Assessing the Effectiveness of Approximate Functional Sizing Approaches for Effort Estimation

Sergio Di Martino[a], Filomena Ferrucci[b], Carmine Gravino[b], Federica Sarro[c]

[a] *Department of DIETI, University of Naples Federico II*
[b] *Department of Computer Science, University of Salerno*
[c] *Department of Computer Science, University College London*

Abstract

Context: Functional Size Measurement (FSM) methods, like Function Points Analysis (FPA) or COSMIC, are well-established approaches to estimate software size. Several approximations of these methods have been recently proposed as they require less time/information to be applied, however their effectiveness for effort prediction is not known.
Objective: The effectiveness of approximated functional size measures for estimating the development effort is a key open question, since an approximate sizing approach may miss to capture factors affecting the effort. Therefore, we empirically investigated the use of approximate FPA and COSMIC sizing approaches, also compared with their standard versions, for effort estimation.
Method: We measured 25 industrial software projects realised by a single company by using FPA, COSMIC, two approximate sizing approaches proposed by *IFPUG* for FPA (i.e. High Level and Indicative FPA), and three approximate sizing approaches proposed by the *COSMIC organisation* for COSMIC (i.e. Average Functional Process, Fixed Size Classification, and Equal Size Band). Then we investigated the quality of the regression models built using the obtained measures to estimate the development effort.
Results: Models based on High Level FPA are effective, providing a prediction accuracy comparable to the one of the original FPA, while those based on the Indicative FPA method show poor estimation accuracy. Models based on COSMIC approximate sizing methods are also quite effective, in particular those based on the Equal Size Band approximation provided an accuracy similar to the one of standard COSMIC.
Conclusion: Project managers should be aware that predictions based on High Level FPA and standard FPA can be similar, making this approximation very interesting and effective, while Indicative FPA should be avoided. COSMIC approximations can also provide accurate effort estimates, nevertheless, the Fixed Size Classification and Equal Size Band approaches introduce subjectivity in

Email addresses: sergio.dimartino@unina.it (Sergio Di Martino), fferrucci@unisa.it (Filomena Ferrucci), gravino@unisa.it (Carmine Gravino), f.sarro@ucl.ac.uk (Federica Sarro)

Published by Elsevier
Information and Software Technology
link to publisher version with DOI: https://doi.org/10.1016/j.infsof.2020.106308

the measurement.

Keywords: Software Effort Estimation, Functional Size Measures, Approximate Sizing Approaches, FPA, COSMIC

1. Introduction

In the software development domain, project managers need to quantify the "size" of a software product to be developed, since this is a key indicator of the development effort/cost expected. Relevant over- or under-estimates of the software size can have a strong impact on the success of a software project [1].

Functional Size Measurement (FSM) methods have been proposed to derive the size of a software from its Functional User Requirements (FURs) [2]. FSM methods have the benefits of being independent from the underlying programming technologies, and being applicable in the early stages of the development process.

The first FSM method was the Function Point Analysis (FPA) proposed in 1979 [3]. Since then, the method has evolved, being eventually standardized by ISO as the ISO/IEC 20926:2009. Also, several variants of FPA have been proposed (e.g., MarkII and NESMA) aiming at improving the accuracy of the measurements or extending its applicability to particular domains [4]. All these methods are collectively considered as the 1^{st} generation of FSM. In 1999, a novel FSM was proposed, i.e. COSMIC, which is considered the first FSM of 2^{nd} generation, due to significant evolutions in the measurement process over FPA. COSMIC complies with the fundamental principles of measurement theory and was designed to be suitable for a broader range of application domains [5, 6, 7, 8].

FSM methods have been extensively applied by practitioners and public agencies to size software systems. Nevertheless, there are some common managerial scenarios where they turn out to be difficult to apply, like:

- in the early phases of a project, when the FURs have not been specified yet at the required level of detail for a precise measurement, or when the documentation is not detailed enough;
- when the size measure is needed but there is insufficient time or resources to measure it using the standard method [9]. Indeed, FSM methods have been sometimes criticized to be time consuming in their application. Empirical studies on the productivity of FPA measurers reported that they proceed at a relatively slow pace, measuring between 200 and 600 Function Points (FPs) per day [10, 11], where it is not uncommon to have projects whose size is thousands of FP.

To support measurers in these scenarios, a number of FSM approximate sizing approaches (a.k.a. simplified size estimation methods) have been proposed (e.g., [4, 12, 13, 14, 15, 16]), with the goal to reduce the time and/or information needed for their application and, at the same time, able to provide an useful approximation of the functional size of the software being measured.

The International Function Points User Group (IFPUG)[1] published in 2015 a *Usage Tip* document dedicated to *Early Function Point Analysis and Estimation* [17], proposing two approximate sizing methods.The same was done by the COSMIC organization[2] with a first set of approximate sizing methods in 2007 [18], and in a further evolution, in 2015 [9].

Previous studies have investigated the effectiveness of approximate FSM methods by assessing whether the computed software size is comparable to the one achievable by the corresponding standard methods (e.g., [14, 19, 20, 21]), and most of the work provided a positive answer. Nevertheless, to the best of our knowledge, there is no study in the literature assessing the direct prediction of development effort, starting from approximate measures. Since correlation may not be always transitive [22, 23], to fill this gap in the literature, in this paper we investigate to what extent approximate measures can be used to estimate software development effort compared to those measures obtained using a standard method. This is a key aspect to investigate, since an approximate size, considering less information, may fail to measure some crucial aspects of a software system, thus biasing the development effort estimation [13]. Thus, in this work we empirically assess the effectiveness of 5 official FSM approximations in predicting software development effort.

In particular, we considered two approximations for FPA, namely *Indicative FPA* and *High Level FPA*, and three for COSMIC, namely *Average Functional Process*, *Fixed Size Classification*, and *Equal Size Bands*, plus the original FPA and COSMIC methods, applied to a dataset of 25 industrial applications developed by an Italian software company, containing also information on the actual development efforts. It is worth noting that this is the first study comparing these measures on a same dataset.

As a first step, we aimed at understanding whether these approximate sizing methods might be used in the context of an early and rapid development effort estimation. To this aim, as widely done in previous work, we built some effort estimation models using the size measures calculated with the 5 approximate sizing methods, and compared the accuracy of these estimates against some standard baselines widely used in the literature (e.g., [8, 24, 25, 26]). The rationale is that a sizing method is considered a good size measure for effort estimation if it leads to significantly better predictions than techniques which do not consider the software size as a cost driver [24, 27]. To this aim, we formulated the following first two research questions:

RQ1: Can FPA approximate sizing methods provide good early size measures for effort prediction?

RQ2: Can COSMIC approximate sizing methods provide good early size mea-

[1]IFPUG maintains the definition of FPA and certificates FP measurers: http://www.ifpug.org

[2]COSMIC Organization maintains the definitions of the COSMIC method: https://cosmic-sizing.org/

sures for effort prediction?

Furthermore, we investigated the differences in effort prediction accuracy when using approximate functional sizing methods with respect to the standard FPA and COSMIC. Indeed, we aimed at quantifying how much the use of approximate sizing measures, rather than an exact counting, might affect the estimates accuracy, which motivates two further research questions:

RQ3: Are size measures obtained using FPA approximate sizing methods as effective as Function Points, for effort estimation?

RQ4: Are size measures obtained using COSMIC approximate sizing methods as effective as COSMIC, for effort estimation?

To answer all these research questions, we used Simple Linear Regression (SLR) as estimation technique to build the prediction models. To evaluate and compare the accuracy of these models, we used the Mean of Absolute Residuals (MAR) [28, 29]. Furthermore, we verified if the differences in the absolute residuals obtained with the different estimation models were statistically significant, by using both statistical test and the effect size [30].

To answer RQ1 and RQ2, as done in previous work (e.g.: [8, 24, 25, 27]), we benchmarked the accuracy of the effort estimation models based on the investigated measures against two widely used baselines, i.e., the mean and the median of previous project efforts developed by a software company. The use of mean/median effort of past projects is often used as a baseline, since it does not require any actual sizing of the software to be developed. Rather, it relies on the idea that, if the mean/median effort of past projects gives a good indication of the expected effort of a new software to develop, there is no need to use more sophisticated methods. In our case, since all the investigated approximated methods involve a sizing phase of the new system, if they are not able to overcome the trivial mean/median of past projects, they cannot be considered good early size measures for effort prediction.

To answer RQ3 and RQ4, we compared the estimation accuracy of the models based on the approximate sizes against those obtained using the standard FPA and COSMIC methods.

Summarizing, this paper contributes to the body of knowledge by representing, to the best of our knowledge, the first study investigating the accuracy of a large number of different approximate sizing approaches proposed by the two main international organisations in functional sizing (IFPUG and COSMIC Organization) when used to directly predict software development efforts.

The rest of the paper is structured as follows: Section 2 provides the reader with background knowledge on FPA, COSMIC and their approximations, and on the related work. The design of our empirical study is described in Section 3, while its results and discussion are provided in Section 4. Conclusion and future work conclude the paper.

2. Background

In this section we describe the FSM methods compared in our empirical study (see Section 3) and discuss the related work.

2.1. Function Points Analysis

The Function Point Analysis (FPA) is the first FSM method proposed in the literature [3] as a measure (the Function Points) to quantify the "functionality" provided by a software, from the end-user point of view, independently from the underlying technologies. The original FPA formulation was then revised in 1983 [31], and in 1986 FPA became managed by the *International Function Point Users Group* (IFPUG) [32]. Since then, the method is named IFPUG FPA (or simply IFPUG, for short) and has been standardized by ISO as ISO/IEC 20926:2009.

More in details, with IFPUG, the size of a software system can be considered as the (weighted) sum of simple unitary elements (its FURs), whose measurement is easier than the whole system. In particular, to identify the set of functionalities provided by the software, each FUR is decomposed into Base Functional Components (BFC), which can be of the following five types:

- Internal Logical Files (ILF) are logical, persistent entities maintained by the application to store information of interest.
- External Interface Files (EIF) are logical, persistent entities that are referenced by the application, but are maintained by another software application.
- External Inputs (EI) are logical, elementary business processes that cross into the application boundary to maintain the data on an Internal Logical File.
- External Outputs (EO) are logical, elementary business processes that result in data leaving the application boundary to meet a user requirements (e.g., reports, screens). To qualify as EO, the processing logic must contain at least one mathematical formula or calculation, create derived data, change the behaviour of the system, or maintain one or more ILFs [32].
- External Inquires (EQ) are logical, elementary business processes that consist of a data trigger followed by a retrieval of data that leaves the application boundary (e.g., browsing of data). The processing logic of an EQ must not contain mathematical formulas or calculations, nor create any derived data or change the behaviour of the system, and no ILF is maintained [32].

The first two types deal with data storage and are called *Data BFC*. The other three are referred to as *Transactional BFC*, dealing with business processes.

