Published in: Information and Software Technology
This version of the article has been accepted for publication, after peer review and is subject to Elsevier terms of use. The Version of Record is available online at: http://dx.doi.org/10.1016/j.infsof.2017.02.005

A Study on the Statistical Convertibility of IFPUG Function Point, COSMIC Function Point and Simple Function Point

Abedallah Zaid Abualkishik

American University in the Emirates

Filomena Ferrucci

Università degli Studi di Salerno

Carmine Gravino

Università degli Studi di Salerno

Luigi Lavazza

Università degli Studi dell'Insubria

Geng Liu

Hangzhou Dianzi University

Roberto Meli

DPO

Gabriela Robiolo

Universidad Austral

Abstract

Background Several functional size measurement methods have been proposed. A few ones –like IFPUG and COSMIC methods– are widely used, while others –like Simple Function Points method– are interesting new proposals, which promise to deliver functional size measures via a faster and cheaper measurement process.

Objectives Since all functional size measurement methods address the measurement of the same property of software (namely, the size of functional specifications), it is expected that measures provided in a given measurement unit can be converted into a different measurement unit. In this paper, convertibility of IFPUG Function Points, COSMIC Function Points, and Simple Function Points is studied.

Method Convertibility is analyzed statistically via regression techniques. Seven datasets, each one containing measures of a set of software applications expressed in IFPUG Function Points, COSMIC Function Points and Simple Function Points, were analyzed. The components of functional size measures (usually known as Base Functional Components) were also involved in the analysis.

Results All the analyzed measures appear well correlated to each other. Statistically significant quantitative models were found for all the combinations of measures, for all the analyzed datasets. Several models involving Base Functional Components were found as well.

Conclusions From a practical point of view, the paper shows that converting measures from a given functional size unit into another one is viable. The magnitude of the conversion errors is reported, so that practitioners can evaluate if the expected conversion error is acceptable for their specific purposes. From a conceptual point of view, the paper shows that Base Functional Components of a given method can be used to estimate measures expressed in a different measurement unit: this seems to imply that different functional size measurement methods are 'structurally' strongly correlated.

Keywords: Functional Size Measurement, IFPUG Function Points, COSMIC Function Points, Simple Function Point, Convertibility, Base Functional Components (BFC)

---

1. Introduction

Functional Size Measurement (FSM) aims at providing a measure of the size of functional user requirements (FUR). Several FSM methods have been proposed [1, 2, 3, 4, 5, 6]. A few of the proposed FSM methods, like the IFPUG method [1, 7] and the COSMIC method [2, 8] are widely used, especially as a basis for estimating software development effort. However, applying the IFPUG and COSMIC methods is relatively time- and effort-consuming. In particular, the need to analyze in detail every elementary (in IFPUG terminology) or functional (in COSMIC terminology) process can easily require a sizeable amount of work. Consequently, approximate estimation methods (AEM's) have been proposed for both IFPUG [9] and COSMIC methods [10, 11, 12, 13].

AEM's provide an estimate of the functional size of given FUR based on a subset of the elements of the FUR that should be considered to carry out the full-fledged measurement. In practice, AEM's yield approximate estimates of functional size (with some estimation error) at a fraction of the measurement cost [14, 15, 16].

The Simple Function Points (SiFP) method [6] aims at providing functional size measures at a much smaller cost than IFPUG or COSMIC methods. Unlike AEM's, the SiFP method provides real measures, rather than estimates. Therefore, this method is potentially quite interesting for practitioners, given the little measurement cost it involves.

The situation sketched above suggests two research activities:

- Since all FSM methods aim at measuring the same property of software (i.e., functional size), it is expected that a measure expressed in a measurement unit can be converted into a measure expressed in another measurement unit. In fact, convertibility is explicitly mentioned in the ISO standard that specifies the required features of functional size measurement [17].

  Convertibility is useful also for very practical reasons. For instance, an organization that is willing to move from an FSM method to another one needs to convert historical data into the new functional size measurement unit. Similarly, to merge historical datasets that include measures performed according to different FSM methods, some sort of conversion is needed, to make the resulting dataset homogeneous.

- Most AEM's use a subset of the data required to perform standard FSM measurement. Accordingly, studying the relationship between standard measures and the data used by AEM's is interesting to determine which AEM's are feasible and what level of accuracy can be expected by every AEM. Moreover, studying the relationship between the Base Functional Components (BFC's) of a method and another method's measure provides useful indications concerning the structural similarity among FSM methods.

In this paper we analyze the convertibility of IFPUG Funtion Points, COSMIC Funtion Points and SiFP Funtion Points. Other FSM methods are excluded from the study for two reasons: 1) some methods are not very popular, therefore there is little interest in their convertibility; 2) to carry out convertibility analysis we need one or more datasets in which every software application has been measured according to different FSM methods: we were able to find datasets containing IFPUG, COSMIC and SiFP measures, but not other types of measures.

In general, there are multiple procedures to convert a measure from a given measurement unit into another measurement unit. However, the type of datasets that were available for this study supported only statistical convertibility.

The statistical convertibility between functional size measures has been widely studied, as discussed in Section 10. This paper contributes to improving the knowledge concerning convertibility among functional size measures in multiple respects:

- In addition to IFPUG Function Points and COSMIC Function Points, SiFP Function Points are studied; this involves that conversion between COSMIC Function Points and SiFP Function Points (which was never addressed before) is also studied; similarly, the relationship between SiFP Function Points and IFPUG elements like File Type Referenced (FTR) had never been studied before.
- A systematic and comprehensive study of the relationships between measures and their components is performed. This can be very beneficial to understand the strengths and weaknesses of AEM's. In the paper, we

study some relationships (like the one between COSMIC Function Points and data movements, presented in Section 7.3) that had never been published before.

- The correlation of a measure with the elements of other measures is studied, to get insights into possible alternative AEM's methods. Also in this case, some relationships studied in the paper had never been explored before (see for instance the relationships based on FTR's in Section 7.6).

- The study is based on seven datasets. This allows to spot commonalities and variations in the convertibility models. Even more important is that analyzing seven independent datasets and finding that in all cases a statistically significant model can be found increases the confidence that a correlation between the considered measures actually holds.

The paper also improves our capacity to address practical problems, as shown in Section 4.2, where two usage scenarios are described.

The paper is structured as follows: Section 2 briefly describes the three FSM methods considered in this paper. In Section 3 the datasets used for the analysis are described. The research reported in the paper is described in Section 4: Section 4.1 describes the focus of the research, Section 4.2 illustrates usage scenarios, and Section 4.3 illustrates the research method used. Section 5 reports about the correlations we found between SiFP Function Points and IFPUG Function Points on one side and COSMIC Function Points on the other side. Section 6 reports about the correlations found between SiFP Function Points and IFPUG Function Points. Section 7 reports about the correlations that we found among the BFC's of the considered measures. The results of the analyses are discussed in Section 8. Section 9 discusses the threats to the validity of the study. Related work is accounted for in Section 10. Section 11 draws the conclusion and outlines future work.

2. Functional Size Measurement Methods

In this section a brief introduction to the FSM methods considered in this paper is given. Readers are referred to the official documentation for further details [1, 2, 8, 7, 6].

2.1. A Case Study

Throughout this section, we use a slightly modified version of the Warehouse management software (WMS) by Fetcke, as an example. The detailed specifications can be found in the paper by Fetcke, which is available on-line [18].

The WMS is used by a company that operates several warehouses, where customers' goods are stored. Customers can deposit items into storage locations in the warehouse. After the items have been kept in the warehouse for some period of time, they can be retrieved by their owners. The customers get billed for the storage service.

The Entity/Relationship diagram representing the entities involved in the WMS is given in Figure 1. The entities and their attributes are described in Figure 2. Both figures are from [18]. Attributes Owner and Storage place are references to entities Customer and Place, respectively.

Figure 1: Entity/Relationship diagram of the WMS [18].

The WMS allows the user to perform several operations, such as adding a new customer, deposit an item, receive payment, print the customer item list, and many others. Here we report the specifications of the *Add customer*, *Delete customer* and *Change customer data* operations, which will be used to illustrate the functional measurement methods. The complete functional requirements of the WMS can be found in [18].

The *Add customer* operation adds an instance of Customer data to the system's repository. The attributes Name and Address have to be given. The

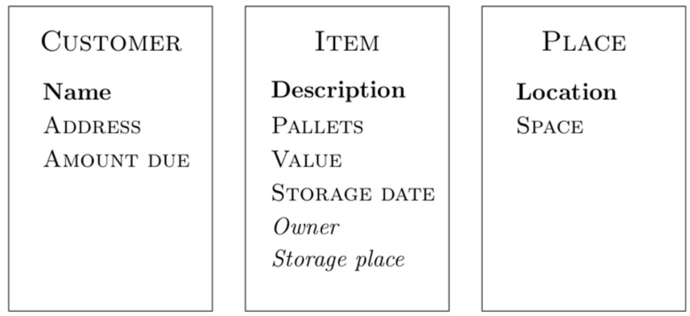

Figure 2: Entities of the WMS [18].

Amount Due is initialized to zero. If an instance of Customer with the Name entered already exists, no new instance of Customer is created, the repository is not changed, and an error message is displayed.

The *Delete customer* operation removes an instance of Customer data from the system's repository, given the attribute Name of the customer. Customer data are removed if the Amount Due attribute is zero and the customer does not own any stored Items. An error message is displayed, if the record cannot be removed or if there is no instance of Customer with the given name.

The *Query customer's items* operation is used to visualize the stored items that a given customer owns. The user enters the name of the customer. If that customer exists, an output screen lists the description, pallets, value and storage date of all the items the customer owns. Otherwise, an error message is displayed.

To ease functional measurement, the main actions involved in the operations described above can be summarized as done in Table 1. Many notations can be used to describe functional specifications and ease measurement; for instance, a representation of the *Add Customer* operation via UML sequence diagrams can be found in [19].

In our case study, we assume that the list of available storage places is created and managed by another application: the WMS just reads the list of available places, but cannot modify it.

Table 1: Characteristics of WMS operations that are relevant for functional measurement.

Operation	Action	Entity	Attribute
Add Customer	Input	Customer	Name, Address
	Read	Customer	Name
	Create	Customer	Name, Address, Amount due
	Output	Message	(message content)
Delete Customer	Input	Customer	Name
	Read	Customer	Name, Amount due
	Read	Item	Owner
	Delete	Customer	(object)
	Output	Message	(message content)
Query Customer's Items	Input	Customer	Name
	Read	Customer	Name
	Read	Item	Description, Pallets, Value, Storage date, Owner
	Output	Item	Description, Pallets, Value, Storage date
	Output	Message	(message content)

2.2. The IFPUG method

Function Point Analysis (FPA) was originally introduced by Albrecht to measure the size of data-processing systems from the end-user's point of view, with the goal of estimating the development effort [20].

The initial interest sparked by FPA along with the recognition of the need for improvement in its counting practices led to founding the IFPUG (International Function Points User Group).

The IFPUG provides guidelines for measuring [1], makes measurement rules evolve along with the evolution in software technologies, and oversees the standardization of the measurement method (http://www.ifpug.org/).

The IFPUG method is now an ISO standard [7] in its "unadjusted" version. So, throughout the paper we refer exclusively to Unadjusted Function Point (UFP) version, even when we talk generically about IFPUG Function Points or IFPUG FP's. The conversion of adjusted Function Points is not considered at all, because adjusted measures account for factors –dealing with the software product or process– that are not considered by COSMIC and SiFP methods.

Albrecht's basic idea –which is still at the basis of the IFPUG method– is that the "amount of functionality" released to the user can be evaluated by taking into account the data used by the application to provide the required functions, and the transactions (i.e., operations that involve data crossing the boundaries of the application) through which the functionality is delivered to the user. Both data and transactions are evaluated at the conceptual level, i.e., they represent data and operations that are relevant to the user. Therefore, IFPUG Function Points are counted on the basis of the user requirements specification. The boundary indicates the border between the application being measured and the external applications and user domain.

FUR's are modeled as a set of BFC's, which are considered the elementary units of FUR's. Each of the identified BFC's is then measured; finally, the size of the whole application is obtained as the sum of the sizes of BFC's.

IFPUG BFC's are data functions, which are classified into internal logical files (ILF) and external interface files (EIF), and elementary processes (EP), also known as transactional functions, which are classified into external inputs (EI), external outputs (EO), and external inquiries (EQ) according to the activities carried out within the process and its main intent. Each function, whether a data or transactional one, contributes a number of Function Points that depends on its "complexity." Each function is weighted on the basis of its complexity according to given tables.

Weights for ILF's are defined as $w_{ILF}(f) = tab_{ILF}(f.RET, f.DET)$. That is, the weight of f is given by a table, and the entries to be used are the RET's (Record Element Types), which indicate how many types of information (e.g., sub-classes, in object-oriented terms) can be contained in the given ILF, and DET's (Data Element Types), which indicate how many types of elementary information (e.g., attributes, in object-oriented terms) can be contained in the given ILF. Weights for EIF's are defined in the same way, but via a different table tab_{EIF}.

Weights for EI's are defined as $w_{EI}(f) = tab_{EI}(f.FTR, f.DET_{I/O})$. That is, the weight of f is given by a table, and the entries to be used are the number of FTR's of f –i.e., the number of types of logical data files used while performing the required operation– and the number of DET's, i.e., the number of types of elementary data that f sends and receives across the boundaries of the application. The weights for external outputs and queries are similarly defined, via tables tab_{EO} and tab_{EQ}.

