Impact of Base Functional Component Types on Software Functional Size Based Effort Estimation

Luigi Buglione¹ and Cigdem Gencel²

¹ École de Technologie Supérieure (ETS) / Engineering.it

Luigi.Buglione@computer.org

² Blekinge Institute of Technology, Department of Systems and Software Engineering
cigdem.gencel@bth.se

Abstract. Software effort estimation is still a significant challenge for software management. Although Functional Size Measurement (FSM) methods have been standardized and have become widely used by the software organizations, the relationship between functional size and development effort still needs further investigation. Most of the studies focus on the project cost drivers and consider total software functional size as the primary input to estimation models. In this study, we investigate whether using the functional sizes of different functionality types, represented by the Base Functional Component (BFC) types; instead of using the total single size figure have a significant impact on estimation reliability. For the empirical study, we used the projects data in the International Software Benchmarking Standards Group (ISBSG) Release 10 dataset, which were sized by the COSMIC FSM method.

Keywords: Functional Size Measurement, Effort Estimation, COSMIC, Base Functional Component, International Software Benchmarking Standards Group (ISBSG).

1 Introduction

Forty years after the term “software engineering” was coined [28] great effort has been put forth to identify and fine tune the “software process” and its proper management. Unique tools and techniques have been developed for software size, effort, and cost estimation to address challenges facing the management of software development projects [16][42][45].

A considerable amount of these efforts have been put on software size measurement based on the fact that software size is the key measure. Function Point Analysis (FPA) was designed initially in 1979 [1] by Albrecht. This method was aimed at overcoming some of the shortcomings of measures based on Source Lines of Code (SLOC) for estimation purposes and productivity analysis, such as their availability only fairly late in the development process and their technology dependence.

FPA method was based on the idea of determining size based on capturing the amount of functionality laid out on software functional requirements. They take into account only those elements in the application layer that are logically ‘visible’ to the

user and not the technology or the software development methodology used. Since the introduction of the concept, the topic of FPA evolved quite a bit. Many variations and improvements on the original idea were suggested [11], some of which proved to be milestones in the development of Functional Size Measurement (FSM).

FPA was designed in a business application environment and has become a *de facto* standard for this community. During the following years, a large number of variants for both business application software and for other application domains (such as real-time, Web, Object Oriented, and data warehouse systems)¹ were developed. In the '90s, work was initiated at the International Organization for Standardization (ISO) level to lay the common principles and foundations for regulating *de jure* standards in FSM. Between 1998 and 2005, the 14143 standard family was developed [31]² [33]-[37] with four instantiations matching with those requirements; the Common Software Measurement International Consortium Full Function Points (COSMIC FFP) [38][46] the International Function Point Users Group (IFPUG) FPA [39][43], MarkII FPA [40][44] and the Netherlands Software Metrics Association (NESMA) FSM [41] methods. A fifth FSM method, the Finnish one by FISMA [48], will be standardized in a while. The evolution of current FSM methods is shown in Figure 1.

Fig. 1. Evolution of the main Functional Size Measurement (FSM) methods

Among those, COSMIC³ [46] adopted in 2003 as ISO 19761 [38], has been defined as a 2nd generation FSM method as a result of a series of innovations, such as a better fit with both real-time and business application environments, identification and measurement of multiple software layers, different perspectives of functional users from which the software can be observed and measured, and the absence of a weighting system.

Due to these constructive progresses, FSM has begun to be widely used for software size measurement. The number of benchmarking data on the projects which were measured by FSM methods has significantly increased in well-known and recognized benchmarks such as the one by ISBSG [13] with more than 4,100 projects. On the other hand, one of the major uses of software size measurement is its use in software effort estimation for software management purposes. However, effort estimation still remains a challenge for software practitioners and researchers.

¹ Please refer to [42] and [11] for a detailed list and a history of FSM-like methods.

² Part 1 (14143-1) has recently been updated (February 2007) [32] from its first release [31] (1998).

³ From version 3.0, the old name of this method (COSMIC-FFP) is become simply 'COSMIC'.

Effort estimation based on the functional size figures have just begun to emerge as more empirical data are collected in benchmarking datasets as in ISBSG dataset. The nature of the relationship between functional size and effort has been explored in many studies (see Section 2). The project related attributes such as ‘Team Size’, ‘Programming Language Type’, ‘Organization Type’, ‘Business Area Type’ and ‘Application Type’ were considered in the estimation models. However, the common conclusion of these studies was that although different models are successfully used by different groups and for particular domains, none of them has gained general acceptance by the software community due to the fact that no model is considered to perform well enough to fully meet market needs and expectations.

