

MetricViews

Published twice a year by the International Function Point Users Group (IFPUG), headquartered in Princeton Junction, New Jersey, U.S.A.

MetricViews August 2015

Editor

Paul Radford

IFPUG Board of Directors

President

Kriste Lawrence

Vice President

Tom Cagley

Secretary and Director of Communications & Marketing

Mauricio Aguiar

Treasurer

Debra Maschino

Immediate Past President

Joe Schofield

Director of Applied Programs

Christine Green

Director of Sizing Standards

Dácil Castelo

Director of International & Organizational Affairs

Pierre Almén

Director of Education & Conference Services

Luigi Buglione

IFPUG Office

Executive Director

Connie Holden

Association Coordinators

Nicole Lauzon

Michele Giovine

Views and opinions presented in *MetricViews* articles may not represent those of the International Function Point Users Group (IFPUG).

Please submit all articles, news releases and advertising to:



IFPUG/MetricViews

191 Clarksville Road

Princeton Junction, NJ 08550

(609) 799-4900

ifpug@ifpug.org

Ten Fundamental Questions about Function Point Analysis

By *Guilherme Siqueira Simões, CPFS and Carlos Eduardo Vazquez, CPFS*

1. What is Function Point Analysis? What is A Function Point?

Function Point Analysis (FPA) is a software measurement technique based on the users point of view. It measures the software functions and the Function Point (FP) is its measuring unit. The method has as an objective to become independent of the technology being used to build the software. In other words, FPA measures what the software does and not how the software was developed.

This being said, the measurement process (also called function point counting) is based on a standard evaluation of the user's functional requirements. This standard procedure is described in the IFPUG *Counting Practices Manual*.

The main estimation techniques used for software development projects assume that the software size is an important driver for the estimation of its development effort. Thus, knowing its size is one of the first steps in the effort, duration and cost estimation.

At this point it is important to know that function points do not measure effort, productivity nor cost directly. It is exclusively a software functional size unit. This size, along with other variables, is what could be used to derive productivity, estimate effort and cost of software projects.

2. Who created Function Points Analysis? Why it was created?

Function Point Analysis (FPA) was invented in the 1970s as a result of a project developed by the researcher Allan Albrecht of IBM. His job involved a productivity analysis for software projects developed by a service unit of IBM. To do this he developed a method to measure software independently of the programming language used, checking only the external aspects of the software, primarily based on the user's vision.

3. Is the Function Point Analysis technique owned by some company?

No. Despite having emerged in IBM, the result of this project was opened to the whole software community.

Nowadays, the standard recognized for Function Point Analysis (FPA) is defined in the IFPUG *Counting Practices Manual* (CPM) maintained by the International Function Point Users Group (IFPUG).

IFPUG is a nonprofit entity composed by people and companies from all over the world, with the purpose of promoting better management of development and software maintenance processes through the use and effective understanding of Function Point Analysis.

4. What are Function Point Analysis benefits?

We can highlight several benefits on applying function point analysis:

- Provides a tool for estimating costs and resources for developing and maintaining software. By carrying out a count or estimating function points early in the life-cycle of a software project, it's possible to determine its functional size. This measurement can be used as input for many models of effort, time and cost estimation.

(continued on next page)

(Ten Fundamental Questions, continued from page 4)

- Supports the analysis of productivity and quality, either directly or in conjunction with other metrics such as effort, cost and defects.
- Supports project scope management. A challenge of any project manager is to control “scope creep”, or the increase of the scope. By making estimates and measurements of function points of the project at every stage of its life cycle, it is possible to determine whether the functional requirements increased or decreased. Evaluation can be made of whether this variation corresponds to new requirements or to requirements that already existed but were not well understood until more detailed examination.
- Offers a tool to support contract negotiation. Function points can be used to generate several service level indicators (SLA – Service Level Agreement) in software development and maintenance contracts. Besides that, it allows contract establishments by using a unit price per function point, where a unit represents a tangible asset to the client. This modality allows for a better risk and responsibilities distribution between the client (who manages the scope and cost) and the provider (who manages the productivity and quality).
- Complements requirements management to assist in verifying the unambiguous and completeness of the specified requirements. The process of counting function points favors a structured and systematic analysis of the requirements specification and brings similar benefits of a peer review process.

5. Is it necessary to be a software developer to do Function Point Analysis?

Absolutely not. The great advantage of Function Point Analysis is that it is based on the USERS POINT OF VIEW, allowing its concepts to be understood by the developer and the user. To measure the function points it is necessary only to know the requirements that the software must attend to and the rules presented by the CPM.