Finally, the size expressed in Unadjusted Function Points (UFP's) is obtained by summing the contribution of data and transaction functions, as shown in formula (1).

$$Size_{UFP} = \sum_{f \in ILFs} w_{ILF}(f) + \sum_{f \in EIFs} w_{EIF}(f) + \sum_{f \in EIs} w_{EI}(f) + \sum_{f \in EOs} w_{EO}(f) + \sum_{f \in EQs} w_{EQ}(f) \quad (1)$$

In formula (1), $ILFs$ denotes the set of all ILF's, $EIFs$ denotes the set of all EIF's, etc. $w_{FT}(f)$ indicates the weight assigned to a data or transaction function f according to the function type FT (i.e., EI, EO, EQ, ILF or EIF).

To clarify how the IFPUG method works, we apply it to the fragment of the WMS described in Section 2.1. However, we warn the reader that the description provided here is not sufficient to learn all the details of the measurement. Further details about the IFPUG method can be found in the manual [1, 7].

The logical data files that are maintained or used by the application are the three entities described in Figures 1 and 2.

To measure the size of each data file we need to consider its characteristics, as follows.

The *Customer* is an internal logical file, because it is created, modified, read and possibly deleted by the WMS application. It contains just one type of data, that is, all customers are characterized by the same set of data elements; thus, the *Customer* ILF has one RET. The elementary data types that characterize the RET are the attributes shown in Figure 2, i.e., *Name*, *Address* and *Amount Due*; thus, the the *Customer* ILF has three DET's. According to IFPUG tables, an ILF having one RET and three DET's is a low complexity one, and contributes 7 UFP to the application's size measure.

The *Item* is an internal logical file, because it is created, modified, read and possibly deleted by the WMS application. It contains just one type of data, that is, all items are characterized by the same set of data elements; thus, the *Item* ILF has one RET. The elementary data types that characterize the RET are the attributes shown in Figure 2, i.e., *Description*, *Pallets*, *Value*, *Storage date*, *Owner* and *Storage place*; thus, the *Item* ILF has six DET's. According to IFPUG tables, an ILF having one RET and six DET's is a low complexity one, and contributes 7 UFP to the application's size measure.

The *Place* is an external interface file, because it is created and modified by an external application. It contains just one type of data, that is, all places are characterized by the same set of data elements; thus, the *Place* EIF has one RET. The elementary data types that characterize the RET are the attributes shown in Figure 2, i.e., *Location* and *Space*; thus, the *Place* EIF has two DET's. According to IFPUG tables, an EIF having one RET and two DET's is a low complexity one, and contributes 5 UFP to the application's size measure.

Using the terminology of formula (1) we count: $w_{ILF}(Customer) = 7$ UFP, $w_{ILF}(Item) = 7$ UFP and $w_{EIF}(Place) = 5$ UFP.

The main intent of the *Add customer* operation is to update the set of customers, therefore it is an EI. According to the specifications, *Add customer* reads and creates instances of *Customer*. Therefore, it involves only one FTR. To perform the *Add customer* operation, the user supplies the Name and Address of the new customer. The system replies with a confirmation or diagnostic message. Hence, we have three DET's (name, address and message) that cross the application's boundaries, plus the invocation of the function. According to IFPUG tables, an EI having one FTR and four DET's is a low complexity one, and contributes 3 UFP to the application's size measure.

The main intent of the *Delete customer* operation is to update the set of customers, therefore it is an EI. According to the specifications, the *Delete customer* reads and deletes instance of *Customer*, but before removing a customer it has to check that no item of that customer is still stored in the warehouse, so it also need to read *Item* data. Therefore, we have two FTR's (*Customer* and *Item*). The user supplies the name of the customer to be removed, and the system issues the usual confirmation or diagnostic message; hence, we have 2 DET's crossing the applications' boundaries (the name and the message), plus the invocation of the function. According to IFPUG tables, an EI having two FTR's and three DET's is a low complexity one, and contributes 3 UFP to the application's size measure.

The main intent of the *Query customer's items* operation is to display information that is stored in the system, without any specific processing; accordingly, it is an EQ. According to the specifications, the *Query customer's items* reads *Customer* data to check that the given name actually identifies an existing customer, then reads the *Item* data to be displayed. Therefore, we have two FTR's (*Customer* and *Item*). The user supplies the name of the customer whose items' data have to be shown, and the system either displays the required information (Description, Pallets, Value and Storage date) or issues a diagnostic message; hence, we have 6 DET's crossing the applications' boundaries (the Name Description, Pallets, Value and Storage date attributes and the message), plus the invocation of the function. According to IFPUG tables, an EQ having two FTR's and seven DET's is an average complexity one, and contributes 4 UFP to the application's size measure.

So, we have: $w_{EI}(Add\ Customer) = 3$, $w_{EI}(Delete\ Customer) = 3$ UFP and $w_{EQ}(Query\ Customer's\ Items) = 4$ UFP.

The results of the counting are given in Table 2. So, if the WMS would involve only the operations examined here, its size would be 29 UFP.

2.3. COSMIC

The COSMIC method assumes a model of software in which FUR's are mapped into unique functional processes, initiated by functional users. The concept of functional process in COSMIC is practically coincident with the concept of elementary process (or transaction) in the IFPUG method. Each

Table 2: Summary of WMS size measurement in UFP.

Function	Type	Complexity	Size [UFP]
Customer	ILF	Low	7
Item	ILF	Low	7
Place	EIF	Low	5
Add Customer	EI	Low	3
Delete Customer	EI	Low	3
Display Customer's Items	EIQ	Average	4
Total	–	–	29

functional process consists of sub-processes that involve data movements. A data movement concerns a single persistent data group (DG) type. A data group is defined as "a distinct, nonempty, non-ordered, and non-redundant set of data attributes where each included data attribute describes a complementary aspect of the same object of interest." A data group is considered persistent if its value is stable between two consecutive functional process executions. Data movements are classified into Entry and Exit (i.e., I/O movements) and Read and Write (to persistent storage), that are defined as follows:

- An Entry (E) moves a data group from a functional user across the boundary into the functional process where it is required.
- An Exit (X) moves a data group from a functional process across the boundary to the functional user that requires it.
- A Read (R) moves a data group from persistent storage within each of the functional process that requires it.
- A Write (W) moves a data group lying inside a functional process to persistent storage.

Each data movement (i.e., Entry, Exit, Read, or Write) is counted as 1 COSMIC Function Point (CFP).

The size of a software application in CFP is the sum of the sizes of its functional processes. The size of each functional process is the number of involved data movements.

$$Size_{CFP} = \sum_{f \in FPr} (entries(f) + exits(f) + reads(f) + writes(f)) \qquad (2)$$

where FPr is the set of all functional processes.

To clarify how the COSMIC measurement process works, we apply it to the fragment of the WMS described in Section 2.1. However, we warn the reader that the description of the COSMIC method provided here is not sufficient to learn all the details of the COSMIC method. Further details on the COSMIC method can be found in the manuals [2, 8].

The data group types involved in the application are the *Customer*, *Item* and *Place*, as evident from Figure 1. In the COSMIC method, data groups do not contribute directly to the size measure, but they need to be clearly identified, since they are the object of data movements.

The *Add customer* operation is a functional process that involves the following movements: a *Customer* Entry (the name of customer to be added); the *Customer* Read (to check if the customer is already registered in the system); the *Customer* Write (when it is created), the message Exit. The functional process involves 4 data movements, hence it contributes 4 CFP to the size of the application.

The *Delete customer* operation is a functional process that involves the following movements: a *Customer* Entry (the name of the customer to be deleted), the *Customer* Read (to check if the customer is registered), the *Item* Read (to check that no items of the customer are still stored in the warehouse), the *Customer* Write (when it is deleted), the message Exit. The functional process involves 5 data movements, hence it contributes 5 CFP to the size of the application.

The *Query customer's items* operation is a functional process that involves the following movements: a *Customer* Entry (the name of the customer), the *Customer* Read (to check if the customer is registered), the *Item* Read (to retrieve the data to be displayed), the *Item* Exit (to display the required data), the message Exit. The functional process involves 5 data movements, hence it contributes 5 CFP to the size of the application.

The results of the counting are given in Table 3. So, if the WMS would involve only the operations examined here, its size would be 14 CFP.

Table 3: Summary of WMS size measurement in terms of CFP.

Functional process	Movements	Size [CFP]
Add Customer	*Customer* Entry, *Customer* Read, *Customer* Write, message Exit	4
Delete Customer	*Customer* Entry, *Customer* Read, *Customer* Write, *Item* Read, message Exit	5
Display Customer's Items	*Customer* Entry, *Customer* Read, *Item* Read, *Item* Exit, message Exit	5
Total		14

2.4. The Simple Function Point (SiFP) method

The idea that effective functional size measures can be based on the analysis of just a subset of the elements of the IFPUG software model is at the base of the definition of the SiFP method.

The SiFP method was defined by Meli [21] and subsequently published by the Simple Function Point Association in an official Reference Manual, which is available in the public domain [6]. The SiFP method was defined based on the observation –derived from the experience with IFPUG method– that to get a measure of the functional size of an application

1) it is not necessary to identify several types of transactions (classified according to the primary intent and involved activites) and files (classified as internal or external);
2) a notion of complexity –based on the details of data and transactions (see Section 2.2)– is not relevant to the goal of representing functional size and of estimating effort or costs.

So, the SiFP method adopts a model of the software to be measured that is greatly simplified with respect to the model used by IFPUG. In fact, the SiFP method defines only two BFC's, known as:

1. Unspecified Generic Elementary Process (UGEP). This element corresponds to the IFPUG concept of elementary process and to the COSMIC concept of functional process. It is named "unspecified" since it is not classified as input, output or query. It is named "generic" since it is not differentiated in terms of internal complexity.
2. Unspecified Generic Data Group (UGDG). This element corresponds to the IFPUG concept of logical data file and to the COSMIC concept of data group. It is named "unspecified" since it is not classified as internal or external. It is named "generic" since it is not differentiated in terms of internal complexity.

Therefore, the SiFP model of software is a proper subset of the IFPUG model of software: all the elements of the SiFP model appear in the IFPUG model as well, while the opposite is clearly not true: for instance, there is no notion of FTR in the SiFP model.

The IFPUG method requires analyzing the details of logical data functions and transaction functions, to determine their complexity, hence their size. The SiFP method does not require this activity, therefore applying the SiFP method is substantially cheaper than applying the IFPUG method: a survey of experts' opinions showed that savings around 30% can be expected [16].

The size of a software application –expressed in SiFP– is

$$Size_{SiFP} = 4.6 \times \#UGEP + 7 \times \#UGDG \tag{3}$$

where #UGEP is the number of UGEP's and #UGDG's is the number of UGDG's. In practice, all UGDG's (logical data) are assumed to be of weight (i.e., size) 7 SiFP, while all UGEP's (elementary processes) are assumed to be of weight (i.e., size) 4.6 SiFP.

Note that, by definition, $\#UGEP = \#EI + \#EO + \#EQ$ and $\#UGDG = \#ILF + \#EIF$. Therefore, to convert FP measures into SiFP measures, one can simply compute $SiFP = 4.6\ (\#EI + \#EO + \#EQ) + 7\ (\#ILF + \#EIF)$, if the list of transacton and data functions was recorded. The statistical models described in this paper are useful for legacy data, when no measurement documentation is available, so that you just know the size in UFP (i.e., in IFPUG Function Points) of a set of applications.

It is important to highlight that the SiFP method defines *a new autonomous measurement unit*, which can be adopted and used independently from the IFPUG method. SiFP measures are not intended to be used where IFPUG measures are expected, e.g., in IFPUG Function Point-based models of development effort. Instead, SiFP-based models (i.e., models that adopt SiFP Function Points as independent variables) can be built so that SiFP Function Points can be used directly for effort estimation (like in [22, 23]). Similarly, SiFP Function Points can be used instead of IFPUG Function Points for various purposes, besides effort estimation: for instance, one could measure defect density as the number of bugs per SiFP.

If the SiFP method were simply an estimation method for IFPUG Function Points, it would be implicitly assumed that one SiFP equals one UFP. Instead, since SiFP is a new measurement unit, a study of convertibility between SiFP and other functional size measurement units is needed.

To clarify how the SiFP method works, we apply it to the fragment of the WMS described in Section 2.1. Actually, the counting is very simple: we just have to note that we have three UGDG's (the *Customer*, *Item* and *Place*) and three UGEP's (*Add Customer*, *Delete Customer* and *Display Customer's Items*). So, if the WMS would involve only the operations examined here, its size would be $3 \times 7 + 3 \times 4.6 = 34.8$ SiFP. The details of the counting are given in Table 4.

3. The datasets used in this study

Seven datasets were analyzed. The list of datasets is given in Table 5, while the descriptive statistics of those datasets are given in Table 6.

Table 4: Summary of WMS size measurement in SiFP.

BFC	Type	Size [SiFP]
Customer	UGDG	7
Item	UGDG	7
Place	UGDG	7
Add Customer	UGEP	4.6
Delete Customer	UGEP	4.6
Display Customer's Items	UGEP	4.6
Total	–	34.8

Table 5: The datasets used in this study.