The general approach of the existing studies is the functional size of a software system is expressed as a single value obtained by a specific FSM method. This single value is derived from a measurement function in all ISO-certified FSM methods, and it is the result of adding together the functional sizes of different Base Functional Component (BFC)⁴ Types to obtain a total functional size. Each BFC Type represents different type of functionality to be provided to the users.

In our previous study [47], we made an analysis on the ISBSG dataset to test our hypothesis which states that the effort required to develop the unit size of each of the BFC Types, which provide different user functionalities is different and hence contributes to total effort at different levels. The results showed that using the functional sizes of each BFC Type as inputs to effort estimation improve the estimation reliability. In that study, we considered ‘Application Type’ to form the homogenous sub-groups of projects for the statistical analysis.

In the study presented here, we further investigate the contribution of different functionality types represented by BFC Types to total development effort. We again utilized the project data, which were measured by COSMIC-FFP [46] in the ISBSG dataset Release 10 [13]. In this case, we formed the sub-groups of projects with respect to ‘Development Type’. Then, we made Pareto analysis to further investigate the effect of the size of the projects on the estimation reliability. We also analyzed the distribution of different BFC Types in different Application Types.

The paper is organized as follows: Section 2 presents some background on functional size measurement and related work on its relationship to project effort. Section 3 presents the data preparation process. Section 4 presents the data analysis and Section 5, the conclusions of this study.

2 Related Work

There is a large body of literature on software effort estimation models and techniques in which a discussion on the relationship between software size and effort as a primary predictor has been included, such as [2][5][6][14][15][17][18].

Other factors related to non-functional characteristics of software projects are also included in many estimation models. Significant variations on the impact of other project cost drivers have been observed. Therefore a number of experimental studies were performed to investigate their impact on the size-effort relationship. Among the

⁴ BFC Type: A defined category of BFCs. A BFC is an elementary unit of an FUR defined by and used by an FSM method for measurement purposes [31].

cost drivers investigated, 'Team Size', 'Programming Language Type', 'Organization Type', 'Business Area Type', 'Application Type' and 'Development Platform' have been found to affect the size-effort relationship at different levels of significance [23][24][25][26][27][29]. Among these, the most significant are reported in [23][24] to be 'Team Size', 'Business Area Type' and 'Application Type'.

Constructive Cost Model (COCOMO) II [6], the revised version of the original COCOMO [5] takes into account the cost drivers in the estimation models and provide for measuring functional size and converting this result to SLOC. However, 'backfiring' of SLOC from functional size still can not account for the extra uncertainty introduced by adding another level of estimation [7][8][9].

In [22], Leung and Fan discuss both the strengths and weaknesses of effort estimation models. They evaluated the performance of existing models as well as of newer approaches to software estimation and found them as unsatisfactory. Similarly, in a number of studies, such as [2][19][20][21], related work on effort and cost estimation models is assessed and compared. They concluded that the models, which are being used by different groups and in different domains, still have not gained universal acceptance.

Most of the above approaches use functional size as the primary predictor and consider other project parameters in effort estimation. Abran et al. [3] used the 2003 version of the ISBSG repository to build estimation models for projects sized by the FPA method. They defined the concept of a software functional profile as the distribution of function types within the software. They investigated whether or not the size-effort relationship was stronger if a project was close to the average functional profile of the sample studied. For each sample, it was noted that there was one function type that had a stronger relationship with project effort. Moreover, the sets of projects located within a certain range of the average profile led to estimation models similar to those for the average functional profile, whereas projects located outside the range gave different regression models, these being specific to each of the corresponding subsets of projects.

In [4], the impact of the functional profile on project effort was investigated using the ISBSG repository. The ISBSG projects included in this analysis were sized by COSMIC method. In COSMIC, a functional profile corresponds to the relative distribution of its four BFC Types for any particular project. It was observed that the identification of the functional profile of a project and its comparison with the profiles of their own samples can help in selecting the best estimation models relevant to its own functional profile.

In [10], the types of functionalities a software system can provide to its users are identified, and a multidimensional measure which involves measuring the functional size of each functionality type is defined. It was suggested that experimental studies should be conducted to find the relationship between the functional size of each functionality type and the effort needed to develop the type of functionality that can pioneer new effort estimation methods.

