2011

Research and Development

Simple Function Point

A new Functional Size Measurement Method fully compatible with IFPUG FP ©

Data Processing Organization Srl

DPO

TABLE OF CONTENTS

TRACKING OF CHANGES

Issue Date	Version	Description of Change
18/04/2011	V001	Not Applicable
08/06/2011	V002	Insights about the correlation between IFPUG and SiFP

Edited	**Research Team**
Checked	**Valentina Martiri** **Franco Perna**
Approved	

ACRONYMS AND DEFINITIONS

Scope	Scope of measurement	The scope defines the functionality that will be included in a particularly simple measurement of function points. The scope: • defines a subset of the software to be measured; • is determined by the objective set for the measurement; • identifies which functions should be included in the measurement so as to provide relevant answers to the stated purpose; • could include more than one MSA.
MSA	Measurable Software Application	An aggregate of particular software functions from a user point of view, homogeneous in terms of level of logical abstraction. It is a functional group suitable for the functional measurement. It may be different from other types of aggregates responding to technical goals (i.e. client-server distribution)
BFC	Base Functional Component	Basic elements of the functional measurement.
DET	Data Element Type	A field not duplicated identifiable for users
EI	External Input	Elementary Process - Input type
EIF	External Interface File	Logical group of data used in read-only and therefore outside the application boundary.
EO	External Output	Elementary Process - Output type
EQ	External Inquiry	Elementary Process - Query type
FP	Function Point	Functional Size for a software product
FPA	Function Point Analysis	
FSMM	Functional Size Measurement Method	
IFPUG	International FP User's Group	
ILF	Internal Logical File	Logical Data group used in read / write operations inside the application boundary.
ISBSG	International Software Benchmarking Standards Group	International organism which has an open, non proprietary benchmarking database
SiFP	Simple Function Point	
SiFPA	Simple Function Point Association	International metrical Association for development and promotion of the SiFP method
UFP	Unadjusted Function Point	Measure not adjusted by the VAF
UGDG	Unspecified Generic Data Group	data type BFC
UGEP	Unspecified Generic Elementary Process	transactional BFC
VAF	Value Adjustment Factor	Value Adjustment Factor

0 BASIC MOTIVATIONS FOR A NEW FSMM

0.1 OVERVIEW OF EXISTING FUNCTIONAL SIZE MEASUREMENT METHODS

Up to the date of June 2011 there are two main functional size measurement methods for software that compete in the market: IFPUG and COSMIC. ISO has also accepted other 3 methods (Nesma, UKSMA and FISMA) as international standards but their circulation is limited if compared to the market leaders. All methods include the identification of BFCs at a very detailed level and the appreciation of complexity is based on a very granular detail of functional information. IFPUG method identifies either transactional BFCs (EI,EO,EQ) and permanent data BFCs (ILF,EIF), while COSMIC method identifies only transactional BFCs (Entry, eXit, Read, Write), although in order to identify them accurately it is also necessary to identify business objects that have a strong correspondence with data elements but does not contribute directly to the numerical value in CFP. Both methods have a detailed and extensive set of rules to identify BFCs and complexity of elementary processes.

0.2 ADVANTAGES AND DISADVANTAGES OF EXISTING FSMM

IFPUG method main Pros:

- It is consolidated by a several decades use
- Many benchmarking data are publicly available

IFPUG method main Cons:

- it requires a very detailed level of granularity for measurement data
- it provides a wealth of rules not always easily applicable
- it does not provide a layered model compatible with recent development architectures based on software components
- FP do not possess the distributive property of the calculations - FP(A union B) is generally less than FP(A) + FP(B) where A and B are two sets of requirements considered in the first case as a single application and in the second case as two separate applications, and this is because of the elimination rule for BFC which are identical within the same measurable software application (MSA), for example all the shared logical files.
- it induces uncertainty in the final value of a measurement exercise depending on interpretations of rules (multimedia issue, decoding tables etc.). Values may fluctuate from 5 to 20% or even more
- it is not usually considered appropriate to technological software
- it is relatively easy to use it for ex novo development projects but difficult to apply to ordinary maintenance, not well documented enhancement projects and for keeping asset management measures always updated.