6. Who uses Function Point Analysis in the world?

IFPUG has affiliates in more than 40 countries around the world.

Companies such as IBM, Unisys, Xerox, HP, CitiGroup, Tata Consulting Services, Lockheed Martin EIS, Booz Allen & Hamilton, Nielsen Media Research, Banco do Brasil, Citibank, HSBC, Indra, Bank of Canada, Ralston Purina Co., Banco de la República (Central Bank of Colombia), Northrop Grumman Corp, Samsung SDS Co Ltd, BASF Corporation, Banco Central de Chile, Accenture, IBM, Petrobras, Pepsi Co, Compuware, Price Waterhouse Cooper, Vale, Banco Santander, Petrobras and Telefonica, among others, are using function points for software project management.

7. What tools are suitable for support and/or to automate the use of FPA?

The first point to note in this issue is that there are no tools available that automatically count function points reliably. However there are tools available that can support and partially automate the process of function point counting and also to store and manage the results of the counts.

The simplest tool to be used to record a function point count is a spreadsheet. Despite being the first and simplest tool to be used by many professionals, its use begins to be impractical as the number of counts increases. The control of the counting repository is usually manual, and with the increasing amount of data, the task becomes costly.

When the organization realizes that the spreadsheet no longer meets its needs, a natural course of action is to search tools with more capabilities on the market. IFPUG has a certification process for the tools to support the function point counts. According to this process, the tools can be classified into three categories:

Type 1: The user does the function points count manually and the software provides functionalities for data collection and calculations.

Type 2: The software provides the functionalities for data collection and calculations, and the user and the system do the

(continued on next page)



CHARISMATEK
SOFTWARE METRICS
www.charismatek.com

**Function Point
WORKBENCH**

Release 7.0L out now!

For the people who count.

(Ten Fundamental Questions, continued from page 5)

interactive function points count, using questions submitted by the system and actions being taken automatically depending on the answers provided.

Type 3: The software automatically produces a function point count using various sources of information, such as the database application, the application itself and artifacts of the development tools. The user can enter the data interactively, but his involvement is minimal during the count. Almost by definition, these tools do not provide a true function point representation (refer to Point 8 below). It is important to note that there are no such tools certified.

Although there are several options of tools on the market to support the use of function points, many organizations choose to develop an in-house tool integrated with its systems of internal control. Some reasons for this may be:

- The perceived cost to develop an internal solution is less than the cost of acquisition and maintenance of packages available on the market (obvious nonsense – some counting tools have over 1500 function points of functionality and sell for less than \$1/FP – this is the solution provided by programmers-who-want-to-have-fun).
- Lack of local support for the solution, due to the fact that most tools on the market are foreign (given the limited market size and cost of development of a useful tool, only a few are available and mostly from English speaking markets; however, some of these do provide at least reporting support to multiple languages).
- The need to integrate with internal systems.

8. Why automatic tools cannot correctly count function points?

There are some software products that from a program model or its source code calculate its size in function points. However, comparisons between the results produced by different tools for the same system frequently have an unacceptable variation. These numbers, also often differ greatly from a manual count.

The answer to this variation is in how these tools calculate the number of function points. Some are based on files, screens, reports and other elements to derive a number. Although there is often a direct relationship between these objects and data functions and transactions functions of Function Point Analysis (FPA), it must be remembered that the technique measures only the logical functions of the system. And these tools have difficulties distinguishing logic functions and physical functions. For example, not every file or table from a program file corresponds to an internal logical file or external interface file. Or even an elementary process can be implemented through multiple screens. To do the measurement in a correct way, the software should have enough

intelligence to make this judgment. That is, this software has the skill to read the program and interpret the user's requirements. However, there is no software with this artificial intelligence.

There are software products to support the process of counting function points that automate a part of the process, but the decision and analysis of that should be considered is responsibility of the human user who enters the data, and not of the software.

9. What kind of software can be measured by Function Points?

FPA is a technique to measure the functionalities that the software provides to the users; and this measurement is always made on an external perspective, the users' perspective. However, it is important to say that the concept of user for FPA is not only the one of the end-user of the software. The user for the FPA is any person or thing that interacts with the software at any time. In other words, the user for FPA can be both the person acting as end-user to the software and another software that uses the services of the software in analysis.

Considering that the objective of any software is to offer one or more services (functions) to someone (person or thing); it is concluded that every and any software can be measured by Function Points.