As next step, the "complexity" of each BFC is assessed, by characterizing further attributes, such as the number of Data Element Types (DETs) and Record Element Types (RETs) handled within each BFC. More in details, a DET is defined as a "unique user recognizable, non-repeated field" [32], while a RET is a "user recognizable subgroup of data elements within an ILF or EIF" [32]. A proper characterization of DETs and RETs is a non-trivial task, requiring time and experience.

Once these two information have been derived, they must be combined with a reference table provided in the IFPUG method [32], which indicates the "complexity" of each BFC, like in Table 1. Then, another table provided in the IFPUG method provides the correspondences between the complexity of a BFC, and its size in terms of Function Points (FPs). As an example, an ILF with Low complexity yields to 7 FPs, a Medium to 10 FPs, and a High complexity to 15 FPs. The sum of all the FPs for all the identified BFCs gives the total functional size of the software system.

Table 1: Table to compute the complexity of a ILF/EIF BFCs, given the number of DETs and RETs

	DETs		
RETs	1-19	20-50	>=51
1	Low	Low	Medium
2-5	Low	Medium	High
>5	Medium	High	High

For more details about the use of the IFPUG method, readers may refer to the counting manual [32].

2.2. FPA Approximate sizing methods

Many FPA approximations have been proposed over the years, from different institutions, companies and researchers. In our investigation we focused on the two proposals by IFPUG, the organization that endorses the FPA standard. More in details, in 2015 IFPUG published a guide, including a basic empirical study, on how to approximate the application of FPA, depending on the amount of information and time available to the measurer [17]. These two methods are detailed in the following.

2.2.1. High Level FPA

High Level FPA is suited to be applied in the case where the BFCs to be sized are identified, but details on their DETs and RETs are not available, due to time constraints or missing details in FURs. In this case, IFPUG suggest to proceed as follows:

- determine all the Base Functional Components of the system to be measured (ILF, EIF, EI, EO, EQ);
- rate the complexity of all the ILFs and EIFs as Low;

- rate the complexity of all the EI, EO, EQ as Average;
- assign the corresponding Function Points according to the standard IFPUG tables, and accumulate.

Thus, the key difference between the High Level FPA and the standard FPA is that the complexity of each BFC, a time consuming task, is no more computed on the actual system, but rather it is assigned by default, according to previous statistical evidence [17].

2.2.2. Indicative FPA

Indicative FPA is meant to be applied when only a very limited amount of information on the system to be measured is available. Thus, rather than providing a size of the software, it is more meant to produce very quickly a rough estimate of its size, to be intended as a ROM (Rough Order of Magnitude) [17]. In this case, IFPUG suggests to proceed as follows:

- determine only all data functions (ILF, EIF);
- the indicative functional size of the software is 35 x number of ILFs + 15 x number of EIFs.

This function has been proposed by IFPUG based on a projected ratio of Transactional BFC for each Data BFC . According to IFPUG, *"experience has shown that it is a suitable approximation"* [17].

2.3. COSMIC

Even if there is a large empirical evidence of the general effectiveness of FPA, it is worth noting that this method is not compliant with the measurement theory. Indeed, some steps of the process improperly mix different types of scales [33], as well as the way the "weights" of the BFCs are defined in the reference tables has been object of large discussion in the literature (e.g., [34, 35]).

The COSMIC method is a FSM method proposed in 1999 to overcome some of the limitations of FPA. Indeed, the basic idea underlying the COSMIC method is that, for many types of software systems, the most of the development efforts are due to the implementation of proper functions to move data from/to persistent storage and from/to users. Thus, the number of data movements can be a meaningful predictor of the final system size [5], expressed in COSMIC Function Point (CFP), the COSMIC measurement unit. More in details, the sizing process defined by the COSMIC method consists of the three following phases:

1. The *Measurement Strategy* phase defines the *purpose* of the measurement, the *scope* (i.e., the set of FURs to measure), the *functional users* of each piece of software (i.e., the intended senders/recipients of data to/from the software to be measured), and the *level of granularity* of the available artefacts.

2. The *Mapping Phase* is a crucial process, where each FUR is expressed in the form required by the *COSMIC Generic Software Model*. This model, needed to identify the key elements of a FUR to be measured, assumes that (I) each FUR can be mapped into a unique *functional process*, meant as a cohesive and independently executable set of data movements, (II) each functional process consists of sub-processes, and (III) each sub-process may be either a data movement or a data manipulation. As an approximation for measurement purposes, data manipulation sub-processes are not separately measured; the functionality of any data manipulation is assumed to be accounted for the data movement associated with it. Moreover, to measure data movements, three other concepts have to be identified: (I) a *Triggering Event*, i.e., an action of a functional user triggering one or more functional processes, (II) a *Data Group*, i.e., a distinct set of data attributes, where each attribute describes a complementary aspect of the same object of interest, and (III) a *Data Attribute*, i.e., the smallest piece of information, within an identified data group, carrying a meaning from the perspective of the interested functional user. Data movements 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.
3. The *Measurement Phase* identifies and counts the data movements of each functional process. Each E, X, R or W is counted as 1 COSMIC Function Point (CFP). Thus, the size of the application within a defined scope is given by the sum of the sizes of all the functional processes within the scope.

For more details about the COSMIC method, readers are referred to the COSMIC Measurement Manual [5].

2.4. COSMIC Approximate sizing methods

As for FPA, also for COSMIC some Approximate sizing methods have been proposed in the literature (e.g. [36]). In our analysis we focused on three approximations proposed by the COSMIC Organization in the official COSMIC method documentation [37], i.e., the Average Functional Process (AFP), the Fixed Size Classification (FSC), and the Equal Size Bands (ESB). They are detailed in the following.

2.4.1. Average Functional Process

Average Functional Process (AFP) is the first approximation suggested by the COSMIC community to obtain early size estimations of the software. It is a data-driven approach, requiring information from a set of previous projects measured with the full COSMIC method to speed up the measurement of a new project, whose actual requirements are known only in terms of functional processes, but not of data movements. It consists of two macro steps, as follows:

1. Sampling and calculation of the size of an average functional process
   (a) Identify a sample of requirements of past projects, already measured with the standard COSMIC method, whose characteristics are similar to the actual requirements of the software to be approximately measured.
   (b) Identify the functional processes of these sample requirements, and determine their average size in CFP (e.g., average size = 8 CFPs. 8 is then the *scaling factor* for this approach).
2. Approximation using the calculated average of the sample
   (a) Identify and count all the functional processes of the actual requirements of the software to be measured (e.g., = 40 functional processes).
   (b) The approximate functional size of the software to be measured is approximated by *number of functional processes x scaling factor* (e.g., $40 \times 8 = 320$ CFPs).

Let us remark that this approximate sizing method requires that some previous projects have been fully measured using the standard COSMIC method within the same company and the same domain.

2.4.2. Fixed Size Classification

The AFP approximation assumes that all the functional processes have the same size, equal to the average size of the functional processes of similar requirements in past projects. The Fixed Size Classification (FSC) approximation is an evolution of AFP, letting the measurer to subjectively specify a size class (e.g., Small, Medium, or Large) for each functional process to be measured. Since a corresponding size in CFP, or scaling factor, is assigned to size class, like in Table 2 [37], it is very fast to derive an approximate total size of the software system to be measured. As an example, using Table 2, once a functional process has been classified by the Measurer as "Small", it is assigned a scaling factor of 5, so that its size is 5 CFP.

The COSMIC manual suggests that, to support the Measurer in making a conscious choice of size, the step between the classes should be taken to be fairly wide, like 5 CFP [37]. When well calibrated, this approach should give a more accurate functional size than the AFP method [37].

According to the COSMIC manual, the FSC approach has been extensively and successfully used by a large business organization in the Netherlands, but no public information on the accuracy of this approximation is available in the literature [37].

Table 2: List of Size Classes of Functional Processes, with corresponding Size in CFP. If necessary, the table may be extended with one or more additional sizes, such as very large of 20 CFPs, and so on [37].

Functional Process Classification	Size in CFP
Small	5
Medium	10
Large	15

2.4.3. Equal Size Bands

The Equal Size Bands (ESB) approximation can be seen as an improvement of FSC when sufficient size information from past project is available for an accurate calibration relevant to the new measurement. Like the FSC approach, also ESB classifies each functional process into one of a small number of size classes, or *bands*. The difference is that ESB choses the reference sizes of the bands in a calibration process based on past data, so that the total size of all the functional processes in each band is the same. As an example, if we define to have three bands, then the total size of all the functional processes in each band will be the 33% of the total size of the software being measured. Thus, also ESB is a data-driven method, requiring a set of previous projects developed by the same company in the same domain and measured with the standard COSMIC method.

According to the COSMIC manual, the ESB approach can lead to more accurate result than FSC, if sufficient size data is available for an accurate calibration [37]. Vogelezang and Prins did a deep investigation on this approach, discussing the distribution of functional processes over different classes with calibration using measurements on 37 business applications, each of total size greater than 100 CFP [15].

2.5. Further Related Work

In the literature, several studies have investigated the effectiveness of FSM methods to predict software development effort. The results in general show that FSM are correlated with effort, thus being effective predictors of the development effort. Another common result is that usually COSMIC can provide better estimates than FPA (see e.g., [8][38][39][40][41][42][43][44]).

As for approximate FSM methods, many proposals have been done in the past. A comprehensive review of the literature was done by Morrow et al. [20], where the authors also empirically assessed the effectiveness of two simplifications of the NESMA variant of FPA, in terms of correlation with the full NESMA method. In general, only very few studies have empirically investigated the use of approximate size measures for the managerial task to estimate the development effort of a software system. Popovic et al. [45], investigated the effectiveness of several FSM to predict effort estimation on 30 software projects, including also an approximation of NESMA. However, the main aim of this paper was to show the great potential of using functional measurement for effort

prediction, able to achieve an estimate accuracy close or even better than accuracies found in the practice [45]. Ohiwa et al. [46] used a data set containing data from 36 projects collected by the Economic Research Association from 2008 through 2012, finding that there was a good correlation between development effort and size estimates provided by a NESMA approximation. On the other hand, most of the papers dealing with variants of FSM methods focus on evaluating the accuracy of these measurements, by comparing these sizes with the ones calculated using the original methods [47]. Furthermore, in the majority of cases the accuracy of the proposed approaches has been documented mainly by the researchers who defined the approximate methods. For example, van Heeringen et al. [14] verified the accuracy of *Estimated NESMA* and *Indicative NESMA* methods and the results of their analysis revealed that the NESMA estimated approach and the COSMIC Equal Size Bands approach provided accuracy results that are comparable with those achieved fully applying the same FSM method [14].

Other two studies assessed functional size estimation techniques like *FP Prognosis* [48] and *Early & Quick Function Point* method [49]. As a consequence, there is a lack of independent evaluations of approximate FP estimation methods. Lavazza and Liu performed an empirical study to compare the accuracy of some approximate estimation methods [47] showing that only a few of them (i.e., NESMA indicative, Early & Quick Function Points and ISBSG average weights) provide size estimations characterized by a 10% sizing errors on average.