In [47], Gencel and Buglione explored whether effort estimation models based on the BFC types, rather than those based on a single total value would improve estimation models. They observed a significant improvement in the strength of the size-effort relationship.

3 Data Preparation

In this study, the projects data in the ISBSG 2007 Repository CD Release 10 [13] were used for the statistical analysis. The ISBSG Repository includes more than 4,106 projects data on a very wide range of projects. Among those, 117 projects were sized using COSMIC-FFP. The projects cover a wide range of applications, development techniques and tools, implementation languages, and platforms. Table 1 shows the filtration process with respect to the project attributes defined in the ISBSG dataset.

Table 1. Filtration of ISBSG 2007 Dataset Release10

Step	Attribute	Filter	Projects Excluded	Remaining Projects
1	Count Approach ⁵	= COSMIC-FFP	3,989	117
2	Data Quality Rating (DQR)	= {A B}	5	112
3	Quality Rating for Unadjusted Function Points (UFP)	= {A B}	21	91
4	Development Type	= {New Development}	22	34
		= {Enhancement}		30
		= {Re-development}		5

In the first step, we filtered the dataset with respect to the ‘Count Approach’ attribute to obtain the projects measured by COSMIC. This step provided 117 projects.

In the second step, we analyzed these 117 projects with respect to ‘Data Quality Rating (DQR)’ to keep only the highest quality data for statistical analysis. In the ISBSG dataset, each project has a Quality Tag⁶ (A, B, C or D) assigned by the ISBSG reviewers based on whether or not the data fully meet ISBSG data collection quality requirements. Considering this ISBSG recommendation, 5 of the projects with a C and D rating were ignored, leaving 112 projects following this filtration step.

In the third step, we verified the availability of fields of size by functional type (or BFC) in the data set, for each of the 112 projects from step 2, since these fields are necessary for this study. The verification indicates that this information is not available for 21 of the projects, leaving 91 projects for the next step.

Since many factors vary simultaneously, the statistical effects may be harder to identify in a more varied dataset than in a more homogeneous one. Therefore, in Step 4, we built a series of homogeneous subsets considering the ‘Development Type’ attribute. We built homogeneous subsets for ‘New Development’, ‘Enhancement’ and ‘Re-development’ projects out of the 91 remaining projects. While forming the

⁵ No further filter has been considered with respect to the COSMIC versions.

⁶ A: The data submitted were assessed as sound, with nothing identified that might affect their integrity; B: The submission appears fundamentally sound, but there are some factors which could affect the integrity of the submitted data; C: Due to significant data not being provided, it was not possible to assess the integrity of the submitted data; D: Due to one factor or a combination of factors, little credibility should be given to the submitted data.

subsets, we removed the outlier projects which have very low productivity values. Since the data points for the Re-development projects were too few for statistical analysis (5 projects), we removed them from further analysis.

While exploring the nature of the relationship, we did not consider the impact of ‘Application Type’. In our previous study [47] we observed that the strength of relationship between functional size and effort are much lower when we formed homogeneous subsets with respect to Application type (0.23 for Subset 1; 0.56 for Subset 2 and 0.39 for Subset 3). But, we observed increases in R^2 values (0.23 to 0.41 for Subset 1; 0.56 to 0.60 for Subset 2 and 0.39 to 0.54 for Subset 3) when the functional sizes of each of the BFC Types are taken into account for effort estimation purposes instead of total functional size which motivated us to further investigate the effects of BFC Types on the strength of the relationship.

4 Statistical Data Analysis and Results

The primary aim of this study is to explore whether or not an effort estimation model based on the components of functional size rather than on only a total single value of functional size would improve estimation models and if so formulating the estimation model.

In this study, the two sub-datasets are first analyzed to determine the strength of the relationship between the total functional size and the development effort by applying a Linear Regression Analysis method. Then, the strength of the relationship between the functional sizes of the COSMIC BFC Types used to determine total functional size and development effort is analyzed by applying a Multiple Regression Analysis method. These findings are compared to the models representing the relationship between total functional size and effort. All the statistical data analyses in this study were performed with the GiveWin 2.10 [12] commercial tool and its sub modules and the Microsoft-Excel ‘Data Analysis ToolPak’⁷.