COSMIC method main Pros:

- It is suitable for a wider range of types of software (business, real time networks ..)
- It introduces the concept of layers and layered architectures
- It is relatively independent by how the boundaries among software applications are set up

COSMIC method main Cons:

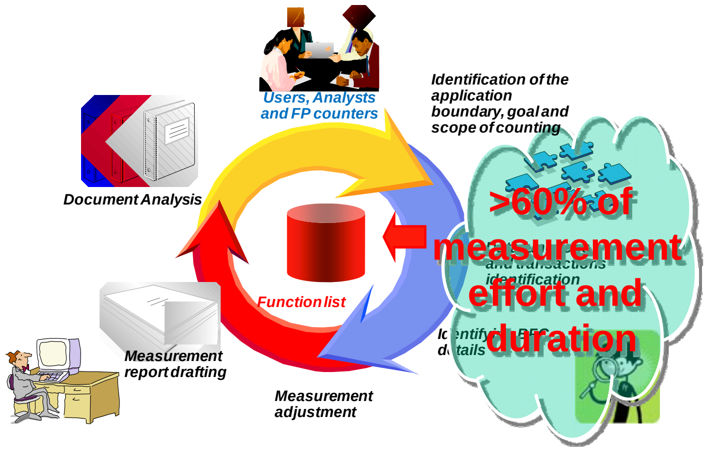
- it requires a very detailed level of granularity for measurement data
- it provides a wealth of rules not always easily applicable
- the numerical value depends strongly on the selected viewpoint and on the identification of business objects of interest

0.3 Shared trade offs of existing methods

Keeping a repository of baseline measures always up to date after the countless small evolutionary maintenance activities, that are part of the ordinary life, is virtually impossible. Small changes to the software are not reported at the baseline level and so the measures worsen and the idea that they are of little use to the business is spread.

Cost of measurement, although extremely low for any measurement instance with respect to the entire software lifecycle cost, is considered still too high by the managers especially if we consider the hidden costs of attracting, maintaining and improving specific competencies on functional measurement.

For IFPUG FSMM the cost of the BFC identification and complexity attribution is the most part of the measurement effort.



0.4 Market needs

Today's market requires fast, agile, very challenging size measurement methods with low impact on production processes, which require not too specialized skills, that are reliable in results, not dependent on counters opinions, technology and adequately related to effort, cost, duration of a software project.

The current FSMMs are only partially compliant with these needs.

1 SIMPLE FUNCTION POINT

1.1 RESEARCH PROJECT'S GOALS

To define a new Functional Size Measurement Method consistently with the framework of the ISO 14143 family of standards, totally numerically compatible with the IFPUG FP method when applied on the same object of measurement, but:

- easier to apply
- easier to learn
- less susceptible to different interpretations
- less susceptible to "manipulation" of measurements
- designed to allow an easier update of existing measurement assets
- designed to allow an immediate conversion of existing assets counted with the IFPUG method

1.2 BASIC FEATURES DESCRIPTION

The new Simple Function Point Metrics (SiFP) has a capability to measure the functional user requirements with the same precision as the standard IFPUG method at the application level (not at the single BFC level) being fully compatible with it in terms of numerical value: that is, the conversion ratio between a SiFP measure and a IFPUG FP measure is essentially equal to 1. This means that all the results of analysis conducted on the basis of the IFPUG measures can be used for SiFP as well, from productivity data collected in the ISBSG data base to the FP unitary prices on the exchange market, from the defect rates to asset valuations.

<u>Simple Function Point method is not a technique for estimating the IFPUG Function Point metrics but it is a functional alternative to it, easily convertible.</u>

1.3 THE MODEL BEHIND THE MEASURE

The SiFP metrics' underlying model coincides with the IFPUG 4.x model with respect to the general approach, the boundary, the scope, the elementary process and logical data definitions, its practices and formulas. It is different for the introduction of the layer concept (optional) and the use of only two BFCs known as:

- Unspecified Generic Elementary Process (UGEP)
- Unspecified Generic Data Group (UGDG)

The first one is a transactional object type while the second is a data object type. No more differentiation is needed among EI, EO, EQ, ILF and EIF, nor definition of primary intent or inherent complexity.

1.4 THE PROCESS OF MEASUREMENT

1.4.1 IDENTIFYNG THE OBJECT TO BE MEASURED - DEFINING BOUNDARIES

A Measurable Software Application (MSA) is defined as "an aggregate of specific functional features conceived from a user point of view, homogeneous in terms of level of logical abstraction."