Regarding the use of approximate COSMIC methods, to the best of our knowledge there is only one study investigating their accuracy in correlation with estimating the development effort [36], where the authors empirically validated the AFP approximation on the same dataset we use herein, revealing that AFP led to significantly worse effort estimates than those obtained by using the COSMIC method [36]. Other studies have evaluated the effectiveness of COSMIC approximations for estimating the COSMIC functional size of an application versus measuring it [50, 51]. As an example, recently Lavazza and Morasca have assessed the Average Functional Process and the Equal Size Bands methods as well as two new approaches for defining bands in the Fixed Size Classification method [50]. The methods were evaluated on a set of applications previously measured according to the COSMIC method, so that the information to apply approximations was available. The results of the performed analysis show that the method using bands provide good estimations and the level of their accuracy can be obtained based on the number of bands used and by quantifying the ability to classify each functional process in the correct band. Differently, for Average Functional Process method the analysis revealed that its estimation errors are too large to be acceptable [50].

There is also a line of work investigating how to obtain model-based early and rapid estimation of COSMIC functional size as the work by del Bianco *et al.* [52] and by De Vito and Ferrucci [16].

As a consequence, the present paper is the first study empirically assessing the effectiveness of a number of official FSM approximations in predicting the

software development effort.

3. Empirical Study Design

In this section we describe the design of the empirical study. We present the dataset and the estimation technique used. Then, we formalize the null hypotheses for our research questions, followed by details on the validation method and evaluation criteria, and describe the main threats to the validity of the study.

3.1. Dataset

The data used in our study was obtained from an Italian medium-sized software company, which mainly develops enterprise information systems for local and central government (e.g., health organizations, research centres, industries, and other public institutions). In particular, this software company is specialized in the design, development, and management of solutions for Web portals, enterprise Intranet/Extranet applications (such as Content Management Systems, e-commerce, work-flow managers, etc.), and Geographical Information Systems. It has about fifty employees, it is certified ISO 9001, and it is also a certified partner of Microsoft, Oracle, and ESRI. From this company, we obtained information on 25 Web applications they developed in the past years, such as e-government, e- banking, Web portals, and Intranet applications. The technologies exploited for the development were SUN J2EE or Microsoft .NET technologies. Oracle was the most commonly adopted DBMS, but also SQL Server, Access and MySQL were employed in some of these projects.

In order to collect the data we needed, time sheets were used to keep track of the Web application development effort. In particular, for each project and every day, each team member annotated the information about his/her development effort, and weekly each project manager stored the sum of the efforts for the team. Furthermore, we defined a template to be filled in by the project managers, to obtain from the company all the significant information needed to calculate the values of the size measure in terms of FPA, COSMIC and their approximations. To avoid misunderstandings, the project managers were trained on the use of the questionnaires. To cross-check the provided data, one of the researchers involved in this experimentation analysed the filled templates and the software documentation of the involved projects. The same researcher calculated the values of the size measure in terms COSMIC method and its approximations. With regard to the calculation of the size in terms of FPA, the project managers of the software company are highly skilled on this method, having always applied this size to measure their applications. One of the researchers involved in this experimentation calculated the approximate FPA sizes, using documentation and information provided by the project managers. In the threats to validity section further considerations about the collection of the data will be done.

In Table 3 we report some summary statistics for the 25 software systems we used in the experiments, in terms of development effort and the seven considered size measures.

Table 3: Descriptive statistics of variables

Variable	Content	Min	Max	Mean	Median	St.Dev.
EFF	Total actual development effort, in person-hours	782	4537	2577	2686	988.136
FP	Size in Function Points by applying the standard FPA method	110	973	399.96	336	216.366
HLFPA	Size in Function Points by applying the High Level FPA approximate sizing method	123	1043	442.2	360	235.649
IFPA	Size in Function Points by applying the Indicative FPA approximate sizing method	105	1610	311.2	190	345.499
CFP	Size in COSMIC by applying the standard COSMIC method	163	1090	602.04	611	268.473
CFPafp	Size in COSMIC by applying the AFP approximate sizing method	183	1417	610.8	545	299.676
CFPfsc	Size in COSMIC by applying the Fixed Size Classification approximate sizing method	180	1335	650.6	640	295.481
CFPesb	Size in COSMIC by applying the Equal Size Bands approximate sizing method	133	942.5	518.5	513	232.041

3.2. Estimation Technique

To build the effort prediction models based on the considered size measures, we employed as an estimation technique the Simple Linear Regression (SLR), which is a model-based approach widely and successfully employed in the industrial context and in several research work to estimate development effort (see e.g., [8][38][43][53][54])[3].

SLR allows us to build models explaining the relationship between the independent variable, i.e. the employed size measure, and the dependent variable, i.e. the development effort. Thus, SLR allows us to obtain models of this type:

$$EFF = a + b \times Size \tag{1}$$

where EFF is the dependent variable, $Size$ is the independent variable (e.g., CFP), b is the coefficient that represents the amount the variable EFF changes when the variable $Size$ changes 1 unit, and a is the intercept.

Once such a model is obtained from a dataset of past projects, given a new software project for which an effort estimation is required, the manager has to size it using the same unit of measure of the model, and to use this value in the regression equation to get the effort prediction.

Note that when variables were highly skewed they were transformed in order to comply with the assumptions underlying linear regression [68] (i.e., linearity between dependent and independent variable, normality and statistical independence of error terms, constant variance of error terms). In particular, we applied a log-transformation (i.e., natural log [69]), thus introducing a variable $Log(X)$ if the original variable was X.

[3]We did not use other estimation techniques, such as machine learners [55, 56], search-based approaches [57, 58, 59, 60, 61, 62, 63, 64] and their combination [65, 66, 67], since this work aims at comparing FSM methods rather than estimation techniques.

To evaluate the goodness of fit of a regression model, several indicators can be used. Among them, we considered the square of the linear correlation coefficient, R^2, shows the amount of the variance of the dependent variable explained by the model related to the independent variable.

The coefficient of determination normally ranges from 0 to 1 and the higher the value the higher the goodness of fit of the model. Other useful indicators we used, are the F value and the corresponding p-value (denoted by $SignF$), whose high and low values, respectively, denote a high degree of confidence for the prediction (i.e., there is a relationship between our independent and dependent variables) [68] [70]. A practical interpretation is that if R^2 is big, i.e., near 1 (meaning that a linear model fits the data well) then the corresponding F statistic should be large, meaning that we should have a strong evidence that the coefficient of the independent variable is non-zero [71]. The further the F value is from 1 the better it is. However, how much larger the F-value should be depends on both the number of predictors (in our case 1) and the number of data points (in our case 25). Generally, when the number of data points is large, an F-statistic that is only a little bit larger than 1 is already sufficient [70] [72].

We also verified the stability of each SLR model, by analysing the presence of influential data points (i.e., extreme values which might unduly influence the models obtained from the regression analysis). As suggested in [27], we further analysed the residuals plot and used Cook's distance to identify possible influential observations. In particular, the observations in the training set with a Cook's distance higher than $4/n$ (where n represents the total number of observations in the training set) were removed to test the model stability, by observing the effect of their removal on the model. If the model coefficients remained stable and the adjusted R^2 improved, the highly influential projects were retained in the data analysis. This is a common procedure applied in previous work (e.g., [42] [66] [73]).

3.3. Baselines

Regarding the baselines, we used two constant models, as done in several previous studies (e.g., [24, 66, 74, 75, 76, 77]). The first constant model, *MeanEFF*, considers the mean of the previous 24 project efforts (namely, EFF_{o_1}, ..., $EFF_{o_{i-1}}$, $EFF_{o_{i+1}}$, ..., $EFF_{o_{25}}$) as the predicted effort, for each observation o_i in the dataset (for $i = 1, ...25$), while the second model, *MedianEFF*, exploits the median of the previous 24 project efforts.

This was done because, as suggested by Mendes and Kitchenham [69] [27], if an estimation method does not outperform the results achieved by using very naive benchmarks, such *MeanEFF* and *MedianEFF*, it cannot be transferred to industry. Indeed there would be no value for a company in dealing with sophisticated estimation methods to predict development effort compared to simply using the average effort of its own past projects as the estimated effort for a new project.

All the models employed in our study are summarized in Table 4.

Table 4: Estimation Models Compared in Our Empirical Study.

Model Name	Brief Description	Equation
MFP	Model based on the size computed with FPA	EFF = a + b × FP
MHLFPA	Model based on the size computed with HLFPA	EFF = a + b × HLFPA
MIFPA	Model based on the size computed with IFPA	EFF = a + b × IFPA
MCFP	Model based on the size computed with COSMIC	EFF = a + b × CFP
MCFPafp	Model based on the size computed with AFP	EFF = a + b × CFPafp
MCFPfsc	Model based on the size computed with FSC	EFF = a + b × CFPfsc
MCFPesb	Model based on the size computed with ESB	EFF = a + b × CFPesb
MeanEFF	Model based on mean of effort of previous projects	EFF = $Mean\{EFF_{o_1}, ..., EFF_{o_{25}}\}$
MedianEFF	Model based on median of effort of previous projects	EFF = $Median\{EFF_{o_1}, ..., EFF_{o_{25}}\}$

3.4. Null Hypotheses

The first two research questions have been defined to perform a sanity check, by verifying that the models based on the approximate sizes provide significantly better effort estimations than the simple baselines described in Section 3.3. To this end, we formulated the following null hypotheses for each considered models X∈{MHLFPA, MIFPA, MCFPafp, MCFPesb, MCFPfsc}:

Hn1$_X$: The effort predictions obtained with X are not statistically significantly better than those achieved with *MeanEffort*.

Hn2$_X$: The effort predictions obtained with X are not statistically significantly better than those achieved with *MedianEffort*.

The alternative hypotheses are:

Ha1$_X$: The effort predictions obtained with X are statistically significantly better than those achieved with *MeanEffort*.

Ha2$_X$: The effort predictions obtained with X are statistically significantly better than those achieved with *MedianEffort*.

Thus, given a model X, we can positively answer to RQ1 (RQ2, respectively) if can reject Hn1$_X$ and Hn2$_X$ for X.

The third and fourth research questions have been defined to verify the effectiveness of the approximations with respect to their standard methods. Thus, we formulated the following null hypotheses:

Hn3$_Y$: There is not statistically significant difference between the effort predictions obtained with MFP and Y.

Hn4$_Z$: There is not statistically significant difference between the effort predictions obtained with MCFP and Z.

where Y∈{MHLFPA, MIFPA} and Z∈{MCFPafp, MCFPfsc, MCFPesb}

The alternative hypotheses are:

Ha3$_Y$: There is statistically significant difference between the effort predictions obtained with MFP and Y.

Ha4$_Z$: There is statistically significant difference between the effort predictions obtained with MCFP and Z.