4.1 Total Functional Size - Effort Relationship

For the Linear Regression Analysis [30], we have the independent variable as Functional Size and the dependent variable as the Normalized Work Effort (NW_Effort) as given by the following formula;

$$NW_Effort = B_0 + B_1 FunctionalSize \quad (1)$$

where B_0 and B_1 are the coefficients to be estimated from a generic data sample. Normalized Work Effort variable is used so that the effort data among the projects which do not include all the phases of the development life cycle are comparable.

Figure 2 shows the relationship between Normalized Work Effort and COSMIC Function Points (CFP). For the New Development Projects dataset, the R^2 statistic is better than that for the Enhancement Project datasets.

⁷ <http://office.microsoft.com/en-gb/excel/HP052038731033.aspx>

a) Sub-dataset 1: New Development Projects (n=34)

b) Sub-dataset 2: Enhancement Projects (n=30)

Fig. 2. The Relationship between Normalized Work Effort and COSMIC Functional Size

A significance test is also carried out in building a linear regression model. This is based on a 5% level of significance. An F-test is performed for the overall model. A $(Pr > F)$ value of less than 0.05 indicates that the overall model is useful. That is, there is sufficient evidence that at least one of the coefficients is non-zero at a 5% level of significance. Furthermore, a t-test is conducted on each β_j ($0 \leq j \leq k$). If all the values of $(Pr > |t|)$ are less than 0.05, then there is sufficient evidence of a linear relationship between y and each x_j ($1 \leq j \leq k$) at the 5% level of significance. The results of the linear regression analysis are given in Table 2.

For subsets 1 and 2, the Total Functional Size is found to explain about 76% and 71% of the NW_Effort respectively. See [50] for an exhaustive discussion and detailed explanation about the meaning of the statistical variables. Because two subsets obtained proper R^2 values against a quite high number of data points, they were not split by size ranges⁸ or by application types. In this case a further split, the too reduced number of data points would not assure a statistical significance of the obtained results.

⁸ See [51] for a size range classification applying Pareto Analysis, applied on ISBSG r9 data repository.

Table 2. Regression Analysis Results (Normalized Work Effort – Total Functional Size)

Subset 1: New Development Projects							
	Coeff	StdError	t-value	t-prob	Split1	Split2	reliable
Constant	-49.78763	24.48831	-2.033	0.0504	0.0363	0.4419	0.7000
Functional Size	0.58882	0.05787	10.174	0.0000	0.0000	0.0000	1.0000
R² = 0.7639							
	value	prob					
normality test	28.5832	0.0000					
Subset 2: Enhancement Projects							
	Coeff	StdError	t-value	t-prob	Split1	Split2	reliable
Constant	-196.24813	83.73519	-2.344	0.0264	0.2963	0.0081	0.7000
Functional Size	3.13900	0.38040	8.252	0.0000	0.0004	0.0000	1.0000
R² = 0.7086							
	value	prob					
normality test	4.3408	0.1141					

4.2 Functional Sizes of BFC Types – Size-Effort Relationship

The COSMIC method [38][46] is designed to measure the software functional size based on its Functional User Requirements (FURs). Each FUR is decomposed into its elementary components, called Functional Processes⁹. The BFCs of this method are assumed to be Data Movement Types, which are of four types; Entry (E), Exit (X), Read (R) and Write (W). The functional size of each Functional Process is determined by counting the Entries, Exits, Reads and Writes in each Functional Process, and the Total Functional Size is the sum of the functional sizes of the Functional Processes.

In this study, the Multiple Regression Analysis method [30] was used to analyze the relationship between the dependent variable Normalized Work Effort and the functional sizes of each BFC Type as the dependent variables. The following multiple linear regression model [30] that expresses the estimated value of a dependent variable y as a functions of k independent variables, $x_1, x_2, \dots, x_k$, is used:

$$y = B_0 + B_1x_1 + B_2x_2 + \dots + B_kX_k \quad (2)$$

where B_0, B_1, B_2, B_k are the coefficients to be estimated from a generic data sample. Thus, the effort estimation model can then be expressed as:

$$NW_Effort = B_0 + B_1(E) + B_2(X) + B_3(R) + B_k(W) \quad (3)$$

where, NW_Effort (Normalized Work Effort) is the dependent variable and E, X, R and W are the independent variables representing the number of Entries, Exits, Reads and Writes respectively. In building a multiple linear regression model, the same significance tests as discussed in the previous section are carried out. Table 3 shows the multiple regression analysis results.

⁹ Functional Process: “an elementary component of a set of FURs comprising a unique, cohesive and independently executable set of data movements” [38].

