

August 2015

INTERNATIONAL  
FUNCTION POINT  
USERS GROUP

191 Clarksville Road  
Princeton Junction, NJ 08550  
USA

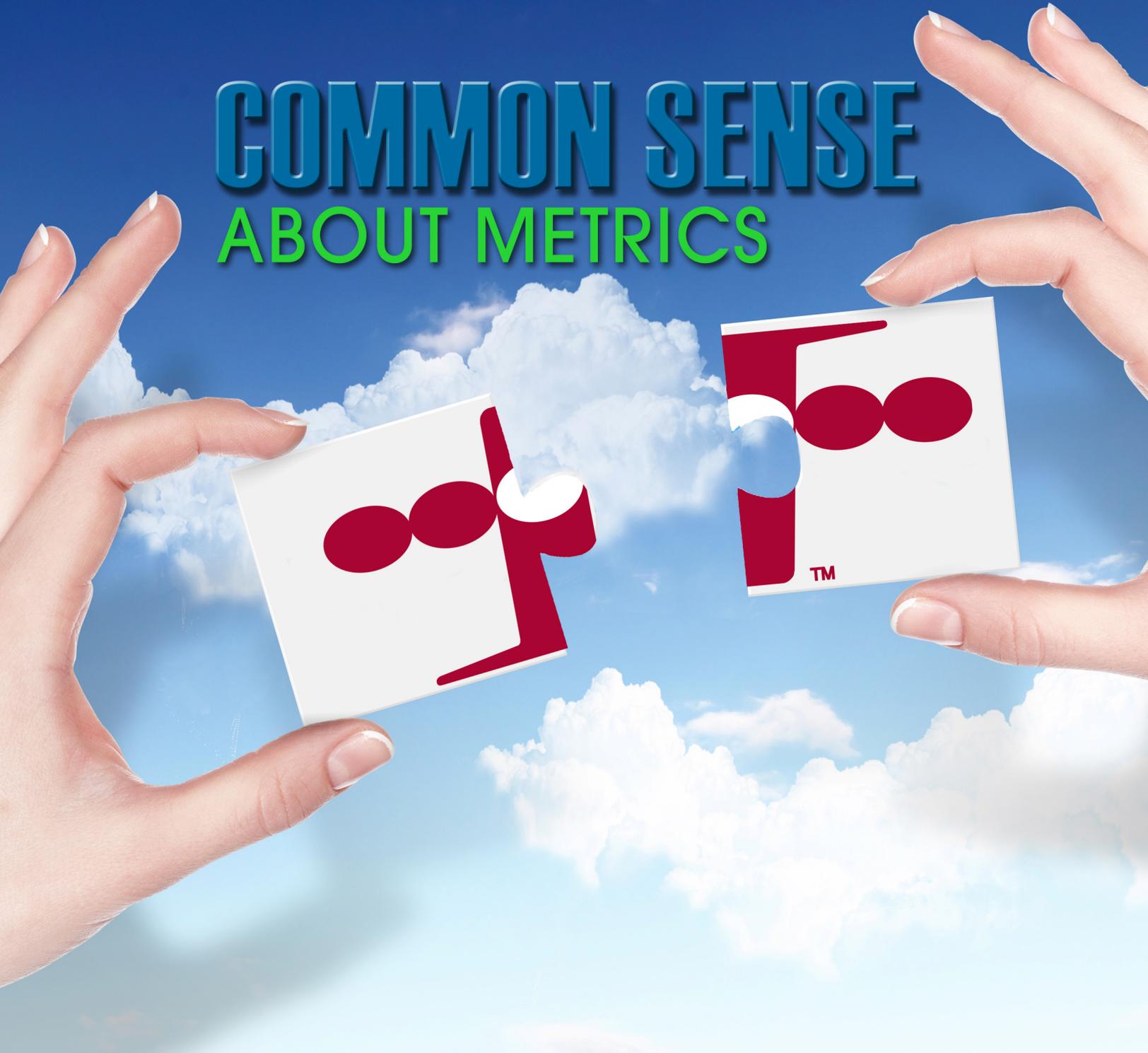


Vol. 9 • Issue 2

# Metric Views

[www.ifpug.org](http://www.ifpug.org)

## COMMON SENSE ABOUT METRICS



## Inside...

|   |       |
|---|-------|
| In this Edition.....  | 2     |
| Message from the President .....  | 2     |
| From the Editor's Desk.....   | 3     |
| Fundamental Questions About<br>Function Point Analysis.....   | 4     |
| Function Point Based Pricing<br>Model in Brazil.....  | 7     |
| 10 Common Challenges in<br>Measurement Process.....   | 11    |
| How Function Points and SNAP<br>Work Together .....   | 14    |
| SNAP Going Beyond Non<br>Functional Requirements.....   | 16    |
| Using Project History to Produce<br>Effective Estimates .....   | 18    |
| Component Sizing.....   | 23    |
| Doubts About Objectivity of Logical<br>Determination of Uniqueness of<br>Elementary Process in FP Analysis..... | 28    |
| Committee Reports.....  | 33-36 |
| Conference and Education Committee  |       |
| Communications and Marketing<br>Committee   |       |
| Functional Sizing Standards<br>Committee  |       |
| Non-Functional Sizing Standards<br>Committee  |       |
| International Membership Committee  |       |
| International Standards (ISO)<br>Committee – Committee Highlight.....   | 35    |
| Board of Directors.....   | 37    |
| Behind the Scenes .....   | 37    |
| Committee Rosters .....   | 38    |
| CFPS Certification Listing .....  | 38    |
| CFPP Certification Listing .....  | 40    |
| CSP Certification Listing.....  | 41    |
| CFPS Fellow Certification Listing .....   | 41    |
| Certification Matters .....   | 42    |
| ISMA10 In the USA .....   | 43    |
| ISMA11 & ISMA12 Information .....   | 44    |

## IN THIS EDITION

- a lot of good old common sense
- a few golden rules
- some good advice
- some examples of how metrics and practical estimating can be established
- some good information on SNAP and how to apply it
- an interesting, thought provoking piece on the success of IFPUG in Brazil and its implications for the future
- some esoteric stuff as well, delving into different modes of measurement as well as discussion of some of the inconsistencies applying within our own
- and, of course, information on what has been happening within IFPUG working groups
- what will be happening, where and when
- and look out for the next ISMA in Sao Paulo



### Message from the President

*Kriste Lawrence*

### “Common Sense about Metrics.”

We are already in the month of August and are getting ready for the upcoming IFPUG Board of Director elections. This year, there is only one Board position open.

Last year's election resulted in the Board make-up being five international members; Mauricio Aguiar of Brasil, Christine Green of Denmark, Dácil Castelo of Spain, Luigi Buglione of Italy, and Pierre Almén of Sweden. The other four members are from the United States; Joe Schofield from Arizona, Tom Cagley from Ohio, Debra Maschino from Georgia, and myself from New York. The 2015 IFPUG Annual Meeting will be held in Cracow (sometimes spelled Krakow), Poland, in keeping with our plans to go out and meet IFPUG's membership. I can't wait to see what happens with this year's election!

IFPUG as an organization is always changing but has an underlying common sense about metrics and how our products and services are used. I started this article discussing the upcoming elections and our current Board of Directors. It is a fact that an IFPUG Board member must have common sense about metrics and must have common sense about IFPUG as an organization. A Board member must always have IFPUG's best interests in mind regardless of potentially competing influences or priorities.

One year ago, the IFPUG Board decided to switch around some of the directorates and the committees that they support. We moved the Non-functional Sizing Standards Committee to work with the renamed Director of Counting Standards and the Functional Sizing Standards Committee. This switch has promoted a great partnership between the two committees and has seen them work together to develop better resources for our members. The switch was just “common sense.”

*(continued on next page)*

*(President's Message, continued from page 2)*

Another thing we did recently was to eliminate the fee for white papers when purchasing a new or renewed Membership. It just makes "common sense" for our members to obtain white papers for their personal use at no cost. Additionally, earlier in the year we announced a partnership with the IT Metrics and Productivity Institute (ITMPI) which gave our members a free membership to all of ITMPI's webinars and reference materials. Partnerships such as ITMPI are another way of IFPUG showing its "common sense" to our members.

I would like to thank the Board and the Committees for their support throughout my tenure as President. Both groups work tirelessly for the IFPUG membership by working toward defining our future and developing new materials. Tom Cagley will be assuming the role of President of the Board on November 1, 2015.

Under Tom's direction, the new Board will continue to enhance IFPUG's presence around the world and will strive to meet more and more of our members. I hope that you, as members, look for ways in which you can engage the Board and other IFPUG members to make an even stronger connection and look for ways to further instill "common sense" in all that we do.

**Kriste Lawrence**  
**IFPUG President**

perceive around us – but it is a bigger problem and a greater drain on resources than most of the trivial issues that occupy the daily attention of our newspapers and on-line twitterfests.

I have my own personal list of (Australian) project disasters due, primarily, to a deliberate refusal of stakeholders to adopt common sense. Billions of wasted dollars. I have encountered similar or worse responses all around the world, which would dwarf this Aussie mountain of wasted cash and people. Yet history repeats – again and again.

And the hidden cost is vast. We all pay for it, either through our taxes, our workloads or simply the added cost on products and services we purchase.

Whilst a dependence upon instinct and a preferred outcome as a planning process is common, it can be seen to be overcome in some fields where measurement is trusted. When the engineers tell us that the bridge must be re-inforced to enable a heavier load, we believe them – and pay the extra cost. When they tell us there is a cost for better braking technology in our cars – we pay the extra cost. And clients (specific people) for whom we have performed work in the past are more than delighted that they can gain such insight into past and future cost implications.

So TRUST is our biggest problem.

Where people trust us to deliver the results they need, the cost of measurement is relatively immaterial. It is the foundation on which all other work becomes possible.

As a community of people, as guardians of a measurement technique, as professionals within the Information Technology industry, it is up to us to address this issue. We need to identify WHY we have failed to gain TRUST as a group and then directly address those issues.

This applies equally to any measurement centre within an organisation as it does to the place of measurement in the world at large – it is only the answers and responses that differ. In both cases, the answer lies within. For IFPUG to achieve a universally trusted role, we must address those issues of conflicting directions and needs of all our participants and demonstrate the way forward, with certainty, confidence and the full support of ALL our people.

We must start this process of change very soon, or history will pass us by.

**Paul Radford**  
**Communications and Marketing Committee**

---



## From the Editor's Desk

### Mid-Year Madness

**Paul Radford**

Madness surrounds us.

In the media, priority is given to the trivial and the popular. And, not to put too fine a point on it, rational discourse and courtesy do not appear to be the main winners in geopolitics or commerce. But underpinning good economics must be accountability.

But the madness of the software industry rarely wavers. Perhaps the madness of refusing sensible measurement before risking tens of millions of dollars on vague software wish lists is less visible and less directly painful than much of what we

## 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)*



**Release 7.0L out now!**

**CHARISMATEK**  
SOFTWARE METRICS

[www.charismatek.com](http://www.charismatek.com)

**Function Point  
WORKBENCH**

*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)*

*(Pricing Model in Brazil, continued from page 7)*

active support from IFPUG starting 1999. As a result, a Brazilian was elected to the IFPUG Board of Directors in 2000. After a couple of years another Brazilian joined the IFPUG Board of Directors. A Brazilian – this author – was elected IFPUG President from 2005 to 2007. Also as a part of the Second Wave, the second Brazilian function point analysis book was published in 2003 [6].

The CFPS exam came to Brazil for the second time in 2001 and became a regular event, being held twice a year in three or four major Brazilian cities until its automation in 2008. A total of 1098 exam candidates sat for the exam in Brazil from 2001 to 2007. Many received the CFPS designation.

A possible reason why The Second Wave was more successful than the first is that the former came because of Brazilian government initiatives. The Brazilian government was concerned

with the cost of software development. Therefore, they were looking for ways of managing and possibly reducing that cost. Caixa Economica Federal (“CAIXA”), a Brazilian government bank, was the leader of that movement. CAIXA was the first government organization to transition its software development projects to a “price per function point” model. They launched a large function point based RFP in 1999. Other government agencies such as the Brazilian Post Office (“Correios”) and the Brazilian IRS Data Processing Service (“SERPRO”) soon followed. A government directive known as “IN04” was issued in 2008 (updated in 2010) and stated that IT services should not be billed based on effort (person-hours) or job positions (persons-month) [7, 8]. This confirmed function points as the measure of choice for software development contracts in the Brazilian government. Those facts got the attention of private organizations

such as banks, telecom companies, airlines, and others that would soon adopt variations of the method.

The success of function point analysis in Brazil led IFPUG to hold its 2010 annual conference in Sao Paulo, Brazil’s largest city. ISMA Cinco (ISMA is the International Software Measurement & Analysis Conference. “Cinco” is Portuguese for “Five”) was successful both in terms of international participation as well as local attendance. BFUG has held an annual software measurement and analysis conference in Brazil since 2009, featuring international speakers from the software & systems measurement arena. ISMA 8 was held in Rio (2013), and ISMA 11 will be held again in Sao Paulo (2015).

In November 2010, the Brazilian government, through its Ministry of Planning, published guidelines to the application of function point analysis to software development contracts [11].

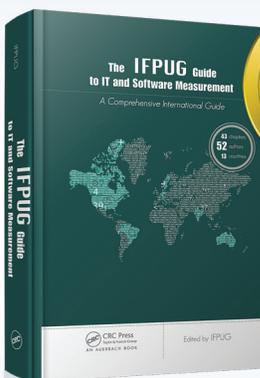
As more organizations adopt the “price per function point” method in Brazil, it is likely that more measurement-related methods and techniques will become popular. For instance, the COCOMO II estimating model and the Practical Software & Systems Measurement framework are currently used in Brazil; The Netherlands Software Metrics Users Association (NESMA) methods for early size estimating and enhancement counts are also used; the COSMIC measurement method is increasing its popularity, as well as the new IFPUG Software Non-functional Assessment Method (SNAP).

### 3. Challenges

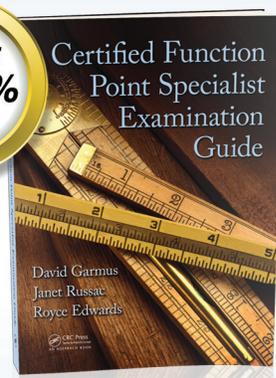
Even though there are many benefits associated with the use of functional sizing in software development contracts there are still many issues to be resolved. Brazilian acquirers and suppliers from both the government and private sectors have addressed those challenges.

## Your Authoritative Guides to Software Measurement

### The IFPUG Way



**SAVE 25%**



Enter promo code **GWP18** at checkout and **SAVE 25% + Free Shipping**

Sign Up For Exclusive Discounts Only Available To Members of the CRC Press Online Community

[www.CRCPRESS.COM](http://www.CRCPRESS.COM)



CRC Press  
Taylor & Francis Group

*(continued on next page)*

(Pricing Model in Brazil, continued from page 8)

### 3.1. Predictive versus Prescriptive Models

One issue has to do with the differences between predictive and prescriptive models. Because most of the software development market still uses conventional effort-based pricing, specialists tend to focus on predictive models to estimate effort. However, in Brazil the most important use of functional sizing is in pricing and billing. Even though there are similarities between predictive and prescriptive models, there are also differences to consider before using them. A few of the differences are listed on Table 1.

**Table 1:**  
Differences in Estimating vs. Pricing

| Estimating (Predictive Model)   | Pricing (Prescriptive Model)  |
|---|---|
| 1. Model inputs must be known in advance  | 1. Model inputs may be known before or after the fact                                 |
| 2. Model is expected to give approximate results  | 2. Model is expected to give exact results  |
| 3. Different estimators may produce different values (depending on their expertise and skill) | 3. Different model operators must produce the same values                             |
| 4. Model inputs do not need to be objective – may depend on estimator’s opinion/assessment    | 4. Model inputs must be objective – must not depend on estimator’s opinion/assessment |

Both estimating and pricing models share the goal of producing values that should be as close to the actual values as possible. Even though neither will be able to match the actual values exactly, both are expected to approximate them in the end.