Thus, we can positively answer to RQ3 (RQ4, respectively) when:

- we cannot reject $Hn3_Y$ ($Hn4_Z$, respectively).
- we can reject $Hn3_Y$ ($Hn4_Z$, respectively) and the errors in the predictions obtained using Y (Z) are less than the ones achieved with MFP (MCFP, respectively).

We cannot positively answer the third (and forth) research question if the model based on a FPA (COSMIC) approximation provides significantly worse effort estimations than the model based on standard FPA (COSMIC).

3.5. Validation Method and Evaluation Criteria

To verify whether the values obtained with the prediction models are useful estimates of the actual development effort, we carried out a cross validation, which means that the original dataset was divided into different subsets of training and validation sets. Training sets are used to build SLR estimation models, and the corresponding validation sets are used to validate these models. In particular, we applied a leave-one-out cross validation, which means that the original data set of 25 observations was divided into 25 different training sets (containing 24 data points each) and 25 different validation sets (containing one data point each).

For each validation set i (for i=1...25), we calculated the absolute residual as follow:

$$AbsoluteResidual(AR)_{o_i} = |EFF_{o_i} - Predicted_{o_i}| \quad (2)$$

where EFF_{o_i} is the actual development effort of the observation o_i in the validation set i, and $Predicted_{o_i}$ is the predicted effort for o_i, using the estimation model built on the training set i.

As for the evaluation criteria, we employed the Mean of the Absolute Residuals (MAR) obtained for the 25 observations in the dataset, as done in many similar works (e.g., [24][25][26]).

To answer our research questions, we tested the null hypotheses by applying the T-test on the distributions of the obtained absolute residuals and the Wilcoxon signed rank test when absolute residuals were not normally distributed [78]. To verify the normality of distributions, we used the Shapiro Wilk Test [79], by considering as a null hypothesis the normality of error terms. So, to test $Hn1_X$ ($Hn2_X$, respectively) we applied the statistical test on the distribution of the 25 absolute residuals obtained with X∈{MCFPafp, MCFPesb, MCFPfsc, MHLFPA, MIFPA} and the distribution of the 25 absolute residuals achieved with MeanEFF (MedianEFF, respectively). Similarly, to test $Hn3_Y$ and $Hn4_Z$ we applied the Wilcoxon test.

Table 5: Effect size classification

Effect size	A12 statistics
small	over 0.56
medium	over 0.64
large	over 0.71

For all the statistical tests performed in our analysis, we decided to accept a probability of 5% of committing a Type-I-Error, as customary in Software Engineering empirical studies [80]. Furthermore, since we formulated different hull hypotheses for answering each of the research questions (and thus performed multiple statistical tests, e.g., four in case of RQ1), we have applied a Bonferroni correction in order to classify the p-values as significant (i.e., to reject the null hypotheses and positively answer the research question) [78, 81]. As a consequence, if n is the number of tests performed, the correction applied is: $\alpha_{cor} = \frac{0.05}{n}$ (e.g., $\frac{0.05}{4} = 0.0125$ in case of RQ4).

To have also an indication of the practical/managerial significance of the results, we verified the effect size. Effect size is a simple way of quantifying the standardized difference between two groups. In our analysis we considered the Vargha and Delaney's $\hat{A}_{12}$ statistics as non-parametric effect size measure [82]. According to Vargha and Delaney, a difference between two populations can be classified in small, medium, and large as in Table 5; an effect size lower than or equal to 0.56 can be considered negligible. Since we are interested in *any* improvement in predictive performance, no transformation of the $\hat{A}_{12}$ is needed [26, 58, 83, 84].

3.6. Threats to Validity

The way the measurement of the functional size is accomplished represents a crucial task in studies similar to ours. Indeed, the collection of the information to calculate FPA (and its approximations), COSMIC (and its approximations), and Effort can bias the construct validity.

As mentioned in Section 3.1, the software company involved in our study has always applied FPA to measure its applications. So, we obtained all the information for the single BFCs (i.e., ILF, EIF, EI, EO, and EQ) of the FPA analysis and computed HLFPA and IFPA sizes as explained in Section 2.

Concerning the use of COSMIC approximations, a possible threat can be related to the way we applied the AFP. Indeed, for each step of the leave-out cross validation, we took into account all the requirements of the projects in the training set, rather than choosing only those more similar to the ones to be measured. However, in this way we did not introduce subjectivity issues due to an arbitrary selection. ESB and FSC measure the functional processes by classifying them in three classes, i.e., small, medium, or large. Thus, a possible threat can be related to this classification. In our case, we have verified that we misclassified slightly more than 10% of the functional processes in the application of ESB and FSC. Furthermore, we are aware that some measurements were

not performed as part of the company projects but for the research purpose by the authors.

We only consider those functional size measurement methods familiar to either the project managers of the company and the researchers involved in this study. Other FSM methods, such as NESMA, can be taken into account in future work in order to consolidate the results achieved here.

Another possible threat regards the subjectivity introduced when performing the measurement. As described in Section 3.1, a single researcher involved in this experimentation calculated the values of size measure in terms of COSMIC method and its approximations as well as the values of size measure in terms of FPA approximations (using documentation and information provided by the project managers). Thus, his level of expertise may have influenced the comparisons between approximate methods, particularly for the approximate COSMIC methods. Since the project managers of the company involved in our study did not have previous experience for these kinds of measurement we could not involve them in this task. Assessing the impact of subjectivity in measurement could be subject of future investigation. To this end a number of developers/managers can be involved to perform the measurement task, with the aim of analysing the variation of their size estimations on a given project and of quantifying the impact of subjectivity on comparing different of FSM methods (and their approximations).

Regarding the collection of the effort data, each project team member annotated the information about his/her development effort every day, while each project manager stored the sum of the efforts for the team weekly. Thus, possible threats to effort data collection have been mitigated by the procedure in place.

With regard to conclusion validity, we verified all the required assumptions and carefully applied the linear regression analysis to built the estimation models and the statistical tests to analyze differences in the distributions of errors. We are aware that the number of observations in our dataset is not so high and it could represent another threat to conclusion validity. Nevertheless, we want to point out that some researchers have proposed: "A rule of thumb in regression analysis is that 5 to 10 observations are required for every variable in the model" [53]. On the other hand, the fact we did not reject some null hypothesis could be the result of the low power of the tests as the number of observations is small. However, as widely recognized in the empirical research field, replications, using also larger datasets, should be performed to confirm results of a study.

To mitigate possible threats to internal validity, we did not applied particular selections of the Web applications from the software company. With the aim of dealing with the reliability of the data and lack of standardization, the same questionnaires were used to collect the information to calculate the sizes in terms of the considered measures and the development efforts for all the Web applications in our dataset. Furthermore, we instructed the project managers on how to fill them in and correctly report the information required.

With regard to external validity, we think that the type of analyzed Web applications did not bias validity of the achieved results. Indeed, taking into

account their functionality and complexity the Web applications considered in our study can be seen as representative sample. However, it is recognized that the results obtained for a given software company might not hold in others since each company might be characterized by some specific project and human factors, such as development process, developer experience, application domain, tools, technologies used, time, and budget constraints [71]. We want also to observe that the historical data of a given company and the way it was collected can impact the variability of the effort data as well as of the calculated approximated functional sizes, which can differ from those characterizing other companies. Thus, the application of measurement methods and effort collection of the specific company can influence the relationships between size and effort. Another threat could be related to the fact that our results cannot be generalized to those datasets collecting data from multiple different companies (i.e., cross-company datasets) [85]. To mitigate this possible threat we advocate replications of our study using data collected by other (single- and cross)-companies.

4. Empirical Study Results

In this section we present the results of the empirical analysis we carried out to answer our research questions.

4.1. RQ1: FPA approximate sizing methods vs. Baselines

The first results are about the comparison of the effort predictions obtained by the SLR models built on the approximations of FPA (MHLFPA and MIFPA, respectively) with those achieved by the constant models MeanEFF and MedianEFF.

We started by verifying the assumptions underlying SLR, namely the existence of a linear relationship between the independent variable and the dependent variable (*linearity*), the constant variance of the error terms for all the values of the independent variable (*homoscedasticity*), the normal distribution of the error terms (*normality*), and the statistical independence of the errors, in particular, no correlation between consecutive errors (*independence*).

All the tests performed (intended as statistically significant at 95% confidence level) and analysis are reported in the Appendix. Then, as designed (see Section 3.2), we verified the presence of influential observations by using the residuals plot and Cook's distance. In particular, the observations in the training set with a Cook's distance higher than $4/n$ (where n represents the total number of observations in the training set) were removed to test the model stability, by observing the effect of their removal on the model. If the model coefficients remained stable and the adjusted R^2 improved, the highly influential projects were retained in the data analysis. From the results of this analysis, no observations were removed from the original data set.

The models obtained are reported in Table 6, together with the values of the indicators used to assess the model. We can observe that MHLFPA and

Table 6: Results of SLR for RQ1 and RQ3

Model	Equation	R^2	F	Sign. F
MHLFPA	EFF = $74.9 \times \text{HLFPA}^{0.58}$	0.501	23.1	<0.001
MIFPA	EFF = $539.99 \times \text{IFPA}^{0.272}$	0.17	4.701	<0.001
MFP	EFF = $87.42 \times \text{FP}^{0.56}$	0.472	20.55	<0.001

Table 7: Results in terms of MAR for RQ1 and RQ3

MHLFPA	MIFPA	MeanEFF	MedianEFF	MFP
561	760	843	872	580

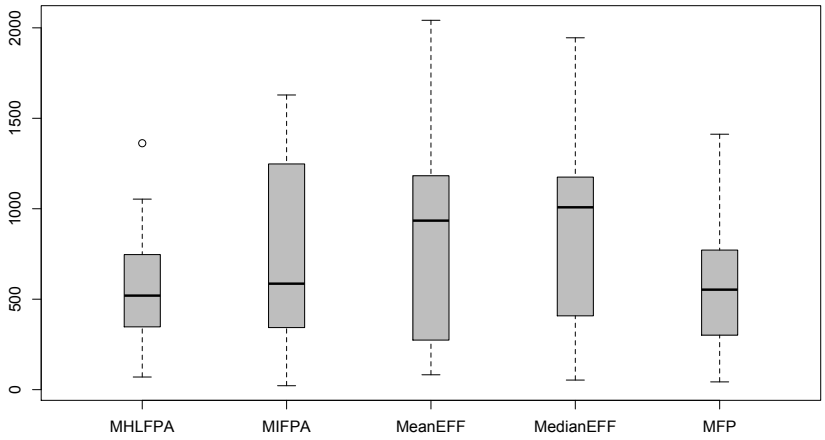


Figure 1: Boxplots of absolute residuals for RQ1 and RQ3.

MIFPA are characterized by a Sign F value lower than 0.05, that is, the models are significant. However, the R^2 and F values are quite low.

To compare FPA approximations and the constant models MeanEFF and MedianEFF, we performed a leave-one-out cross-validation as described in Section 3.5, whose results, in terms of MAR, are reported in Table 7. We can observe that MHLFPA provided good results. Indeed, the obtained MAR is much lower than the one achieved by the baselines. The scenario is different for MIFPA, as it provides only marginally better results than the baselines.