Table 3. Multiple Regression Analysis Results (Normalized Work Effort – Functional Sizes of BFC Types)

Sub-dataset 1: New Development Projects dataset

Observations: 34

	Coeff	StdError	t-value	t-prob
Constant	-31.83818	18.46448	-1.724	0.0953
E	0.72694	0.38916	1.868	0.0719
X	0.01875	0.25507	0.073	0.9419
R	-0.03702	0.24675	-0.150	0.8818
W	2.21199	0.42239	5.237	0.0000

R² = 0.8919

	value	prob
normality test	13.2388	0.0013

After F presearch testing,

	Coeff	StdError	t-value	t-prob	Split1	Split2	reliable
Constant	-32.10285	17.75256	-1.808	0.0803	0.1592	0.0360	0.7000
E	0.74298	0.23129	3.212	0.0031	0.0004	0.0000	1.0000
W	2.17018	0.30448	7.128	0.0000	0.0000	0.4214	0.7000

R² = 0.8918

Sub-dataset 2: Enhancement Projects Dataset

Observations: 30

	Coeff	StdError	t-value	t-prob
Constant	-46.26395	67.37480	-0.687	0.4986
E	-0.47787	1.91093	-0.250	0.8046
X	7.37899	1.40681	5.245	0.0000
R	-1.76768	1.35114	-1.308	0.2027
W	8.08448	2.59471	3.116	0.0046

R² = 0.8755

	value	prob
normality test	3.3048	0.1916

After F presearch testing, specific model of WE;

	Coeff	StdError	t-value	t-prob	Split1	Split2	reliable
X	7.61616	1.31971	5.771	0.0000	0.0000	0.0000	1.0000
R	-2.51783	0.99965	-2.519	0.0180	0.1747	0.0129	0.7000
W	7.55544	2.47507	3.053	0.0050	0.1043	0.0058	1.0000

R² = 0.8713

In Table 4, the results from the two approaches are summarized. The results show that the R^2 is higher using the four BFC Types rather than the single total COSMIC FPs (+16.7% for new development; +23.6% for enhancement projects).

Another observation from the regression analysis results is that the functional sizes of not all BFC Types are found to be significant in estimating the effort. Two of the BFC Types, i.e. Entry and Write for New Development projects and Exit, Read and Write for Enhancement projects were found to model Normalized Work Effort.

Table 4. Comparison of the Results

Sub-datasets	# of Data Points	R ² (Using Total Functional Size (CFP))	R ² (Using BFC Types)	Increase ¹⁰ (%)
Sub-dataset 1: New Development	34	0.76	0.89	+16.7%
Sub-dataset 2: Enhancement	30	0.71	0.88	+23.6%

So, the next two questions were; 1) What about the prediction capability of an estimation model using only the BFC Types found to be significant in estimating the effort, not necessarily all the four ones at a time? 2) Is there a correlation between the contribution of BFC Types to total functional size and the BFC Types which are found to be significant in estimating the effort? Table 5 shows the results for Question 1.

Table 5. Comparison of the Results

		R ²	FORMULA
Sub-dataset 1: New Development Projects (n=34)	Total functional size (CFP)	0.7639	$Y=0.5888*CFP-49.788$
	E/X/W/R	0.8919	$Y=0.72694*E+0.011875*X-0.03702*R+2.21199*W-31.83818$
	E/X	0.8918	$Y=0.74298*E+2.17018*W-32.10285$
Sub-dataset 2: Enhancement Projects (n=30)	Total functional size (CFP)	0.7086	$Y=3.139*CFP-196.25$
	E/X/W/R	0.8755	$Y=-0.47787*E+7.37899*X-1.76768*R+8.08448*W-46.26395$
	X/R/W	0.8713	$Y=7.61616*X-2.51783*R+7.55544*W$

Thus, for New Development projects, the functional sizes of only E and W types of BFCs and for Enhancement Projects, X, R and W types can as better estimate the effort as when the functional sizes of all four types are used. In order to answer Question 2, we analyzed the distribution of the BFC Types with respect to the Development Type (see Figure 3).

The contribution to total functional size to Enhancement projects by **R** type BFC is the greatest, while **X** and **E** types contribute more for New Development projects. In terms of BFC Types, **E**, **X** and **W** types are predominant in New Development projects, while **R** in Enhancement ones.

Thus, we could not find a correlation between the level of contribution of BFC Types to total functional size and the ones which are found to be significant in estimation capability of an estimation model.