The term "Measurable" is needed in order to differentiate this kind of aggregation from other aggregations used by ICT organizations for technical or managerial goals.

The identification of any MSA is required to identify and eliminate from its measure all the identical

BFCs. Within a specific MSA, indeed, one must not count the same BFCs appearing in several places of the functional specifications. They must be counted - if appearing in different MSA – within each MSA. The boundaries setting is also important for the inter-MSA communication.

To find the boundary of a MSA one should aggregate data and functionalities based on the presence of organizational, functional and semantic affinity. The identification of the application boundaries should follow the principles of structured design of software known as: minimizing coupling and maximizing cohesion. In other words, the functional and operational interdependencies between different MSA should be minimal or nil. While inside a MSA there should not be parts that are not completely autonomous and independent from the operational and logical point of view. The "container"-type MSA, in which the different functions are characterized by the fact that they cannot be elsewhere, should be reduced to minimum.

1.4.2 IDENTIFYNG THE LAYERS

Current software architectures are characterized by the distribution of data and processing logic components among separate and cooperating technological platforms. The execution of a process is often implemented dynamically in the most appropriate element of the architecture at any given time. This organization allows to reuse generalized components (often called services), through standardization and specialization, in order to construct any new transaction. Models that describe these architectures use the concept of layers which is a way to aggregate those components on the basis of homogeneity of representation and usage. A layer is linked, therefore, to a certain level of abstraction in the representation of data and related functions, this, in turn, determine a different concept of user associated with that particular layer.
Any Software Application can be distributed across multiple layers, each one containing generalized software (technical or business) components designed to give specific support to the implementation of reusable and specific functional or non-functional requirements. A MSA may only belong to one and one layer so the measurement is consistent in terms of level of abstraction and it does not depend on the internal modular organization but only on the external functional User requirements.

1.4.3 DEFINING THE GOAL OF MEASUREMENT

This aspect influences the scope of the measurement.

1.4.4 ESTABLISHING THE SCOPE OF MEASUREMENT

The scope of measurement consists of all those "parts" of one or more MSA whose measurement gives an answer to the established goal.

1.4.5 MAPPING THE FUNCTIONAL REQUIREMENTS

The measurement process provides a mean for the creation of a mapping of functional requirements on a SiFP abstract model of Functional Requirements for their measurement. The mapping involves the identification of an architectural reference model and the application of specific Guidelines, which contextualize the mapping in specific areas of use.

1.4.6 IDENTIFYNG BASE FUNCTIONAL COMPONENT (BFC)

Functional requirements must be identified in the BFC, which is the undifferentiated elementary processes and the undifferentiated logical entities.

1.4.6.1 UNSPECIFIED GENERIC DATA GROUP

These are data type elements that represent business objects of interest for a particular set of functional users. They correspond to ILF and EIF of the IFPUG method.

1.4.6.2 *UNSPECIFIED GENERIC ELEMENTARY PROCESS*

These are elements that represent automated elementary transactional capabilities for the user. The definition of an elementary process is similar to IFPUG and COSMIC approach. They correspond to the EI, EO, EQ of the IFPUG method.

1.4.7 BFC WEIGHTS

The values associated with each BFC are the following:

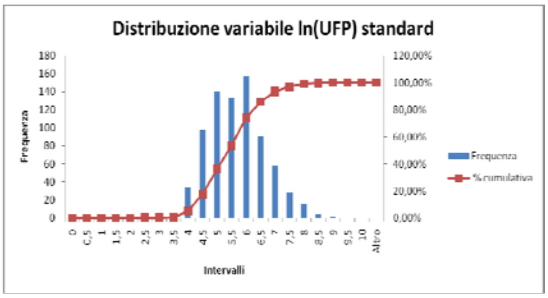
UGEP = 4,6 SiFP
UGDG = 7,0 SiFP

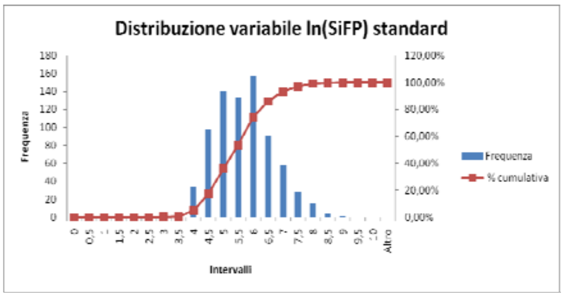
1.4.8 CALCULATION FORMULAS

The calculation formulas are similar to those from IFPUG, except for the measurement of assets, due to the invariance of the complexity of the BFC calculated before and after an enhancement maintenance project. To assure compliance with ISO 14143 guidelines, SiFP method does not introduce any adjustment factor like the IFPUG VAF.