These trends are graphically confirmed by the boxplots of absolute residuals, shown in Figure 1. Indeed, on one hand, the boxplot of MHLFPA is much smaller, has a median closer to zero, and the box length and tails are shorter than the ones of the baselines. On the other hand, the box length and tails of boxplot for MIFPA are quite similar to the ones of the baselines, thus confirming that the difference in terms of effort predictions between MIFPA and the baselines

Table 8: Wilcoxon test p-values (and effect size between brackets) for RQ1 and RQ3

	MeanEFF	MedianEFF	MFP
MHLFPA vs	0.009 (small)	0.004 (small)	0.325 (negligible)
MIFPA vs	0.474 (negligible)	0.339 (negligible)	0.101 (negligible)

is small.

Also the statistical significance tests of the distribution of absolute residuals confirm the above findings. In particular, the tests reveal that $Hn1_X$ and $Hn2_X$ can be rejected for X=MHLFPA, i.e., the residuals of HLFPA are significantly smaller than MeanEFF and MedianEFF (i.e., p-values are less than 0.05/4=0.013), however with a small effect size (see Table 8). Differently, $Hn1_X$ and $Hn2_X$ cannot be rejected for X=MIFPA, since the Wilcoxon test p-values are greater than 0.013 (see Table 8). The effect size in this case is negligible. Concluding, we cannot positively answer our first research question for both FPA based approximations:

> RQ1: Only HLFPA approximate sizing approach provided good early size estimations, for effort prediction.

4.1.1. Discussion

From these findings, the main conclusion is that HLFPA is able to provide significantly better predictions than the models used as baselines yet with small effect size, thus it can be used to measure the projects delivered by the company in our investigation.

On the other hand, IFPA led to poor results on our dataset, with errors in the effort predictions that are comparable to those of the basic constant models. Thus, its adoption is not advisable for the company in our investigation. A possible explanation for these poor results might be found in the components of a software system that are measured by the IFPA approximate sizing method. Indeed, this approximation requires only to count the *Data BFC* (i.e., ILF and EIF), with a strong emphasis on the first one (i.e. ILFs managed by the application), that are multiplied by a factor of 35. In our dataset, seven projects were providing forms and reports only for external data (i.e., EIF), without having any local database or file. For these applications, the estimations provided using the Indicative FPA approximate sizing method were really unsatisfactory, with an error of the predicted effort over the actual one ranging from 20% to 77% (avg. 44%). If we remove these seven projects without ILF, the results are slightly better, but still there is no statistically significant difference between Indicative FPA and the constant models.

From this experience, given the type of software projects of our dataset, there are no practical benefits in using the Indicative FPA sizing method, not even

Table 9: Results of SLR for RQ2 and RQ4

Model	Equation	R^2	F	Sign. F
MCFPafp	EFF = 771.798 + 2.955×CFPafp	0.80	93.96	<0.001
MCFPfsc	EFF = 583.796+ 3.064×CFPfsc	0.839	120.1	<0.001
MCFPesb	EFF = 22.8×CFPesb$^{0.76}$	0.852	132.4	<0.001
MCFP	EFF = 512.43 + 3.429×CFP	0.87	151.3	<0.001

Table 10: Results in terms of MAR for RQ2 and RQ4

MCFPafp	MCFPfsc	MCFPesb	MeanEFF	MedianEFF	MCFP
387	336	295	843	872	276

to get a Rough Order of Magnitude, since its results are comparable with the trivial mean or median of past efforts of the projects developed in the software company.

4.2. RQ2: COSMIC Approximate sizing methods vs. Baselines

This research question investigates whether the approximate sizing methods of a more modern functional size, as is COSMIC, are able to outperform the naive baselines in terms of development effort predictions.

Again, we started by verifying the assumptions underlying SLR: linearity, homoscedasticity, residual normality, and independence (details are reported in the Appendix). As result of the analysis, no transformation of the original data was performed. Furthermore, we verified the presence of influential data points and no observation was removed.

The regression equations obtained are shown in Table 9 together with the values of the indicators used to evaluate them. We can observe that the models obtained using the three approximations of COSMIC, namely MCFPafp, MCFPesb, and MCFPfsc, are characterized by a Sign F-value lower than 0.05, that is, the models are significant. The values of R^2 and F for the models are quite high. In particular, R^2 is greater than 0.8 for all the models.

To compare the prediction accuracy of MCFPafp, MCFPesb, and MCFPfsc against the baselines, we performed a leave-one-out cross-validation, whose results are reported in Table 10. From these numbers, we can see that the MAR values obtained with the COSMIC approximate sizing methods are by far smaller than those achieved with the baselines. This means that the COSMIC approximations provide much better development effort predictions than the constant models. These results are confirmed by the boxplots of absolute residuals shown in Figure 2. Indeed, the boxplots of MCFPafp, MCFPesb, and MCFPfsc have a median closer to zero and their box lengths and min-max ranges are shorter than those of MeanEFF and MedianEFF.

Finally, also the statistical significance tests confirm these finding. Indeed, the results shown in Table 11 reveal that $Hn1_X$ and $Hn2_X$ can be rejected for all the COSMIC approximations, i.e., MCFPafp, MCFPesb, and MCFPfsc

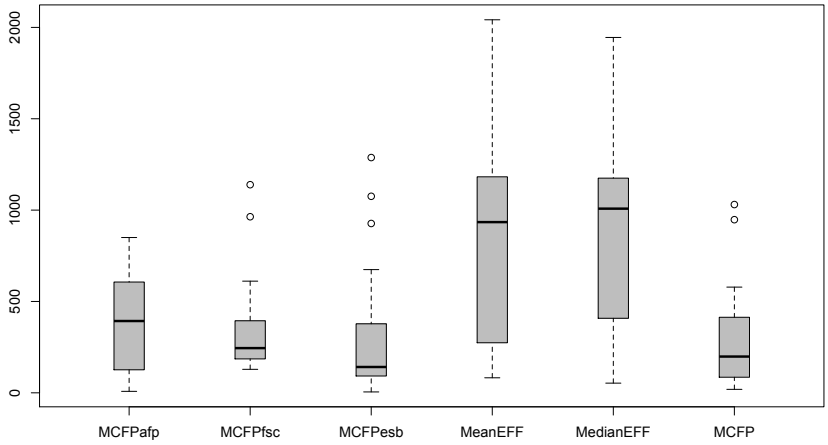


Figure 2: Boxplots of absolute residuals for RQ2 and RQ4.

Table 11: Wilcoxon test p-values (and effect size between brackets) for RQ2 and RQ4

	MeanEFF	MedianEFF	MCFP
MCFPafp vs	<0.001 (large)	<0.001 (large)	0.016 (small)
MCFPfsc vs	<0.001 (large)	<0.001 (large)	0.011 (small)
MCFPesb vs	<0.001 (large)	<0.001 (large)	0.692 (negligible)

provide significantly smaller absolute residuals than MeanEFF and MedianEFF (i.e., p-values are less than 0.05/6=0.008) with a larger effect size. Thus, we can positively answer our second research question:

RQ2: All the COSMIC approximate sizing methods provided good early size estimations for effort prediction.

4.3. RQ3: FPA Approximations vs. FPA

The third research question aims at comparing the prediction accuracy of the models based on FPA approximate sizing methods against the standard FPA method. From the results shown in Table 7, we can note that, surprisingly, the MAR value obtained with MHLFPA is marginally lower than the one achieved with the standard FPA. Furthermore, the Wilcoxon test revealed that

the differences in the absolute residuals are not significant (see Table 8), with a very small effect size (i.e., we cannot reject $Hn3_{MHLFPA}$).

Thus, we confirm the general findings of previous works (e.g., [14, 19, 20]) investigating the HLFPA method (being focused on its correlation with FPA rather than with the development effort), with the conclusion that the HLFPA and standard FPA methods achieve comparable results.

On the other hand, MIFPA provided by far worse effort predictions than the original method, since its MAR value is about 30% higher. As for the statistical significance of the distribution of the residuals, we can see that the p-value is 0.101 and the effect size is negligible. Thus, if we consider a 95% confidence level, we cannot reject the hypothesis $Hn3_{MIFPA}$ of a statistically significant difference between prediction errors of MIFPA and those of MFP.

The boxplots of absolute residuals shown in Figure 1 confirm these results. Indeed, even if the medians of MHLFPA and MIFPA boxplots are quite close, the box length and tails of the MHLFPA boxplot are shorter than those of the MIFPA boxplot.

Taking into account our study design, from the above analysis we can positively answer our third research question since there is no statistically significant difference between effort prediction errors of FPA approximation based models and FPA based model. Nevertheless, we note that the effort prediction errors obtained with MIFPA are by far higher than HLFPA. In conclusion:

> RQ3: FPA approximate sizing methods are as effective as Function Points, for effort estimation.

4.4. RQ4: COSMIC Approximate sizing methods vs. COSMIC

The last research question aims at comparing the prediction accuracy of the models based on COSMIC approximate sizing methods against those of the original COSMIC method.

From the results reported in Table 10, we can observe that there is a clear difference among the three approximations, with the AFP method providing the worse results and the ESB method the best ones. Anyhow, none of these methods was able to provide a MAR smaller than the standard COSMIC.

More in details, the AFP provided an error about 40% bigger than the original method, FSC about 22%, and ESB 7%. This trend is graphically confirmed by the boxplots of absolute residuals, shown in Figure 2, where the boxplot of COSMIC has a median closer to zero than boxplots of MCFPafp and MCFPfsc. As for the comparison between absolute residuals of MCFP and MCFPesb, we can note that the boxplots are quite similar, even if the tails of the MCFP boxplot are less skewed.

The results of the statistical significance tests (see Table 11) suggest that we can reject $Hn4_{MCFPafp}$ and $Hn4_{MCFPfsc}$ because there is a significant difference for MCFPafp and MCFPfsc (i.e., p-values are less than 0.05/3=0.017) yet with a small effect size, while $Hn4_{MCFPesb}$ cannot be rejected because no

significant difference is found for the case of MCFPesb against the standard method. These results highlight that the use of the standard COSMIC method can be preferred to its AFP and FSC approximations, but not to the ESB one. Consequently, we can answer our fourth research question:

> RQ4: Size measures obtained using COSMIC approximate sizing methods are not always as effective as COSMIC, for effort estimation.

4.4.1. Discussion

It is worth noting that the COSMIC approximations present two aspects deserving attention in their application:

1. AFP and ESB are data-driven, meaning that, to be applied, they require some projects previously measured with the standard COSMIC method;
2. FSC and ESB require a subjective classification of the size of each functional process.

In our empirical study, the researcher who labelled the processes based on the project documentation, on average misclassified slightly more than the 10% of the processes. Measurers with different skills might lead to different rates of misclassification, with important impacts on the achievable results. Further investigations on the impact of subjective misclassification should be carried out.