¹⁰ It was calculated as the relative increment: $[(R^2(BFC)-R^2(CFP))/R^2(CFP)]$.

Fig. 3. The distribution of BFC Types by Development Type

5 Conclusions and Prospects

This study has explored whether an effort estimation model based on the functional sizes of BFC Types rather than the total functional size value would provide better results. Our hypothesis was that the development effort for each of the BFC Types, which provide different user functionalities, might be different.

The R^2 statistics were derived from Linear Regression Analysis to analyze the strength of the relationship between total functional size and normalized work effort. The results were compared to the R^2 statistics derived from the Multiple Regression Analysis performed on the Functional Sizes of the BFC Types and Normalized Work Effort. We observed increases in R^2 values (0.76 to 0.89 for New Development projects and 0.71 to 0.88 for Enhancement projects) when the functional sizes of each of the BFC Types are taken into account for effort estimation purposes instead of the total functional size. The results showed a significant improvement, i.e. +16.7 % for new development projects and +23.6% for enhancement projects, in the effort estimation predictability.

Another interesting observation in this study is that the functional sizes of all BFC Types are not found to be significant in estimating the effort. Two of the BFC Types, i.e. Entry and Write for New Development projects and Exit, Read and Write for Enhancement projects were found to better model Normalized Work Effort.

We also analyzed the dominating BFC types in each of the datasets analyzing the frequency distribution. For New Development projects, it is the Entry (33.4%) and Exit (34.3%) that are dominant among the four BFC types. For Enhancement projects Entry (28.1%), Exit (23.8%) and Read (37.1%) that are all dominant. The results of these analysis showed that there is no correlation between the dominating BFC Types

in the dataset and the BFC Types which are found to be significant in estimating the effort.

Our hypothesis in this study was developing different functionality types requires different amounts of work effort and contributes to effort estimation in different levels of significance. The results of this study confirmed our hypothesis. Although we built some estimation formulas based on the data in ISBSG dataset, our aim in this study was not to arrive at a generic formula but rather compare the conventional approach to effort estimation and our approach discussed in this paper. Further research is required to analyze which BFC Types are significant in estimating effort and to conclude the ones to be used for establishing reliable estimation models. Further work should also include comparisons with related work performed with the IFPUG FPA method.

Because of the improvements in the estimation results just using four proxies instead of the solely functional size unit value, the organizational consideration would be the data gathering process. Usually, only the total functional size values are stored, not the whole detail derived from the measurement. However, with a low additional cost in terms of time in the data insertion it would be possible to obtain better estimation premises. In process improvement terms, using the terminology of a well known and proven maturity model as Capability Maturity Models Integration (CMMI) [49], this action would have a positive impact on:

- **PP** (*Project Planning*, Specific Practice (SP)1.4 about the estimation model used for deriving estimates comparing estimated and actual values;
- **MA** (*Measurement & Analysis*, SP2.3) about the storage of project data;
- **OPD** (*Organizational Process Definition*) about the definition of the measurement repository (SP1.4);
- **GP** (*General Practice*) **3.2** (*Collect Improvement Information*), that is the generic practice crossing all the PA (Process Areas) about the capability of collecting info to be used for improving the organizational unit's results.

Thus, starting to consider which BFC Types are significant in estimation instead of using total size figures and using establishing estimation models considering different functionality types is promising. In order to verify these conclusions and find other eventual useful relationships, further studies will also be conducted on the ISBSG dataset for the projects measured by IFPUG FPA.

References

- [1] Albrecht, A.J.: Measuring Application Development Productivity. In: Proc. Joint SHARE/GUIDE/IBM Application Development Symposium, pp. 83–92 (1979)
- [2] Abran, A., Ndiaye, I., Bourque, P.: Contribution of Software Size in Effort Estimation. Research Lab in Software Engineering, École de Technologie Supérieure, Canada (2003)
- [3] Abran, A., Gil, B., Lefebvre, E.: Estimation Models Based on Functional Profiles. In: International Workshop on Software Measurement – IWSM/MetriKon, Kronisburg (Germany), pp. 195–211. Shaker Verlag (2004)

