elementary processes should be or the same or different.

According to the FPA the functional size of application or project is to be calculated based on the logical model of the application or project, which is a logical result of decomposition and that is unique to the application/project and provides unique value of the functional size. However, as you can see in the Examples above, the same application can have different "logical" models, depending on the details of the wording of the functional requirements of the user. (If in FUR you will specify one report we have 1 EO / EQ , and if - two separate reports, we have 2 EO / EQ.) This leads to ambiguously defined functional size value, undermining its credibility.

Therefore, it is completely unacceptable to give one party (in this case to the user) the right to decide on determining of the uniqueness of elementary processes and resulting from this a number of transactional functions, what clearly affects the obtained value of the functional size, measured in FP. According to me, it can distorts the idea of the whole method and make it incompatible.

This may also results from the following example. Imagine that two <u>identical</u> systems are made independently for two different users. One user requires to recognize the elementary processes, differing by one field (DET), as identical, the second one assumes that they are different elementary processes. If we calculate the functional size of such a system in both cases, the result will come out different in each case. What will be, therefore, the real/true functional size of such a system?!

There is also another problem. We should be aware that the value of the functional size determines the cost of the system, and thus the amount of money that the user will pay the developer for the system. I have participated many times in such negotiations and I know that there are often very acute disputes and divergences. So giving one party a certain advantage in deciding on the value of the functional size of the system is absolutely inacceptable, because it can be abused by this party.

5. Proposals for changes. Conclusions

We have discussed two issues in the article:

1. The precise allocation of the roles between FUR and FPA
2. Logical clarification of inaccurate wording (more literary than mathematical or logical) used in the definition of the uniqueness of the elementary process

1. FPA must remain a strictly logical method, and only then its results can properly determine the shape and structure of individual elementary functions (data functions and / or transactional functions)

   Without such an approach, any comparison of the functional size of individual projects or applications will not be reliable, because it would be a functional size, determined directly by the customer whim or intuition, not the real functional size, derived from the correct use of FPA.

   As was said:

   Functional user requirements - a sub-set of the user requirements specifying what the software shall do in terms of **tasks and services**. (ISO 14143-1:2007)
   (*CPM R.4.3.1 Part 5 – Appendices and Glossary page G-4*)

   In other words:
   FUR is a **strictly specified** subset of the User Requirements that describes (in terms of tasks and services) all the software operations required by the user.

   Therefore FUR should be the basis for the implementation of the FPA and cannot replace it in its functionalities and activities.

   If even (quite rare) in FUR decomposition can / must be done to some extent on the level of the elementary process and below, such findings must be confirmed by the decomposition of the system as a whole, using the method of FPA. And this result is only credible /reliable. The user is not a specialist in the use of the FPA method, therefore his suggestions need to be confirmed by such specialists.

   Next issue:

   In the description of the process uniqueness appeared concepts that are logically not accurate, and therefore may be logically inconsistent; it is rather a collection of wishes expressed in literary form, which can be quite freely interpreted.

   There is a need for such fine tuning (refinement) of these "minor variations in DETs, FTRs as well as multiple alternatives, variations or occurrences of processing logic” to restore the integrity of the step of determining the uniqueness of the elementary process that has accurately and most rightly been introduced as a separate step in the current version of CPM.

   This may be done by a listing of these "minor variations in DETs, FTRs as well as multiple alternatives, variations or occurrences of processing logic” (even though such a solution would not be very elegant) or by fine tuning of the definitions of various types of processing logic (mainly because of the precision of processing logic going on here), and such their clarification to automatically ensure the uniqueness of the elementary processes of different processing logic (more elegant solution, but much more labor intensive and difficult to achieve satisfactory results)

   The most optimal solution would be, however, the formulation of determining the elementary process and its uniqueness by a methodology similar to the one presented for determining the data functions (*Entity (In-) Dependency Method, using Data Modeling Concepts*). This task is certainly difficult and it is not known to what extent practically achievable. But dreams of such a solution exist, as it would in an absolute manner make the step of determining the uniqueness of the elementary process logically fully consistent

References:

1. Function Point Counting Practices Manual (CPM) Release 4.3.1, Parts 0 – 5, IFPUG, CPC, New York, January 2010

About the author:

By education I am a mathematician and computer specialist (systems analyst)

I deal with the problems of the FPA for more than 5 years. For the purpose of my company I translated into Polish almost the entire CPM R.4.2.1, then CPM R.4.3.1. I presented these translations, divided thematically, in the form of 15 presentations in MS PowerPoint, which made it easier to access and easier to understand the FPA principles for employees willing to get acquainted with it

E-mail business address: marian.zalcman@asseco.pl
E-mail private address mzalcman@vp.pl