Estimating model inputs must be known (or estimated) in advance; otherwise, it will be impossible to compute the estimated values. On the other hand, pricing model inputs may be known after the fact; it will still be possible to compute prices. For example, an effort-based pricing model will necessarily have its main input (effort) known only after the fact – when the project is complete.

Estimating models are expected to give approximate results. A slight change in the inputs may not be reflected in the outputs. On the other hand, pricing models are expected to give exact results. A small change in the inputs should result in a (hopefully small) change in the outputs.

Different estimators are expected to produce different results. A more skilled and more experienced estimator is expected to produce a better estimate than a novice does, whereas an operator of a pricing model is expected to follow exact rules and produce exactly the same result as any other trained operator.

Estimating models often have subjective parameters. For example, the COCOMO II estimating model has parameters such as ACAP (Analyst Capability) and PCAP (Programmer Capability) with ratings 15%, 35%, 55%, 75%, and 90% [9]. An analyst or programmer team that falls in the 15% level is rated very low – at the estimator’s discretion. Two distinct estimators could potentially disagree on those levels. Estimating models often have to be customized before they can be used for pricing purposes. Pricing models, on the other hand, have no room for ambiguity or subjectivity.

### 3.2. Items Not Covered by Functional Sizing

By definition, functional sizing does not consider non-functional items. This leaves room for unaccounted effort variation in projects where effort is predicted or prescribed using only functional measures. A simple solution is to define several different project types and assign them different productivities, hoping those will account for all non-functional effects. However, there will always be variation even in a well-specified and calibrated model. Another solution would be to use a parametric model

such as COCOMO II, where all variation due to factors other than size would be accounted for by model parameters (there are 22 of them in COCOMO II). This works well for estimating, but suppliers and acquirers are not happy when variation is not accounted for. In Brazil, some suppliers have built tables that transform certain non-functional characteristics into an equivalent number of function points. That

type of solution has been used in government bids [10]. Other ways of dealing with so-called “non-measurable items” for several types of activities including documentation and testing are defined in Roteiro de Metricas de Software do SISP – Versao 2.0 [11]. Many of those methods add equivalent function points to the functional size, or multiply the functional size by a specified factor.

Even though adding “equivalent function points” to the functional size violates several measurement principles, the Brazilian industry has been using workarounds for lack of a better solution. The industry will typically adopt

(continued on next page)

*(Pricing Model in Brazil, continued from page 9)*

an imperfect solution as long as it is (or seems) better than the alternatives. The software measurement community still has a lot of work to do on non-functional assessment models such as IFPUG SNAP to fix this situation.

### 3.3. Sizing Enhancements

A significant part of the software measurement community in Brazil believes that the IFPUG method for sizing enhancement projects is not optimal for pricing models. They prefer to use the NESMA enhancement sizing method instead. That method assigns different weights for added, deleted, and changed function points [12]. According to some Brazilian users, the NESMA method provides more accurate results than the IFPUG method.

### 4. Benefits and Challenges of Using Functional Sizing in Software Development Contracts

The “price-per-function-point” method potentially leads to better productivity and represents an improvement over previous effort-based methods. It brings transparency and objectivity to the negotiation process, being good for any application domain, development process, and technology.

Special care must be taken when determining initial productivities in order to establish a balanced relationship between acquirer and supplier. While a good pricing model will reduce variation to an acceptable level, it is important to note that bad requirements do not favor accurate sizing. Poor requirements will increase the uncertainty in the sizing process. Any functional sizing method may be used, but it is highly recommended that measurement be performed or at least supervised by certified professionals. This will reduce differences in the interpretation of counting rules, especial-

ly between acquirer and supplier. Sizing may also be outsourced to a neutral third party organization in order to improve transparency and minimize conflict. Non-functional items will continue to be a challenge until a non-functional measurement solution is found and accepted by the measurement community. So far, the most promising solution is the IFPUG SNAP method. Most of all, one should keep a win-win attitude and be aware that when using functional sizing in pricing models there will be gains and losses, but at the end of the day things will balance and everybody will win.

### 5. Conclusion

This article has presented a short description of the utilization of software measurement in Brazil. After providing a historical perspective, several relevant topics were addressed, such as the difference between predictive and prescriptive models, and why the latter is so important in Brazil; how Brazilian organizations in the government and private sectors use functional sizing for estimating and pricing; the main technical difficulties that have been encountered and how they have been addressed.

We hope that this report will contribute to raise interest in the use of functional sizing in pricing models, since so much work is still needed to improve those models.

### 6. References

- [1] IFPUG, unpublished internal reports, IFPUG, Princeton Junction, 2015.
- [2] Aguiar, Mauricio, “Domain-Oriented Modeling, Estimation and Improvement for Commercial and IT Systems Domains” (presentation), 26th International Forum on Systems, Software, and COCOMO Cost-Modeling”, USC, Los Angeles, 2011, [http://csse.usc.edu/csse/event/2011/COCOMO/presentations/Panel\\_Aguiar\\_2011-11-03.ppt](http://csse.usc.edu/csse/event/2011/COCOMO/presentations/Panel_Aguiar_2011-11-03.ppt)

- [3] Jones, Capers, “Applied Software Measurement – Global Analysis of Productivity and Quality”, 3rd Edition, McGraw-Hill, New York, 2008, pg. 310.
- [4] Fernandes, Aguinaldo Aragon, “Gerencia Efetiva de Software Atraves de Metricas”, Atlas, São Paulo, 1995.
- [5] Braga, Antonio, “Analise de Pontos de Funcao”, Infobook, Rio de Janeiro, 1996.
- [6] Vazquez, Carlos E., Simoes, Guilherme S. e Albert, Renato M., “Analise de Pontos de Funcao: Medicao, Estimativas e Gerenciamento de Projetos de Software”, Erica, Sao Paulo, 2003.
- [7] Santos, Rogerio S., “Instrucao Normativa No 4”, SLTI, Brasilia, 2008.
- [8] Santos, Maria da Gloria G., “Instrucao Normativa No 04”, SLTI, Brasilia, 2010, <http://www.governoeletronico.gov.br/biblioteca/arquivos/instrucao-normativa-no-04-de-12-de-novembro-de-2010/download>
- [9] Boehm, Barry et al., “Software Cost Estimation with COCOMO II”, Prentice-Hall, New York, 2000.
- [10] Caixa Economica Federal, “Concorrencia No 001/2006, Anexo I: Projeto Basico – Item VIII”, Caixa Economica Federal, Brasilia, 2006.
- [11] MPOG/SLTI, “Roteiro de Metricas de Software do SISP – Versao 2.0”, Ministerio do Planejamento, Orcamento e Gestao, Brasilia, 2012.
- [12] Netherlands Software Metrics Users Association, “Function Point Analysis for Software Enhancement – Guidelines, Version 1.0”, NESMA, 2001.

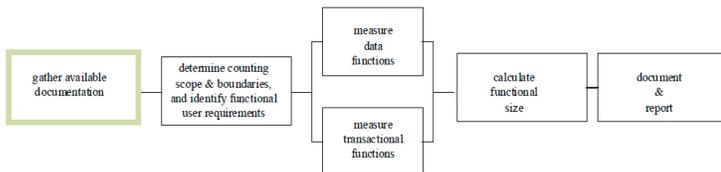
# 10 Common Challenges in the Function Point Measurement Process – and How to Face Them!

By Ricardo Gaspar  
ricardo.gaspar@bndes.gov.br

The benefits of using function point analysis to support management decision-making can only be achieved if organizations address the challenges inherent to the measurement process. Decisions should be made based on data, but what if the numbers are wrong? What if you commit to deadlines based on an unrealistic productivity measured using a wrong number of function points?

Each stage of the measurement process brings issues to be dealt with by organizations who wish to ensure the quality of their measurement data. How can we prevent mistakes in the function point counting process? This article presents 10 challenges in the measurement process with possible ways to handle them based on the Brazilian Development Bank (BNDES) experience on the subject.

## First group of challenges: Is the documentation available?



Accuracy in available documentation will determine the precision of function point measurements. To avoid mistakes, time must be spent to address the following challenges:

### • Challenge #1: Is documentation complete and updated?

Documents with missing or outdated references make it difficult to understand the requirements and can lead to errors in counting function points.

### • Challenge #2: Is documentation in the correct version?

If the function point counting is being conducted at the end of a phase of the project life cycle, the documentation should be on the correct version (related to that moment), even if it had later developments.

### • Challenge #3: What documents are really relevant for counting function points?

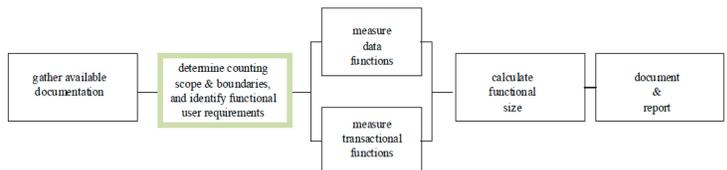
If the documents are scattered among many folders and files, it may be difficult to find the relevant ones for the function point counting, making it difficult to read and understand the requirements, and can lead to errors in the count.

## How to face challenges #1, #2, and #3?

The available documentation gathered must be validated by both the business analyst and a systems analyst responsible for the project. Besides assuring that the documents are complete and updated, they can indicate a list of the documents that are necessary for function point counting and where they are stored (requirements, database model, etc).

It is also highly recommended to use a version control software (such as CVS or SVN) to manage changes to documents and assure that the function point analyst is using the correct version of the documentation.

## Second group of challenges: Are counting boundaries well determined?



(continued on next page)

## CURRENT CONTACT INFORMATION?

To ensure you won't miss out on any IFPUG communications, please log in to your profile on the IFPUG Members Services Area and update your information.

Go to [www.ifpug.org](http://www.ifpug.org)

Send emails to [ifpug@ifpug.org](mailto:ifpug@ifpug.org), call 609-799-4900 or fax 609-799-7032

Write to: IFPUG, 191 Clarksville Road, Princeton Junction, NJ 08550

(10 Common Challenges, continued from page 11)

Positioning of the boundaries influences how logical files will be counted, and it may significantly increase the number of function points.

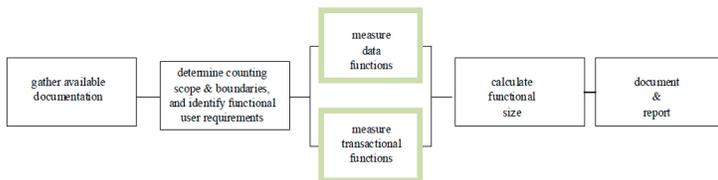
- **Challenge #4: Is there a unique understanding of counting boundary?**

Although CPM [1] states that the application boundary depends on the user view, there can be different opinions among different stakeholders and positioning of the border can become subjective, directly impacting the function point counting.

### How to face challenge #4?

Discussions involving all stakeholders must be made prior to the function point counting and similar criteria should be established for all the projects of the organization. If criteria are changed through time, there will be differences between function point counting in different projects and any kind of comparison would be inaccurate.

### Third group of challenges: Are data and transactional functions being properly measured?



Are there any specific rules for measuring data and transactional functions? Some contracts, for example, state the use of a deflator for data and transactional functions on enhancement projects. If there are specific rules, they must be always applied to allow comparisons between different projects.

Besides, divergences in documentation may result in mistakes when measuring logical files.

- **Challenge #5: On enhancement projects, how to identify the features that have already been counted previously in order to apply a deflator?**

If the function point analyst is not aware of what has already been counted and the project documentation does not explain what was changed, there may be errors on the function point counting.

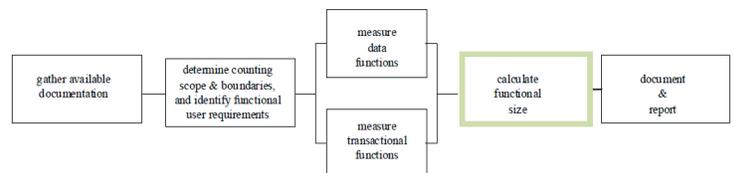
- **Challenge #6: If there is more than one project counting the same logical file in different projects, are the counts compatible?**

It may be necessary to count the same logical file in different projects. However, if the documentation provided is not accurate in one of the projects, it may result in counting inconsistencies. For example, an external logical file may be counted with a different complexity due to documentation problems, even if it has the same data types.

### How to face challenges #5, and #6?

A good practice is to maintain a data function baseline to be accessed and updated at the time of each function point counting. This can be achieved by using a software tool to maintain historical data or by using a spreadsheet that has to be updated in every new function point counting. Comparing spreadsheets is also possible, but it is harder to avoid mistakes unless an automatic tool is used to help.

### Fourth group of challenges: Are there specific rules to calculate functional size?



In some cases, function point based software development contracts involve different rules to calculate functional size.

- **Challenge #7: How to compare two different function point counts?**

Sometimes function point based software development contracts involve different vendors for function point counting and for software development. In those situations, each vendor does a function point count and it is necessary to compare them. Comparison is complex and takes time unless an automatic tool is used to help.

### How to face challenge #7?

There are automated plugins to compare spreadsheets in Microsoft Excel. For example, "Fuzzy Lookup" (<http://www.microsoft.com/en-us/download/details.aspx?id=15011>) can be used to find similarities and differences and make comparison easier in an automatic way.

Other good practice is to create naming rules to perform the count, for example: grouping the functionalities or using alphabetical order. Consequently when comparing spreadsheets, it will be easier to match the functionalities.

(continued on next page)

(10 Common Challenges, continued from page 12)

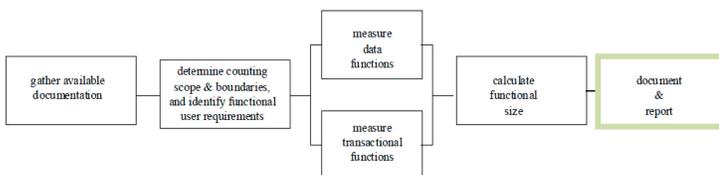
## • Challenge #8: Is there need for specific guidelines for counting function points?

There are new technologies that CPM does not address how to count function points. Some are discussed in IFPUG's white papers such as data warehouse projects. In contractual relations, the establishment of some guidelines may be necessary too.

### How to face challenge #8?

All guidelines should be established prior to the function point count. If guidelines are changed through time, comparisons between projects will not be possible.

## Fifth group of challenges: Is the information being documented and reported properly?



Spreadsheets may be used to document function point counting, but the information must be structured in order to allow extracting the data. Counting tools provide a means of facilitating and linking documentation effectively.

## • Challenge #9: How to structure function point counting data to estimate the effort, cost and productivity of the software development team for use in decision making?

To allow the necessary calculations, the number of function points, the schedule and number of resources of every project must be structured and available.

### How to face challenge #9?

In order to create a function point historical database, every hour spent by professionals involved on the software project must be registered. The organization should evaluate the possibility to buy a software tool to help or develop an internal solution.

## • Challenge #10: If the organization does not have a function point historical database and needs to estimate the effort and cost of a software project, what can be done?

### How to face challenge #10?

If there is no data available, market data can be used (e.g. Capers Jones analysis) [2] to obtain the necessary comparison.

Other option is to measure old projects based on the available documentation and begin the implementation of the function point historical database. Reference to ISBSG data is helpful for most small projects. Alternately, there are a number of estimating tools with historical database information.

The 10 presented challenges in the article represent what BNDES had to deal with on the measurement process. When implementing a measurement process, organizations must identify their own challenges and address them to assure quality decisions based on accurate data.

## References

- [1] Counting Practices Manual, International Function Point Users' Group (IFPUG). 4.3.1.
- [2] Estimating Software Costs, Caper Jones, 9780070659490, 2007. McGraw-Hill Education.



- The leading independent provider of software sizing.
- Certified SNAP Practitioners and Certified Function Point Counters.
- Training, consulting and customized solutions, available in the United States and Europe.
- [www.davidconsultinggroup.com](http://www.davidconsultinggroup.com)

# How Function Points and Snap Work Together

By Charley Tichenor

## Introduction

IFPUG now sponsors two types of software metrics – function points and SNAP. Both are important for software sizing, but what is the difference between function points and SNAP, and why is it important to use both?