ID	dataset name	reference
1	Desharnais	[24, 25]
2	Abualkishik	[26]
3	Robiolo	[27]
4	Liu	[19]
5	van Heeringen	[28]
6	Cuadrado Gallego	[29]
7	Ferrucci	[30]

3.1. The dataset by Desharnais

This dataset was originally published by Desharnais et al. [24] and then further studied in [25]. The dataset consists of 14 industrial Management Information Systems (MIS) that belong to a data processing group of a governmental agency in Canada. All the applications are lying within a single layer. The main activity of the agency involves developing and maintaining various types of business application and MIS. The applications were developed using COBOL, C and Java between 2003 and 2008. The Fetcke case [18] study was added to this dataset to form a dataset of 14 applications.

3.2. The dataset by Abualkishik

The dataset consists of 13 Real-Time applications that have been collected from different resources [26]. Three applications are from an experimental application of COSMIC method by Khelifi [31]. Another application is the traffic

Table 6: Descriptive statistics

Dataset	n	SiFP				IFPUG FP				CFP			
		Mean	Stdev	Median	Range	Mean	Stdev	Median	Range	Mean	Stdev	Median	Range
1	14	304	156	323	(95,659)	310	164	330	(77,646)	308	169	296	(81,579)
2	12	110	52	109	(32,208)	95	39	97	(37,161)	103.5	60	83	(36,222)
3	22	191	151	125	(46,698)	170	133	115	(40,623)	95	58	81	(46,295)
4	15	134	33	127	(86,185)	114	29	107	(73,163)	93	28	86	(50,154)
5	25	528	357	403	(67,1569)	499	354	412	(61,1622)	545	441	445	(66,1864)
6	33	326	114	346	(95,538)	291	99	315	(78,462)	209	75	215	(65,313)
7	25	490	265	394	(148,1188)	400	216	336	(110,973)	602	268	611	(163, 1090)

control system case study that has been published by IFPUG. The remaining 9 applications are from the software measurement course for postgraduate students in University Putra Malaysia; these are avionics, robotics and control systems in which the Software Requirements Specifications have been written according to the IEEE STD 830 template [32].

The applications represented in the dataset are all small real-time systems, modeled as a single layer, so that the comparison of their size expressed in different FSM methods is possible. Finally, the quality of requirement specifications was evaluated via the procedure proposed by Desharnais and Abran [33] to guarantee a reasonable level of granularity for the functional processes. The application of this procedure yielded that 83.2% of the functional processes of all the dataset applications were of good quality, so that the corresponding measures are deemed reliable.

In the analysis reported here, the smallest application was excluded from the dataset, because it is so small (16 UFP) that it is hardly representative of any real-life applications. In the other datasets no such small applications are present.

3.3. The dataset by Robiolo

The dataset is a set of small business projects, most of which were Web applications. They were all new developments, whose requirements specifications were documented in a homogeneous way, namely, via Use Cases.

The involved human resources shared a similar profile: advanced undergraduate students –who had been similarly trained– worked in the academic setting, at the S&T Department of an University and at a CMM level 4 company.

3.4. The dataset by Liu

This dataset collects data from 15 projects, including academic examples used in teaching (8 projects); academic examples used in research (3 projects); project management tools (3 project); and measurement tools (1 project). All the software applications were measured by Liu during his PhD study, following the methodology described in [34]. The quality of the model and the datasets was then checked by two professors who are quite expert in the FSM field.

3.5. The dataset by van Heeringen

This dataset includes data from 26 projects, which were measured during the Sogeti bidding process. In the COSMIC measurements, only the end user measurement viewpoint has been used, to make the outcomes of the analysis comparable. The measurements have been carried out and reviewed by IFPUG certified analysts. The analysts have a considerable amount of experience with the COSMIC method as well. Most of the COSMIC analyses are reviewed by COSMIC entry level certified analysts, so the expected quality of the measurements is high. The projects involved are all situated in the business application domain. The major part of the organizations involved operate in the banking, insurance and government domains.

3.6. The dataset by Cuadrado Gallego

This dataset includes data from 33 software applications. Out of these 33 software applications, one is a case study documented by IFPUG, another application is a case study provided by IBM Rational [35]; the remaining applications were final projects of students attending the Software Engineering course at the University of Alcalá, Madrid, Spain. These applications were measured by a team of three junior measurers; later the measures were verified by another senior measurer and finally by the authors of [29].

3.7. The dataset by Ferrucci

Data were provided by an Italian software company, whose core business is the development of enterprise information systems, mainly for local and central government. The company develops and manages solutions for Web portals, enterprise intranet/extranet applications (such as Content Management Systems, e-commerce, work-flow management systems, etc.), and Geographical Information Systems. The dataset includes data from 25 applications.

4. Description of the research

4.1. Research focus

The core of the measurement process according to the IFPUG method is described in Table 7, where for each phase, the delivered results are specified, and the measures that can be derived from such results are listed. The measures written in bold are those considered in this paper.

Table 7: IFPUG process and products.

Phase	Activity	Results delivered	Avaliable measures
1	Identify EP and LDF	list of EP and LDF	**#EP**, #LDF
2	Classify EP and LDF	list of ILF, EIF, EI, EO, EQ	**#ILF**, **#EIF**, **#EI**, **#EO**, **#EQ**
3	Analyze EP and LDF	for each EP and LDF: DET, RET, FTR	**#FTR**, #RET, #DET
4	Evaluate complexity and weigh EP and LDF	ILF, EIF, EI, EO, EQ	**size of ILF, EIF, EI, EO, EQ**
5	Sum up	–	**UFP**

'LDF' stands for Logical data file.

The core of the measurement process according to the SiFP method is described in Table 8.

The core of the measurement process according to the COSMIC method is described in Table 9.

Table 8: SiFP process and products.

Phase	Activity	Results delivered	Avaliable measures
1	Identify EP and LDF	list of EP and LDF	**#UGEP**, **#UGDG**
2	Compute 4.6 $\#UGEP + 7\ \#UGDG$	–	**SiFP**

Table 9: COSMIC process and products.

Phase	Activity	Results delivered	Avaliable measures
1	Identify Functional processes	list of Func. processes	**#FPr**
2	Identify data groups	list of DG	—
3	Analyze Func. processes	for each Func. process: list of data movements	**E, X, W, R**
4	Sum up	–	**CFP**

The relationships that were studied are summarized in Figure 3, where double-headed arrows indicate convertibility relationships; solid single-headed arrows indicate relationships between a measure and its own elements (e.g., between UFP and FTR); dashed single-headed arrows indicate relationships between a measure and other measures' elements (e.g., between CFP and #ILF, #EIF, #EI, #EO, #EQ). The rightmost column gives the number of the paper section where the analysis is illustrated. We introduced this figure to provide a graphical guidance to the reader to better comprehend the relationship among measures and the BFC's. With convertibility relationships we are interested in analyzing how a measure expressed in a measurement unit can be converted into a measure expressed in another measurement unit [36]. With the relationships between a measure and its BFC's we are interested in investigating whether the measurement process can be simplified by focusing only on some steps of the measurement process. Indeed, if we establish a statistical relationships between a measure and subset of its BFC's, we could approximate the functional size by exploiting only the sizes of the selected BFC's. As an example, using the non-weighted IFPUG BFC's we could estimate the size of an application early and quickly, without performing the whole measurement process as prescribed by the IFPUG manual [9]. With a similar goal, we investigate relationships between a measure and the BFC's of another measure. As an example, we could exploit the non-weighted IFPUG BFC's to quickly estimate the functional size in terms of COSMIC Function Points of an application without employing the COSMIC manual.

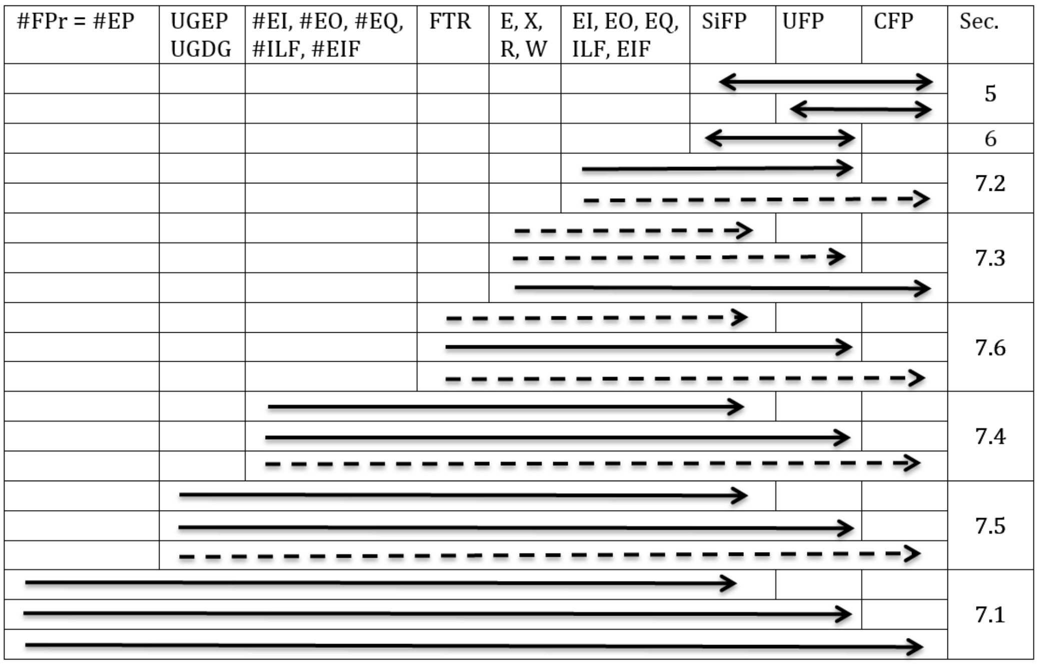

Figure 3: Summary of relationships studied.

4.2. Usage scenarios

Here we illustrate two of the many possible scenarios in which the relationships between measures presented in the paper can be effectively used.

4.2.1. Conversion of historical data

Suppose that an organization has been using FSM method X for some years: a wealth of historical data –expressed in the X measurement unit– has been collected. Now, this organization has decided to switch to FSM method Y (for some reason, e.g., because Y is believed to allow for faster measurement). Since the historical dataset is a valuable asset, the organization wants to convert the data in the dataset from X to Y. There are several ways to perform the conversion:

1. Repeating the measurement using Y instead of X. In this way you get the most accurate measures, but you have to perform the full-fledged Y measurement process, which could be quite expensive.
2. Using the available documentation of X measures to get Y measures. For instance, if X is IFPUG Function Points and Y is COSMIC Function Points, data groups can be identified by looking at logical data files, functional processes are identified by considering transactions, FTR indicate possible reads and writes, etc. This procedure is cheaper than the previous one and almost equally accurate.
3. Performing a conversion based on a statistical model. This can be done either by using models at the measure level, i.e., $Y = c\ X$, as in Table 11, or models at the BFC level, i.e., $Y = c_1\ X_1 + c_2\ X_2 + ... + c_n\ X_n$ (where X_i is a BFC of the X method), as in Table 17.

The conversion described at point 3 above can itself be carried out in two ways:

- Using external conversion constants; that is, Y measures are computed as $Y = c\ X$, with c taken from the literature.

- Using locally computed constants. This implies that a) a subset of the historical projects are measured in terms of Y, e.g., using the procedures 1 or 2 above, b) a model $Y = d\ X$ is derived, and c) the model is used to convert the subset of historical data not re-measured at step a) into measure Y.

The information provided in this paper can help practitioners with some decisions. For instance, practitioners could decide whether to use procedure 3 instead of 1 or 2, based on the level of accuracy they aim to achieve, according to the data given in Sections 5–7. The same data could help practitioners decide whether to use models from the literature or build their own models.

4.2.2. Early size estimation

Suppose that a project is in a very early stage of development, when the new application to be developed is described very roughly. Suppose also that a manager has to take decisions that require an approximate knowledge of the size of the application to be developed. If the organization in question uses the IFPUG method, our manager could look into Table 19 and see that the size in IFPUG Function Points can be estimated via model $UFP = c \ \#UGEP$, with c in the range [4.58, 6.63]. If the available description of the new application supports identifying the elementary process (usually a mock-up of the user interface is sufficient for this purpose), our manager can get a very early and fast estimation of the size of the application in UFP. In fact, it is between 4.58 $\#UGEP$ and 6.63 $\#UGEP$. The estimation range is large, but it can be obtained in ten minutes by looking at the mock-up, and this kind of estimates is often sufficient to support critical decisions in the earliest stages of development.

4.3. Research methodology

For all the pairs of measures that we analyzed, correlation analysis was first studied. To this end, we used Pearson test when possible, and non parametric Spearman and Kendall tests when the conditions for applying Pearson's test did not hold.

Then, linear regression was applied. In general we applied Ordinary Least Squares (OLS) linear regression. However, the characteristics of some datasets –namely the non normality of data distributions and the presence of outliers– suggested that more robust regression techniques be used. In fact, OLS has a null breakdown point (according to the definition by Hampel [37], the breakdown point is the smallest percentage of contaminated data that can cause the estimator to take on arbitrarily large aberrant values). The Least Median of Squares (LMS) was introduced by Rousseeuw as a *robust* regression, i.e., a regression featuring a high breakdown point [38]. LMS works like OLS, except that the median –rather than the mean– of squared residuals is minimized. In this way, a breakdown point = 50% is achieved.