5. Conclusions

Even if FSM methods, like FPA or COSMIC, are widely adopted in the software development industry, there are some managerial scenarios where they cannot be fully applied, due to lack of time and/or project information. Many FSM approximate sizing methods have been defined in the past to help measurers in these scenarios, requiring less time and information to be applied.

In this paper, we have assessed the use of five official approximate sizing methods, two for FPA proposed by IFPUG [17] and three for COSMIC proposed by the COSMIC Organization [37]. In particular, we investigated their effectiveness when used as a predictor of the software development effort, to the best of our knowledge, the first study of this kind for FSM approximate sizing methods. To perform our empirical analysis we used a dataset of 25 software systems developed by a single company, comparing the absolute residuals obtained by using the regression models based on the approximate sizes, against some standard baselines used in literature, and the absolute residuals of the models based on the measures obtained by using the standard FPA and COSMIC methods.

The results of the investigation show that the High Level FPA approach was able to significantly overcome the baselines, and to provide results that are comparable (and slightly but not significantly better) to the standard method.

Thus, as found also in previous study [19], the use of this approximation seems to be highly advisable. On the other hand, the Indicative FPA approach was not able to beat the baselines, being thus, in our opinion, not advisable to use for a software company, not even to get a Rough Order of Magnitude of the size of the system to be developed.

Regarding COSMIC, in the official manual, the three considered approximations are presented as subsequent improvements (i.e., FSC requires more effort to be applied than AFP, but should provide more accurate sizes, and the same for ESB over FSC). Our results fully confirm this trend. Indeed, AFP leads to good effort predictions, which are improved by FSC, and then by ESB. Nevertheless, AFP and FSC lead to significantly worse effort predictions than the standard COSMIC method, while ESB provides comparable accuracy. On the other hand, COSMIC approximate sizing methods have two criticalities: (I) AFP and ESB require some previous projects measured with the full method to carry out some calibrations, while (II) FSC and ESB involve a subjective classification of the size of functional processes, which might impact on the measurement accuracy.

The experimental results presented herein hold for the company involved in our study and they should be assessed on further data as soon as it becomes available. Indeed, replications of a study, in different setting and with larger datasets, are always required in this field. Thus, as future work, we plan to collect and analyze data from other companies.

Moreover, we intend to further investigate the approximate sizing methods proposed in the COSMIC documentation [37]. In particular, we intend to verify pros and cons of the different approaches not only with respect to the accuracy of the estimation but also with respect to the information required to apply the methods and then the time needed to carry out the measurement, as well as to quantify the impact of misclassifications done by the measurer. In the future, we also intend to investigate whether the type of applications considered can influence the obtained estimation models.

References

[1] I. Sommerville, Software Engineering, 9th Edition, Addison-Wesley, Harlow, England, 2010.

[2] ISO, ISO/IEC 14143-1:2007: Information technology - Software measurement - Functional size measurement (2007).

[3] A. Albrecht, Measuring Application Development Productivity, in: Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium, 1979, pp. 83–92.

[4] C. Gencel, O. Demirors, Functional size measurement revisited, ACM Trans. Softw. Eng. Methodol. 17 (3) (2008) 15:1–15:36.

[5] A. Abran, J. Desharnais, A. Lesterhuis, B. Londeix, R. Meli, P. Morris, S. Oligny, M. O'Neil, T. Rollo, G. Rule, L. Santillo, C. Symons, H. Toivonen, The COSMIC Functional Size Measurement Method Measurement Manual, version 4.0.1, http://www.cosmicon.com/portal/public/MMv4.0.1.pdf (2015).

[6] F. Ferrucci, C. Gravino, P. Salza, F. Sarro, Investigating functional and code size measures for mobile applications, in: Proceedings of Euromicro Conference on Software Engineering and Advanced Applications, 2015, pp. 365–368.

[7] F. Ferrucci, C. Gravino, P. Salza, F. Sarro, Investigating functional and code size measures for mobile applications: A replicated study, in: P. Abrahamsson, L. Corral, M. Oivo, B. Russo (Eds.), Product-Focused Software Process Improvement, Springer International Publishing, Cham, 2015, pp. 271–287.

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

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

[10] C. Jones, A new business model for function point metrics, Capers Jones & Associates Llc.

[11] L. Lavazza, On the effort required by function point measurement phases, International Journal on Advances in Software Volume 10, Number 1 & 2, 2017.

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

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

[14] H. van Heeringen, E. van Gorp, T. Prins, Functional size measurement accuracy versus costs is it really worth it?, in: Software Measurement European Forum, 2009.

[15] F. Vogelezang, T. Prins, S. N. BV, Approximate size measurement with the cosmic method factors of influence, in: Software Measurement European Forum, 2007, pp. 167–178.

[16] G. De Vito, F. Ferrucci, Approximate COSMIC size: The quick/early method, in: Proceedings of Euromicro Conference on Software Engineering and Advanced Applications, 2014, pp. 69–76.

[17] uTip - Early Function Point Analysis and Consistent Cost Estimating, IFPUG v. 1.0, 2015/7/1, http://www.ifpug.org/uTips/uTip003EarlyFPAandConsistentCostEstimating.pdf (2015).

[18] A. Abran, B. Londeix, M. O?Neill, L. Santillo, F. Vogelezang, J. Desharnais, P. Morris, T. Rollo, C. Symons, A. Lesterhuis, et al., The cosmic functional size measurement method, Advanced and Related Topics, Version 3.

[19] F. G. Wilkie, I. McChesney, P. Morrow, C. Tuxworth, N. Lester, The value of software sizing, Information and Software Technology 53 (11) (2011) 1236–1249.

[20] P. Morrow, F. G. Wilkie, I. McChesney, Function point analysis using nesma: simplifying the sizing without simplifying the size, Software Quality Journal 22 (4) (2014) 611–660.

[21] A. Z. Abualkishik, F. Ferrucci, C. Gravino, L. Lavazza, G. Liu, R. Meli, G. Robiolo, A study on the statistical convertibility of IFPUG function point, COSMIC function point and simple function point, Information and Software Technology 86 (2017) 1–19.

[22] A. E. Castro Sotos, S. Vanhoof, W. Van Den Noortgate, P. Onghena, The transitivity misconception of pearson's correlation coefficient., Statistics Education Research Journal 8 (2).

[23] F. Ge, P. Jing, T. Dongke, Z. Z. Julia, F. Changyong, Two paradoxes in linear regression analysis, Shanghai archives of psychiatry 28 (6) (2016) 355.

[24] M. J. Shepperd, S. G. MacDonell, Evaluating prediction systems in software project estimation, Information and Software Technology 54 (8) (2012) 820–827.

[25] W. B. Langdon, J. Dolado, F. Sarro, M. Harman, Exact mean absolute error of baseline predictor, MARP0, Information and Software Technology 73 (2016) 16 – 18.

[26] F. Sarro, A. Petrozziello, Linear programming as a baseline for software effort estimation, ACM Trans. Softw. Eng. Methodol. 27 (3) (2018) 12:1–12:28.

[27] E. Mendes, B. Kitchenham, Further Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications, in: Proceedings of International Software Metrics Symposium, IEEE press, 2004, pp. 348–357.

[28] D. Conte, H. Dunsmore, V. Shen, Software engineering metrics and models, The Benjamin/Cummings Publishing Company, Inc., 1986.

[29] I. Myrtveit, E. Stensrud, Validity and reliability of evaluation procedures in comparative studies of effort prediction models, Empirical Software Engineering 17 (1-2) (2012) 23–33.

[30] V. Kampenes, T. Dyba, J. Hannay, I. Sjoberg, A systematic review of effect size in software engineering experiments, Information and Software Technology 4 (11-12) (2007) 1073–1086.

[31] A. J. Albrecht, J. E. Gaffney, Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, IEEE Transactions on Software Engineering 9 (6) (1983) 639–648.

[32] IFPUG, International Function Point Users Group (IFPUG)- release 4.3.1 - www.ifpug.org (2010).

[33] C. R. Symons, Function point analysis: difficulties and improvements, IEEE transactions on Software Engineering 14 (1) (1988) 2–11.

[34] A. Abran, P. N. Robillard, Function Points: a study of their measurement processes and scale transformations, Journal of Systems and Software 25 (2) (1994) 171–184.

[35] B. Kitchenham, Counterpoint: the problem with Function Points, IEEE Software 14 (2) (1997) 29–31.

[36] L. D. Marco, F. Ferrucci, C. Gravino, Approximate COSMIC size to early estimate web application development effort, in: Proceedings of Euromicro Conference on Software Engineering and Advanced Applications, 2013, pp. 349–356.

[37] A. Abran, B.Londeix, M. O'Neill, L. Santillo, F. Vogelezang, J.-M. Desharnais, P. Morris, T. Rollo, C. Symons, A. Lesterhuis, S. Oligny, G. Rule, H. Toivonen, The COSMIC Functional Size Measurement Method, Version 3.0, Advanced and Related Topics, http://www.cosmicon.com/portal/public/COSMIC%20Method%20v3.0%20Advanced%20&%20Related%20Topics.pdf (2007).

[38] E. Mendes, S. Counsell, N. Mosley, Comparison of Web Size Measures for Predicting Web Design and Authoring Effort, IEE Proceedings-Software 149 (3) (2002) 86–92.

[39] G. Costagliola, S. Di Martino, F. Ferrucci, C. Gravino, G. Tortora, G. Vitiello, A COSMIC-FFP Approach to Predict Web Application Development effort, Journal of Web Engineering 5 (2) (2006) 93–120.

[40] S. Di Martino, F. Ferrucci, C. Gravino, Estimating Web Application Development Effort Using Web-COBRA and COSMIC: An Empirical Study, in: Proceedings of Euromicro Conference on Software Engineering and Advanced Applications, ACM press, 2009, pp. 306–312.

[41] S. Abrahão, L. De Marco, F. Ferrucci, C. Gravino, F. Sarro, A COSMIC measurement procedure for sizing web applications developed using the OO-H method, in: Proceedings of the Workshop on Advances in Functional Size Measurement and Effort Estimation, ACM, 2010, pp. 2:1–2:8.

[42] S. Abrahão, L. D. Marco, F. Ferrucci, J. Gómez, C. Gravino, F. Sarro, Definition and evaluation of a COSMIC measurement procedure for sizing web applications in a model-driven development environment, Information and Software Technology 104 (2018) 144–161.

[43] S. Di Martino, F. Ferrucci, C. Gravino, E. Mendes, Comparing Size Measures for Predicting Web Application Development Effort: A Case Study, in: Proceedings of Empirical Software Engineering and Measurement, IEEE press, 2007, pp. 324–333.

[44] F. Ferrucci, C. Gravino, S. Di Martino, A Case Study Using Web Objects and COSMIC for Effort Estimation of Web Applications, in: Proceedings of Euromicro Conference on Software Engineering and Advanced Applications, IEEE press, 2008, pp. 441–448.