It is essential for software development project managers to quantitatively forecast the size of their software to be able to forecast the associated cost and schedule to deliver that software. The software industry has had the “function point” metric for several decades as an available methodology for measuring the size of software. Most software applications are basically forms of data processing systems. Function points measure the functional size of this data processing capacity. As the amount of functionality increases, the function point count also increases, and the cost and time to develop the software also increase in a corresponding fashion. Because function points basically measure data processing capabilities, or the functionality provided by software, function points are often referred to as “functional” software metrics.

A function point can be thought of as one standard unit of software data processing capacity. It is analogous to a “gallon” of gasoline, a “cord” of wood, or a “meter” of length. An application of 1000 function points has twice the data processing capability of an application having 500 function points.

## SNAP Overview

Although some form of data processing is the main purpose of most software applications, and certainly significant work effort is apportioned to data processing functionality, current trends are showing that software customers are requiring software features that do not directly involve the flow and storage of data through the application – at least as defined by the IFPUG Function Point Counting Practices Manual. These other kinds of requirements can therefore be referred to as “non-functional.” Software development projects can contain functional requirements, technical requirements, training requirements, quality requirements, performance requirements, support requirements and maintenance requirements. These project deliverables can be developed at various phases of the project. SNAP offers a methodology to measure and quantify non-functional requirements.

Non-functionality is important to recognize, as it can provide much value to the user and require much work effort to deliver. IFPUG has developed a methodology for sizing software non-functionality, called the “Software Non-functional Assessment Process,” or “SNAP.” SNAP recognizes 4 categories and 14 subcategories of non-functionality, as show in Figure 1, extracted from the SNAP Assessment Practices Manual (APM), Part 1, “Categories and Sub-categories.”

|   |
|---|
| <b>Category 1: Data Operations</b>                        |
| 1.1 Data Entry Validation                                 |
| 1.2 Logical and Mathematical Operations                   |
| 1.3 Data Formatting                                       |
| 1.4 Internal Data Movements                               |
| 1.5 Delivering Added Value to Users by Data Configuration |
| <b>Category 2: Interface Design</b>                       |
| 2.1 User Interfaces                                       |
| 2.2 Help Methods  |
| 2.3 Multiple Input Methods                                |
| 2.4 Multiple Output Methods                               |
| <b>Category 3: Technical Environment</b>                  |
| 3.1 Multiple Platforms                                    |
| 3.2 Database Technology                                   |
| 3.3 Batch Processes                                       |
| <b>Category 4: Architecture</b>                           |
| 4.1 Component Based Software                              |
| 4.2 Multiple Input / Output Interfaces                    |

Figure 1. These are the SNAP categories and sub-categories.

An internationally conducted beta test of the SNAP methodology, version 2.1, was completed in the fall of 2012, with the SNAP methodology achieving statistical significance and an  $r^2$  of .89 when correlating SNAP points per application measured with the work effort in staff hours to build those SNAP points – with the exception of the Help Methods subcategory. The Help Methods subcategory had to be removed from the data because it did not correlate.

## Counting Function Points and SNAP Points

The size of software, therefore, may have both functional and non-functional aspects. Some requirements may be only functional, and measured by function points. Some may be only non-functional, and measured by SNAP. Some requirements may contain both functional and non-functional aspects; in such a case, the requirement will have a functional size, measured in function points, and a SNAP size, measured in SNAP points, and the segregation should be agreed upon by both the user/customer and development teams.

Here is an example of how a hypothetical general requirement can be decomposed into function points and SNAP points.

(continued on next page)

*(How Function Points and Snap Work Together, continued from page 14)*

In a travel application for international flight booking, a “passport type” field is being added to an existing external input. A new code data table is therefore added to store passport type data (regular, diplomatic, etc.). Based on passport type, new data entry validation rules will be programmed to ensure that the passport type data is being entered correctly. In addition, and to improve the look and feel of the screen, it was requested that all the fields should add byte padding, to ensure a 30 character display field length for name field. The passport’s name should be displayed as FirstName.MiddleName.LastName.

To size this enhancement, count function points for adding the new field in the external input. Recall that code data is not countable under function points – and for the purposes of this example, the passport type data will be treated as a code table. Also, neither data entry validation nor data formatting directly involves the flow and storage of data. These aspects of software, therefore, are non-functional. Count SNAP points for the new code data table (Figure 1, subcategory 3.2), additional SNAP points for the new data entry validation (Figure 1, subcategory 1.1), and SNAP points for the data formatting (Figure 1, sub-category 1.3).

(Editors Note: this example does highlight the inconsistencies created by excluding code tables from within an IFPUG function point count – consequent work effort then has to be accommodated within an additional metric).

### SNAP Improves Accuracy of Estimation

In strictly metrics terms, before SNAP, the work effort for this general requirement would only have been based on the function point count of the enhanced external input. The work effort (for the development life cycle) for the code table, the data entry validation, and data formatting would theoretically have been unaccounted for since they were non-functional. Now with the SNAP metric, the life cycle development effort for the code table, data entry validation, and data formatting are accounted for and better estimates for the cost and schedule to build this requirement are possible. Also, the developers may have more confidence in the metrics program to fairly represent their work efforts. Early information shows that SNAP typically accounts for about 15% of the work effort.

Here is an example similar to my experience with auditing projects. This hypothetical project team requests additional funding for a planned enhancement, and senior management wants to know if this team has spent its budget wisely. We performed a function point and SNAP analysis of that project and learned the following information.

|   |           |
|---|-----------|
| Functional size (# function points)         | 971       |
| Non-functional size (#SNAP points)          | 632       |
|   |           |
| Benchmark cost per function point           | \$825     |
| Avg. fully burdened labor rate              | \$100     |
| Non-functional rate = 1.27 hours/SNAP point | \$127     |
| Total project actual cost                   | \$901,283 |

Figure 2. This is the hypothetical audit data collected.

Our team counted the function points and SNAP points, realized the comparable benchmark cost per function point, and our organization’s average SNAP delivery rate was 1.27 hours per SNAP point. The project team’s average fully burdened labor rate was \$100 per hour, so their SNAP delivery cost was \$127 per SNAP point. See Figure 2.

Here is how the audit was completed. See Figure 3. The benchmark cost for the functionality was  $\$825 * 971$  function points, or \$801,075. The expected non-functional cost was  $\$127 * 632$  SNAP points, or \$80,264. Adding these together, the total expected cost of the application was \$881,339. Their actual cost to deliver was \$901,283. These two costs are comparable – within about 2% ( $[\$901,283 - \$881,339] / \$901,283$ ) -- so it appears that the project team managed its money resources wisely. Also notice that the gap between expected costs and actual cost was about 11% ( $[\$901,283 - \$801,075] / \$901,283$ ) before using SNAP, and about 2% while using SNAP – which means that the variance between expected cost and actual cost was a result of spending effort on non-functional aspects, and not poor productivity of the development team.

| Breakout            | Functional | Non-functional | Total     |
|---------------------|------------|----------------|-----------|
| Benchmark           | \$801,075  |                |           |
| Expected            |            | \$80,264       |           |
| Total expected cost |            |                | \$881,339 |
| Actual cost         |            |                | \$901,283 |

Figure 3. This is the hypothetical audit results.

### Further Reading

The size value of software, then, is a combination of both its functional part and non-functional parts, such as 500 function points and 250 SNAP points. For those who are mathematically inclined, it is much like the complex numbers where the value of a number has two parts – its real part and imaginary part -- such as  $500 + 250i$ . Function points and SNAP points are not to be somehow combined such as 750 points of some kind because they measure two different kinds of things.

*(continued on next page)*

*(How Function Points and Snap Work Together, continued from page 15)*

IFPUG offers two case studies which further detail how SNAP can be used. The first SNAP case study, “Assessing the Size of Extensive Mathematical Operations Using SNAP” shows how algorithms are sized using SNAP. This helps to solve the problem of sizing software that is algorithmic-intensive such as large computer simulations and software requiring extensive real time calculations. The second SNAP case study, “How to Use Function Points and SNAP to Improve a Software Acquisitions Contract,” shows how to use a combination of both function points and SNAP to leverage economic incentives to reduce the costs of acquiring software.

The SNAP methodology is overseen by the IFPUG Non-functional Sizing Standards Committee (NFSSC). This

committee is very active, and has provided the community with such products as the recently approved version 2.3 of the SNAP APM, a SNAP certification program, and Train-the-Trainer materials.

Being relatively new there are still some gray areas in the SNAP methodology and the NFSSC is working to clarify them. Also, research is ongoing to improve the Help Methods subcategory.

Add SNAP to your portfolio of software metrics. Your cost and schedule forecasts will improve, your software project audits will improve, and your developers will greatly appreciate that more of their work will be formally recognized.

## SNAP: Going Beyond Sizing Non Functional Requirements

*By Amit Javadekar*

*Amit Javadekar leads the Estimation Centre of Excellence at Infosys Limited*

*Amit\_Javadekar@infosys.com*

The Function Points model is the most popular method for sizing functional requirements. Over the past 35 years that the model has been in use various perceived limitations of the model have been highlighted by industry experts. Even within the scope of the FP model (i.e. sizing functional requirements) there are aspects like usage of heavy calculations and complex algorithms, data movements within the application, enabling/disabling functionality through configuration changes to an application etc. which are not separately considered by the model. Greater adoption of the FP model has been hampered over the years due to the excessive focus on these perceived limitations by those not willing to appreciate the usefulness of the model.

While the SNAP model has been designed to size the non-functional requirements (NFR) it also has the capability to accommodate some of the aspects of the FP model that have been highlighted in the past. This article shows how SNAP can be used to address some of these aspects of FP.

### Limitations of the Function Points Model

While many perceived limitations of the FP model have been raised by industry practitioners, this article will focus on the ones that are aired most often, in the author’s experience, during any discussion on FP. Some of these aspects are:

1. Complex algorithms and heavy calculations that are part of a transaction’s processing logic are not separately considered as part of the functional sizing
2. Functionality enabled/disabled through ‘application configuration’ type of work does not fetch additional size
3. FP only considers interactions between the (external) user and the application. Interactions between various internal parts of the application are not considered by the FP model
4. Repositioning of User Interface (UI) elements without adding/deleting/modifying any of them is not included in the sizing process

5. If the same output is created in multiple formats or methods (e.g. MS-Excel and PDF), no additional size is calculated for the multiple formats (i.e., only one format is included for the size calculation)

### Using SNAP to overcome these limitations

Most of the situations mentioned above are not true NFRs. Rather they form part of the user’s functional requirements. But since the FP model does not directly address such requirements there was no standard way to size them. However, with the advent of SNAP it is now possible to size such requirements using the SNAP framework as described below:

#### 1. Heavy calculations and complex algorithms as part of a transaction’s processing logic

Using the Logical and Mathematical Operations sub-category under Data Operations category, it is now possible

*(continued on next page)*

(SNAP: Going Beyond Sizing, continued from page 16)

to get a SNAP size for a transaction in addition to its FP size. This sub-category helps size processing logic having “extensive mathematical and logical operations”. Since most heavy calculations and complex algorithms are nothing but a series of extensive mathematical and logical operations this sub-category helps size such processing logic using SNAP Points and adds to the transaction’s size.

## 2. Functionality enabled/disabled through configuration settings included in the application

In many applications (and certainly in commercial off the shelf products) new sets of data can be processed by a transaction by making changes to some configuration settings in the application. This new data is processed by the application without making any changes to the transaction’s processing logic (i.e. the code). E.g. in an order management system, currently only products of type ‘F’ (Finished Goods) can be ordered by the customer. If there is a new requirement that allows products of type ‘S’ (Semi-finished Goods) to be ordered by the customer, then it is possible to do so by configuring (adding) the Valid Products File to include product type ‘S’ in addition to existing value of ‘F’. This will allow Semi-finished Goods to be ordered without making any changes to the transaction’s processing logic. Using FP this functionality would have been sized only once, but with SNAP sub-category Delivering Added Value to Users by Data Configuration under the Data Operations category it is possible to get additional SNAP size for the new value that has been configured.

## 3. Interactions between internal parts of the application

The FP model only considers data movements (through transactions) that cross the application boundary. However there are many instances where data movements happen internally within an

application but these are not considered by FP, for example:

- a) Caching frequently used data records at the client end in a client-server architecture by copying them periodically from the server end. This allows for faster processing of these records.
- b) Archiving data within the application at set intervals.
- c) Data movements from the Middleware layer to the User Interface or Database layer.

Such data movements can now be sized in SNAP using the Internal Data Movements sub-category under the Data Operations category and included as part of the application’s size

## 4. Repositioning of UI elements

Changes in business requirements at times necessitate changing the screen or report layout without actually adding/deleting any fields in the UI, for example:

- a) In a List of Active Projects report, a High Risk project should be highlighted in Red colour instead of Black.
- b) Rearranging the Order Entry screen layout such that general information about the order is displayed at the top and all product related information is displayed at the bottom of the screen. No new fields are added or existing ones deleted.

c) The field where the cursor is currently positioned should be highlighted with a Green border.

Situations like the ones described above are difficult to size in FP since it considers only those transactions as impacted if they have any functional changes i.e. some UI elements (DETs) have been added/deleted or FTRs have been added/deleted. SNAP provides the solution to sizing the above described requirements through the User Interfaces sub-category under the Interface Design category.

## 5. Same functionality in multiple output formats/methods

This is arguably the one perceived limitation/criticism of FP that is talked about the most. In almost all applications there are situations where the same output (EO/EQ) is displayed in multiple formats like MS-Excel or PDF files. Since it is the same functionality, FP would size the transaction only once, either as an EO or as an EQ. However, using SNAP, additional size would be obtained for the one (or more) additional formats in which the output is displayed. This is possible in SNAP by using the Multiple Output Methods sub-category under the Interface Design category.

## Summary

Given below is a table showing the appropriate categories and sub-categories of SNAP that address some of the perceived limitations of FP.

| #  | FP Limitation   | SNAP Category    | SNAP Sub-Category                                     |
|----|---|------------------|---|
| 1. | Heavy calculations and complex algorithms as part of a transaction's processing logic     | Data Operations  | Logical and Mathematical Operations                   |
| 2. | Functionality enabled/disabled through configuration settings included in the application | Data Operations  | Delivering Added Value to Users by Data Configuration |
| 3. | Interactions between internal parts of the application                                    | Data Operations  | Internal Data Movements                               |
| 4. | Repositioning of UI elements  | Interface Design | User Interfaces                                       |
| 5. | Same functionality in multiple output formats   | Interface Design | Multiple Output Methods                               |

(continued on next page)

(SNAP: Going Beyond Sizing, continued from page 17)

## Conclusion

SNAP provides an excellent mechanism to size Non Functional Requirements. It helps overcome one of the major criticisms of the FP model that it focuses only on the Functional view of the application and does not adequately consider the Non Functional view. However, even within the func-

tional view of the application, FP has been criticized over the years for intentionally not taking into consideration some commonly occurring situations as described above. The design of the SNAP model is such that these criticisms can be addressed by using SNAP in conjunction with FP not just to size NFRs but also to size FURs

that are not considered in the FP model. This should help address the concerns of many FP critics who hesitate in using FP by pointing to its long standing perceived limitations.

## References

SNAP Assessment Practices Manual Release 2.2

---

## Using Project History to Produce Effective Estimates

By Dan Horvath

We can thank Albert Einstein for the popular definition of insanity: *doing the same thing over and over again and expecting different results*. If that same thing includes producing inaccurate estimates resulting in projects that do not meet their time, effort or scope objectives, better results can indeed be obtained by doing something different: utilizing project history. Edmund Burke tells us: *Those who don't know history are destined to repeat it*. Learning from, and making use of project history can surely help us to achieve better estimation as well as improved project execution results. These results can be achieved with little effort and in a relatively brief period of time.

The objectives for at least some Enterprise Project Management Offices (EPMOs) include goals such as: *Improve the predictability, agility, and speed of project delivery*. To an EPMO's Estimation Center of Excellence (ECO), this means improving the accuracy of estimates, particularly those made early in the project life cycle. To accomplish this, a project benchmarking process should be initiated in order to collect pertinent data into a project history database as projects close.

While it's possible to produce estimates based on industry data, as specific historical data becomes available, company specific information can be used. For the purposes of this article, the QSM Slim Suite of tools is used for historical data collection and analysis, as well as for project estimation. But other methods and tools may be used as well. The article will outline some findings and challenges regarding the process for collecting and using company specific project history data to produce better estimates, and thereby improving predictability, agility, and speed of project delivery.