We are interested to achieve distributional robustness, that is, to minimize the impact of skewed distributions and/or outliers on regression estimates. Although conceptually distinct, distributional robustness and outlier resistance are, for practical purposes, synonymous.

In this paper, whenever OLS regression does not appear to guarantee distributional robustness, due to the characteristics of the dataset being analyzed, we use LMS regression.

With both OLS and LMS regressions, we used regression through the origin, in order to obtain models of type $y = a \times x$ instead of models of type $y = a \times x + b$. Models of type $y = a \times x$ were used because when the size is zero with one measurement unit, it should be zero with other measurement units as well. Moreover, conversions based on a simple constant are simpler to understand and use for practitioner.

To compare the accuracy of two given models, we compared the absolute relative residuals via Mann-Whitney (Wilcoxon rank sum) test.

All the results described in the paper are statistically significant. The statistical significance threshold was set at $\alpha = 0.05$, as usually done in Empirical Software Engineering studies [39].

5. Study of the correlation among SiFP, IFPUG and COSMIC Function Points

Here we report the description of the analysis and the results obtained for each dataset considered in this study.

Figure 4 shows the SiFP and UFP measures compared to CFP measures. Circles represent data points in the SiFP-CFP plane, while crosses represent points in the UFP-CFP plane. The distribution of points in Figure 4 suggests that both SiFP and UFP are fairly correlated with CFP for Abualkishik, Robiolo, Liu, Cuadrado Gallego, and Ferrucci datasets. In the case of Desharnais and van Heeringen datasets we can note a stronger correlation. This visual impression is confirmed by the correlation test results reported in Table 10. Observe

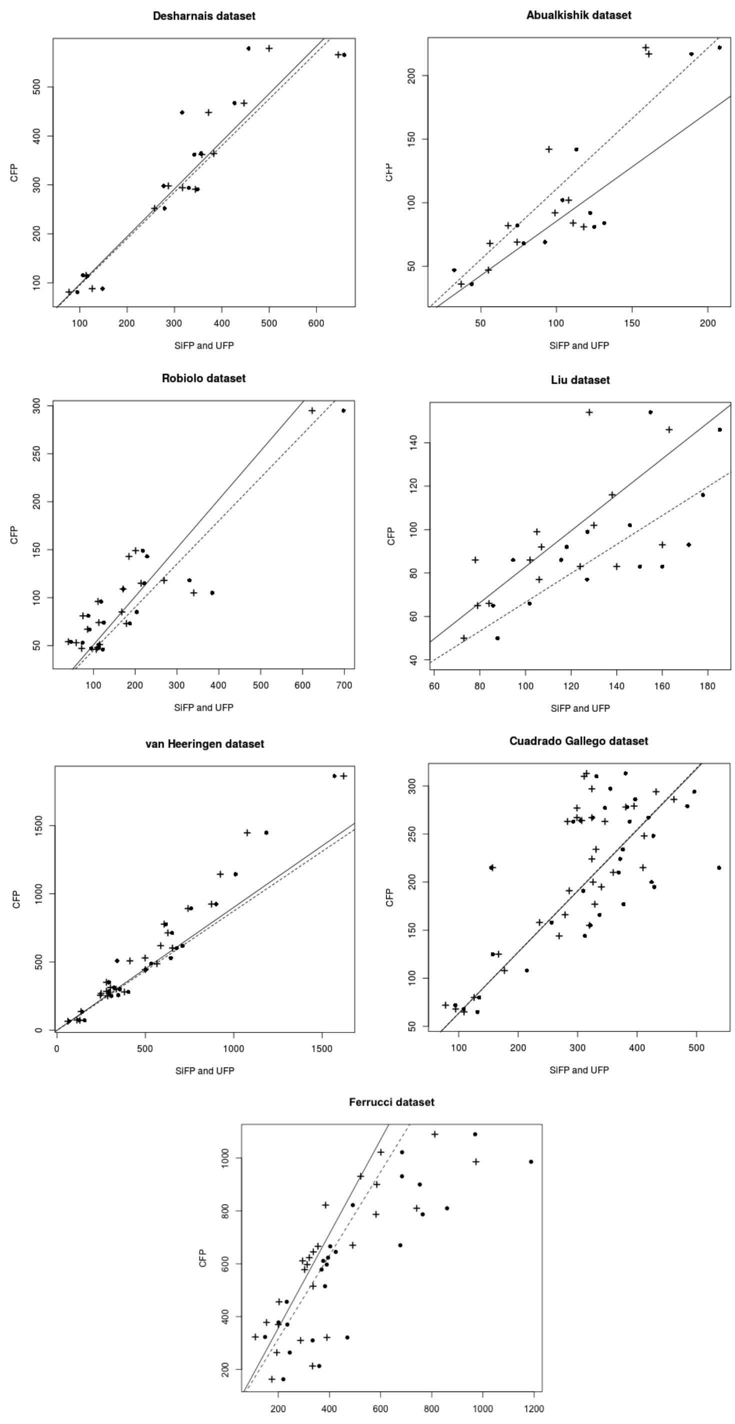

Figure 4: SiFP vs. CFP and UFP vs CFP regression lines for the analyzed datasets

that we exploited Pearson's test for Desharnais and Liu dataset since the distributions of the measures are normal according to Shapiro's test. Differently, for the remaining datasets the distributions of the measures are not normal and we applied Spearman's and Kendall's tests (see Table 10). From Table 10 we can note that the correlation of CFP with SiFP and UFP is very strong for Desharnais and van Heeringen datasets, while it is less strong –though quite evident– in the cases of Robiolo, Cuadrado Gallego, and Ferrucci datasets.

For the same reason, we performed the analysis using OLS regression when the distributions of the measures are normal. On the contrary, we carried out LMS regression when measures are not normally distributed. The models found and the corresponding accuracy indicators are summarized in Table 11.

The regression lines are also shown in Figure 4. It is possible to see that regression lines are very close to each other for Desharnais, van Heeringen, and Cuadrado Gallego datasets: for Desharnais dataset the two lines are both close to the y=x line. As for the van Heeringen dataset, we can also observe that most applications have size smaller than 750 UFP, thus LMS regression ignores the few larger applications, so that the resulting model does not fit well such applications. It is therefore prudent to consider the model valid only up to 750 SiFP (or 750 UFP).

The magnitude of relative errors (MRE) distributions of the two models built for each dataset are described via boxplots in Figure 5. The blue diamonds indicate the MMRE values (mean MRE).

It can be seen that –although with differently shaped distributions– both SiFP and UFP correlations to CFP are fairly accurate for Desharnais, Liu, van Heeringen, and Cuadrado Gallego datasets; differences greater than 20% are exclusively due to outliers for Desharnais dataset. Differently, neither SiFP nor UFP correlations to CFP are very accurate for Abualkishik, Robiolo and Ferrucci datasets.

To test which of SiFP and UFP correlates better to CFP, we run the Mann-Whitney (Wilcoxon rank sum) test on MRE for each dataset, and found that the test accepts the equivalence hypothesis for SiFP and UFP MRE for each dataset, thus supporting the hypothesis that there is no difference between the distributions of MRE for SiFP- and UFP-based models.

Table 10: CFP vs. SiFP and CFP vs. UFP correlation tests.

	SiFP vs. CFP			UFP vs. CFP		
Dataset	Kendall	Spearman	Pearson	Kendall	Spearman	Pearson
Desharnais	—	—	0.926	—	—	0.967
Abualkishik	0.636	0.804	—	0.636	0.811	—
Robiolo	0.532	0.760	—	0.579	0.788	—
Liu	—	—	0.722	—	—	0.679
van Heeringen	0.805	0.931	—	0.865	0.965	—
Cuadrado Gallego	0.454	0.609	—	0.486	0.628	—
Ferrucci	0.693	0.862	—	0.687	0.831	—

Table 11: CFP convertibility models.

Dataset	Model	regr.	outl.	MMRE	MdMRE	Pred(25)	Error range
Desharnais	CFP = 0.951 × SiFP	OLS	4/14	15.9%	11.3%	85.7%	(-33%, 60%)
Desharnais	CFP = 0.973 × UFP	OLS	5/14	10.1%	6.6%	92.9%	(-19%, 39%)
Abualkishik	CFP = 1.108 × SiFP	LMS	0/12	29.9%	25.7%	50.0%	(-24%, 74%)
Abualkishik	CFP = 0.855 × UFP	LMS	0/12	21.0%	18.7%	58.3%	(-43%, 24%)
Robiolo	CFP = 0.451 × SiFP	LMS	0/19	27.3%	25.9%	47.4%	(-61%, 65%)
Robiolo	CFP = 0.506 × UFP	LMS	0/19	28.0%	22.7%	57.9%	(-63%, 64%)
Liu	CFP = 0.666 × SiFP	OLS	1/15	15.7%	14.6%	80.0%	(-33%,28%)
Liu	CFP = 0.829 × UFP	OLS	3/15	15.7%	12.1%	80.0%	(-31%,43%)
van Heeringen	CFP = 0.874 × SiFP	LMS	0/25	19.3%	14.9%	64.0%	(-42%,86%)
van Heeringen	CFP = 0.901 × UFP	LMS	0/25	18.7%	16.7%	68.0%	(-33%,59%)
Cuadrado Gallego	CFP = 0.637 × SiFP	LMS	0/33	20.8%	20.4%	60.6%	(-54%,59%)
Cuadrado Gallego	CFP = 0.635 × UFP	LMS	0/33	17.2%	13.0%	69/7%	(-54%,31%)
Ferrucci	CFP = 1.580 × SiFP	LMS	0/25	39.8%	19.3%	52.0%	(-28%,167%)
Ferrucci	CFP = 1.784 × UFP	LMS	0/25	36.4%	20.6%	52.0%	(-39%,180%)

6. A study of the correlations between SiFP and UFP

The correlation between SiFP and IFPUG FP was first studied via Pearson's test (for datasets featuring normally distributed data) and Kendall's and

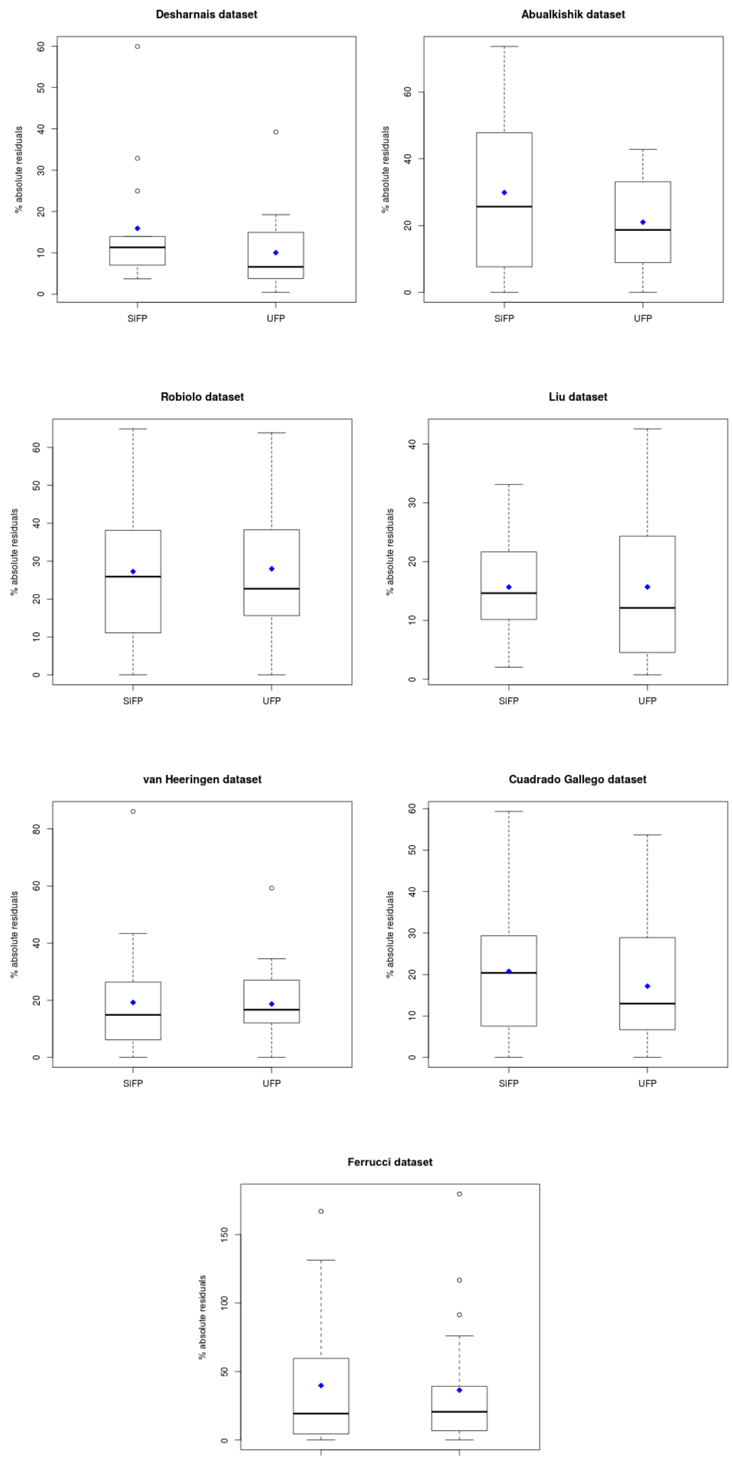

Figure 5: SiFP vs. CFP and UFP vs CFP magnitude of relative error boxplots

Spearman's tests (for datasets featuring not normally distributed data). Results of correlation tests are given in Table 12.

