Definition of When Requirements Should be Used to Count Function Points in a Project

Denis St-Pierre, Jean-Marc Desharnais, Alain Abran and Barbara Gardner

This paper is aimed at clarifying the Function Point concept of user requirements in a project.

According to the IFPUG CPM 4.0, Function Points are based on user requirements. However, this concept of user requirements is not defined in the CPM and its interpretation may vary across organizations and counters. This could lead to a variance in the count. This paper is aimed at defining a type of requirements to be used in the process of counting Function Points for a project.

During a project life-cycle, user requirements will evolve rapidly in the early phases of a project. The user has to decide which functions will be a part of the application and which functions will not. The decisions of the users regarding the functions of the project are influenced by many things:

- The needs of the organization.
- The risk (business and technical) associated with the project.
- The resources available (budget, staff, ...) in the organization for the project.
- The technology available in the organization.
- The influence of developers (technical staff) through comments and suggestions. etc.

This paper focuses on the influence of the developers.

At the very beginning of a project, the global justification of the project is determined. The global justification is the highest level of specification and is usually very short, for example:

- The organization needs an application to comply with a new tax law.
- The organization needs an application to manage inventory more efficiently.
- The organization needs an application to manage human resources more efficiently.

Starting with the global justification, the user develops requirements that become more and more precise with time. At a certain point, the user will consult software developers to help him write the detailed requirements. Before the user consults, software developers can get a head start by coming up with their own development and implementation requirements based on the global justification. The results of the discussions between the user and the software developers can lead to enhanced requirements. Therefore, we propose a model with three categories of requirements:

Initial User Requirements;
Initial Technical Requirements;
Final Functional Description Requirements.

This paper reviews these three categories of requirements and then the adoption of the Final Functional Description Requirements are proposed as the most reliable type of requirements to be used to count Function Points in a project.

INITIAL USER REQUIREMENTS

This category represents user requirements prior to the various joint working sessions held with software developers; it has the following characteristics:

May not be complete:
For example, Initial User Requirements may lack functions necessary for referential integrity.

May lack "utility" functionality:
For example, necessary validation reports or inquiries may be lacking.

May be impossible to implement or very difficult to use:
For example, a user may ask for an on-line inquiry that requires an hour of CPU processing.

May be too general:
For example, requirements may not include the number of fields.

May not be formal enough.

May vary for different users, if more than one user is responsible for the project:
The requirements of a specific project may vary from one user to another if they do not have the same functional needs.

May state requirements without regard for current and/or future application boundaries.

May be expressed in a framework or a terminology different from the Function Point one.
For example, Initial User Requirements may refer largely to the physical or manual aspects of the system.

EXAMPLE:
In the Human Resources Department of a corporation, a user expresses his needs:
"Whenever I'm working with an employee, I want to be able to view the employee's information by entering his/her name".

Such requirements imply the development of an inquiry screen and a group of data on employees.[1]

Function types of the Initial User Requirements example:
EQ inquiry on a specific employee
ILF employee group of data

INITIAL TECHNICAL REQUIREMENTS

The second category represents the software developers requirements developed from the global justification of the project. This category is referred to as Initial Technical Requirements. The task of the software developers, among others, is to organize the requirements into existing applications, if any. The Initial Technical Requirements may include elements which are necessary for the implementation, but which are not used in the Function Point counting rules (e.g. temporary files, index, etc.). this category has the following characteristics:

May be technology dependent.

May be out of touch with the functional needs of the users:
Software developers tend to add functions not asked for by the users.

May use terminology not familiar to the users.
For example, software developers may refer to physical files rather than to logical groups of data.

[1] To keep this example simple, it is assumed that:
-The employee group of data is maintained internally
-The other employee functions (create, update and delete eployee) are not described here.

Functionality may be determined by putting too much emphasis on technical constraints:

For example, some developers tend to limit the scope of the requirements by focusing on the computing capacity (CPU) currently available in the organization.
Boundaries are determined according to the technical architecture of the other applications of the organization.

Continuing with the same example, says the IS analyst:
"I recognize the need for an employee inquiry. Besides, I believe that an index is necessary to speed up the retrieval of a specific employee".

Function types of the Initial Technical Requirements example:
EQ inquiry on a specific employee
ILF employee group of data
ILF index on the employee file[2]

FINAL FUNCTIONAL DESCRIPTION REQUIREMENTS

The third category of requirements is the result of joint sessions with the user and the software developers. This category is referred to as Final Functional Description Requirements. The joint sessions are necessary to achieve a consistent and complete functional description of the application to be developed. This category is the last version of the functional requirements before the beginning of the development phase in a project life-cycle, and has the following characteristics:

- Terminology which can be understood by both users and software developers.
- Is expressed in a way which is not related to any technology.
- More complete than two other categories:
  Ideally, this category would include all the functions which will be implemented.
  Ideally, this category would include all details necessary to count Function Points.
- Integrated description of all the users responsible for the project.
- Formal enough to count Function Points.
- Consistent.
- Each process and group of data is "approved" by the user.
- The feasibility and usability is "approved" by the software developers.
- Is integrated functionally with the other applications of the organization.
- It does not include redundant functionality, unless explicitly asked for by the user.
- Is the last functional version of the requirements prior to development.

Continuing with the same example:
User: "Whenever I'm working with an employee, I want to be able to view the employee's information by entering his/her name".
IS analyst: "I recognize the need for a single employee inquiry. But, since many employees may have the same name, it is not possible to specify and employee by typing his/her name. Therefore, I suggest an on-line employee list (name,location and social security number) to select a unique employee.
Besides, I believe than an index is necessary to speed up the retrieval of a specific employee".

continued on page 22

[2] According to the IFPUG CPM, 4.0 index filed are not counted becuase they do not represent the user's view. In this example, the index file has been counted as an ILF to illustrate a typical error of software developers.

Definition of When Requirements Should be Used to Count Function Points in a Project

continued from page 7

User: "I agree that the employee selection list is necessary in this case, and it will also be used for purposes other than selecting an employee. However, the index is a technical solution, not a functional description of my project".
Result of the discussions between the user and the IS analyst:
Include the on-line list of employees
Exclude the employee index

Function types of the Final Functional Description Requirements example:
EQ inquiry on a specific employee
EQ on-line list of employees
ILF employee group of data

The Functional Description Requirements is the last version of the requirements before the beginning of the development phase. It is only then that there is an agreement that the documented requirements are considered complete, formal and approved. This type of requirements allows for a count that may be closer to the count of the application once developed.

CATEGORY COMPARISONS

At an early stage, the count of Initial Users Requirements may be the only one available. In spite of the disadvantages mentioned previously, in good conditions this count can be very useful to produce an early estimate. Nevertheless, in general, a count of the Initial User Requirements is less likely to be accurate[3] than a count of the Final Functional Description Requirements.

A count of the Initial Technical Requirements may be significantly different from the count of the application once developed because of the emphasis on details relevant only to software developers.

CONCLUSION

To make a point, we have described in this paper the two extremes on the spectrum of requirements (figure 1): initial requirements (user or technical) and final requirements. In reality, requirements of a project may exhibit a mix of the characteristics described here for initial and final requirements.

The example presented in the paper may be oversimplified. Nevertheless, we hope that it illustrates that to determine the reliability of a count, it is necessary to determine the type of requirements that were used to count.

This paper is only aimed at stimulating discussion. More discussions and research are needed to resolve the type of requirements that should be used to count Function Points in a project.

[3] Accuracy is determined by using the difference between the count of the chosen category and the count of the application immediately development.