### Equal Opportunity Benchmarking: Collection of Data for Projects of all Shapes, Sizes

We could simply call it historical data collection, and gathering and storing the information is indeed the main goal. But the word benchmarking implies something more: according to the Oxford Dictionaries, to *Evaluate or check (something) by comparison with a standard*. Regardless of the precise definition, we must gather and store, but also compare the project data being evaluated with other project data already in the historical database. Reports and charts are produced, and these results should be shared with the project manager (PM).

Although project data is automatically collected in an enterprise project management application such as Microsoft Project, benchmarking requires that certain additional data is needed as well. Most notable of these data elements is project size. A benchmarking process is put in place to ensure that all data is collected, and to store it all in a usable format.

Besides the objective to help make estimates more accurate, there are additional reasons to collect project history data. The other reasons include aiding the organization in performing analysis regarding how projects perform based on size, team, technology and business area.

### The Beginning of History

It's easy to become discouraged if there is a lack of company specific project historical data. Bear in mind that *you have to start somewhere*, and the sooner the better. Once the process has begun, the data can be put to use rather quickly. Employing knowledgeable and experienced consultants can help get the ball rolling.

The project history data may be stored in a spreadsheet, a custom developed application and database, or a sophisticated commercial tool.

(continued on next page)

(Using Project History, continued from page 18)

## Types of Projects

It certainly helps if the Information Technology organization is highly project oriented. At these types of organizations, virtually all work above a certain threshold (say, 500 hours) is considered a project, and all projects are managed through a central point.

Most projects fall into one of the following general categories: application development, infrastructure, package implementation, R & D and process development. Of course there are hybrids of these, and also a few that don't seem to fit any of the categories. There are other ways of classifying projects as well, such as by technology type, business function, sponsoring platform, application and work type.

It is challenging to estimate, as well as to plan and manage so many different types of projects. In order to improve the predictability, agility, and speed of project delivery, an ECOE (Estimation Center of Excellence) is not only charged with performing the estimation, but also to continuously improve the quality of the estimates. To assist in this endeavor, the team should try to benchmark as many as possible of each project type. As each project closes, the PM creates a request for project benchmarking services in order to initiate the process. The majority of projects ought to be benchmarked.

The ECOE should aim to conduct the project benchmarking as soon as possible after the project closes in order to catch team members before their reassignments are too far along, and also to reach team members before the memory of the project begins to fade. On the other hand, the benchmarking cannot be performed before the project is closed, since effort, duration and even scope must be complete.

## Type of Data to Collect

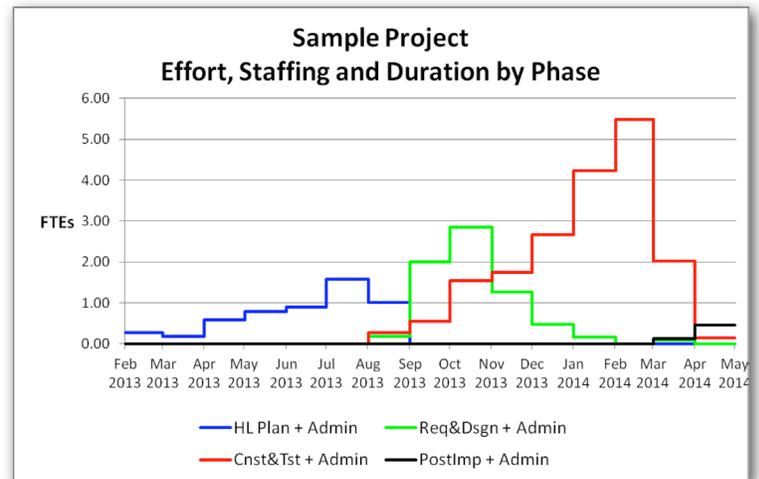
In *Software Estimation, Demystifying the Black Art*, Steve McConnell recommends beginning historical data collection with a minimum of four data points:

- Size (lines of code, function points, etc.)
- Effort, in hours or months
- Time, in calendar months
- Defects by severity

The ECOE – or measurement group involved - may also decide to collect several additional metrics, including:

- Staffing / Full Time Equivalents (FTEs) by month
- Effort and Time by Software Development Methodology phase
- Number of Requirements and/or Story Points
- Qualitative Project Assessment Factors (scores of scope control, resource availability, dependencies, etc.)

Classification, such as the type of project, and other project identification factors need to be recorded as well. A productivity index, a value based on the size, effort and duration, can also be calculated. An example of the effort and duration data extracted from the project management system follows:



This chart shows some of the data collected during a typical benchmarking session. Effort, staffing (in terms of FTEs) and duration, by month and phase, are displayed as extracted from the project management system.

## What's in it for me?

It would be understandable if some of the project managers (PM's) were to ask why, after closing the project, they then need to go through the benchmark process; it does involve some additional work for them. They should be informed that the information is used as a feedback loop in order to improve estimation.

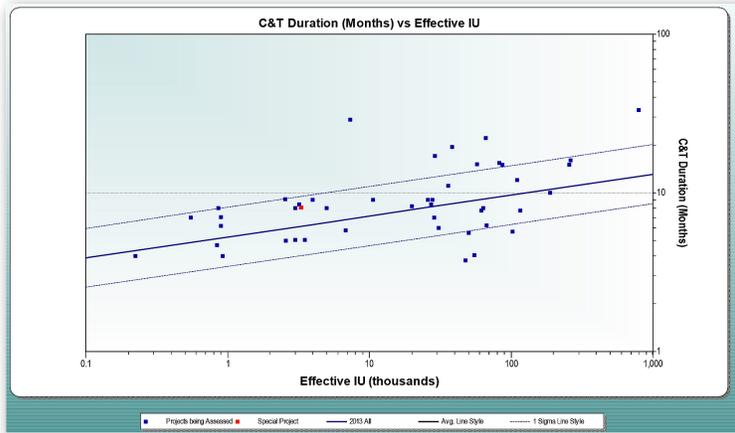
But the PMs are also provided useful information as a result of the benchmarking session. After the effort and duration data is validated; after the quality and project assessment factors are reviewed and recorded; after the sizing is determined, the data can be compared to other company projects. If there are similar types of projects, they can be contrasted in further detail.

Reports and charts can then be produced in order to present the benchmarking data graphically. Some examples follow. For each chart, the size is reflected as Implementation Units (IUs) on the horizontal axis, and the metric it's measured against (duration, effort, etc.) is on the vertical axis. Other benchmarked projects in the project history database are represented as dots, and the current project is highlighted in red. The trend line shows the average and one standard deviation above and below that value.

(continued on next page)

(Using Project History, continued from page 19)

The example below demonstrates the relationship between size and duration. Similar illustrations can be performed for other relationships of interest, such as productivity or effort cost.



The schedule trend shows construction and test duration versus size. In this example, the current project took longer than expected for projects of its size.

## Sizing: the Elephant in the Room

Vince Lombardi, the legendary coach of the Green Bay Packers is famous for saying, *Winning isn't everything, it's the only thing.* Let's try to modify that for sizing: *Sizing isn't everything, it's the only thing.* Maybe.

In regards to project estimation, size is the major driver of effort and duration. And of course effort and duration are the major things that anyone requesting an estimate wants to know. Other factors that can play a role in estimation include quality, risk, cost and staffing, but based on project history, these can also be derived from effort, duration and especially size. Therefore, accurate sizing is of critical importance during the project history benchmarking process. It is the most important metric in the project history database. The reasons are twofold: as noted, size is used as the basis for most secondary metric calculations, and size has the greatest impact or weight for these calculations. The most critical success factor for the entire endeavor has been identified as implementation and use of a consistent sizing methodology.

The following methods of sizing have been identified and used (by this author: there are several other sizing methods currently popular and many other measures that have been applied and tried and not found universally useful):

- Function Points and SNAP Points
- Source Lines of Code (SLOC)
- Story Points
- Story Cards
- Components (list of items created or changed)

- Requirements
- Complexity Points (a generally home-grown measure of complexity and size)
- A range of other measures applied and tried and not found universally useful

Each method has its advantages and disadvantages. When practical, multiple methods have been used when measuring the same projects in order to validate one versus another. One goal has been to be able to use the same methodology for both benchmarking and for estimating. Since the estimates are done early in the life cycle, this makes some of the methods impractical.

Function Point Analysis (FPA) is an International Organization for Standardization (ISO) standard sizing methodology, and Function Points have been selected by the Information Systems Audit and Control Association (ISACA) as the number one sizing metric. FPA can be combined with Software Non-functional Assessment Process (SNAP) to provide the complete sizing for most software development projects. When sizing projects that involve infrastructure (hardware and/or software), software upgrades or process development, however, additional or alternative sizing methods need to be considered.

After some measure of analysis, examination and experimentation, some organizations may decide to employ an 'all-of-the-above' approach to project sizing. This is mostly due to the wide variety of different types of projects as noted above. Some of this method's advantages are:

- 1) It is usable at any point in the project life cycle, including initiation
- 2) It is as simple as possible for PMs, developers and project planners to understand and use
- 3) It is robust enough to encompass all types of projects

Whichever method or component is used, it is helpful to have the size stored in consistent units. One such unit is Implementation Units (IUs). IUs are considered the smallest unit of work produced by a system. An IU is equivalent to a line of code. All components, function points, story points, requirements, etc. are converted to IUs using an appropriate gearing factor.

Depending on the sizing methodology, care must be taken to ensure consistency and to avoid double-counting. An ECOE office or similar should be charged with scrutiny and inspection of the application. Regarding the development and use of the components, much is driven by the categories of projects. Note that application development projects amount to less than 50% of the total. Package implementations and infrastructure projects may make up a large percentage.

(continued on next page)

(Using Project History, continued from page 20)

## Project Initiation and Estimation Process: Making Use of the Data

When and How to Estimate

*Vote early and vote often* is a phrase often associated with Chicagoans, such as Mayors Richard J. Daley, William Hale Thompson and, believe it or not, Gangster Al Capone. For the ECOE, the phrase is modified slightly, to *Estimate early and estimate often*. In fact, estimation is often performed at three points in the project lifecycle:

- Stage Gate 1 / rough order or magnitude (ROM) – A high-level, top-down estimate used to determine feasibility for approval for high level planning (HLP). It is typically created by the planning and estimation or ECOE team.
- Stage Gate 2 / Budgetary – A detailed estimate at the conclusion of the HLP phase used for approval for project execution. It is created by the project team.
- Stage Gate 3 / Definitive – A detailed estimate produced when the project execution is 40% complete. It is also created by the project team.

In terms of the expected level of accuracy for the three types of estimates, please refer to the diagram below. In this example, the Cone of Uncertainty shows that the ROM estimate is expected to be between 100% above and 25% below the final cost. The Budgetary estimate should be from up to 35% above to 10% below the final cost. The Budgetary estimate is expected to range from 20% above to 10% below the final cost.

The ROM estimate is top-down, and can be performed using the Slim-Estimate tool in conjunction with the project history data stored in the SLIM-DataManager file. The Budgetary and Definitive estimates are detailed at the task level and are therefore a little more bottom-up. The results for these estimates should be validated against project history as much as possible. Since the focus of this article is using project history to produce estimates, and since that history is mostly used when developing the ROM, the ROM estimate will be the emphasis of the remainder of this section.

The Cone of Uncertainty used for many ECOEs. The percentages at each stage as well as the project hours for the ROM are shown as examples.

## Analogous and Parametric Estimation

The Stage Gate 1 or rough order of magnitude (ROM) estimate is high-level and top-down for a very good reason: not enough information is available for a detailed bottom-up estimate. The ROM is used to determine whether the project should proceed to the HLP (High Level Planning) phase. Depending on the organization, some ROMs do require a relatively high expectation of accuracy. In fact, for some projects, a *very* rough order of magnitude (VROM) estimate may be created even before the ROM. In some instances, the title, *rough order of magnitude* might indeed be somewhat a misnomer.

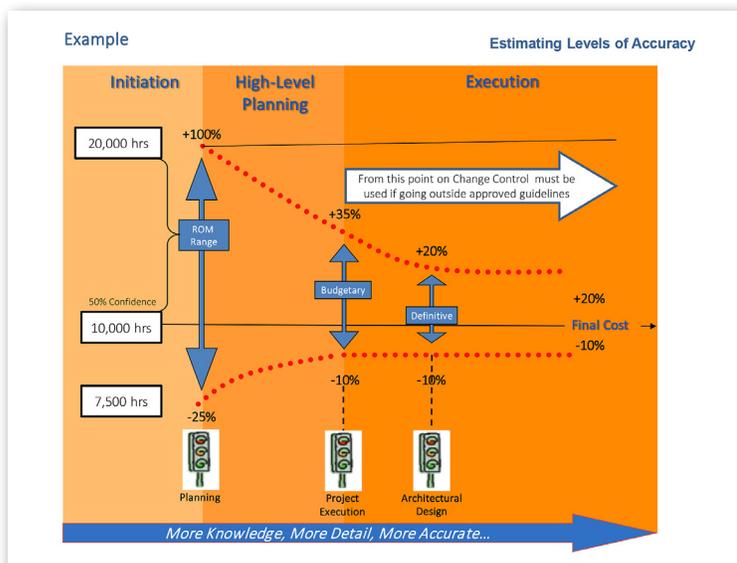
Although the amount of known information about the project is still low at initiation, an estimate produced during this period can nonetheless be quite useful and possibly necessary. In addition to aiding steering teams in their decision about whether to move forward, it also helps the newly assigned project manager determine areas of impact as HLP begins. Finally, it helps in the long range planning of resource requirements.

The project history data is applied when creating the ROM estimate. For Budgetary and Definitive estimates, the focus is on the detailed tasks to be included. At this point in the project, the team should be able to reconcile and describe any discrepancies between the estimates and the actual data from previous projects.

The simplest and, surprisingly, also one of the most useful methods of project estimation is analogous estimation. The project sponsor and others are asked to identify any similar projects that they are aware of. If the information about the previous projects is in the project history database, it can be immediately put to use for an analogous estimate.

Otherwise, if similar project(s) have not been previously benchmarked, the actual data can be extracted from the project management application as well as the quality statistics application. This process provides effort, staffing, duration and defect metrics, but not size. Even this somewhat incomplete picture can still provide valuable input to the new estimate.

No two estimates are identical because, by definition, no two projects are identical. This is why the word, analogous is used to describe these types of estimates. According to Cambridge Dictionaries, analogous means *a comparison between things that have similar features, often used to help explain a principle or idea*. The key word is similar. Comparisons such as these are often heard:



(continued on next page)

(Using Project History, continued from page 21)

- *It's similar to project A, but this time we won't be doing feature m or feature n.*
- *This one will be like project B, but it will require twice the construction and testing effort.*
- *We believe the new project will require more effort than project C, but less than project D.*

The ECOE should listen carefully to these types of comments when performing an analogous estimate. Although most are quite valid, some are not. Some skepticism is warranted when a comment such as this is heard: *it will be similar to project F, but we have improved our efficiency and therefore anticipate that our effort and duration will be less than half.*

If similar projects are not identified, project history data is applied towards a parametric estimate. This term is straightforward enough: Of, *relating to, or expressed in terms of a parameter or parameters*, according to Oxford Dictionaries. For estimation, the parameters being applied are those of previous projects.

Most estimating tools enable users to apply either industry data or organization specific project history data collected using the benchmarking process. Once the organization has collected a sizable set of project history, the industry data may no longer be needed. In addition, it's possible to apply subsets of the data. When enough data is available for the type of project being estimated, that subset is used. By slicing the data closely enough, the parametric estimate becomes closer to an analogous one.

Project oriented organizations will have a process to initiate and perform the ROM estimation for potential projects. The primary intent is to create a ROM estimate and to prepare other documentation for Stage Gate 1 and high level planning.