[45] J. Popović, D. Bojić, A comparative evaluation of effort estimation methods in the software life cycle, Computer Science and Information Systems 9 (1) (2012) 455–484.

[46] S. Ohiwa, T. Oshino, S. Kusumoto, K. Matsumoto, Towards an Early Software Effort Estimation Based on the NESMA Method (Estimated FP), in: The IT Confidence conference 2nd Int. Conf. on IT Data Collection, Analysis and Benchmarking, 2014.

[47] L. Lavazza, G. Liu, An empirical evaluation of simplified function point measurement processes, International Journal on Advances in Software 6.

[48] M. Bundschuh, Early project estimation with early function point prognosis, in: International Conference Software Measurement & Analysis, 2006, pp. 18–20.

[49] R. Meli, Early & quick function point method - an empirical validation experiment, in: Int. Conf. on Advances and Trends in Software Engineering, 2015.

[50] L. Lavazza, S. Morasca, Empirical evaluation and proposals for bands-based COSMIC early estimation methods, Information and Software Technology 109 (2019) 108–125.

[51] F. Valds Souto, Analyzing the performance of two cosmic approximation sizing techniques at the functional process level, Science of Computer Programming 135 (2016) 105–121.

[52] 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.

[53] T. Menzies, Z. Chen, J. Hihn, K. Lum, Selecting Best Practices for Effort Estimation, IEEE Transactions on Software Engineering 32 (11) (2006) 883–895.

[54] B. Kitchenham, E. Mendes, G. Travassos, Cross versus Within-Company Cost Estimation Studies: A systematic Review, IEEE Transactions on Software Engineering 33 (5) (2007) 316–329.

[55] K. Srinivasan, D. Fisher, Machine learning approaches to estimating software development effort, IEEE Transactions on Software Engineering 21 (2) (1995) 126–137.

[56] L. C. Briand, I. Wieczorek, Resource Estimation in Software Engineering, John Wiley & Sons, Inc., 2002.

[57] F. Sarro, Search-based predictive modelling for software engineering: How far have we gone?, in: Proceedings of the 11th International Symposium on Search-Based Software Engineering, SSBSE, 2019, pp. 3–7.

[58] F. Sarro, A. Petrozziello, M. Harman, Multi-objective software effort estimation, in: Proceedings of the International Conference on Software Engineering, 2016, pp. 619–630.

[59] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, Genetic programming for effort estimation: An analysis of the impact of different fitness functions, in: Proceedings of the International Conference on Search Based Software Engineering, 2010, pp. 89–98.

[60] F. Ferrucci, C. Gravino, F. Sarro, How multi-objective genetic programming is effective for software development effort estimation?, in: Proceedings of the International Conference on Search Based Software Engineering, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 274–275.

[61] F. Ferrucci, M. Harman, F. Sarro, Search-based software project management, in: G. Ruhe, C. Wohlin (Eds.), Software Project Management in a Changing World, Springer Berlin Heidelberg, 2014, pp. 373–399.

[62] F. Sarro, F. Ferrucci, C. Gravino, Single and multi objective genetic programming for software development effort estimation, in: Proceedings of the Annual ACM Symposium on Applied Computing, SAC '12, ACM, 2012, pp. 1221–1226.

[63] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, E. Mendes, Investigating tabu search for web effort estimation, in: Proceedings of Euromicro Conference on Software Engineering and Advanced Applications, 2010, pp. 350–357.

[64] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, Using tabu search to estimate software development effort, in: A. Abran, R. Braungarten, R. R. Dumke, J. J. Cuadrado-Gallego, J. Brunekreef (Eds.), Software Process and Product Measurement, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 307–320.

[65] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, E. Mendes, How effective is tabu search to configure support vector regression for effort estimation?, in: Proceedings of the International Conference on Predictive Models in Software Engineering, 2010, p. 4.

[66] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, E. Mendes, Using tabu search to configure support vector regression for effort estimation, Empirical Software Engineering 18 (3) (2013) 506–546.

[67] E. Kocaguneli, T. Menzies, J. W. Keung, On the value of ensemble effort estimation, IEEE Transactions on Software Engineering 38 (6) (2012) 1403–1416.

[68] K. Maxwell, Applied Statistics for Software Managers, Software Quality Institute Series, Prentice Hall, 2002.

[69] E. Mendes, B. Kitchenham, A Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications, in: Proceedings of Conference on Evaluation and Assessment in Software Engineering, 2004, pp. 47–55.

[70] D. Montgomery, E. Peck, G. Vining, Introduction to Linear Regression Analysis, John Wiley and Sons, Inc., 1986.

[71] L. C. Briand, J. Wüst, Modeling Development Effort in Object-Oriented Systems Using Design Properties, IEEE Transactions on Software Engineering 27 (11) (2001) 963–986.

[72] R. Freund, W. Wilson, P. Sa, Regression Analysis: Statistical Modeling of a Response Variable, Elsevier Academic Press, 2006.
URL https://books.google.it/books?id=Qtx2lAEACAAJ

[73] E. Mendes, S. Di Martino, F. Ferrucci, C. Gravino, Effort estimation: how valuable is it for a Web company to use a cross-company data set, compared to using its own single-company data Set?, in: Proceedings of the 6th International World Wide Web Conference, ACM press, 2007, pp. 83–93.

[74] F. Ferrucci, E. Mendes, F. Sarro, Web effort estimation: the value of cross-company data set compared to single-company data set, in: Proceedings of the International Conference on Predictive Models in Software Engineering, 2012, pp. 29–38.

[75] F. Ferrucci, C. Gravino, F. Sarro, Conversion from IFPUG FPA to COSMIC: within-vs without-company equations, in: Proceedings of Euromicro Conference on Software Engineering and Advanced Applications, 2014, pp. 293–300.

[76] F. Ferrucci, C. Gravino, F. Sarro, Exploiting prior-phase effort data to estimate the effort for the subsequent phases: a further assessment, in: Proceedings of the International Conference on Predictive Models in Software Engineering, 2014, pp. 42–51.

[77] L. L. Minku, F. Sarro, E. Mendes, F. Ferrucci, How to make best use of cross-company data for web effort estimation?, in: Proceedings of ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2015, pp. 172–181.

[78] W. J. Conover, Practical Nonparametric Statistics, 3rd Edition, Wiley, 1998.

[79] P. Royston, An extension of Shapiro and Wilk's W test for normality to large samples, Applied Statistics 31 (2) (1982) 115–124.

[80] C. Wohlin, P. Runeson, M. Host, M. Ohlsson, B. Regnell, A. Wesslen, Experimentation in Software Engineering - An Introduction, Kluwer, 2000.

[81] J. L. Devore, N. Farnum, Applied Statistics for Engineers and Scientists, Duxbury, 1999.

[82] A. Vargha, H. D. Delaney, A critique and improvement of the cl common language effect size statistics of mcgraw and wong, Journal of Educational and Behavioral Statistics 25 (2) (2000) 101–132.

[83] G. Neumann, M. Harman, S. M. Poulding, Transformed vargha-delaney effect size, in: Proceedings of the International Conference on Search Based Software Engineering, 2015, pp. 318–324.

[84] F. Sarro, M. Harman, Y. Jia, Y. Zhang, Customer rating reactions can be predicted purely using app features, in: Proceedings of IEEE International Requirements Engineering Conference, 2018, pp. 76–87.

[85] E. Mendes, M. Kalinowski, D. Martins, F. Ferrucci, F. Sarro, Cross- vs. within-company cost estimation studies revisited: an extended systematic review, in: Proceedings of International Conference on Evaluation and Assessment in Software Engineering, 2014, pp. 12:1–12:10.

[86] J. Freund, Mathematical Statistics, Prentice-Hall, Upper Saddle River, NJ, 1992.

[87] T. Breush, A. Pagan, A simple test for heteroscedasticity and random coefficient variation, Econometrica 47 (1992) 1287–1294.

Table A.12: Pearson's correlation and Spearman' rho test results to assess Linearity

Effort	vs	CFP	CFPafp	CFPesb	CFPfsc	FP	HLFPA	ILFPA
Pearson	statistics	0.932	0.896	0.899	0.916	0.782	0.8	0.456
	p-value	<0.01	<0.01	<0.01	<0.01	<0.01	0.022	<0.01
Spearman	statistics	0.942	0.909	0.912	0.9	0.774	0.79	0.79
	p-value	<0.01	<0.01	<0.01	<0.01	<0.01	0.092	<0.01

Table A.13: Breush-Pagan test results to assess Homoscedasticity

Effort	vs	CFP	CFPafp	CFPesb	CFPfsc	FP	HLFPA	ILFPA
	statistics	0.109	0.106	0.117	0.48	0.601	1.134	0.626
	p-value	0.741	0.745	0.733	0.489	0.438	0.287	0.429

Appendix A. Testing Linear Regression Assumptions

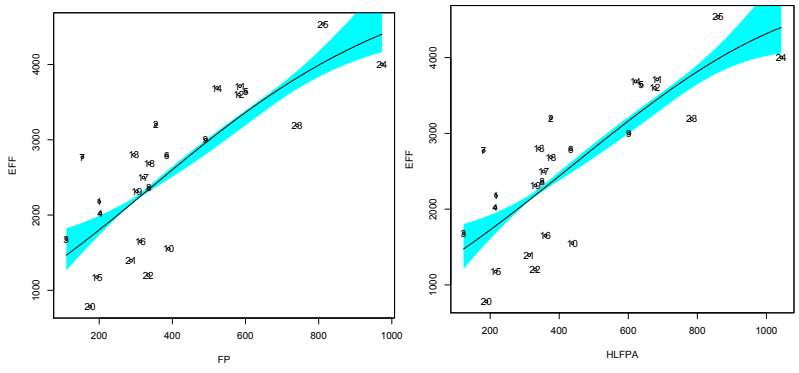
In the following, for each research question, we report on the analysis we carried out to verify the four linear regression assumptions (i.e., *linearity*, *homoscedasticity*, *normality*, and *independence*) for the construction of the effort estimation models used in our study.

Appendix A.1. RQ1

Linearity. Linearity was assessed both graphically, as shown in Figure A.3 (b) and (c), and by exploiting the Pearson and Spearman' rho tests [78]. The scatter plot in Figure A.3 (b) shows a positive linear relationship between EFF and HLFPA, confirmed by the Pearson's correlation test (statistic=0.8 with p-value =0.022) and the Spearman' rho test (statistic=0.79 with p-value =0.092). On the other hand, as for EFF and IFPA, the Pearson's correlation test (statistic=0.456 with p-value <0.01) and the Spearman' rho test (statistic=0.79 with p-value <0.01) revealed a weak linear correlation.