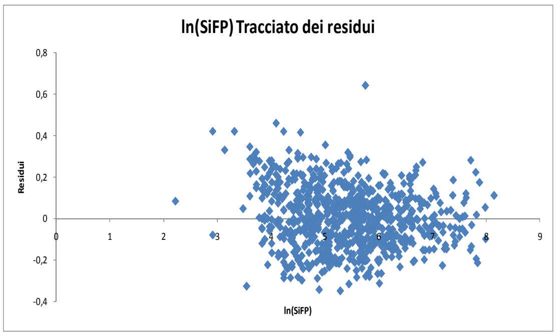
1.5 CONVERSION BETWEEN IFPUG FP AND SIFP

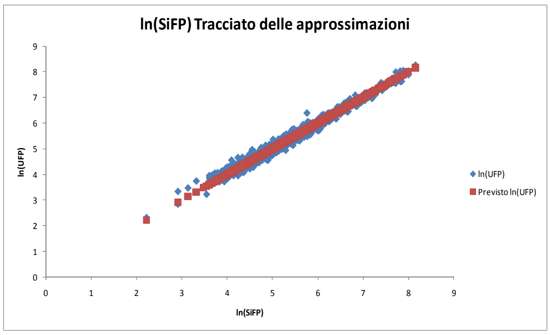
To check the convertibility of the IFPUG FP to a SiFP measure a sample of 766 ISBSG points has been used for which ISBSG provided the details of counting in terms of BFCs. The distribution of the FP and SIFP values (after log transformation) is very close to the normal distribution allowing to apply the tools of statistical analysis.

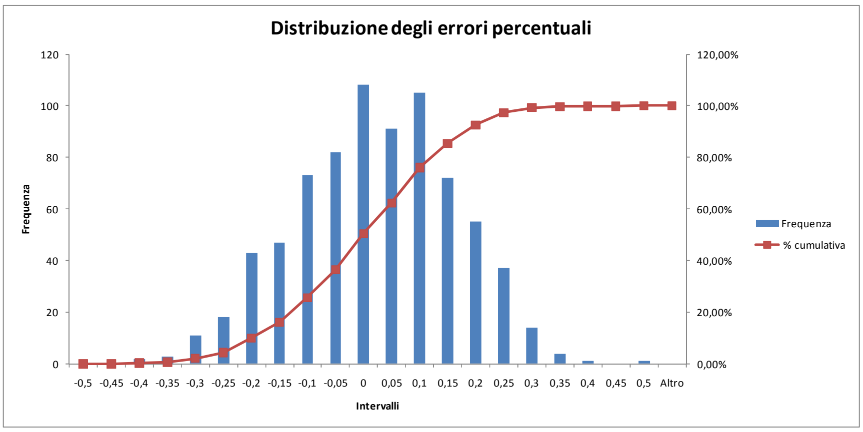




The linear regression on logarithmic transformed data (UFP vs. SiFP) indicates a coefficient of exchange rate of 0,9988 that's to say, practically: 1 UFP correspond to 1 SiFP with a corrected correlation index equals to 0.9776. The single minimum difference (in absolute value) is 0%, the maximum is 47%, the average is 12%, the median is 10%. For 80% of the projects the difference is below 19%. This result indicates that the two metrics are nearly coincident. The residue analysis is regular and distributed normally.







The asset difference - or the difference between the sum of all measures with the sign made with the IFPUG method and the sum of all measures with the sign made using the SiFP method - is equal to -1123 FP out of 284'005 FP corresponding to -0,4%, which means that the absolute errors are compensated by combining the counts as if they were belonging to a large application portfolio. In fact, the overall SiFP FP measurement is 1123 less than the IFPUG value on a total of 284'005 UFP!