## How Well Does it Work?

Outliers, Damned Outliers and Statistics

Mark Twain's famous quote, *There are lies, damn lies and statistics* becomes more relevant in this context when we replace lies with outliers. As we examine the project history data, we may notice that some projects do appear to be outliers. Statisticians inform us that a project more than three standard deviations away from the mean may be considered an outlier. But other questions, such as examination of *which* metrics appear incorrect, must be considered. The best solution to the question of outliers appears to be to leave them in the data, but to only use those data points when they appear to be of a similar project type to that being estimated. In other words, filter out projects that don't apply, including possible outliers.

Other questions about the history data also arise. How many of each type of project are enough? McConnell suggests about twenty, but it's always best to err on the side of more, rather than less data.

Once enough data has been collected, it will be possible to perform some analysis on the information. Do projects of one work type perform better than another? Are the projects of one year more efficient than those of the previous year? Do projects become more efficient as their size increases?

An organization should also measure the accuracy of its estimates. The ECOE ought to be able to observe that the trend is positive, with more and more project estimates falling within the proper range in the Cone of Uncertainty.

## Challenges

Consistency in sizing often continues to be a challenge. Part of the problem may be resistance to change. The organization that has been simply estimating hours and duration may have some difficulty in the transition to estimation based on size; getting everyone to think in terms of what will be produced (in terms of products) instead of how much effort it will take to produce it will often be difficult. The culture change will eventually take place, but it will take some time.

Bundling is another challenge. Some projects combine two or more initiatives together for more efficient management. For example, two different Microsoft client based products may need to be upgraded, and the two upgrades are managed together as one project. When it comes time to estimate a subsequent upgrade on one of those products by itself, the information within the previous project is not available because of the bundling; it's now nearly impossible to separate out the metrics of one upgrade versus that of the other. One answer is to build project plans more strictly by product so that it becomes clear how much effort, time, etc. went into each one.

Yet another challenge is to use defect tracking and prediction effectively. To do this, there is a need to use consistent methods to gather the defects pre and post release. But the most important challenge is to overcome resistance to change in general. The EPMO may have been managing (including initiating, working and closing) projects in certain ways for several years. To begin using sizing for those unfamiliar with the concept, and to introduce other changes often proves somewhat difficult. But, for more predictable delivery, it's worth it.

# Component Sizing

By Chuck Wesolowski, Chief Architect, Vencore

## Introduction

Component based development is used extensively in Service Oriented Architectures (SOA), where components form the fundamental modeling elements of software design. These are deployed in a net-centric environment, using a variety of technologies to realize the coupling and to enable communication.

Components exhibit a characteristic called coupling that is related to the number of connection points with other components, expressed in terms of their interfaces.

Component specification is the process that produces the Architectural Design of the system from its functional requirements. The component specification indicates the key software configuration items that must be managed and constructed during the software development process, without specifying the actual implementation details.

This article explains the key elements of component specification, and defines the measurements necessary to determine software size and complexity from both the functional and technical perspectives.

## Components

“A component represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment. A component defines its behavior in terms of provided and required interfaces. As such, a component serves as a type whose conformance is defined by these provided and required interfaces.”

Components are often indicated graphically using the Unified Modeling Language (UML). The key elements are the components themselves, and their interfaces.

### 1.1 Component Notation and Specification

The simplest component is one that provides its services using one interface. For example the following illustration depicts a four function calculator service component using a UML class diagram that clearly distinguishes between the implementation of the service, indicated by the Calculator component, and the interface to the component that specifies the services provided.

There are two ways of indicating interface realization using UML class notation. These are depicted in the following diagram.

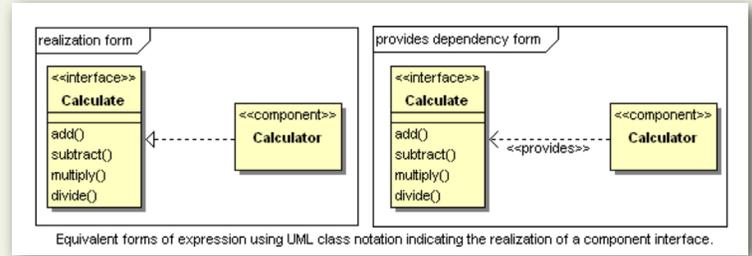


Figure 1-- UML Class Notation

The left hand side of the preceding diagram uses the realization relationship, which is valid for any class type. The right hand side uses provides dependency notation, which is specifically used when the realizing class is a component. In either case, the component implements the functionality specified by the interface. The Calculator is the thing that adds, subtracts, multiplies, and divides. The calculator’s functionality is accessed via the Calculate interface

An alternate way of depicting the Calculator uses UML component notation. The so-called black box view involves indicating provided interfaces using a “lollipop” that specifies the interface name, without listing its operations. This form is commonly used when depicting the manner in which a system’s components are architecturally related, or coupled to each other.

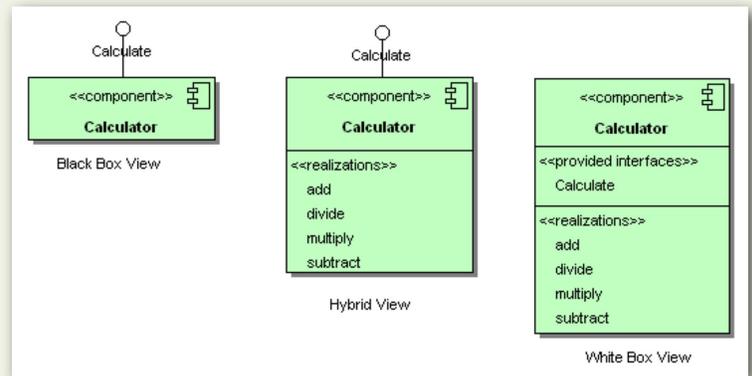


Figure 2-- UML Component Notation

(continued on next page)

(Component Sizing, continued from page 23)

Lastly, a component may simply be specified textually; for example:

```
Interface Calculate
Operations: add, subtract, multiply, divide
Component Calculator
Provides: Calculate
```

## 1.2 Functional and Technical Requirements

A component specification illustrates the difference between functional and technical requirements. The interface specifies the functional requirements of the software component. “An interface specifies a contract; any instance of a classifier that realizes the interface must fulfill that contract.”

A realization represents an allocation of functional requirements to a particular system component, and thus traces the system’s functional requirements to its architectural design. Components and their interfaces represent the system’s technical software requirements; hence the process of component specification constitutes the first phase of transition from analysis to implementation.

## 1.3 Coupling Factor

The Coupling Factor (CF) of a component is measured by the number of its required and provided interfaces, and expressed as their sum, as indicated in the following diagram.

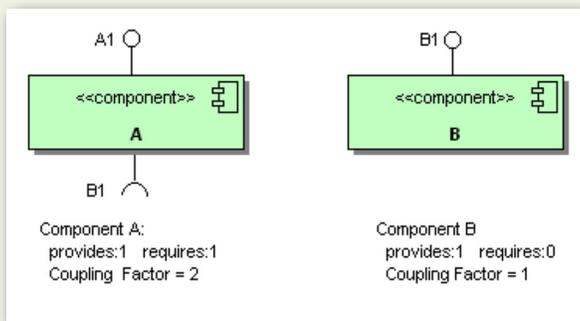


Figure 3 – Component Coupling Factor

The Coupling Factor is a kind of technical size measurement for one component. The number of interfaces provided and required is an indication of technical complexity. Note that while provided interfaces are indicated by a “lollipop,” required interfaces are indicated by a “socket.”

## 1.3.1 Coupling Points

Components are coupled when one component requires the service of another. A coupling point connects two components via their required and provided interfaces. As previously mentioned, UML uses a “ball and socket” notation in a component diagram to indicate coupling between components. The complexity of the coupling point is the number of operations provided by the interface.

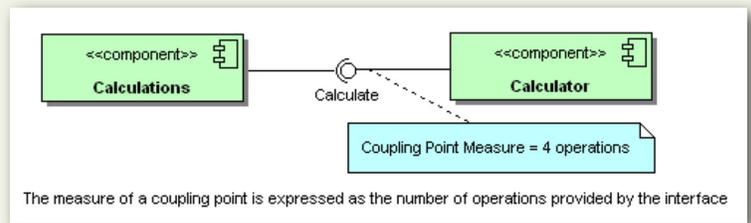


Figure 4 – Coupling Point Metric

The coupling point metric represents a coarse measure of the functionality offered by the component on a particular interface.

## 1.3.2 Dependency Cycles

Components can be characterized as independent, dependent, or co-dependent based on an examination of a dependency graph of the component architecture.

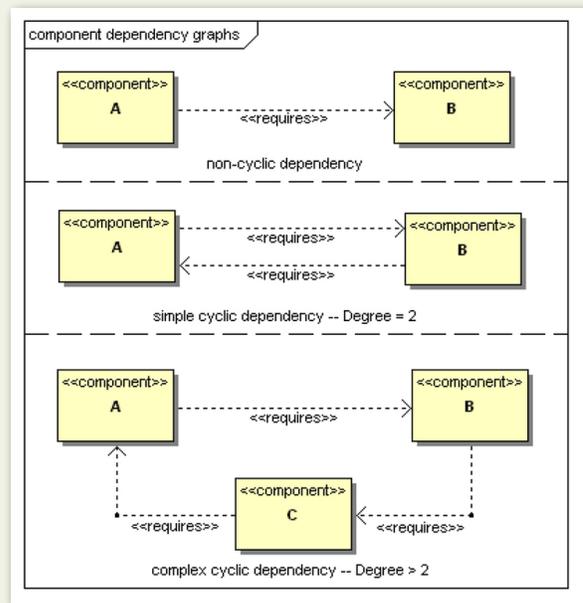


Figure 5 – Component Dependency Cycles

(continued on next page)

(Component Sizing, continued from page 24)

The preceding diagram illustrates the differences between cyclic and non-cyclic dependencies. The top part indicates a non-cyclic dependency, in this case A is dependent, and B is independent. This is the same pattern illustrated by the Calculator example from Figure 4.

We note that the bottom two examples illustrate cyclic dependencies. Components involved in a cycle are co-dependent.

A cyclic dependency is measured by the number of components involved, called the Degree of the Cycle. The middle part of the diagram shows a simple cycle, which is defined as involving exactly two components. A complex cyclic dependency has a degree greater than 2, and is illustrated in the bottom part of the diagram. The number of cycles and the degree of each cycle reflects architectural complexity.

### 1.3.3 Types of Component Coupling

Components may be loosely or tightly coupled. This characteristic is observable as a cycle in the dependency graph of any two coupled components.

The preceding class diagram illustrates the principle that tightly coupled components are co-dependent, which means that neither can exist independently within the system architecture. The following diagram illustrates the same concept using a component diagram, and demonstrates that the type of coupling between components is an independent measure of technical complexity.

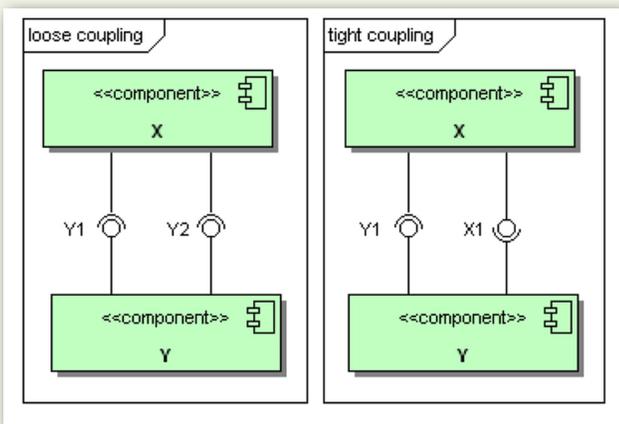


Figure 6 – Loose vs. Tight Coupling

The coupling type is different despite the fact that the coupling metric for components X and Y is identical; both X and Y have a Coupling Factor of 2. Tight coupling is an indication of higher technical complexity than loose coupling, and the number of tightly coupled components as a proportion to the total number of components enables a quantitative evaluation of the “flexibility” of an architecture.

The tight coupling between X and Y on the right hand side of the preceding illustration is an example of a simple dependency cycle because it only involves two components. The following illustration demonstrates the existence of a complex dependency cycle. Note that there is loose coupling between any two components in the architecture, yet none of these components can exist independently.

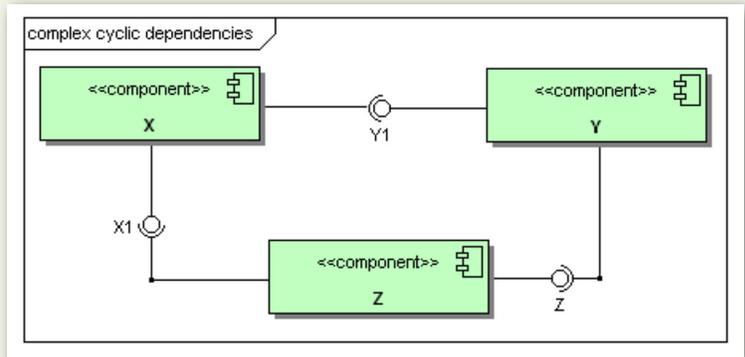


Figure 7-- Complex Dependency Cycle

We can see that the number of components involved in a dependency cycle affects technical complexity by reducing the degree of component independence within the architecture. It is important to capture the number of dependency cycles in component model, as well as the number of components involved in a cycle. The more components involved in a cycle reflects the higher the technical complexity and risk associated with the architecture.

### Distribution of Functionality

Returning to the example of the Calculator component, we see that all four functions are specified by the Calculate interface which is realized by the component.

(continued on next page)

(Component Sizing, continued from page 25)

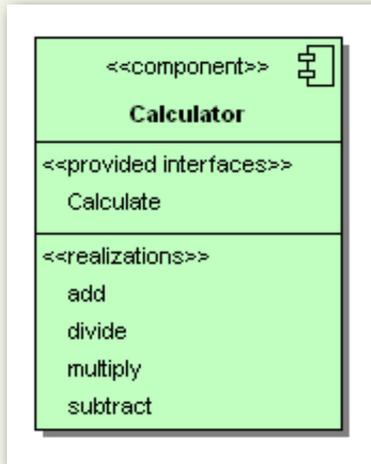
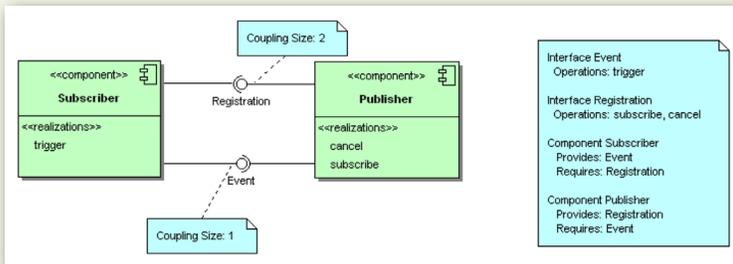


Figure 8-- Component View of Allocated Functionality

We take note that the maximum number of interfaces provided by a component is equal to number of operations realized by the component. Thus we see that the technical size is independent from the functional size.

The following diagram illustrates this principle using UML component notation, where the functionality allocated to the component is listed, as well as the interfaces that it provides.



The following diagram illustrates coupling using SOA lingo. The Choreography Layer is also called the Business Process Composition Layer, or the Orchestration Layer, depending on the author.

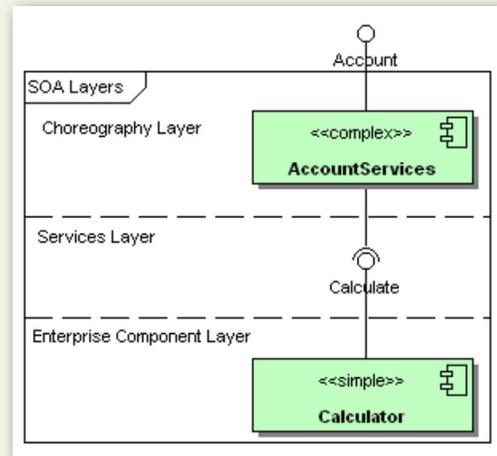


Figure 9 – Component Coupling in a SOA

## 1.4 Component Types

We use the UML stereotypes `<<simple>>` and `<<complex>>` to distinguish component types. A simple component has no required interfaces, and is therefore autonomous from a component based architecture perspective. A component that requires one or more interfaces is called a complex component. This is formally expressed as follows using Object Constraint Language (OCL).