Table 12: SiFP–UFP correlation tests.

Dataset	Kendall	Spearman	Pearson
Desharnais	—	—	0.990
Abualkishik	—	—	0.961
Robiolo	0.860	0.961	—
Liu	—	—	0.974
van Heeringen	0.886	0.967	—
Cuadrado Gallego	0.798	0.930	—
Ferrucci	0.900	0.980	–

Table 13: SiFP–UFP convertibility models.

Dataset	Model	regr.	outl.	MMRE	MdMRE	Pred(25)	Error range
Desharnais	SiFP = 0.957 × UFP	OLS	1/14	7.2%	5.9%	100%	(-22%, 13%)
Abualkishik	SiFP = 1.132 × UFP	OLS	1/12	15.7%	7.6%	91.7%	(-19%, 92%)
Robiolo	SiFP = 1.046 × UFP	LMS	0/19	7.8%	6.7%	100%	(-21%,6%)
Liu	SiFP = $\text{UFP}^{1.033}$ (*)	OLS	0/15	4.8%	4.3%	100%	(-9%,10%)
van Heeringen	SiFP = 1.034 × UFP	LMS	0/25	7.0%	3.8%	92.0%	(-27%, 25%)
Cuadrado Gallego	SiFP = 1.121 × UFP	LMS	0/33	8.0%	7.3%	97.0%	(-15%,30%)
Ferrucci	SiFP = 1.221 × UFP	LMS	0/25	5.0%	4.9%	100%	(-11%, 13%)

(*) no statistically significant linear model could be found for this dataset; this model was obtained after log-log transformation of data.

Regression analysis yielded the models that are summarized in Table 13. The distributions of the models' MRE are described via boxplots in Figure 6. The blue diamonds indicate the MMRE values (mean MRE). Outliers are not shown to keep the figure readable; however, the presence of outliers can be guessed by looking at Table 13: the only dataset having a data point not shown in Figure 6 is the dataset by Abualkishik (this also explains why MMRE is so much greater than MdMRE for this dataset).

7. Analysis of the correlations involving BFC's

Not all the datasets provide the same information. However, it was possible to carry out several studies concerning the correlation of functional size measures

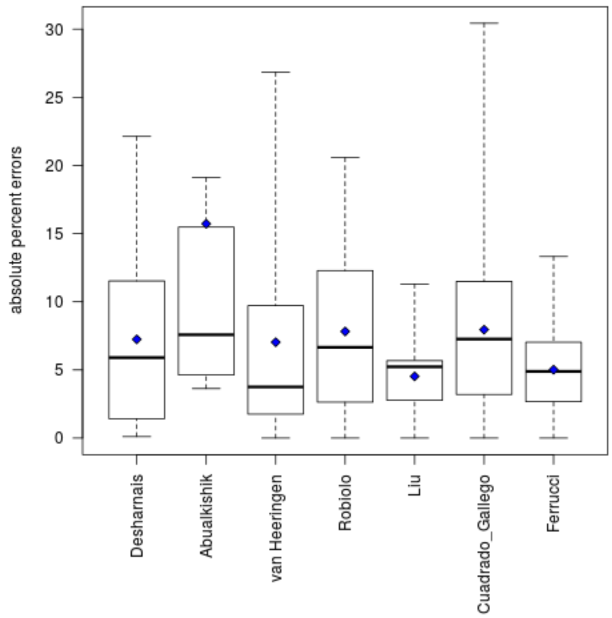

Figure 6: SiFP vs. UFP magnitude of relative error boxplots for all the analyzed datasets.

with different types of IFPUG or COSMIC measure elements (e.g., ILF and Exit). Most studies reported in the following sections are supported by three or more datasets.

In what follows, we generally show the statistically significant models featuring MMRE not greater than 30%; this is a somewhat arbitrary choice, but necessary to limit the number of models to be described. Concerning multivariate models, only models having uncorrelated independent variables are given.

7.1. Correlation of measures with the number of COSMIC Functional Processes

This analysis was supported by the datasets by Desharnais, Abualkishik, Robiolo, Liu, Van Heeringen and Ferrucci.

The models found are illustrated in Table 14. It is interesting to note that in several cases UFP and SiFP are estimated more accurately than CFP. This result is not surprising, since the number of functional processes is generally also the number of transactions, or #UGEP. Anyway, who wants to perform

Table 14: Correlation of measures with the number of COSMIC Functional Processes (#FPr)

Dataset	Model	regr.	MMRE	MdMRE	Pred(25)	Error range
Desharnais	UFP = 6.57 #FPr	OLS	10.5%	3.6%	79%	-36%..37%
Desharnais	CFP = 6.74 #FPr	OLS	18.3%	13.3%	71%	-36%..53%
Desharnais	SiFP = 6.47 #FPr	OLS	8.3%	6%	93%	-33%..16%
Abualkishik	UFP = 5.22 #FPr	OLS	19.8%	19.3%	67%	-53%..30%
Abualkishik	SiFP = 6.22 #FPr	OLS	9.8%	5.9%	83%	-29%..12%
Robiolo	UFP = 5.6 #FPr	LMS	34.6%	34.3%	32%	-68%..68%
Robiolo	CFP = 4.21 #FPr	LMS	19.6%	14%	74%	-60%..49%
Robiolo	SiFP = 7.29 #FPr	LMS	32.8%	28.6%	47%	-62%..89%
Liu	UFP = 6.33 #FPr	OLS	18.5%	14.6%	67%	-41%..30%
Liu	CFP = 5.92 #FPr	OLS	20.3%	10.1%	67%	-27%..57%
Liu	SiFP = 7.45 #FPr	OLS	15.3%	11.5%	73%	-35%..26%
Van Heeringen	UFP = 6.46 #FPr	LMS	20.2%	17.5%	64%	-54%..62%
Van Heeringen	CFP = 7.50 #FPr	LMS	20.6%	18.7%	64%	-25%..47%
Van Heeringen	SiFP = 6.71 #FPr	LMS	18.3%	16.5%	76%	-56%..66%
Ferrucci	UFP = 4.34 #FPr	LMS	27.7%	19.5%	52%	-68%..113%
Ferrucci	CFP = 7.94 #FPr	OLS	11.9%	10.7%	88%	-28%..33%
Ferrucci	SiFP = 5.52 #FPr	LMS	25.9%	22%	52%	-66%..101%

approximate estimations of the functional size based on the number of functional processes should not necessarily target CFP to get the most accurate estimates.

7.2. Correlation of measures with IFPUG BFC's

This analysis was supported by the datasets by Robiolo, Liu and Ferrucci. In this analysis, we do not consider SiFP, since it would hardly make sense to correlate a measure (SiFP) that does not involve evaluating the 'complexity' of data and transactions with BFC's that incorporate the evaluation of 'complexity'.

The models found having no more than two independent variables are illustrated in Table 15. Those having more than two variables are more accurate, but are less interesting, since it is quite clear that the more information is available (via more work), the more accurate will be the measure (in the extreme, the model of UFP vs. all IFPUG BFC's will be perfectly accurate).

The models in Table 15 show that it is possible to obtain fairly good approximations of the functional size measured in UFP by means of only two BFC's (e.g., the model based on ILF and EI has MMRE=15.9% for the dataset by Robiolo, the model based on ILF and EO has MMRE=11.2% for the dataset by Liu, and the model based on EI and EQ has MMRE=17.1% for the dataset by Ferrucci).

The size expressed in CFP can be estimated at similar levels of accuracy only for Liu's dataset (see for instance the model based on EI and EQ).

7.3. Correlation of measures with COSMIC data movements

This analysis was supported by all datasets, except Van Heeringen's and Cuadrado-Gallego's.

The models found are illustrated in Table 16.

It can be seen that estimates based on just Exit data movements are fairly accurate for all the datasets. This is good news for organizations that need to estimate the size in CFP without going through the whole measurement process, since counting Exit data movements is clearly faster and easier than counting also the Entry, Read and Write data movements.

7.4. Correlation of measures with unweighted IFPUG BFC's

This study concerns the correlation of functional size measures with unweighted IFPUG BFC's, i.e., #ILF, #EIF, #EI, #EO and #EQ. This analysis was supported by the datasets by Robiolo, Liu, Van Heeringen and Ferrucci.

In this case, models with more than two independent variables are interesting, because they indicate the possibility of getting estimates of the standard functional size measures (namely, UFP and CFP) based on a set of counts that are quite easy to obtain. In fact, most approximate functional size estimation methods (like NESMA estimated [3], as well as the official IFPUG early estimation method [9]) are based on unweighted IFPUG BFC's.

We found 154 models; for space reasons, here we report only a selection of such models.

The models of CFP featuring the lowest MMRE are given in Table 17. These models indicate that it is possible to estimate the functional size in CFP based on early products of the IFPUG measurement process.

A selection of the models of UFP are given in Table 18. It is easy to see that fairly accurate models are available for all the datasets. These results support the idea at the base of the SiFP method: it is hardly necessary to apply the full-fledged IFPUG process to get values that are fairly close to the actual measures. The models that involve all the independent variables (i.e., #ILF, #EIF, #EI, #EO and #EQ) –not reported in Table 18– feature MMRE $< 10\%$.

The models found involving SiFP are not reported, since they tend to reproduce the definition of SiFP Function Points, i.e., 7 (#ILF + #EIF) + 4.6 (#EI + #EO + #EQ), with MMRE close to zero.

Finally, it should be stressed that all the models presented in this section are suitable as approximate estimation methods, since they require only a small set of data that are obtained quite easily. In fact, it is sufficient to identify the logic data and the transactions involved in the software application to be measured, and classify data as internal or external, and transactions as input, output or inquiries, according to IFPUG rules. These activities are much faster and cheaper than the full-fledged IFPUG process [16], since the most expensive and time-consuming activities —namely analyze each logic data file and each transaction to determine their "complexity"— are not required.

7.5. Correlation of measures with unspecified generic data and processes

This analysis concerns the correlation of functional size measures with unspecified and generic data (UGDG) and processes (UGEP), as defined in Section 2.4. This analysis was supported by all datasets.

The models found are illustrated in Table 19. For SiFP, only univariate models are given, since —as could be expected— bivariate models tend to reproduce the SiFP computation formula (3).

The models illustrated in Table 19 suggest a few interesting observations:

- For all datasets it was possible to estimate UFP measures with MMRE better than 20% (with the exception of Abualkishik's dataset, for which MMRE=20.1%). Even more interestingly, the accuracy of UFP estimates based on UGEP and UGDG does not appear worse than the accuracy of estimates based on weighted IFPUG BFC's (see Table 15). This observation seems to support the idea at the basis of SiFP, that classifying and weighting data and transactions does not add value to functional size measures.

- CFP measures are often correlated with UGEP and only in one case (the dataset by Robiolo) with UGDG. Only for the datasets by Cuadrado-Gallego and Ferrucci a model CFP = f(UGEP, UGDG) was found. This observation seems to confirm that —since data do not enter in the definition of the COSMIC measure— also CFP estimates are hardly based on data measures.

- SiFP measures can be estimated with fairly good accuracy based on UGEP alone. This means that one could simplify even further the process of measuring SiFP, by skipping the measurement of UGDG.

7.6. Correlation of measures with IFPUG FTR

This study concerns the correlation of functional size measures with the number of File Type Referenced according to the IFPUG method. The analysis was supported by the datasets by Desharnais, Abualkishik and Cuadrado Gallego.

The models found are illustrated in Table 20. In this case it is particularly interesting that COSMIC Function Points appear better correlated to FTR (an IFPUG concept) than IFPUG and SiFP Function Points. This finding can be partly explained by considering that FTR (the number of references to logic data files) is conceptually correlated to COSMIC Read and Write data movements.

8. Discussion of results

In this section we provide an interpretation of the results obtained from the analyses described above. We also provide some suggestions for the practical envisioned usage of the mentioned results.

8.1. Convertibility

In Section 4.2.1 we described why an organization could possibly be interested in converting their historical measures from a given unit into another. Here we describe how to perform such conversions, according to the results of the analyses described in Sections 5–7.

8.1.1. Convertibility among functional size measures

The examined FSM measures appear statistically correlated. For all the available datasets it was possible to derive statistically significant models that associate measures expressed in a given functional size measurement unit to measures expressed in another unit (see Tables 11 and 13).

The conversion accuracy achieved is sufficiently good to allow for practical usage of the conversion models. However, it is up to project managers and other people interested in conversion, to evaluate if the accuracy levels are acceptable or not for their specific purposes. To this end, the data provided in Tables 11 and 13 provide clear indications about the level of accuracy that can be expected.