- [4] Abran, A., Panteliuc, A.: Estimation Models Based on Functional Profiles. III Taller Internacional de Calidad en Tecnologías de Información et de Comunicaciones, Cuba, February 15-16 (2007)
- [5] Boehm, B.W.: Software Engineering Economics. Prentice-Hall, Englewood Cliffs (1981)
- [6] Boehm, B.W., Horowitz, E., Madachy, R., Reifer, D., Bradford, K.C., Steece, B., Brown, A.W., Chulani, S., Abts, C.: Software Cost Estimation with COCOMO II. Prentice Hall, New Jersey (2000)
- [7] Neumann, R., Santillo, L.: Experiences with the usage of COCOMOII. In: Proc. of Software Measurement European Forum 2006, pp. 269–280 (2006)
- [8] De Rore, L., Snoeck, M., Dedene, G.: COCOMO II Applied In A Banking And Insurance Environment: Experience Report. In: Proc. of Software Measurement European Forum 2006, pp. 247–257 (2006)
- [9] Rollo, A.: Functional Size measurement and COCOMO – A synergistic Approach. In: Proc. of Software Measurement European Forum 2006, pp. 259–267 (2006)
- [10] Gencel, C.: An Architectural Dimensions Based Software Functional Size Measurement Method, PhD Thesis, Dept. of Information Systems, Informatics Institute, Middle East Technical University, Ankara, Turkey (2005)
- [11] Gencel, C., Demirors, O.: Functional Size Measurement Revisited. Scheduled for publication in ACM Transactions on Software Engineering and Methodology (July 2008)
- [12] GiveWin 2.10, <http://www.tspintl.com/>
- [13] ISBSG Dataset 10 (2007), <http://www.isbsg.org>
- [14] Hastings, T.E., Sajeev, A.S.M.: A Vector-Based Approach to Software Size Measurement and Effort Estimation. IEEE Transactions on Software Engineering 27(4), 337–350 (2001)
- [15] Jeffery, R., Ruhe, M., Wieczorek, I.: A Comparative Study of Two Software Development Cost Modeling Techniques using Multi-organizational and Company-specific Data. Information and Software Technology 42, 1009–1016 (2000)
- [16] Jones, T.C.: Estimating Software Costs. McGraw-Hill, New York (1998)
- [17] Jørgensen, M., Molokken-Ostvold, K.: Reasons for Software Effort Estimation Error: Impact of Respondent Role, Information Collection Approach, and Data Analysis Method. IEEE Transactions on Software Engineering 30(12), 993–1007 (2004)
- [18] Kitchenham, B., Mendes, E.: Software Productivity Measurement Using Multiple Size Measures. IEEE Transactions on Software Engineering 30(12), 1023–1035 (2004)
- [19] Briand, L.C., El Emam, K., Maxwell, K., Surmann, D., Wieczorek, I.: An Assessment and Comparison of Common Software Cost Estimation Models. In: Proc. of the 21st Intern. Conference on Software Engineering, ICSE 1999, Los Angeles, CA, USA, pp. 313–322 (1998)
- [20] Briand, L.C., Langley, T., Wieczorek, I.: A Replicated Assessment and Comparison of Software Cost Modeling Techniques. In: Proc. of the 22nd Intern. Conf. on Software engineering, ICSE 2000, Limerick, Ireland, pp. 377–386 (2000)
- [21] Menzies, T., Chen, Z., Hihn, J., Lum, K.: Selecting Best Practices for Effort Estimation. IEEE Transactions on Software Engineering 32(11), 883–895 (2006)
- [22] Leung, H., Fan, Z.: Software Cost Estimation. Handbook of Software Engineering, Hong Kong Polytechnic University (2002)
- [23] Angelis, L., Stamelos, I., Morisio, M.: Building a Cost Estimation Model Based on Categorical Data. In: 7th IEEE Int. Software Metrics Symposium (METRICS 2001), London (April 2001)
- [24] Forselius, P.: Benchmarking Software-Development Productivity. IEEE Software 17(1), 80–88 (2000)