Component Types:

Simple inv:  $\text{count}(\text{required interfaces}) = 0$

Complex inv:  $\text{count}(\text{required interfaces}) > 0$

This constitutes a rule for classifying components, stated as an invariant. An invariant is an assertion about the system that is always true. For example, a simple component will invariably have a required interface count of 0.

## 1.5 Component State

State information is data maintained by the component. The component stores this information in some form of memory. The information persists after execution of an interface operation completes, and it is often called persistent data.

**Your Authoritative Guides to Software Measurement**  
*The IFPUG Way*

**SAVE 25%** Plus **FREE SHIPPING**  
 Promo Code **GWP18**

www.CRCPRESS.com **CRC Press**  
 Taylor & Francis Group

*The IFPUG Guide to IT and Software Measurement: A Comprehensive International Guide*  
*Certified Function Point Specialist Examination Guide*

(Component Sizing, continued from page 26)

The Calculator is a purely functional service component, in that it does not maintain any state information. Each operation is independent, meaning that no operation is related to another via dependencies on persistent data. A stateless component is said to exhibit low communicational (or internal) cohesion.

At the other end of the spectrum is a component that maintains an enterprise data object. In the vocabulary of requirements analysts to maintain is to manage the life-cycle of an object, and is often expressed as the Create, Read, Update, Delete (CRUD) analysis pattern – a find operation is available included, although it is not part of the acronym.

Interface Objects

Operations: create, read, update, delete, find

Component ObjectServer

Provides: Objects

Note that like the Calculator, this component provides all of its operations on one interface.

Often this kind of component is specified with two interfaces for technical reasons like so:

Interface Object

Operations: read, update, delete

Interface ObjectFactory

Operations: create, find

Component ObjectServer

Provides: ObjectFactory, Objects

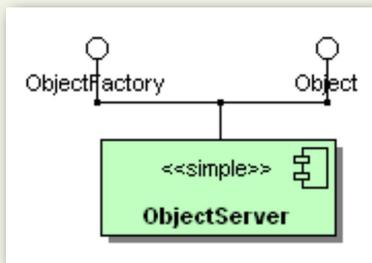


Figure 10 Service with multiple interfaces

This technique is called the Factory Pattern, a factory being a thing that manufactures Objects, hence the create operation is assigned to that interface. What is important to note is that the number of operations did not change (functional requirements), while the number of interfaces provided (technical requirements) did.

## 1.6 Measuring Components

A component is measured by counting the number of interfaces that it provides and requires, as well as the number of operations that it implements. The former is related to technical requirements, while the latter is related to functional requirements.

The scope of the measurement is quantified by the number of components measured. In the following UML class diagram the number of components, hence the scope is 2. The reader will recall the “hollow arrowhead” symbol from Figure 1 as a realization. The “normal arrowhead” is called a dependency. These are the logical equivalents of “provides” and “requires.”

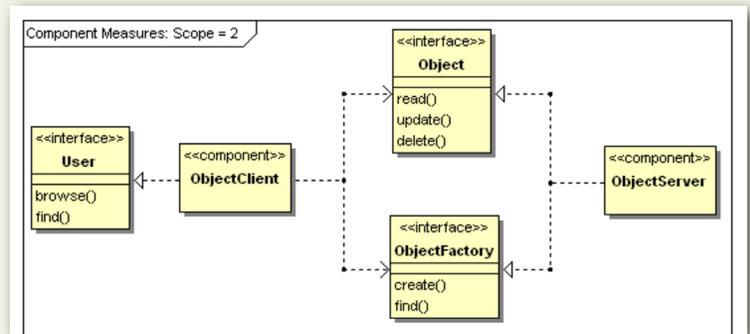


Figure 11—Measurement Scope indicated using Class Notation

| Measurement Scope: 2 | Technical Measurements |          | Functional Measurements |          |
|----------------------|------------------------|----------|-------------------------|----------|
|                      | Interfaces             |          | Operations              |          |
|                      | provided               | required | provided                | required |
| Component            |                        |          |                         |          |
| ObjectServer         | 2                      | 0        | 5                       | 0        |
| ObjectClient         | 1                      | 2        | 2                       | 5        |

## 1.7 References

### 1.7.1 ISO 14143:2007 Definitions

Base Functional Component (BFC): elementary unit of Functional User Requirements defined by and used by an FSM Method for measurement purposes.

BFC Type: defined category of BFCs

Boundary: conceptual interface between the software under study and its users.

(continued on next page)

*(Component Sizing, continued from page 27)*

**FSM Method:** specific implementation of FSM defined by a set of rules, which conforms to the mandatory features of this part of ISO 14143.

**Functional Domain:** class of software based on the characteristics of Functional User Requirements which are pertinent to FSM.

**Functional Size:** size of software derived by quantifying the Functional User Requirements.

**Functional Size Measurement (FSM):** process of measuring Functional Size.

**Functional User Requirements (FUR):** sub-set of the User Requirements describing what the software does, in terms of tasks and services.

1.7.2 OMG Unified Modeling Language (v2.4) Definitions

**Component:** A component represents a modular part of a system that

encapsulates its contents and whose manifestation is replaceable within its environment. A component defines its behavior in terms of provided and required interfaces. As such, a component serves as a type whose conformance is defined by these provided and required interfaces

**Interface:** An interface specifies a contract; any instance of a classifier that realizes the interface must fulfill that contract.

“Since interfaces are declarations, they are not instantiable. Instead, an interface specification is implemented by an instance of an instantiable classifier, which means that the instantiable classifier presents a public facade that conforms to the interface specification. Note that a given classifier may implement more than one interface and that an interface may be implemented by a number of different classifiers.”

## Some Doubts about the Objectivity of Logical Determination of the Uniqueness of the Elementary Process in the Function Point Analysis

*By Marian Zalczan, Ph.D., ASSECO Poland, Rzeszów*

### Table of Contents

1. Introduction
  2. The relationship between the FPA and FUR
  3. Very suitable direction of changes made in CPM R4.2.1 & R.4.3.1
  4. Doubts related to the lack of precision in determining the uniqueness of the elementary process
  5. Proposals for changes
- Conclusions  
References  
About the author

### 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)*

*(continued on next page)*

*(Some Doubts about the Objectivity, continued from page 28)*

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 as 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).

### 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

*(continued on next page)*

*(Some Doubts about the Objectivity, continued from page 29)*

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 (Section 4).

## 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.

*(continued on next page)*

*(Some Doubts about the Objectivity, continued from page 30)*

*(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 always consist of one EI, one EO and one EQ, as other EIs, EOs and EQs may be only treated as “minor differences” (or “minor variations“ in DETs or FTRs as well as “multiple alternatives, variations or occurrences” of processing logic) in comparison to the others (if only so requested by the user in the FUR)!

**Note 2:** 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 Examples. In

both of them, the elementary processes should be 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 distort the idea of the whole method and make it incompatible.

This may also result from the following example. Imagine that two identical 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 unacceptable, 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.

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.

*(continued on next page)*

*(Some Doubts about the Objectivity, continued from page 31)*

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 become acquainted with it.*

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

## NEW IFPUG MEMBER BENEFIT!

A new benefit has been added for IFPUG members.  
All White Papers are now available for FREE through the  
online store to all IFPUG members.

Take advantage of this benefit today!

## Certification Committee

*By Teresa Cristina De Spagna Zenga Beraldo*

The first half of 2015 has been a very busy time for the Certification Committee (CC). There are a lot of interesting things happening and we would like to share some of them with you:

- The market continues to pursue Function Point Certification; we have been pleased to congratulate 78 new Specialists (CFPS) and 32 new Practitioners (CFPP) from January to March, 2015.
- In the same period, 31 extensions of CFPS Certifications were approved through the Certification Extension Program.
- The number of CFPS exams doubled during the three months from January to March 2015 compared to the six months from July to December 2014.
- The country with the highest Function Point Certification Exams growth in their own language was Italy, with an increase of 231% from January to March 2015.

A year and a half after the announcement of the Certified SNAP Practitioner (CSP) exam, the Certification Committee organized and ran, in January 2015, a survey among the Certified SNAP Practitioners. We gathered very useful information, giving helpful insight into the maturity level of the SNAP Certification.

We are also working on several ideas and are looking for opportunities to increase IFPUG membership. We expect to bring good news regarding these ideas soon!

And last but not least, we would like to welcome Lori Limbacher (Holmes) who joined the Certification Committee in March!

### IFPUG to hold CFPS and CSP Manual Exams!

**CSP exam to be held in Sao Paulo, Brasil  
on November 17, 2015**

**Two sessions will be held:**

**9:30-11:30 am and 12:30-2:30 pm.**

**CFPS exam to be held in Madrid, Spain  
on September 24, 2015.**

**Additional information can be found  
on the IFPUG website.**

## Communications and Marketing Committee

*By David Thompson, Chair*

### Social media harmonization, marketing to encourage members to take the CFPS exam, promotion of ISMA<sup>10</sup>, plus website update and eBlasts

During the first six months of 2015, the Communication and Marketing Committee (CMC) focused on harmonizing our social media accounts and promoted their use. Our LinkedIn accounts were re-organized to create two official groups: The IFPUG Official Group and the IFPUG SNAP Official group. And we implemented a Social Media poster WordPress plug-in that automatically generates posts for Twitter, Facebook, and LinkedIn every time a new post is added to the IFPUG website. These have generated increased reader views of the social media accounts, extending IFPUG's reach into the metrics community.

Challenged by the IFPUG Board in 2014 to develop an approach to encourage more members to take the CFPS exam, the CMC developed a campaign to solicit and post on the website, testimonials from CFPSs on the value of certification. To date, we have accepted and posted nineteen testimonials, that you can see in the right sidebar of the home page on the IFPUG website. We expect that this will encourage other members, who are thinking about certification to make plans and prepare to take the exam.

In February, March and April the CMC, through the website, the weekly eBlasts and social media, engaged in an intense marketing campaign to publicize the ISMA<sup>10</sup> week of conference, classes and exams, held at the Sheraton Airport Hotel in Charlotte, NC. As a result, attendance was up from the attendance in the prior US-based conference held in 2012 in Phoenix.

During the 6 month period, The CMC processed 52 web update requests and sent 30 eBlasts, on diverse topics, including those specifically targeted for the ISMA<sup>10</sup> conference.

Additional accomplishments included posting, on the website of five recorded webinars on selected chapters in the newest IFPUG Book, *The IFPUG Guide to IT and Software Measurement*. These are linked from the website page, **The Newest IFPUG Book**.

Looking to the future, the committee has been tasked with developing a marketing plan, as described in a Marketing Strategy document, which was presented by our Board Liaison, Mauricio Aguiar at the IFPUG board meeting in April. And we are considering a re-organization of the website's Frequently Asked Questions page as proposed by the chair of the Conference and Education Committee.

Look for more on this in the second half of 2015!

*(continued on next page)*

(Committee Reports, continued from page 33)

## Conference and Education Committee Report

**By Peter Thomas, Chair**

The Conference & Education Committee (CEC) spent much of the first half of the year working with CMA (Creative Marketing Alliance - <http://thinkcma.com/>) on the ISMA<sup>10</sup> Conference held in the US on April 27-30, 2015. The aim of ISMA is, as from its acronym, to be the 'International Software Measurement & Analysis' conference. That includes putting together the 'Measurement' and the 'Analysis' parts in order to have valuable information for the decision-making process. To enable knowledgeable speakers to share their experience with you, we have partnered with other organizations, TI Metricas in Brazil for ISMA<sup>11</sup> and the Italian Software Metrics Association ([www.gufpi-isma.org/](http://www.gufpi-isma.org/)) for ISMA<sup>12</sup>. Details are available elsewhere in this publication and on the IFPUG website.

We are working with other organizations for additional events and are seeking organizations who wish to partner with us. In particular we would like to hold an event in India. Please contact [cec@ifpug.org](mailto:cec@ifpug.org) if your organization can help us.

The CEC is also preparing a series of recorded webinars from the authors of the IFPUG Book, *The IFPUG Guide to IT and Software Measurement*. Some are already available to IFPUG members others will be published in the second half of 2015. You will find information on purchasing the book on the IFPUG website. There are forty-three chapters by fifty-two authors from thirteen different countries, providing a comprehensive view on IT and Software measurement.

Just a reminder, Conference content can be found in the "Knowledge Base" on the Members' Services website at no charge. Please consider volunteering for an IFPUG Committee to give your support and ideas. Send an email to [ifpug@ifpug.org](mailto:ifpug@ifpug.org) or complete the volunteer form.

## Functional Sizing Standards Committee

**By Dan French, Chair**

The Functional Sizing Standards Committee (FSSC) has had a busy and productive first half of 2015. Recently we published uTip#3 – "Early FPA and Consistent Cost Estimating", which can be found here." The FSSC is also releasing the previously completed uTip #2 – "Project Testing" which discusses the use of Function Point Analysis to properly size a testing project and can be found here. Another publication from the FSSC that will be available by the time you read this is the

"Embedded Systems" white paper, which covers the application of IPFUG Function Point Analysis in the embedded systems application domain. We've also released an update to the iTip #6 – "Shared Data Real-time Responses." The Data Analytics white paper, which is an addendum to the "Hints to Counting Enterprise Data Warehouses" white paper published by the New Environments Committee (NEC) in 2007, will be released in the near future as well.

In April the FSSC held its annual meeting prior to the ISMA<sup>10</sup> conference in Charlotte, NC. In addition to reviewing uTips, iTips and white papers, the committee discussed a number of counting related issues and topics, including some publicly available tools, as well as current and future projects. Currently the FSSC is working on: a collaboration on Shared Data guidance for IFPUG FPA, a white paper on Business Process Management (workflow) applications, a white paper on Integrated External Outputs (EO)/External Inquiry (EQ), and a review/update of case studies published under prior versions of the Counting Practices Manual. The FSSC will be sending out a survey to the IFPUG membership to identify other potential topics for uTips, iTips, or white papers. Please be sure to complete the survey when you receive it.

If you are interested in volunteering to serve on the FSSC or assisting with an FSSC project, please be sure to complete and submit a volunteer form.

## International Membership Committee Report

**By Pierre Almén, Acting Chair**

Since fall 2014, I have been acting as the new Chairman for the International Membership Committee (IMC). IMC has focused on reaching out to the IFPUG members by implementing country representatives in Brazil, China, India, Italy and Spain. All of the IFPUG members or potential members in those countries now have an easy way of asking questions about membership, certification etc. and if wanted they can do that in their native language. To accomplish this, we added four new members to the committee. A warm welcome to new country representatives Anjali Mogre (India), Cao Ji (China), Iván Pinedo (Spain) and Saurabh Saxena (India)! Together with existing members Gianfranco Lanza from Italy and Márcio Silveira from Brazil, we are really an international committee.

To get feedback from all IFPUG members, we are now working with a survey that is planned to be sent out fall 2015. We are also collecting information about universities etc that have classes for Function Points and software measurement.

When you have membership questions, please contact us by using the contact information available on the IFPUG website.

(continued on next page)

(Committee Reports, continued from page 34)

## Non-Functional Sizing Standards Committee (NFSSC)

By **Talman Ben-Cnaan**

*The SNAP (Software Non-functional Assessment Process) method of non-functional sizing continues to evolve*

### 1. SNAP Assessment Practices Manual v2.3 was Released

The new release of the Assessment Practices Manual (APM) provides better clarification on how to use SNAP in organizations that are currently using IFPUG Function Point Analysis.

The APM provides detailed guidelines of joint counting of the Functional size and the Non-functional size. For each subcategory, a set of rules and examples were added to better clarify how to count Function Points and SNAP Points together.

In addition, examples were added to the sub-categories, covering more complex areas. Graphical description was added to examples.

An APM 2.3 Quick Reference Guide was created and is also available on the IFPUG on-line store.

### 2. SNAP Users Forum

A virtual meeting of SNAP users was held in June 2015. Users from the US, Brazil, Spain, France and India discussed common challenges of implementing SNAP in their organization.