It is important to note that different datasets support different models. Consider for instance models of CFP vs. SiFP: for the dataset by Robiolo we have $CFP = 0.451 \times SiFP$, while for the database by Ferrucci we have $CFP = 1.58 \times SiFP$. This means that if CFP sizes of applications in the dataset by Ferrucci were computed using the model obtained from the dataset by Robiolo, we would get an average error $\frac{0.451-1.58}{1.58} = -71\%$, that is, the conversion would grossly underestimate the actual measures in CFP. This type of phenomen can happen with any pair of datasets including quite different applications' measures. Accordingly, if an organization wants to perform a conversion of measures from one unit to another, it is strongly advised to use models derived locally, using measures from applications that are similar to those whose measures have to be converted. This observation is coherent with the results of the empirical analysis by Ferrucci et al. [40], which showed that conversion among UFP and CFP based on company-specific data performed better (in terms of accuracy) than conversion based on data that originated outside of the company interested in the conversion.

8.1.2. Convertibility using BFC's

As shown in Sections 7, several statistically significant models link functional size measures with BFC's of other measures. Therefore, an organization that owns historical data at the BFC level could consider exploiting BFC-based convertibility models. For instance, if the measures or the sheer number of EI, EO, EQ, ILF and EIF were recorded, an organization can convert their IFPUG measures into COSMIC measures via the models given in Section 7.2 (Table 15) or Section 7.4 (Table 17), or Section 7.5 (Table 19). If the number of FTR was also recorded, the models given in Section 7.6 (Table 20) could be used as well.

As expected, estimation based on more detailed information yields more accurate results. For instance, it can be obseved that all the models that compute CFP based on unweighted IFPUG BFC's (given in Table 17) feature values of MMRE that are smaller than those of the models that compute CFP based on UFP (given in Table 11). Therefore, the recommendation here is to use models based on the most detailed data that are available.

As noted in Section 8.1.1 above, different datasets support different models, hence using local data is advisable.

8.2. Early approximate estimation of size measures

In Section 4.2.2 we described why an organization could take advantage of early approximate measurement methods. Here we describe how to build early approximate size estimation models according to the results of the analyses described in Sections 5–7.

In fact, the statistically significant models that relate measures to their BFC's suggest that measures can be estimated on the basis of their BFC's. This result supports the –already quite popular– usage of approximate estimation methods [41]. Quite interestingly, the good correlation with BFC's was confirmed for all measures for all the available datasets.

The results illustrated in Section 7 confirm the validity of already established practices, like estimating COSMIC Function Points on the basis of the number of functional processes as described in the "Guideline for Early or Rapid COSMIC Functional Size Measurement by using approximation approaches" [10]. However, our results also suggest new ways of estimating functional size measures. For instance, in Section 7.3 (Table 16) it is shown that all datasets support models that allow estimating CFP based on just the number of Exit data movements. The accuracy of these models is also quite acceptable (the maximum MMRE is 16.5%, for Ferrucci's dataset). This observation is particularly interesting because it shows that in cases when Functional Process specifications are characterized only in terms of inputs and outputs, a good estimation of the size in CFP is possible. This result was also observed in a previous study where it was highlighted that software code size (in terms of bytes) correlated well with the functional size obtained considering only entry and exit data movements which were automatically measured from component diagrams [42].

8.3. Observations on the nature of functional size measures

The examined FSM measures appear statistically correlated. For all the available datasets it was possible to derive statistically significant models that associate measures expressed in a given functional size measurement unit to measures expressed in another unit.

In several cases, functional size measures appear well correlated to BFC's of other measures. For instance, statistically significant models featuring small MMRE and MdMRE were found linking IFPUG Function Points and COSMIC

Exit data movements (see Table 16). Similarly, COSMIC Function Points appear correlated to unweighted IFPUG BFC's (see Table 17). This phenomenon suggests that –notwithstanding the differences in the definitions– FSM methods tend to provide essentially equivalent indications.

Concerning more specifically the relationship between SiFP and UFP, the results reported in Section 6 show that the two measures are very strongly correlated. These results confirm the early findings [22, 23] on convertibility between SiFP and UFP, and support the hypothesis that the SiFP method (which requires a faster and cheaper measurement process than the IFPUG method) can be used as a replacement of the IFPUG method.

9. Threats to validity

Concerning internal validity, we observe that the necessary conditions for causality are satisfied in our analysis. The fact that we found statistically significant correlations not only between measures, but also between measures obtained via an FSM method and the BFC's of other methods strengthens the evidence that the correlations and associations described throughout the paper derive from (fairly strong) causal relationships.

Concerning external validity, we may wonder to what extent the results presented here can be generalized. To this end, we observe that the number and variety of datasets that were analyzed provide a reasonably wide sample, that is expected to be representative of a wide range of software applications. Indeed, even if 4 out of 7 datasets in our study included applications developed by students (i.e., Abualkishik, Robiolo, Liu, and Cuadrado Gallego datasets) they were not manipulated so that they could be more easily measured [26] [27] [19] [29]. As for the type and size of the applications included in the 7 datasets, from the description provided by the researchers who previously employed them [24] [25] [26] [27] [19] [28] [29] [30], they can be considered representative of industrial software applications. Nevertheless, it is worth highlighting that different datasets tend to provide different conversion models. This means that –although the methodology used for deriving conversion models appears to be applicable to any dataset– the quantitative models depend on specific characteristics of the software applications that contribute data to the various datasets. It is likely that such characteristics –not considered by current FSM methods– affect convertibility. Consider for instance the fact that the granularity of COSMIC data movements is the data *group*, while in IFPUG measurement the *elementary* data that cross the boundary of the application affect size measures. As a consequence, the number of elementary data in a data group affects convertibility, but such datum is neither required for COSMIC nor for IFPUG measurement, thus it is generally not available when building conversion models.

All regression models provide an interpretation of the available data, that is, of a training set. As such, they cannot be used to reason about data that are outside the range covered by the training set. Our models are no exception: so, for instance, the models illustrated in Figure 4 apply in the [100,500] UFP range, approximately.

It can also be observed that our datasets are characterized by few large projects: Figure 4 shows that none of the proposed models is actually valid for applications larger than 1000 UFP.

Unfortunately, it is very difficult to find software applications that were measured using two or more different FSM methods. In fact, FSM is expensive, and software development organizations are not willing to spend extra money to get a second measure of an application. However, since most FSM methods define measures at the elementary or functional process level, we expect that applications involving many processes will not necessarily behave differently than applications having less processes. On the contrary, as observed in [43], it is the nature of processes (e.g., being more or less data and process intensive) that could affect convertibility models.

Construct validity threats do not apply to this study, since there was no option in choosing the functional size measures: their definitions are given (in two cases they are standards). The relevance of the study derives largely from the fact that the analyzed measures are widely accepted or promising proposals.

Finally, we note that a potential threat could come from the accuracy of measures. In fact, all functional size measures are to some extent subjective: we limited this risk by considering only datasets containing measures provided by experienced measurers.

10. Related work

After the COSMIC method was proposed, both researchers and practitioners were curious to know whether historical size measures expressed in UFP could be easily converted into CFP. To this end, the existence of statistical models correlating IFPUG (or NESMA [3]) Function Points and COSMIC Function Points was studied.

One of the first reviews of some provided conversion formulas was done in [44], also reporting on the application and comparing of COSMIC and IFPUG methods in order to analyze how related, consistent and reliable the formulas could be.

The results of the main studies investigating statistical conversions of IFPUG function points and COSMIC function points are synthetically reported in Table 21.

A few studies (Abran et al. [46], Lavazza and Morasca [49]) observed that the relationship between UFP and CFP could be better modeled via piecewise linear models, i.e., they proposed models composed of two segments in the UFP-CFP plane: one that applies to smaller applications and a second one that applies lo larger applications. The discontinuity point is around 200 FP according to Abran, and around 300 FP according to Lavazza and Morasca. The authors agree on the fact that the ratio CFP/UFP is larger for larger applications.

Cuadrado Gallego et al. [36] also investigated the existence of non linear models (namely, models obtained after log-log transformation of variables): in this way they obtained an approximated conversion factor of 1:1, i.e., 1 UFP = 1 CFP, within a confidence range of 0.9 to 1.1.

Lavazza [50] proposed a systematic approach to the conversion process. He pointed out that most of the conducted convertibility studies used single type of statistical linear regression analysis over each dataset. These datasets are usually characterized by a certain degree of skewness, outliers and heteroscedasticity. Thus, it is essential to use suitable statistical tools: besides log-log regression, it is suggested to use robust statistical regression, namely Least Median of Squares, to decrease the dependence on outliers. After applying the proposed approach to a few datasets, Lavazza concludes that –based on the available data– it is not possible to represent convertibility by means of a single type of equation, or even finding a single type of regression analysis that yields the best results.

Some authors also considered the correlations involving the BFC's of functional size measures. The dataset published in [28] was used to study the relationships between IFPUG BCF's – like the non-weighted number of transactions– with CFP measures, and COSMIC BFC's –like the number of functional processes– with UFP measures [51]. Strong correlations among the examined types were found: this fact supports the idea that in presence of a set of applications that are quite homogeneous –i.e., same application domain and characteristics– it is possible to obtain fairly precise estimation expressed either in UFP or CFP, based on a small number of BFC's.

Some of the models proposed in [51] simplify the application of the measurement method. For example, using the non-weighted IFPUG BFC's allows estimating the size of an application early and quickly, without performing the whole measurement process as prescribed by the IFPUG manual. Similarly, it is possible to obtain early and quickly COSMIC Function Points based on the number of IFPUG elementary processes. Actually, early and quick methods to estimate the functional size of applications have been proposed for both IFPUG [9] and COSMIC [10] methods.

Gencel and Bideau [43] applied the convertibility models from the literature to a specific application, and found that they provided poor results. The reported reasons for these results had already been identified by Gencel and Demirors [52]. The first reason is that if the software being measured has a high proportion of files that are not referenced much by the processes, measures expressed in UFP tend to be greater than those expressed in CFP. The second factor is the IFPUG “cut-off” phenomenon (i.e., the fact that no process can account for more than 7 UFP), while with the COSMIC method size limitations do not apply to processes. Gencel and Bideau concluded that it should be feasible to convert UFP to CFP for application types where the transactions are not referencing a lot of data.

Results similar to those given in [51] were found for the COSMIC method when size estimation is based on models of the requirements that are at different levels of detail [19]. For instance, a model could support counting the number of functional process, but could not provide details about every process: in such case, the size in CFP could be estimated based on the number of functional processes, as shown here in Section 7.1. In other cases, models not considered here –e.g., those involving the number of Data Groups per Functional Process– were found in [19].

Abualkishik et al. [25] conducted an exploratory study to examine the accuracy of conversion types used in the literature, according to the principles of measurement theory. Consequently, they proposed a new conversion type based on FTR’s to obtain CFP. The proposed type yielded better results than traditional approach. In conclusion, they suggest that multiple conversion types are used to obtain optimal results.

Finally, an analysis of the correlation between IFPUG Function Points and IFPUG BFC’s was carried out, using the ISBSG dataset [53]. It was found that EI are strongly correlated to UFP, and that development effort estimation based on EI is as accurate as effort estimation based on UFP [54].

11. Conclusion

The choice of using or not using FSM methods depends on several contingent factors. For instance, there are countries (like Italy) where providing functional measures is compulsory, when bidding for public administration contracts. Several organizations build cost estimation models based on historical data including functional size measures. The state of the art is characterized by several different practices: a couple of standard measure definitions (namely, IFPUG and COSMIC Function Points) are widely known and used; however, since under specific conditions both IFPUG and COSMIC measurement methods are relatively time-consuming and expensive, approximate estimation methods have been proposed. These methods call for less thorough and detailed analysis of software functional specifications: therefore, they provide faster and cheaper approximations of IFPUG or COSMIC measures. Based on the same idea, the Simple Function Point method defined a new functional size measure that –being itself simple– does not need for estimation methods.

In presence of all these paths to functional size measures, practitioners need to get quantitative information that can support decisions. For instance: if an organization is considering switching from IFPUG to SiFP (e.g., because SiFP method is cheaper than IFPUG), it is likely interested in converting historical data from UFP into SiFP. Is such conversion possible, and how accurate is it? To provide reliable answers to such questions, we performed a systematic statistical analysis of all the suitable datasets we could find. These datasets are partly of industrial nature. For instance, the dataset by Ferrucci contains measures that were collected by an organization that develops web applications and uses the data for effort estimation.

All the analyzed measures appear well correlated to each other. Statistically significant quantitative models were found for all the combinations of measures, for all the analyzed datasets. Several models involving Base Functional Components were found as well.

In conclusion, the results given in this paper represent a valuable contribution for researchers investigating FSM methods and practitioners employing those methods in that: 1) the amount of data analyzed is definitely greater than in any past study, so that more general and reliable conclusions are possible; 2) SiFP measures are involved in the analysis, so that some evidence supporting the equivalence of SiFP and UFP is made available; 3) the correlation of measures to their BFC's was verified, so that we have evidence that estimation methods are statistically well founded; 4) the correlation between measures and other methods' BFC's was observed, so that we have evidence that the analyzed FSM methods are structurally correlated to each other.

A final contribution of the paper can be summarized in the following indications for those organizations that do not use FSM methods and, thus, could not be interested in conversion: 1) if you start using an FSM method, it is not exceedingly important which one you chose, since the measures they yield appear quite well correlated (see Table 10). Considerations about the cost of measurement or the suitability of a method to the development process are probably more relevant for a choice; 2) if you chose a FSM method and then you change, you will not lose your historical data.