An independent audit was also conducted on a sample of 140 other projects not included in the ISBSG DB providing very similar results.

The transition from assets counted by IFPUG FP into SiFP values is immediate if one knows only the number of EI, EO, EQ, ILF and EIF.

1.6 Size / Effort Correlation

Functional size is usually considered useful for contributing significantly to determine effort, duration, cost, staff of a new development or functional maintenance of a software application.

Until now, the ICT community has accepted an implicit assumption like the following:

> To obtain an acceptable statistical correlation between effort - required to develop or functionally enhance software applications - and functional size it is necessary to consider the numerosity and the internal complexity of BFC.

Studies conducted by DPO on a sample of about 800 projects, counted with the IFPUG method have shown that this assumption, at least in the context of this methodology, is not true.

It's true the following statement, instead:

> The accuracy of a model of correlation between actual effort and the software functional size does not decrease when considering only the number of BFC in each of the two classes (data or transaction).

This finding makes the whole system of rules aimed at the differentiation between IFPUG EI, EO, EQ, ILF and EIF and at the determination of the complexity of the single BFC (DET, RET, FTR), useless.

The consequences of this finding are truly extraordinary in terms of impact on the method and process of measurement of Function Points.

Using only the number of BFC, however, does not allow an immediate use of all models and results obtained by the application of the IFPUG method. The research, therefore, had also, as an essential goal, to find a weight for the new adopted BFC that would have made the two metrics (IFPUG FP and SiFP) reliably and immediately convertible. Of course due to the different level of detail the conversion is one way only. We can convert EI,EO,EQ,ILF and EIF into UGEP and UGDG but not the opposite.

The correlation between effort and SiFP is identical to that between IFPUG UFP and effort for development and for the enhancement maintenance. The two measures can be used interchangeably in the determination of cost models and with the same market unitary prices.

With respect to effort measured in working hours, the two logarithmic models are related to each other as y = 0,9915x + 0,0476 with correlation coefficient R2 = 0.9777. The single minimum difference (in absolute value) is 0%, the maximum is 37%, the average is 8%, the median is 7%. For 90% of the projects the difference is below 17%. The final difference of a portfolio of work hours is -51'364 hours out of 2'437'087 equals to -2%.

IFPUG method and SiFP method have the same level of (im)precision in estimating the project effort and then, for any specific intervention it is not correct to think that the "true" value of effort is the one deriving from IFPUG based model. Actually in around half of cases (46%) is the SiFP value to be closer to the true effort value reported, so pulling out a project at random from the 766 surveyed by ISBSG there is a nearly identical probability that the final effort is closer to that estimated by the SiFP method compared to the IFPUG one. The cost model build on the SiFP and IFPUG method have the same reliability.

The table below shows that even the maximum error committed (the absolute difference between the percentage of working hours and the actual costs estimated by the two methods) is less in the case of SiFP based model. 85% of the projects has a error of less than 157% against 167% of the IFPUG based model.

min	0%	0%
average	107%	106%
median		
max	3166%	2822%
percentile threshold	85%	85%
percentile	167%	157%

By calculating an exchange rate based on the FP value for effort, therefore, an average error of 107% is committed with the IFPUG based model and of 106% with the SiFP based model. There is no reason, therefore, for which we should consider the IFPUG derived value in a specific case more reliable than the one estimated by the SiFP based model.

By replacing IFPUG method with the SiFP method at the portfolio level (i.e. on a number of projects greater than 30-40) any specific differences are netted and economic valuations are practically identical.

Replacing the IFPUG asset value with a SiFP value, maintenance fees and any reports to shareholders may remain identical (absolute difference in asset <0.4%)

1.7 CONCLUSIONS

SiFP method meets all the objectives set for the research project. A SiFP measure is:

- easier to apply
- easier to learn
- less susceptible to different interpretations
- less susceptible to "manipulation" of measurements
- designed to allow an easier update of existing measurement assets
- designed to allow an immediate conversion of existing assets counted with the IFPUG method

In particular, we observe that the alignment of asset value as a result of the continuing operations of small functional enhancements conducted within the routine maintenance involves a very low cost because there is no need to document DET, RET and FTR changed but only added or deleted BFC from the baseline .