Here is a short list of common issues:

- How to calculate productivity when the Elementary Process contains both Functional User Requirements (FUR) and Non-functional Requirements (NFR) and users cannot separate the effort needed to provide the FUR and NFR? Users were looking for a Function Point/Snap Point (FP/SP) ratio in terms of effort and/or SNAP productivity
- Users stated that SNAP Productivity is not consistent across some sub-categories. This issue was identified in sub-categories 1.5 and 3.2.

(The NFSSC is searching for better counting formulas based on data from users).

- Suggestions were raised to use different counting formulas and hence increase SNAP accuracy. The NFSSC captured the suggestions and will analyze their benefits (i.e., improve accuracy without making the SNAP counting process more complex).
- Users asked for more examples in some areas.
- Vendors of Software Estimation Tools expressed interest in working with the NFSSC to incorporate SNAP into their estimation tools.

### 3. A Joint IFPUG / COSMIC Glossary of Non-Functional Requirements

The COSMIC and IFPUG organizations have long-established methods of measuring the size of the functional requirements for the software product, but the problems of gathering consistent data on the non-functional requirements and on the project requirements and constraints suffer from a lack of commonly-agreed and clear definitions. Views differ on how to measure non-functional requirements.

Although COSMIC and IFPUG tackle these problems in different ways, The NFSSC and COSMIC have collaborated in creating a collective Glossary of terms for non-functional requirements and for project requirements and constraints.

The common Glossary will be presented in the 2015 IWSM Mensura conference in Cracow, Poland, October 2015.

The SNAP counting method does not change as a result of the common Glossary. Moreover, we have proven again the strength of SNAP in counting ALL NFR, and irrespective of the way NFRs are defined!

## 20 years of IFPUG Participation in ISO Standards!

## International Standards (ISO) Committee

By **Carol Dekkers, CFPS (Fellow), PMP, CMC, Chair**

It's a time for celebration! 2015 marks the 20th year of formal IFPUG involvement in ISO/IEC standards development and after years of development, writing, negotiating, collaboration and lots of meetings, we're at a point where function points and ISO standards in the area have stabilized. Congratulate yourself as an IFPUG member for all we've accomplished!

Let's start with a short summary of the 20 years of IFPUG and ISO/IEC accomplishments and where the future might lie with IFPUG involvement.

### Background

As you may or may not know, IFPUG and ISO is an important liaison/participation that provides mutual benefit to participating member countries (ISO participation is by "national body" – IFPUG participates as a member of the United States Technical Advisory Group (TAG) to ISO/IEC JTC1 SC7 Software and Systems Engineering and also as a Category "C" liaison member to SC7). For 99% of you, this information may



(continued on next page)

(Committee Reports, continued from page 35)

be irrelevant because ignorance is bliss (for non-native English speakers, the idiom “Ignorance is bliss” simply means that being ignorant of information is not always a bad thing. In ISO standards terms, this means that our work is important, however, for many U.S. domiciled corporations, international standards are not a priority). For the remaining 1%, ISO standards may be an important or emerging knowledge area – and one in which you may want to become more involved.

**THE most important benefit** of our association and participation in International Standards is that the **IFPUG standard(s) remain relevant and current with the software and systems engineering (ISO/IEC JTC1 SC7) standards.** 2015 marks the 20th year of IFPUG involvement in ISO/IEC standards development, and perhaps it makes sense to look at IFPUG and ISO/IEC accomplishments since then.

## 20 years of ISO/IEC standards relevant to IFPUG

Formalized processes and procedures involving a changing landscape of approximately 60+ countries (each “P”rimary member country gets one vote, and ongoing participation is required to maintain Primary status,) govern what becomes an ISO/IEC standard and how it gets there. But it’s not just following the processes that are involved. For every standard (and there are thousands of ISO standards), it takes a culmination of hundreds of hours of meetings in person and by email, plus dedication to do development work between meetings, international travel and domestic travel (each country has internal meetings to formulate a national body “position” in addition to international meetings), plus diplomacy, negotiation and collaboration skills on the part of experts who participate (the majority of experts donate their time without reimbursement from an employer.)

**It’s all for a good cause: creating and fostering international best practices in software and systems development that can be used and followed globally.**

Here’s a summary of the standards that IFPUG has played a major part in developing:

- **ISO/IEC 14143-n (six standards -1 through -6):** This is a six part suite of framework standards that cover definitions, concepts, guidelines and other relevant topics pertinent to “Functional Size Measurement (FSM).” This suite was first published in the late 1990’s and is reviewed and updated (as needed) on a five year maintenance cycle. This suite of standards creates consistency and repeatability for measurement professionals and developers of new FSM methods (of which there are currently 5 ISO/IEC conformant standards outlined below.)
  - o ISO/IEC 14143-1: Information technology — Software measurement — Functional size measurement- Part 1: Concepts and Definitions (the main set of requirements for FSM)
  - o ISO/IEC 14143-2: ... Part 2: Conformity evaluation of software size measurement methods to ISO/IEC 14143-1

- o ISO/IEC 14143-3: (Technical Report) ... Part 3: Verification of functional size measurement methods
- o ISO/IEC 14143-4: (Technical Report) ... Part 4: Reference model
- o ISO/IEC 14143-6: ...Part 6: Guide for use of ISO/IEC 14143 series and related International Standards

**In short:** The 14143 series of standards lays out the core set of requirements for functional sizing of software, and is as relevant today as when they were first published.

- **ISO/IEC 15939 – Software Measurement Framework:** Modeled after the Practical Software and Systems Measurement (PSMM) and the Goal-Question(-Indicator)-Metric approach, this standard provides insights and guidance for practitioners interested in setting up a sustainable measurement program.
- **ISO/IEC FSM Method standards (all ISO conformant):**
  - o ISO/IEC 19761: COSMIC-FFP
  - o ISO/IEC 20926: IFPUG 4.3
  - o ISO/IEC 20968: Mk II Function Point Analysis (UK)
  - o ISO/IEC 24570: NESMA (Netherlands)
  - o ISO/IEC 29881: FiSMA (Finland)

Our current IFPUG method is identical to the ISO/IEC standard version and is available to non-members of IFPUG ([www.ifpug.org](http://www.ifpug.org)) for free download.

- **ISO/IEC 29155: series of Project Performance Benchmarking Framework (Guidance) standards:** based on the International Software Benchmarking Standards Group (ISBSG) body of work with international input.

## Looking forward – Expanding our Influence into the Cloud and beyond

With newer agile methodologies, cloud computing, SAAA and other software and systems concepts becoming mainstream, it’s timely that we (IFPUG) leverage the power and capabilities of Functional Size Measurement and our IFPUG 4.3 FSM method in these arenas. As such, Steve Woodward has been meeting with NIST committees (National Institute of Standards and Technology) to ensure that functional sizing is considered where it may fit in with cloud computing standards. While this means that IFPUG would expand our participation beyond the ISO/IEC JTC1 SC7 software and systems engineering domain, it provides IFPUG with the opportunity to remain at the forefront of standards development. Steve met recently with ITU-T in Geneva to learn what is now going to become a priority since the work on Cloud JCA is complete.

## IFPUG / ISO Newsletter – YOUR Input needed

Thank you for your ongoing support for our IFPUG and ISO efforts (and for reading this far!) We’re working on an IFPUG/ISO newsletter (biannually at least) and we’d like to know what you’d like in it from links to draft documents (to review) to FAQ to anything else. Please send me an email ([dekkers@qualityplustech.com](mailto:dekkers@qualityplustech.com)) with your ideas!

## IFPUG Board of Directors

### **Kriste Lawrence, President**

Hewlett-Packard  
kriste.lawrence@hp.com

### **Tom Cagley, Vice President**

David Consulting Group  
tcagley@earthlink.net

### **Debra Maschino, Treasurer**

NASCO  
debra.maschino@nasco.com

### **Joe Schofield, Immediate Past President**

joescho@joejr.com

### **Mauricio Aguiar, Secretary, Director of Communications and Marketing**

TI Metricas  
mauricio@metricas.com.br

### **Luigi Buglione, Director of Education & Conference Services**

Engineering.IT SpA  
luigi.buglione@eng.it

### **Dácil Castelo, Director of Sizing Standards**

Leda-mc  
dcastelo@leda-mc.com

### **Christine Green, Director of Applied Programs**

Hewlett-Packard  
christine.green@hp.com

### **Pierre Almén, Director of International & Organizational Affairs**

ImproveIT  
Pierrea@coolmail.se

## Behind the Scenes

*By Constance Holden, Executive Director*

It's been a busy year for IFPUG and I hope that you have been able to be involved. Certifications in the CFPS, CFPP and CSP continue to show the excellent standards of the IFPUG membership.

I did want to take this time to review the meaning of being a member of an association. IFPUG was organized back in 1987 by a group of people for the purpose of promoting and encouraging the effective management of application software development and maintenance activities using Function Point Analysis. We must remember that IFPUG is a non-profit organization that is governed by its members who are committed to the IFPUG principals and standards of professionalism. The mission of IFPUG is to be a recognized leader in promoting and encouraging the effective management of application software development and maintenance activities through the use of Function Point Analysis and other software measurement techniques. IFPUG is fortunate to be a truly international organization with members from over 30 countries. Any association is only as strong as its members and IFPUG is no different. I would like to encourage you to volunteer to serve on a committee. Committee members help to make associations strong and guide the association into the future. Do you have ideas or suggestions? We are always interested in what our membership has to say, so please email [ifpug@ifpug.org](mailto:ifpug@ifpug.org) or call us at 609-799-4900.

IFPUG also provides benefits to members which provide a ROI (Return on Investment) that exceeds the cost of membership. Members can receive an electronic copy of the CPM, access to presentations given at past conferences, access to Members' Only forums, documents in the Knowledge Base as well as discounts on conferences and workshops. IFPUG continues to increase member benefits by recently adding free White Papers for members.

Michele Giovine has recently joined the IFPUG HQ office and is assisting with certifications and member services. We are happy to include her on the IFPUG Team.

So, in closing, I thank you for your continued support and I encourage you to participate in the upcoming conferences scheduled in Brasil and Italy.

*Best regards,  
Constance Holden  
Executive Director, IFPUG*



## Committee Rosters

### **Certification Committee**

- Gregory Allen, Pershing – **Chair**
- Mahesh Ananthakrishnan, Cognizant Technology Solutions – **Vice Chair**
- Jim McCauley
- Joanna Soles, WellPoint
- Teresa Beraldo, Banco Bradesco S/A
- Lori Limbacher, Deloitte Consulting, LLP

### **Communications and Marketing Committee**

- Walter David Thompson, Blue Pine Solution Centre – **Chair**
- Antonio Ferre Albero, GFT IT Consulting
- David Herron, David Consulting Group
- Stephen Neuendorf, David Consulting Group
- Paul Radford, Charismatek Software Metrics
- Justin Keswick, Bank of Montreal

### **Conference and Education Committee**

- Peter Thomas, Steria – **Chair**
- Antonio Ferre Albero, GFT IT Consulting

- Toni Ramos, David Consulting Group
- Thiago Silva Da Conceicao

### **Functional Sizing Standards Committee**

- Daniel Bradford French, Cobec Consulting – **Chair**
- Bonnie Brown, Hewlett-Packard – **Vice Chair**
- Diana Baklizky, TI Metricas
- E. Jay Fischer, JRF Consulting
- Steve Keim, David Consulting Group
- Tammy Preuss, AT&T
- Roopali Thapar, IBM
- Peter Thomas, Steria
- Adri Timp, Equens
- Charles Wesolowski

### **International Membership Committee**

- Anjali Mogre, Atos Origin International SAS
- Cao Ji, Beijing Suiji Tech
- Ivan Pinedo, LDA Consulting, S.L.
- Saurabh Saxena, Amdocs Development Centre India Pvt Ltd
- Marcio Silveira, HP
- Gianfranco Lanza, CSI Piedmonte

### **ISO Committee**

- Carol Dekkers, Quality Plus Technologies, Inc. – **Chair**
- Steve Woodward, Cloud Perspectives

### **Non-Functional Sizing Standards Committee**

- Talmon Ben-Cnaan, Amdocs – **Chair**
- Kathy Lamoureaux, HP – **Vice Chair**
- Roopali Anand, IBM
- Abinash Sahoo
- Julian Gomez, Sopra Group Informatica S.A.
- Mousa George Mitwasi, Optum
- Dr. Charley Tichenor
- Jalaja Venkat, iGATE Global Solutions
- Roopali Thapar, IBM

### **IFPUG-ISMA Academic Secretary for Universities:**

- Dr. Maya Daneva, University of Twente

## Congratulations to these NEW Certified Function Point Specialists!

Angela Abreu  
TI Metricas

Sravanti Adipudi  
Accenture

Alessandra Albinati  
Almaviva SpA

Antonio Attadia  
Almaviva SpA

Elia Autiello  
ERICSSON IT SOLUTIONS &  
SERVICES S.P.A.

Ermanno Bagnai  
Almaviva SpA

Armando Baldinacci  
Selex ES SpA

Simona D'Ambrosio Barbarisi  
Almaviva SpA

Elisabetta Belotti  
Almaviva SpA

Alberto Bentivegna  
Almaviva SpA

Marco Bergagnini  
ERICSSON IT SOLUTIONS &  
SERVICES S.P.A.

Andrea Bertanelli  
NTT DATA Italia SpA

Rajat Biswas  
TATA CONSULTANCY SERVICES  
LTD.

Chiara Di Bonito  
Almaviva SpA

Annamaria Bucci  
NTT DATA Italia SpA

Stefano Bullo  
ERICSSON IT SOLUTIONS &  
SERVICES S.P.A.

(continued on next page)

(New CFPS, continued from page 38)