Future work includes:

- Experimenting with additional datasets, hopefully containing several large applications, so as to extend the results given here to larger applications.
- Experimenting with different ways of converting functional measures. More specifically, we could consider conversions that take into account the model that describes the application, as in [55] or in [56].
- If possible, reporting on the actual practice of functional size measure conversion in industry.

Acknowledgments

This work has been partly supported by the FP7 Collaborative Project S-CASE (Grant Agreement No 610717), funded by the European Commission, the "Fondo di Ricerca d'Ateneo" of the Università degli Studi dell'Insubria and the Research Fund of Austral University.

References

[1] International Function Point Users Group (IFPUG), Function point counting practices manual, release 4.3.1 (2010).

[2] Common Software Measurement International Consortium (COSMIC), The COSMIC Functional Size Measurement Method - version 4.0 Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761: 2003) (2014).

[3] International Standardization Organization (ISO), ISO/IEC 24750:2005, Software Engineering NESMA Functional Size Measurement Method, Version 2.1, Definitions and counting guidelines for the application of Function Point Analysis (2005).

[4] C. Symons, Function Point Analysis: Difficulties and improvements, IEEE Transactions on Software Engineering 14 (1988) 2–11.

[5] Finnish Software Measurement Association (FISMA), FiSMA FSM Method 1.1 (2004).

[6] Simple Function Point Association (SiFPA), Simple Function Point Functional Size Measurement Method - Reference Manual SiFP-01.00-RM-EN-01.01 (2014).
URL `http://www.sifpa.org/en/index.htm`

[7] International Standardization Organization (ISO), ISO/IEC 20926: 2003, Software engineering IFPUG 4.1 Unadjusted functional size measurement method Counting Practices Manual (2003).

[8] International Standardization Organization (ISO), ISO/IEC 19761:2011 Software engineering – COSMIC: a functional size measurement method (2011).

[9] A. Timp, uTip - Early Function Point Analysis and Consistent Cost Estimating, uTip IFPUG #03, v.1.0, IFPUG (2015).

[10] COSMIC, Guideline for Early or Rapid COSMIC Functional Size Measurement by using approximation approaches, Guideline, COSMIC (2015).

[11] F. Vogelezang, T. Prins, Approximate size measurement with the COSMIC method Factors of influence, in: Software Measurement European Forum – SMEF, 2007.

[12] L. De Marco, F. Ferrucci, C. Gravino, Approximate COSMIC size to early estimate web application development effort, in: Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on, IEEE, 2013, pp. 349–356.

[13] G. De Vito, F. Ferrucci, Approximate COSMIC size: the quick/early method, in: 40th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), 2014, IEEE, 2014, pp. 69–76.

[14] L. Santillo, M. Conte, R. Meli, Early & Quick function point: sizing more with less, in: 11th IEEE International Software Metrics Symposium (METRICS'05), 2005, pp. 6 pp.–6.

[15] COSMIC Method v3.0 Advanced & Related Topics - [Online]. http://cosmic-sizing.org/cosmic-publications/cosmic-method-documentation/guidelines/ (2008).

[16] L. Lavazza, An investigation on the relative cost of function point analysis phases, in: 11th Int. Conf. on Software Engineering Advances, ICSEA–2016, 2016.

[17] ISO/IEC 14143-1:2007, Information technology – Software measurement – Functional size measurement, ISO, 2007.

[18] T. Fetcke, The warehouse software portfolio: A case study in functional size measurement, Tech. Rep. 1999 20, Département d'informatique, Université du Quebec à Montréal (1999).

[19] V. Del Bianco, L. Lavazza, G. Liu, S. Morasca, A. Z. Abualkishik, Model-based early and rapid estimation of COSMIC functional size–An experimental evaluation, Information and Software Technology 56 (10) (2014) 1253–1267.

[20] A. J. Albrecht, Measuring application development productivity, in: Proceedings of the joint SHARE/GUIDE/IBM application development symposium, Vol. 10, 1979, pp. 83–92.

[21] R. Meli, Simple function point: a new functional size measurement method fully compliant with IFPUG 4. x, in: Software Measurement European Forum, 2011.

[22] L. Lavazza, R. Meli, An evaluation of simple function point as a replacement of IFPUG function point, in: Joint Conf. of the Int. Workshop on Software Measurement and the Int. Conf. on Software Process and Product Measurement (IWSM-MENSURA), IEEE, 2014, pp. 196–206.

[23] F. Ferrucci, C. Gravino, L. Lavazza, Simple Function Points for Effort Estimation: a Further Assessment, in: 31st ACM/SIGAPP Symposium on Applied Computing, Pisa, April 2016, ACM, 2016.

[24] J.-M. Desharnais, A. Abran, J. Cuadrado, Convertibility of Function Points to COSMIC-FFP: identification and analysis of functional outliers, in: Int. Conf. on Software Process and Product Measurement – MENSURA 2006, 2006.

[25] A. Z. Abualkishik, J.-M. Desharnais, A. Khelifi, A. A. A. Ghani, R. Atan, M. H. Selamat, An exploratory study on the accuracy of FPA to COSMIC measurement method conversion types, Information and Software Technology 54 (11) (2012) 1250–1264.

[26] A. Z. Abualkishik, M. H. Selamat, A. A. A. Ghani, R. Atan, J.-M. Desharnais, A. Khelifi, A Convertibility Study on the Conversion between FPA and COSMIC for Real Time Systems, in: 2012 Joint Conf. of the 22nd Int.

Workshop on Software Measurement and the 7th Int. Conf. on Software Process and Product Measurement (IWSM-MENSURA).

[27] L. Lavazza, G. Robiolo, Introducing the Evaluation of Complexity in Functional Size Measurement: a UML-based Approach, in: 4th International Symposium onEmpirical Software Engineering and Measurement ESEM 2010, 16-17 settembre 2010, Bolzano, 2010.

[28] H. Van Heeringen, Changing from FPA to COSMIC A transition framework, in: Proc. of Software Measurement European Forum (SMEF) Conference, Rome, Italy, 2007, pp. 143–154.

[29] J. J. Cuadrado-Gallego, D. Rodríguez, F. Machado, A. Abran, Convertibility between IFPUG and COSMIC functional size measurements, in: Product-Focused Software Process Improvement, Springer, 2007, pp. 273–283.

[30] S. D. Martino, F. Ferrucci, C. Gravino, F. Sarro, Web effort estimation: Function point analysis vs. COSMIC, Information & Software Technology 72 (2016) 90–109.

[31] A. Khelifi, A Set of References for Software Measurement with ISO 19761 (COSMIC-FFP): an Exploratory Study, Département du génie logiciel, PhD thesis, École de Technologie Supérieure, Montréal (2005) 495.

[32] IEEE Computer Society. Software Engineering Standards Committee and IEEE-SA Standards Board, IEEE recommended practice for software requirements specifications, Institute of Electrical and Electronics Engineers, 1998.

[33] J.-M. Desharnais, A. Abran, Assessment of the quality of functional user requirements documentation using criteria derived from a measurement with COSMIC-ISO 19761, in: 20th International Workshop on Software Measurement–IWSM, 2010, pp. 481–496.

[34] L. A. Lavazza, V. Del Bianco, C. Garavaglia, Model-based functional size measurement, in: Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, ACM, 2008, pp. 100–109.

[35] Course registration system, Tech. rep., IBM Rational (2004).

[36] J. J. Cuadrado-Gallego, L. Buglione, M. J. Domínguez-Alda, M. F. de Sevilla, J. A. G. de Mesa, O. Demirors, An experimental study on the conversion between IFPUG and COSMIC functional size measurement units, Information and Software Technology 52 (3) (2010) 347–357.

[37] F. R. Hampel, A general qualitative definition of robustness, The Annals of Mathematical Statistics (1971) 1887–1896.

[38] P. J. Rousseeuw, Least median of squares regression, Journal of the American statistical association 79 (388) (1984) 871–880.

[39] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, Springer Science & Business Media, 2012.

[40] F. Ferrucci, C. Gravino, F. Sarro, Conversion from IFPUG FPA to COSMIC: within-vs without-company equations, in: Software Engineering and Advanced Applications (SEAA), 2014 40th EUROMICRO Conference on, IEEE, 2014, pp. 293–300.

[41] R. Meli, L. Santillo, Function point estimation methods: A comparative overview, in: FESMA, Citeseer, 1999, pp. 6–8.

[42] K. Lind, R. Heldal, A practical approach to size estimation of embedded software components, IEEE Trans. Software Eng. 38 (5) (2012) 993–1007.

[43] C. Gencel, C. Bideau, Exploring the convertibility between IFPUG and COSMIC Function Points: preliminary findings, in: Joint Conf. of the Int.

Workshop on Software Measurement and the Int. Conf. on Software Process and Product Measurement (IWSM-MENSURA), IEEE, 2012, pp. 170–177.

[44] M. F. Rabbi, S. Natraj, O. B. Kazeem, Evaluation of Convertibility Issues between IFPUG and COSMIC Function Points, in: Software Engineering Advances, 2009. ICSEA '09. Fourth International Conference on, 2009, pp. 277–281.

[45] F. Vogelezang, A. Lesterhuis, et al., Applicability of COSMIC Full Function Points in an administrative environment: Experiences of an early adopter, in: 13th International Workshop on Software Measurement–IWSM 2003, Citeseer, 2003.

[46] A. Abran, J.-M. Desharnais, F. Aziz, Measurement Convertibility – From Function Points to COSMIC FFP, COSMIC Function Points: Theory and Advanced Practices.

[47] J. J. Cuadrado-Gallego, L. Buglione, R. J. Rejas-Muslera, F. Machado-Piriz, IFPUG-COSMIC statistical conversion, in: Software Engineering and Advanced Applications, 2008. SEAA'08. 34th Euromicro Conference, IEEE, 2008, pp. 427–432.

[48] F. Ferrucci, C. Gravino, F. Sarro, A case study on the conversion of Function Points into COSMIC, in: Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on, IEEE, 2011, pp. 461–464.

[49] L. Lavazza, S. Morasca, Convertibility of Function Points into COSMIC Function Points: a study using piecewise linear regression, Information and Software Technology 53 (8) (2011) 874–884.

[50] L. Lavazza, A systematic approach to the analysis of function point COSMIC convertibility, in: 20th International Workshop on Software Measurement, ICSM/Mensura, Stuttgart, 2010.

[51] L. Lavazza, Convertibility of functional size measurements: new insights and methodological issues, in: Proceedings of the 5th International Conference on Predictor Models in Software Engineering, ACM, 2009, p. 9.

[52] C. Gencel, O. Demirors, Functional size measurement revisited, ACM Transactions on Software Engineering and Methodology (TOSEM) 17 (3) (2008) 15.

[53] ISBSG, International Software Benchmarking Standards Group, ISBSG repositories (2012).

[54] L. Lavazza, S. Morasca, G. Robiolo, Towards a simplified definition of Function Points, Information and Software Technology 55 (10) (2013) 1796–1809.

[55] L. Lavazza, V. Del Bianco, G. Liu, Analytical convertibility of functional size measures: a tool-based approach, in: Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2012 Joint Conference of the 22nd International Workshop on, IEEE, 2012, pp. 160–169.

[56] O. Demirors, C. Gencel, Conceptual association of functional size measurement methods, IEEE software 26 (3) (2009) 71.

Table 15: Models correlating functional size measures and FPA's BFC's.

Dataset	Model	regr.	MMRE	MdMRE	Pred(25)	Error range
Robiolo	UFP = 2.24 ILF	LMS	24.8%	25.4%	47%	-51%..57%
Robiolo	UFP = 2.58 EI	LMS	27.3%	27.8%	47%	-51%..47%
Robiolo	UFP = 1.79 ILF + 2.71 EIF	LMS	18.5%	15.1%	63%	-54%..25%
Robiolo	UFP = 1.19 ILF + 1.14 EI	LMS	15.9%	8.5%	68%	-47%..18%
Robiolo	UFP = 1.76 ILF + 0.83 EQ	LMS	18.9%	16.5%	68%	-49%..23%
Robiolo	UFP = 1.93 ILF + 3.32 EO	LMS	24.5%	13.5%	63%	-44%..81%
Robiolo	UFP = 2.82 EI + 2.12 EO	LMS	22%	18.8%	63%	-35%..61%
Liu	UFP = 2.54 ILF	OLS	17%	16.1%	73%	-32%..56%
Liu	UFP = 2.41 EI	OLS	15%	13.9%	73%	-27%..38%
Liu	UFP = 1.96 ILF + 1.65 EO	LMS	11.2%	10.5%	100%	-22%..24%
Liu	UFP = 1.11 EIF + 2.44 EI	LMS	13.8%	7%	67%	-26%..40%
Liu	UFP = 2.23 EI + 1.57 EQ	LMS	11.3%	5.3%	87%	-23%..28%
Liu	CFP = 1.96 ILF	OLS	25.3%	17.2%	60%	-52%..107%
Liu	CFP = 1.98 EI	OLS	17%	14.7%	67%	-29%..38%
Liu	CFP = 1.47 EI + 1.93 EQ	LMS	11.9%	5.9%	87%	-37%..3%
Ferrucci	UFP = 2.82 EQ	LMS	22.1%	17.6%	60%	-59%..49%
Ferrucci	UFP = 3.3 ILF + 2.16 EO	LMS	18.9%	15.1%	64%	-53%..27%
Ferrucci	UFP = 2.37 ILF + 2.52 EQ	LMS	17.6%	9.9%	72%	-52%..50%
Ferrucci	UFP = 2.66 EIF + 2.96 EI	LMS	19.1%	10.5%	76%	-72%..61%
Ferrucci	UFP = 1.61 EIF + 2.19 EQ	LMS	21.1%	15.1%	64%	-58%..30%
Ferrucci	UFP = 1.66 EI + 2.08 EO	LMS	17.1%	15.5%	72%	-58%..35%
Ferrucci	UFP = 1.37 EI + 2.14 EQ	LMS	14.1%	7.9%	76%	-40%..38%
Ferrucci	CFP = 5.50 EI	LMS	46.9%	40.1%	44%	-97%..202%
Ferrucci	CFP = 5.59 EO	LMS	48.3%	26.1%	48.3%	-80%..231%
Ferrucci	CFP = 6.65 EIF + 5.1 EI	LMS	42.4%	21.1%	60%	-64%..280%

Table 16: Models correlating functional size measures and COSMIC data movements.