A common mistake for beginners with FPA is to only consider the end-users point of view. In this case some types of software will be partially (or completely) "invisible" to this user. Then they mistakenly conclude that FPA does not work for that kind of software. The most common is for the person to learn the principles of the FPA applied to systems with screens and reports. However, when this person faces some software domains that do not have screens, like batch processing, middleware, software embedded, data warehouse, it is natural to have some difficulties on measuring it.

Let's imagine that the goal was to measure a printer's driver. Well, there is no end-user (person) for this kind of software. In this perspective, the printer's driver is invisible to the end-user. However it exists to offer services to someone; in this case, the operating system. Thus, analyzing the printer's driver in the perspective of the operating system, it is possible to see functions, for example: to start the printer, inform the general situation of the device, eject a sheet of paper, print, alert the level of the ink, etc...

10. Is it possible to use FPA in a project using agile methodology?

Certainly! The FPA is a technique that is independent of the technology used to model or construct software. Therefore, that software will have the same size in function points whether

(continued on next page)

(Ten Fundamental Questions, continued from page 6)

someone use an agile methodology or any other approach to develop it.

What will probably distinguish the measurement of an agile project and other traditional methods are the artifacts that are being used to perform the analysis. In a more conventional approach, for example similar to the Rational Unified Process (RUP), artifacts used for measurement will probably be use case specifications, which are detailed descriptions of the functionality from the viewpoint of a user while interacting with the software.

Agile projects have a greater emphasis on delivering working software than producing a detailed documentation of what will be done. Therefore, it is more likely that user stories will be used in an agile methodology to specify requirements, which are brief descriptions of the desired functionality by the user.

However, user stories are not enough to provide all information necessary to measure function points (although they are sufficient to provide an estimate/approximation of the size in FPs). So how are we able to measure a project?

Sometimes, the developer cannot build the software only with the information provided by the user stories. More detailed requirements are necessary for one to build the desired software. Where can a developer attain more detailed information to build the desired software besides user stories? It is very likely that the developer will turn to the user. The agile methodology advocates that the user join the development team, having a very close interaction with the developers.

Therefore, assuming that the developer attains more detailed information about the requirements to build the software, that same information will be useful when counting FPs.

The Function Point Based Pricing Model in Brazil

By Mauricio Aguiar, TI Metrics

1. Introduction

Founded in 1986, The International Function Point Users Group (IFPUG) is probably the oldest software measurement association in the world. As of April 2015, Brazil was the number one country in IFPUG memberships, with 37% of all IFPUG members, followed by Italy (18%) and the U.S. (17%). IFPUG certifies individuals that pass the IFPUG exams: CFPP and CFPS. As of April 2015, Brazil had 35% of all IFPUG certified individuals, followed by Italy (32%) and India (11%) [1]. These numbers make Brazil the number one country in function point utilization. As a result, several software measurement companies and independent consultants appeared in the Brazilian market in the last decade. Even though there is no publicly available data, a single Brazilian company has claimed to count 60,000 function points per month [2]. At the cost of \$1,000 USD per function point [3], that number could mean \$60,000,000 USD changing hands each month based on function point

counts performed by just one Brazilian software measurement organization.

There are several possible explanations for the growth of software measurement in Brazil in the last several years. The following paragraphs intend to shed some light on this topic.

2. A Very Short History of Software Measurement in Brazil

The first Brazilian book on software measurement was Aguinaldo Aragon's "Gerencia Efetiva de Software Atraves de Metricas" ("Effective Software Management through Metrics") published in 1995. Aragon's book included sections on function point analysis, COCOMO, and linear regression as effort estimating techniques. It also presented several applications of metrics to software management [4].

Function point analysis has been in use in Brazil since the eighties. However, it became more popular in the nineties, when UNISYS Eletronica became its main local sponsor. This was the first

Brazilian function point movement, or the "First Wave". UNISYS Eletronica joined IFPUG in 1989 and started sending employees to IFPUG conferences in 1990. UNISYS Eletronica promoted function point user meetings in Brazil, called ENUPFs, from 1991 to 1994, featuring several international participants. UNISYS Eletronica sponsored a Certified Function Point Specialist (CFPS) exam in Brazil where a few of its employees became CFPS. Unfortunately, for internal reasons that company severely reduced its FPA sponsoring efforts around 1995 so there were no significant changes in the Brazilian measurement market until 1998. The first Brazilian function point analysis book was published in 1996 [5].

In 1998 a group of Brazilians joined IFPUG and founded the Brazilian Function Point Users Group (BFPUG) that had a significant role in promoting software measurement and function point analysis in Brazil. This was the second Brazilian function point movement, or the "Second Wave". BFPUG received

(continued on next page)