- [25] Lokan, C., Wright, T., Hill, P.R., Stringer, M.: Organizational Benchmarking Using the ISBSG Data Repository. *IEEE Software* 18(5), 26–32 (2001)
- [26] Maxwell, K.D.: Collecting Data for Comparability: Benchmarking Software Development Productivity. *IEEE Software* 18(5), 22–25 (2001)
- [27] Morasca, S., Russo, G.: An Empirical Study of Software Productivity. In: *Proc. of the 25th Intern. Computer Software and Applications Conf. on Invigorating Software Development*, pp. 317–322 (2001)
- [28] Naur, P., Randell, B. (eds.): *Software Engineering, Conference Report*, NATO Science Committee, Garmisch (Germany), 7-11 October (1968)
- [29] Premraj, R., Shepperd, M.J., Kitchenham, B., Forselius, P.: An Empirical Analysis of Software Productivity over Time. In: *11th IEEE International Symposium on Software Metrics (Metrics 2005)*. IEEE Computer Society Press, Los Alamitos (2005)
- [30] Neter, J., Wasserman, W., Whitmore, G.A.: *Applied Statistics*. Allyn & Bacon (1992)
- [31] ISO/IEC 14143-1: Information Technology – Software Measurement – Functional Size Measurement – Part 1: Definition of Concepts (1998)
- [32] ISO/IEC 14143-1: Information Technology – Software Measurement – Functional Size Measurement – Part 1: Definition of Concepts (February 2007)
- [33] ISO/IEC 14143-2: Information Technology – Software Measurement – Functional Size Measurement - Part 2: Conformity Evaluation of Software Size Measurement Methods to ISO/IEC 14143-1:1998 (2002)
- [34] ISO/IEC TR 14143-3: Information Technology – Software Measurement – Functional Size Measurement – Part 3: Verification of Functional Size Measurement Methods (2003)
- [35] ISO/IEC TR 14143-4: Information Technology – Software Measurement – Functional Size Measurement - Part 4: Reference Model (2002)
- [36] ISO/IEC TR 14143-5: Information Technology – Software Measurement – Functional Size Measurement – Part 5: Determination of Functional Domains for Use with Functional Size Measurement (2004)
- [37] ISO/IEC FCD 14143-6: Guide for the Use of ISO/IEC 14143 and related International Standards (2005)
- [38] ISO/IEC 19761:2003, Software Engineering – COSMIC-FFP: A Functional Size Measurement Method, International Organization for Standardization(2003)
- [39] ISO/IEC IS 20926:2003, Software Engineering-IFPUG 4.1 Unadjusted Functional Size Measurement Method - Counting Practices Manual, International Organization for Standardization (2003)
- [40] ISO/IEC IS 20968:2002, Software Engineering – MK II Function Point Analysis – Counting Practices Manual, International Organization for Standardization (2002)
- [41] ISO/IEC IS 24570:2005, Software Engineering – NESMA functional size measurement method version 2.1 – Definitions and counting guidelines for the application of Function Point Analysis, International Organization for Standardization (2005)
- [42] Symons, C.: Come Back Function Point Analysis (Modernized) – All is Forgiven! In: *Proc. of the 4th European Conf. on Software Measurement and ICT Control (FESMA-DASMA 2001)*, Germany, pp. 413–426 (2001)
- [43] The International Function Point Users Group (IFPUG). *Function Points Counting Practices Manual* (release 4.2), International Function Point Users Group, Westerville, Ohio (January 2004)
- [44] United Kingdom Software Metrics Association (UKSMA). *MkII Function Point Analysis Counting Practices Manual*, v 1.3.1 (1998)
- [45] Thayer, H.R.: *Software Engineering Project Management*, 2nd edn. IEEE Computer Society Press, Los Alamitos (2001)

- [46] The Common Software Measurement International Consortium (COSMIC). COSMIC-FFP v.3.0, Measurement Manual (2007)
- [47] Gencel, C., Buglione, L.: Do Different Functionality Types Affect the Relationship between Software Functional Size and Effort? In: Proceedings of the Intern. Conf. on Software Process and Product Measurement (IWSM-MENSURA 2007), Palma de Mallorca, Spain, November 5-8, 2007, pp. 235–246 (2007)
- [48] FISMA, PAS Submission to ISO/IEC JTC1/SC7 – Information Technology – Software and Systems Engineering – FISMA v1.1 Functional Size Measurement Method, Finnish Software Metrics Association (2006), http://www.fisma.fi/wp-content/uploads/2007/02/fisma_fsmm_11_iso-final-1.pdf
- [49] CMMI Product Team, CMMI for Development, Version 1.2, CMMI-DEV v1.2, Continuous Representation, CMU/SEI-2006-TR-008, Technical Report, Software Engineering Institute (August 2006), <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>
- [50] Maxwell, K.: Applied Statistics for Software Managers. Prentice Hall, Englewood Cliffs (2002)
- [51] Santillo, L., Lombardi, S., Natale, D.: Advances in statistical analysis from the ISBSG benchmarking database. In: Proceedings of SMEF (2nd Software Measurement European Forum), Rome (Italy), March 16-18, 2005, pp. 39–48 (2005), <http://www.dpo.it/smef2005>