Dataset	Model	regr.	MMRE	MdMRE	Pred(25)	Error range
Desharnais	UFP = 6.41 Entry	OLS	11.3%	5.1%	86%	-38%..50%
Desharnais	UFP = 2.95 Exit	OLS	10%	5.7%	86%	-40%..25%
Desharnais	CFP = 6.38 Entry	OLS	18.3%	10%	64%	-39%..45%
Desharnais	CFP = 2.79 Exit	OLS	12.8%	12%	93%	-44%..19%
Desharnais	CFP = 2.91 Read	OLS	6.8%	4.1%	100%	-21%..19%
Desharnais	SiFP = 6.30 Entry	OLS	9.8%	7.4%	93%	-35%..20%
Desharnais	SiFP = 2.87 Exit	OLS	14.8%	8.6%	71%	-43%..43%
Desharnais	SiFP = 3.19 Read	OLS	24%	18.3%	64%	-48%..75%
Abualkishik	UFP = 5.38 Write	OLS	28.5%	29.4%	42%	-64%..36%
Abualkishik	CFP = 2.77 Exit	LMS	14.7%	17.1%	92%	-43%..20%
Abualkishik	CFP = 2.15 Entry + 1.65 Read	LMS	8.9%	6.6%	92%	-29%..7%
Abualkishik	SiFP = 2.89 Exit	LMS	24.2%	18.4%	67%	-48%..78%
Abualkishik	SiFP = 7.33 Write	OLS	29.7%	32.5%	42%	-60%..58%
Robiolo	UFP = 8.30 Read + 1.78 Write	LMS	28.9%	19.4%	58%	-47%..82%
Robiolo	CFP = 3.31 Entry	LMS	17%	15.5%	79%	-47%..43%
Robiolo	CFP = 3.54 Exit	LMS	13.2%	7.7%	84%	-26%..62%
Robiolo	CFP = 2.31 Read + 2.55 Write	LMS	10%	7.4%	90%	-30%..25%
Robiolo	SiFP = 3.01 Read + 5.99 Write	LMS	27.1%	29.5%	47%	-44%..94%
Liu	UFP = 4.89 Exit	OLS	23.7%	13.2%	67%	-37%..88%
Liu	UFP = 4.77 Read	OLS	23.6%	17%	60%	-52%..65%
Liu	UFP = 5.75 Write	OLS	20.4%	16.7%	80%	-26%..75%
Liu	UFP = 3.17 Entry + 1.79 Read	LMS	18.3%	14.1%	60%	-26%..42%
Liu	CFP = 3.66 Exit	OLS	8.9%	4.9%	87%	-27%..28%
Liu	CFP = 4.04 Read	OLS	16.8%	15.4%	67%	-32%..32%
Liu	CFP = 4.36 Write	OLS	17.4%	13.8%	73%	-49%..31%
Liu	CFP = 2.37 Entry + 1.52 Read	LMS	4.5%	1.4%	100%	-14%..18%
Liu	SiFP = 5.68 Exit	OLS	22.7%	16.8%	67%	-36%..80%
Liu	SiFP = 5.57 Read	OLS	24.2%	23.9%	60%	-48%..59%
Liu	SiFP = 6.69 Write	OLS	20%	15.6%	73%	-29%..69%
Ferrucci	UFP = 0.62 Read + 4.34 Write	OLS	19.7%	13.9%	68%	-59%..15%
Ferrucci	UFP = 4.36 Write + 1.94 Exit	OLS	20.8%	12.7%	64%	-86%..40%
Ferrucci	CFP = 4.83 Entry	OLS	15.4%	13.8%	88%	-53%..24%
Ferrucci	CFP = 1.84 Read	OLS	9%	5.1%	88%	-21%..50%
Ferrucci	CFP = 5.42 Exit	OLS	16.5%	10.2%	80%	-51%..57%
Ferrucci	CFP = 3.99 Entry + 3.19 Write	OLS	11.8%	6%	88%	-61%..21%
Ferrucci	CFP = 1.62 Read + 2.74 Write	OLS	8%	6%	96%	-19%..32%
Ferrucci	CFP = 1.41 Write + 4.96 Exit	OLS	15.1%	10.1%	84%	-55%..59%
Ferrucci	SiFP = 1.7 Entry + 8.44 Write	OLS	23.1%	14%	56%	-89%..42%
Ferrucci	SiFP = 0.67 Read + 7.1 Write	OLS	20%	9.9%	64%	-63%..26%
Ferrucci	SiFP = 6.47 Write + 2.35 Exit	OLS	21.4%	12.6%	60%	-86%..57%

Table 17: Best CFP models based on unweighted FPA BFC's.

Dataset	Model	regr.	MMRE	MdMRE	Pred(25)	Error range
Robiolo	CFP = 7 #ILF + 2 #EI - 0.2 #EQ + 0.3 #EO	LMS	17.2%	10.2%	63%	-35%..43%
Liu	CFP = 5.5 #EI + 8.3 #EQ	LMS	12.4%	10%	87%	-41%..23%
van Heeringen	CFP = 13 #ILF + 7 #EO + 4 #EQ	LMS	14.4%	9.7%	76%	-44%..32%
Ferrucci	CFP = 13.56 #EI + 6.35 #EQ	LMS	28.3%	24.7%	56%	-65%..129%

Table 18: A selection of IFPUG FP models based on unweighted FPA BFC's.

Dataset	Model	regr.	MMRE	MdMRE	Pred(25)	Error range
Robiolo	UFP = 13.54 #ILF + 12.89 #EO	LMS	23.3%	18.7%	58%	-49%..55%
Robiolo	UFP = 5.80 #ILF + 4.20 #EI + 7.53 #EQ	LMS	11.6%	7%	84%	-44%..29%
Robiolo	UFP = 12.37 #ILF+ 8.32 #EO + 4.68 #EQ	LMS	15.8%	11.3%	79%	-48%..37%
Liu	UFP = 13.76 #ILF + 6.20 #EO	LMS	11.4%	10.7%	100%	-23%..24%
van Heeringen	UFP = 10.43 #ILF + 7.93 #EO	LMS	16%	12.1%	76%	-55%..8%
van Heeringen	UFP = 12.12 #ILF + 6.51 #EO + 7.52 #EQ	LMS	13.1%	11.2%	84%	-55%..14%
van Heeringen	UFP = 10.01 #ILF + 5.99 #EIF + 7.86 #EO	LMS	14.1%	7.3%	76%	-52%..34%
Ferrucci	UFP = 21.20 #ILF + 5.42 #EIF + 11.16 #EO	LMS	14.1%	9.3%	80%	-47%..35%
Ferrucci	UFP = 16.5 #ILF + 5.48 #EIF + 6.92 #EQ	LMS	13.4%	10.5%	76%	-40%..41%
Ferrucci	UFP = 12.06 #ILF + 8.27 #EO + 4.48 #EQ	LMS	10.07%	6.5%	84%	-34%..41%
Ferrucci	UFP = 8.23 #EIF + 5.87 #EI + 7.80 #EO	LMS	11.8%	5.7%	84%	-33%..37%
Ferrucci	UFP = 7.83 #EIF + 8.54 #EI + 3.15 #EQ	LMS	11.6%	5.9%	76%	-42%..37%
Ferrucci	UFP = 6.90 #EI + 6.69 #EO + 2.44 #EQ	LMS	10.9%	5.2%	88%	-27%..35%

Table 19: Models correlating functional size measures and unspecified generic data and processes.

Dataset	Model	regr.	MMRE	MdMRE	Pred(25)	Error range
Desharnais	UFP = 24.79 UGDG	OLS	22.1%	15.5%	64%	-33%..76%
Desharnais	UFP = 6.57 UGEP	OLS	10.5%	3.6%	79%	-36%..37%
Desharnais	CFP = 6.74 UGEP	OLS	18.3%	13.3%	71%	-36%..53%
Desharnais	SiFP = 25.03 UGDG	OLS	21%	13.4%	79%	-37%..88%
Desharnais	SiFP = 6.47 UGEP	OLS	8.3%	6%	93%	-33%..16%
Abualkishik	UFP = 5.29 UGEP	OLS	20.1%	19.9%	58%	-62%..32%
Abualkishik	SiFP = 6.31 UGEP	OLS	11.8%	10.8%	92%	-28%..14%
Robiolo	UFP = 14.00 UGDG	LMS	18.4%	12%	68%	-53%..40%
Robiolo	UFP = 6.63 UGEP	LMS	16.8%	12.4%	68%	-34%..43%
Robiolo	CFP = 10.2 UGDG	LMS	28%	22.2%	68%	-50%..133%
Robiolo	SiFP = 15.62 UGDG	LMS	20.1%	17.6%	74%	-57%..35%
Robiolo	SiFP = 8.33 UGEP	LMS	16.3%	14.2%	79%	-28%..46%
Liu	UFP = 6.35 UGEP	OLS	17.7%	14.1%	67%	-37%..30%
Liu	UFP = 3.53 UGEP + 6.70 UGDG	LMS	3%	2.5%	100%	-7%..6%
Liu	CFP = 5.25 UGEP	OLS	18.1%	14.4%	67%	-35%..39%
Liu	SiFP = 7.46 UGEP	OLS	14.4%	10.8%	73%	-30%..26%
Van Heeringen	UFP = 5.97 UGEP	LMS	15.9%	12.3%	68%	-47%..39%
Van Heeringen	CFP = 7.62 UGEP	LMS	23.4%	19.2%	64%	-24%..69%
Van Heeringen	SiFP = 6.63 UGEP	LMS	14%	9.2%	84%	-46%..24%
Cuadrado-Gallego	UFP = 5.30 UGDG + 4.31 UGEP	LMS	8%	7.3%	100%	-23%..19%
Cuadrado-Gallego	CFP = 1.55 UGDG + 3.14 UGEP	LMS	19.3%	16.5%	61%	-64%..42%
Cuadrado-Gallego	SiFP = 26.87 UGDG	OLS	24.9%	19.5%	73%	-62%..100%
Cuadrado-Gallego	SiFP = 6.08 UGEP	OLS	8.4%	6%	94%	-37%..19%
Ferrucci	UFP = 4.58 UGEP	OLS	11.2%	9.8%	92%	-29%..26%
Ferrucci	UFP = 3.33 UGDG + 4.23 UGEP	OLS	6.3%	4.4%	100%	-12%..22%
Ferrucci	CFP = 15.73 UGDG + 6.52 UGEP	OLS	40%	17%	56%	-19%..172%
Ferrucci	SiFP = 5.22 UGEP	OLS	11%	8.7%	84%	-12%..22%

Table 20: Models correlating functional size measures and FPA's FTR.

Dataset	Model	regr.	MMRE	MdMRE	Pred(25)	Error range
Desharnais	CFP = 2.92 FTR	OLS	6.9%	4.1%	100%	-20%..19%
Desharnais	SiFP = 3.19 FTR	OLS	24%	18.3%	64%	-48%..75%
Abualkishik	UFP = 2.83 FTR	OLS	23.2%	22.2%	50%	-54%..40%
Abualkishik	CFP = 3.38 FTR	LMS	13.2%	8.4%	75%	-35%..33%
Abualkishik	SiFP = 3.35 FTR	OLS	20%	18.2%	58%	-36%..39%
Cuadrado-Gallego	UFP = 2.73 FTR	LMS	24%	22.4%	58%	-49%..86%
Cuadrado-Gallego	CFP = 2.46 FTR	LMS	22.2%	14.5%	70%	-25%..75%

Table 21: Convertibility models found by previous studies.

Study	conversion model	n	R^2
Vogelezang and Lesterhuis 2003 [45]	CFP = 1.2 NFP - 87	11	0.99
Abran [46]	CFP = 0.84 UFP + 18	6	0.91
Desharnais [24]	CFP = 0.99 UFP - 3.2	14	0.93
Van Heeringen 2007 [28]	CFP = 1.22 NFP - 64	26	0.97
Cuadrado-Gallego et al. jjcg06, 2008 [47]	CFP = 0.83 UFP -36.61	21	0.7
Cuadrado-Gallego et al. jjcg07, 2008 [47]	CFP = 0.85 UFP + 0.19	14	0.86
Cuadrado-Gallego et al. jjcg0607, 2010 [36]	CFP = 0.68 UFP + 13.03	35	0.85
Ferrucci et al. [48]	CFP = 1.01 UFP + 207	25	0.70

NFP indicates NESMA Function Points.