|   |  |  |  |
|---|--|--|--|
| Giulio Buonomini<br>Selex ES SpA  | Diego d'Ippolito<br>Almaviva SpA   | Stefano Gianelloni<br>FASTWEB                                    | Giuseppe Io Presti<br>ACCENTURE  |
| Maria Cristina Calzetti   | Anitha Damodharan<br>IBM   | Rosalba Giunta<br>Almaviva SpA                                   | Maria Lonetti<br>NTT DATA Italia SpA                                       |
| Luca Cammisa<br>FASTWEB   | Isabela Del Corso<br>TI Metrics Ltda   | Leon Denizard De Carvalho Gomes<br>TI Metrics                    | Robin Lucchesi<br>Aruba SpA  |
| Giorgio Caporaletti<br>Selex ES SpA                                     | Paulo Dias<br>DiasSoft Informatica LTDA                                      | Valentina Grasso<br>ERICSSON IT SOLUTIONS &<br>SERVICES S.P.A.   | Valeria Maiorino<br>Almaviva SpA   |
| Giuliana Capri<br>IBM Italia S.p.A.                                     | Luisa Distaso<br>Indra Italia Spa  | Gianluca Rocco Greco<br>Almaviva SpA                             | Gianluca Marchetti<br>Almaviva SpA   |
| Fiorenzo Nicola Carmenini<br>Almaviva SpA                               | Laura Fabbri<br>NTT DATA Italia SpA  | Dora Grillo<br>Almaviva SpA                                      | Paolo Marchi<br>Selex ES SpA   |
| Anna Carnera<br>Almaviva SpA  | Simona Farina<br>Selex ES SpA  | Francesca Grossi<br>S.I.N. S.p.A.                                | Sauro Marini<br>Selex ES SpA   |
| Pierluigi Casadei<br>MAGGIOLI SPA VAT IT<br>02066400405                 | Roberto Fenaroli   | D'Ubaldo Guglielmo<br>ERICSSON IT SOLUTIONS &<br>SERVICES S.P.A. | Luciana Martins  |
| Alexandre Castagna<br>SISTRAN INFORMÁTICA                               | Maria Cristina Ferrari<br>Almaviva SpA                                       | Bahram Hedayati  | Livio Maurizi<br>Sirti S.p.A.  |
| Marcello Ceccarelli<br>Almaviva SpA                                     | Filippo Fiamma<br>ERICSSON IT SOLUTIONS &<br>SERVICES S.P.A.                 | Theresa Hodge<br>Computer Sciences Corporation                   | Sabrina Mello<br>Spread Sistemas e Automacao Ltda.                         |
| Ho Chuen Chan   | Claudia Filiberti<br>Selex ES SpA  | Sang Hoon Hwang  | Maurizio Menghini<br>Almaviva SpA  |
| Andrea Cherici<br>Almaviva SpA  | Patrizia Forlano<br>FASTWEB  | Sonia Iannone<br>ERICSSON IT SOLUTIONS &<br>SERVICES S.P.A.      | Luca Mignogni<br>ERICSSON IT SOLUTIONS &<br>SERVICES S.P.A.                |
| Daniele Cibir<br>NTT DATA Italia SpA                                    | Giuseppe Miserendino Francesco<br>ERICSSON IT SOLUTIONS &<br>SERVICES S.P.A. | Annuziata Italiano<br>Almaviva SpA                               | Gelsomina Mori<br>Eustema S.p.A  |
| Sonia Ciciotti<br>ERICSSON IT SOLUTIONS &<br>SERVICES S.P.A.            | Giulio Galante<br>ERICSSON IT SOLUTIONS &<br>SERVICES S.P.A.                 | Karol Jastrzebski<br>Accenture                                   | Paolo Mori<br>Almaviva SpA   |
| Francesco Cincotti<br>Almaviva SpA                                      | Sonia Galimberti<br>ERICSSON IT SOLUTIONS &<br>SERVICES S.P.A.               | Krzysztof Klosowski<br>Orange Polska                             | Paula Xavier Muniz<br>SISTRAN INFORMÁTICA                                  |
| Fulvio Contaldi<br>Convergent Technologies Partners,<br>S.p.A.          | Emanuele Galli<br>ERICSSON IT SOLUTIONS &<br>SERVICES S.P.A.                 | Tetyana Komarova<br>NTT DATA Italia SpA                          | Antonella Nalli<br>ERICSSON IT SOLUTIONS &<br>SERVICES S.P.A.              |
| Laura Contino<br>Almaviva SpA   | Silvia Gallo<br>Engineering Ingegneria Informatica<br>SpA                    | Daniele Lagatta<br>Almaviva SpA                                  | Barbara Nardis<br>NTT DATA Italia SpA                                      |
| Antonio Cotroneo<br>Ernst & Young Financial-Business<br>Advisors S.p.A. | Alessandra Gallozzi<br>Almaviva SpA  | Aldo Langianese<br>Almaviva SpA                                  | Roberta Nasti<br>Selex ES SpA  |
| Tao Cui<br>ICBC SDC   | Andrea Garavini<br>NTT DATA Italia SpA                                       | Roberto Laurita<br>NTT DATA Italia SpA                           | Edgard Leuenroth Neto<br>Banestes S/A Banco Do Estado Do<br>Espirito Santo |
| Andrea Curreli<br>FASTWEB   | Vittorio Giacchetti<br>Almaviva SpA  | Michal Lazicki<br>Accenture                                      | Mauricio Neves   |
| Maria Giovanna D'Addario<br>MAGGIOLI SPA VAT IT<br>02066400405          |  | Cesar Lima<br>Prodesp  | Cristiane Oliveira   |
|   |  | David Lipton   | Massimiliano Pace<br>Ericsson Telecomunicazioni S.p.A.                     |

(continued on next page)

(New CFPS, continued from page 39)

|  |  |  |   |
|--|--|--|---|
| Nicola Marcello Parisi<br>Sirti S.p.A.                       | Antonietta Rapuano<br>Engineering Ingegneria Informatica SpA | Barbara Sgaragli<br>Selex ES SpA   | Rafael Tineo<br>Leda Colombia   |
| Emma Di Pasquale<br>NTT DATA Italia SpA                      | Nicolo Rebella<br>Selex ES SpA                               | Cleonice da Silva<br>SISTRAN INFORMÁTICA                                     | Adriana Toda<br>CPM Braxis S.A.                                       |
| Rafael Passaro<br>Prime Informatica Alpha Ltda               | Paolo Ricotta<br>Selex ES SpA                                | Elielton Costa Da Silva<br>Indra Brasil Soluções e Serviços Tecnológicos S/A | Lucia Immacolata Tricarico<br>ERICSSON IT SOLUTIONS & SERVICES S.P.A. |
| Antonio Pellicano<br>ERICSSON IT SOLUTIONS & SERVICES S.P.A. | Maria Romano<br>Almaviva SpA                                 | Pamela Simonovich<br>QSM / Quantitative Software Management, Inc.            | Alberto Livio Troisi<br>Almaviva SpA                                  |
| Daniela Petrucci<br>Almaviva SpA                             | Pietro Rosone<br>FASTWEB                                     | Nitin Kumar Singh<br>Accenture   | Olga Uberti<br>NTT DATA Italia SpA                                    |
| Vincenzo Pinto<br>Almaviva SpA                               | Giuseppe Rotondi<br>ERICSSON IT SOLUTIONS & SERVICES S.P.A.  | Leandro Siniscalchi<br>Cast Informatica S.A. CNPJ:<br>03.143.181/0001-01     | Alessandro Venturi<br>Selex ES SpA                                    |
| Matteo Polelli<br>FASTWEB                                    | Sergio Jimenez Rubio<br>IBM                                  | Elvira Smedile<br>ERICSSON IT SOLUTIONS & SERVICES S.P.A.                    | Claudia Viazzi<br>ERICSSON IT SOLUTIONS & SERVICES S.P.A.             |
| Emanuela Polito<br>Almaviva SpA                              | Marco Salatiello<br>ERICSSON IT SOLUTIONS & SERVICES S.P.A.  | Francesco Sorace<br>NTT DATA Italia SpA                                      | Antonina Vicario<br>Almaviva SpA                                      |
| Daniela Puch<br>BANCO BRADESCO S/A                           | Santa Di Salvo<br>Selex ES SpA                               | Jose Lopez Soriano   | Jefferson Vicente<br>SISTRAN INFORMÁTICA                              |
| Michele Quaini<br>FASTWEB                                    | Carlos Santos<br>SISTRAN INFORMÁTICA                         | Georgia Carvalho Souza<br>CPM Braxis S.A.                                    | Thiago Vilela<br>TI Metrics   |
| Fabrizio Queiroz<br>PD Case Informatica Ltda                 | Jlenya Di Sanza<br>Selex ES SpA                              | Claudio Strazzullo<br>Almaviva SpA   | Cirillo Vincenzo<br>Accenture   |
| Filippo Raciti<br>Selex ES SpA                               | Leonardo Schwindt<br>Ministério Público Federal              | Cristian Talotti<br>ERICSSON IT SOLUTIONS & SERVICES S.P.A.                  | Simona Zarro<br>ERICSSON IT SOLUTIONS & SERVICES S.P.A.               |
| Antonella Ranzini<br>ERICSSON IT SOLUTIONS & SERVICES S.P.A. | Marcello Sgamma<br>NTT DATA Italia SpA                       | Luca Tamburini<br>Almaviva SpA   |   |

## Congratulations to these NEW Certified Function Point Practitioners!

|   |                                       |  |
|---|---------------------------------------|--|
| Alano Arraes<br>Informatica SpA                             | Gabriele Cianciaruso<br>FASTWEB       | Massimo Fratini<br>Selex ES SpA                            |
| Tiziana Angelini<br>Engineering Ingegneria Informatica SpA  | Severino di Cola<br>Selex ES SpA      | Stefano Ferrari<br>ERICSSON IT SOLUTIONS & SERVICES S.P.A. |
| Flavio Barsotti<br>NTT DATA Italia SpA                      | Ciro Coppola<br>Almaviva SpA          | Stefania Format<br>Engineering Ingegneria Informatica SpA  |
| Giuseppina Basile<br>Engineering Ingegneria Informatica SpA | Luigi Diana                           | Daniel Guinda<br>LEDA Consulting, S.L.                     |
| Mariapia Borrelli<br>Engineering Ingegneria Informatica SpA | Alessandra Durante                    |  |
|   | Bianca De Faria<br>BANCO BRADESCO S/A |  |

(continued on next page)

(New CFPP, continued from page 40)

**Renata Guzzardi**  
Engineering Ingegneria  
Informatica SpA

**HA Young Hun**

**Yong Hyun Lee**

**David Lipton**

**Massimiliano Loconte**  
ACCENTURE

**Marina Mancini**  
Almaviva SpA

**Fabricia Marques**  
Magna Sistemas e Consultoria  
S/A

**Luca Mele**  
Accenture

**Maria Carolina de Menezes**

**Rosana Lucia Nascimento**  
Cast Informatica S.A. CNPJ:  
03.143.181/0001-01

**Angela Nicolaio**  
SOGEI

**Claudio Occoffer**  
ERICSSON IT SOLUTIONS &  
SERVICES S.P.A.

**Sonia Olias**  
SOGEI

**Irene Pace**  
Capgemini Italia SPA

**Tamara Papa**  
Selex ES SpA

**Maria Grazia Di Pasquale**  
NTT DATA Italia SpA

**Eduardo Pereira**

**Mauro Ramos**  
BANCO BRADESCO S/A

**Maria Paola Rapanotti**  
Engineering Ingegneria  
Informatica SpA

**Samara Simha Reddy**  
Optum

**Salvatore Ricciardi**  
FASTWEB

**Gianfranco Romeo**  
Ericsson Telecomunicazioni  
S.p.A.

**Maria Di Russo**  
NTT DATA Italia SpA

**Davide Sammartino**  
ACCENTURE

**Cristiane da Silva**  
Elumini

**Leonardo Teixeira da Silva**

**Andrea Silva**

**Wilson Silva**  
Engineering Ingegneria  
Informatica SpA

**Flavia Soares**  
SISTRAN INFORMÁTICA

**Joao Sousa**

**Imma Volpe**  
Almaviva SpA

**Patrizia Volpi**  
Almaviva SpA

## Congratulations to these NEW Certified SNAP Practitioners!

**Ray Boehm**  
Softcomptech, Inc.

**Sainath Manchukonda**  
Optum

**Hari Krishna Marre**  
Optum

**Shrestha Mohanty**  
Optum

## Congratulations for 20 Years of CFPS Certified Function Point Specialist Fellows!

Spring 2015

**E. Jay Fisher**  
JRF Consulting

**James McCauley**

## CERTIFICATION MATTERS!

### CFPS Matters!



“Function Points are used across the world for cost estimation, productivity and defect analysis, estimation, etc. Becoming an IFPUG CFPS has helped me gain in-depth knowledge of this field, answering the famous quote: ‘You Can’t Manage What You Don’t Measure’.”

Jaya Goindani, India

**Schedule your CFPS exam at [www.Prometric.com/IFPUG](http://www.Prometric.com/IFPUG) !**

### CSP Matters



“As a CFPS Fellow I have always valued the contribution the certification had on my career, so when we decided to implement SNAP in the organization, I decided to have many of our team members go through the CSP certification process early on, and it paid-off handsomely. Having gone through the process helped us understand SNAP in great depth and develop and pilot guidelines for implementing SNAP consistently within the organization. Furthermore, having 13 team members achieve CSP certification gave the team great credibility and allowed our stakeholders to trust in the outcome of our sizing activities and support our goal of deploying SNAP.”

M. George Mitwasi, Ph.D., MBA, CFPS Fellow, PMP, CSP | Optum

Director IT – Estimation & Function Point Center of Excellence

**The next CSP exam will be held in Sao Paulo, Brasil, November 2015!**

**Check the IFPUG website for details.**

# ISMA<sup>10</sup> In the USA!

*By Peter Thomas, Conference and Education Chair*



April 2015 marked the arrival of ISMA<sup>10</sup> in Charlotte, North Carolina, USA. The week began with old acquaintances and professionals new to IFPUG gathering on Sunday evening ready for Board, Functional Sizing Standards Committee meetings and workshops on Monday.

The Sheraton Charlotte Airport Hotel did a great job of setting up rooms and ensuring everyone had a good breakfast to start their Monday.

Sheila Dennis and David Herron from David Consulting Group (DCG) led a class on Applying Function Points to Emerging Business Technologies. Everyone attending agreed that by the end of Tuesday they had gained a great insight into Function Point Analysis.

Meanwhile Mary Dale from Q/P Management Group delivered our first vendor version of the SNAP (Software Non-functional Assessment Practices) Workshop. As expected, all went smoothly, and several more champions of this addition to IFPUG's portfolio of IT Measurement methods gained the knowledge they will need to spread the word about this measurement practice.

Wednesday provided an opportunity for the workshop attendees to explore Charlotte, while the Board and Functional Sizing Standards Committee continued to make good progress with their scheduled meetings. More thanks to the Sheraton for arranging a quiet room for the SNAP certification exam and our thanks to George Mitwasi for proctoring. Connie Holden from CMA and Peter Thomas of the Conference and Education Committee were busy with final preparations for presentation day on Thursday. That evening there were several interesting discussions between long standing and new members of IFPUG on matters relating to software measurement and analysis.

The presentations on Thursday were given a great start by the keynote speaker Clark Walton, who spoke on Digital Forensics, in which he revealed how much personal data is readily collected from internet browsing and cell phone usage records. By the end of his presentation, everyone in the room was checking the settings on their phones to try to minimize the personal data they were sharing with the world.

After a brief break, the conference chair, Peter Thomas, introduced Philippe Guerin and Barbara Beach who provided valuable insight on the automation of functional sizing. They stayed for the day and were generous with their business card draw and their time to share the strengths and weaknesses of automated counting.

Carol Dekkers needed very little introduction and shared her experience of measuring Agile. As always, she did a great job in getting the questions coming from the audience.

Next up, Adri Timp provided a glimpse of the work the Functional Sizing Standards Committee is doing with an overview of their latest u-Tip, *Early Function Point Analysis* and consistent cost estimating.

Dan Horvath was a new face for regular conference attendees. He shared, in a great presentation, the best practice his organization has achieved with their estimating process.

The lunch break provided time for discussion and establishing new friendships in the IT measurement community. All too soon Peter Thomas was "sweeping" everyone up to move to the presentation room.

George Mitwasi made a welcome return to the conference to share his organization's experience adopting SNAP. They have gained valuable insight and we look forward to the next update at a future conference.

*(continued on next page)*

(ISMA<sup>10</sup>, continued from page 43)



**leda**MC  
www.leda-mc.com



Joe Schofield met the expectations of regular conference attendees with another entertaining and insightful presentation on Agile measurement.

Eduardo Silva gave an interesting presentation on the critical contribution Function Point Analysis can make to software development processes.

Finally Christine Green presented some thoughts and direct experience in measuring IT. She shared the valuable experience she has had with many leadership teams to enable them to see beyond “cost”.

Peter closed the conference with thanks to the sponsors; IT Metrics, Leda, and CAST and announcing the next two conferences in Sao Paulo, Brasil and Rome, Italy.

## ISMA<sup>11</sup> in Sao Paulo, Brazil

**Date: November 18th, 2015**

**Location: Blue Tree Premium Morumbi  
Av. Roque Petroni Junior, 1000  
Brooklin Novo – Sao Paulo – SP  
Brazil, 04707-000**



The eleventh edition of the IFPUG ISMA Conference will provide a forum for practitioners and researchers to discuss the most recent advances in planning and sustaining measurement programs from both practical and theoretical perspectives. We invite professionals responsible for, involved in, or interested in software measurement to share innovative ideas, experiences, and concerns within this scope.

## SPEAKERS

Dácil Castelo, LEDAmc

Joe Schofield, Sandia National Laboratories  
(ret.) – via WebEx

Kriste Lawrence, HP

Luigi Buglione, Engineering Ingegneria  
Informatica SpA

Márcio Silveira, HP

Pierre Almén, ImproveIT

Steven Woodward, Cloud Perspectives

Tom Cagley, David Consulting Group

**Save the date!**

**See the BFPUG website for more.**

**Visit the IFPUG Website at [www.ifpug.org](http://www.ifpug.org)**

## IFPUG Brings ISMA<sup>12</sup> back to Italy in 2016!

After a 20 year absence, the IFPUG conference will return to Italy on May 3-5, 2016. The conference will be held in the center of Rome and hosted by the Italian Software Metrics Association, GUFPI-ISMA. Workshops will be held May 3rd; a CSP exam on May 4th; and the conference on May 5th. This will be a unique experience in the “Caput Mundi”, the Capital of the World.

A Call for Presentations on topics including Function Point Analysis, Software Non-functional Assessment Process, and other measurement-related items of interest will be announced soon.

Updated information can be found on the IFPUG website.