Homoscedasticity. To verify this assumption, we exploited the scatter plot of residuals (see Figures A.4(b) and (c)) and the Breush-Pagan test. From Figure A.4(b) we can observe that the residuals for HLFPA based model fall within a horizontal band centered on 0 and the assumption (i.e., homoscedasticity of the error terms as null hypothesis) is confirmed by the Breush-Pagan test, since the p-value is 0.287 (statistic=1.134). Differently, the residuals of the IFPA based model are not well distributed within a horizontal band centered on 0. However, the the Breush-Pagan test suggests that the assumptions can be considered verified since the p-value is 0.429 (statistic=0.626)

Normality. From the analysis of the Normal Q-Q plot depicted in Figures A.5(b) and (c), we can note that only some observations are not close to the straight line and they should get more attention, as potential outliers. However, the Shapiro test (with normality of error terms as null hypothesis) suggested that the assumption can be considered as verified for both HLFPA (p-value=0.058, statistic=0.922) and IFPA (p-value=0.358, statistic=0.957) based models.



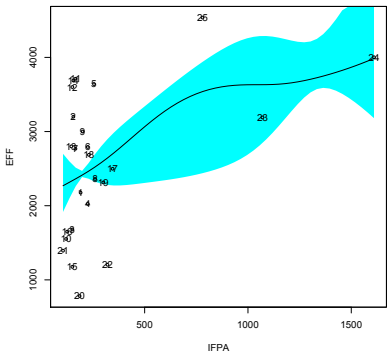


Figure A.3: The scatter plot for EFF and FP (a), EFF and HLFPA (b), and EFF and IFPA (c) resulting from the SLR

Table A.14: Shapiro-Wilk Test results to assess Normality

Effort	vs	CFP	CFPafp	CFPesb	CFPfsc	FP	HLFPA	ILFPA
	statistics	0.959	0.964	0.885	0.941	0.924	0.922	0.957
	p-value	0.389	0.51	0.009	0.159	0.064	0.058	0.358

Table A.15: Results of linear regression for RQ1

Model	Variable	Value	Std.err	t-value	p-value
MHLFPA	Log(HLFPA)	0.58	0.121	4.807	$< .0.01$
	Intercept	4.316	0.721	5.983	$< .0.01$
MIFPA	Log(IFPA)	0.272	0.125	2.168	0.041
	Intercept	6.292	0.687	9.159	$< .0.01$

Independence. The uncorrelation of residuals for consecutive errors has been verified by a Durbin-Watson statistic. For the residuals obtained with MHLFPA, the test provided a value not close to 2 (1.187) and p-value (0.011) less than 0.05, thus, we cannot assume that the residuals are uncorrelated. As for the residuals from MIFPA, the value of the Durbin-Watson statistic can be considered acceptable (1.538) with a p-value=0.089.

The above results suggest that a transformation of the data is required before applying linear regression to build an effort estimation model, for both HLFPA and IFPA. We decided to apply a log transformation as done in previous work (e.g., [54] [68] [69] [73] [66]). The variables log transformed are denoted as Log(HLFPA) and Log(IFPA) and some statistics about the models (values of the intercept and coefficient of variables as well as their standard error) are shown in Table A.15. We have also reported the results of the performed t-statistic, i.e., p-values and t-values of the coefficient and the intercept, in order to evaluate their statistical significance. Let us remember that a p-value less than 0.05 indicates that we can reject the null hypothesis and the variable is a significant predictor with a confidence of 95%. As for the t-value, a variable is significant if the corresponding t-value is greater than 1.5. The obtained models are summarized in the following:

$$Log(EFF) = 0.58 \times Log(HLFPA) + 4.316 \tag{A.1}$$

$$Log(EFF) = 0.272 \times Log(IFPA) + 6.292 \tag{A.2}$$

and when it is transformed back to the original raw data scale we obtain:

$$EFF = 74.9 \times HLFPA^{0.58} \tag{A.3}$$

$$EFF = 539.99 \times IFPA^{0.272} \tag{A.4}$$

Note that in Table 6 when discussing the results of the application of the linear regression we only report the model transformed back to the original raw scale.

Appendix A.2. RQ2

Linearity. Figure A.6(b) shows the scatter plot obtained by considering EFF and CFPafp. It shows a positive linear relationship between the involved

Table A.16: Durbin-Watson statistics results to assess Independence

Effort	vs	CFP	CFPafp	CFPesb	CFPfsc	FP	HLFPA	ILFPA
	statistics	1.543	1.779	1.744	1.537	1.233	1.187	1.538
	p-value	0.109	0.109	0.244	0.104	0.015	0.011	0.089

variables, which is confirmed by the Pearson's correlation test (statistic=0.896 with p-value <0.01) [86] and the Spearman' rho test (statistic=0.909 with p-value <0.01), as reported in Table A.12. Similarly, the scatter plot in Figure A.6(b), (c), and (d) shows a positive linear relationship between the involved variables, i.e., EFF and CFPfsc, and EFF and CFPesb, again confirmed by Pearson's correlation and Spearman' rho tests, reported in Table A.12.

Homoscedasticity. From the scatter plots shown in Figure A.7(b), (c), and (d), we can observe that the residuals fall within a horizontal band centered on 0 for all the approximations. However, some outliers may be noted, e.g., observations 7, 12, 16, and 25. Thus, we further investigated the homoscedasticity assumption, by performing a Breush-Pagan test [87], with the homoscedasticity of the error terms as null hypothesis. This assumption is verified for all the models based on COSMIC approximations, since the p-value of the statistic is greater than 0.05 and therefore the null hypothesis cannot be rejected (see Table A.13).

Normality. The analysis of Normal Q-Q plot for the COSMIC approximations, shown in Figure A.8(b), (c), and (d), revealed that only a couple of observations are not very close to the straight line and they should be investigated. To verify the normality assumption, we used the Shapiro-Wilk Test [79], by considering as null hypothesis the normality of error terms. The results highlight that the assumption can be considered as verified for CFPafp and CFPfsc, since the p-values of are greater than 0.05 and thus the null hypothesis cannot be rejected, but this does not holds for CFPesb (see Table A.14).

Independence. The uncorrelation of residuals for consecutive errors has been verified by a Durbin-Watson statistic, whose results are quite close to 2, with a p-value greater than 0.05, for all the COSMIC approximations. Thus, we can assume that the residuals are uncorrelated.

The above results suggest that a transformation of the data is not required for CFPafp and CFPfsc to apply linear regression. The statistics about the models (values of the intercept and coefficient of variables as well as their standard error and the results of the performed t-statistics) are shown in Table A.17. In particular, the obtained are:

$$EFF = 771.798 + 2.955 \times CFPafp \tag{A.5}$$

$$EFF = 583.796 + 3.064 \times CFPfsc \tag{A.6}$$

Differently, we decided to apply a log transformation to EFF and CFPesb before building the linear regression model for effort estimation.

Table A.17: Results of linear regression for RQ2

Model	Variable	Value	Std.err	t-value	p-value
MCFPafp	CFPafp	2.955	0.305	9.69	< 0.001
	Intercept	771.798	206.628	3.735	0.001
MCFPfsc	CFPfsc	3.064	0.28	10.957	< 0.001
	Intercept	583.796	199.097	2.932	0.008
MCFPesb	Log(CFPesb)	0.758	0.066	11.508	< 0.001
	Intercept	3.127	0.405	7.721	< 0.001

We applied a log transformation and the variables log transformed are denoted as Log(CFPesb) and Log(EFF) and the obtained models are:

$$Log(EFF) = 0.76 \times Log(CFPesb) + 3.127 \tag{A.7}$$

and when it is transformed back to the original raw data scale we obtain:

$$EFF = 22.8 \times CFPesb^{0.76} \tag{A.8}$$

Again, the statistics about the performed t-statistics are shown in Table A.17.

Appendix A.3. RQ3

Linearity. From the scatter plot in Figure A.3(a) we can observe a positive linear relationship with the variable EFF, confirmed also by the Pearson's correlation test (statistic=0.782 with p-value <0.01) and the Spearman' rho test (statistic=0.8 with p-value <0.01).

Homoscedasticity. The scatter plot in Figure A.4(a) suggests that the residuals obtained with the model based on FP fall within a horizontal band centered on 0, with some outliers, e.g., observations 7, 20, and 22. The Breush-Pagan Test, with the homoscedasticity of the error terms as null hypothesis, revealed that the null hypothesis cannot be rejected, since the p-value (0.44) of the statistic (0.596) is greater than 0.05.

Normality. The Normal Q-Q plot in Figure A.5(a) is characterized by an S-shaped pattern, revealing that there are either too many or two few large errors in both directions, i.e., the residuals have an excessive kurtosis [70]. The results of the Shapiro-Wilk test revealed that the null hypothesis can be rejected since the p-value (0.022) of the statistic (0.904) is less than 0.05 (i.e., the assumption is not verified).

Independence. The analysis of the Durbin-Watson statistics to verify the uncorrelation of residuals for consecutive errors highlighted minor cases of positive serial correlation since a value not close to 2 (1.207) was obtained with a p-value (0.0128) smaller than 0.05. Thus, the assumption cannot be considered verified.

The above results suggest to perform a transformation of the data before to apply linear regression to built an effort estimation model based on the Function Points measure.We applied a log transformation and the variables log transformed are denoted as Log(FP) and Log(EFF) and the obtained model is:

$$Log(EFF) = 0.56 \times Log(FP) + 4.471 \tag{A.9}$$

Table A.18: Results of linear regression for RQ3

Model	Variable	Value	Std.err	t-value	p-value
MFP	Log(FP)	0.564	0.124	5.533	0.001
	Intercept	4.4707	0.731	6.117	< 0.001

and when it is transformed back to the original raw data scale we obtain:

$$EFF = 87.42 \times FP^{0.56} \tag{A.10}$$

The other statistics about the models (standar error values of the intercept and coefficient as well as the results of performed t-statistics) are shown in Table A.18.

Appendix A.4. RQ4

Linearity. Figure A.6(a) reports the scatter plot obtained by considering EFF and CFP. It shows a positive linear relationship between these variables, confirmed also by the Pearson's correlation test (statistic=0.932 with p-value <0.01) and the Spearman' rho test (statistic=0.942 with p-value <0.01).
Homoscedasticity. From the scatter plot shown in Figure A.7(a), we can observe that the residuals fall within a horizontal band centered on 0. However, some outliers may be noted, e.g., observations 7 and 16. Thus, we further investigated the homoscedasticity assumption by performing a Breush-Pagan test, with the homoscedasticity of the error terms as null hypothesis. This assumption is verified for the CFP, since the p-value (0.741) of the statistic (0.110) is greater than 0.05 and therefore the null hypothesis cannot be rejected.
Normality. The analysis of Normal Q-Q plot for CFP in Figure A.8(a) revealed that only some observations were not very close to the straight line and they should get closer attention ("outliers"). However, the results of the the Shapiro-Wilk Test revealed that the assumption can be considered to be verified since the p-value (0.389) of the statistic (0.959) was greater than 0.05 and thus the null hypothesis cannot be rejected.
Independence. The uncorrelation of residuals for consecutive errors has been verified by a Durbin-Watson statistic. For CFP the test provided a value acceptable since quite close to 2 (1.543) and p-value (0.109) greater than 0.05. Thus, we can assume that the residuals are uncorrelated.

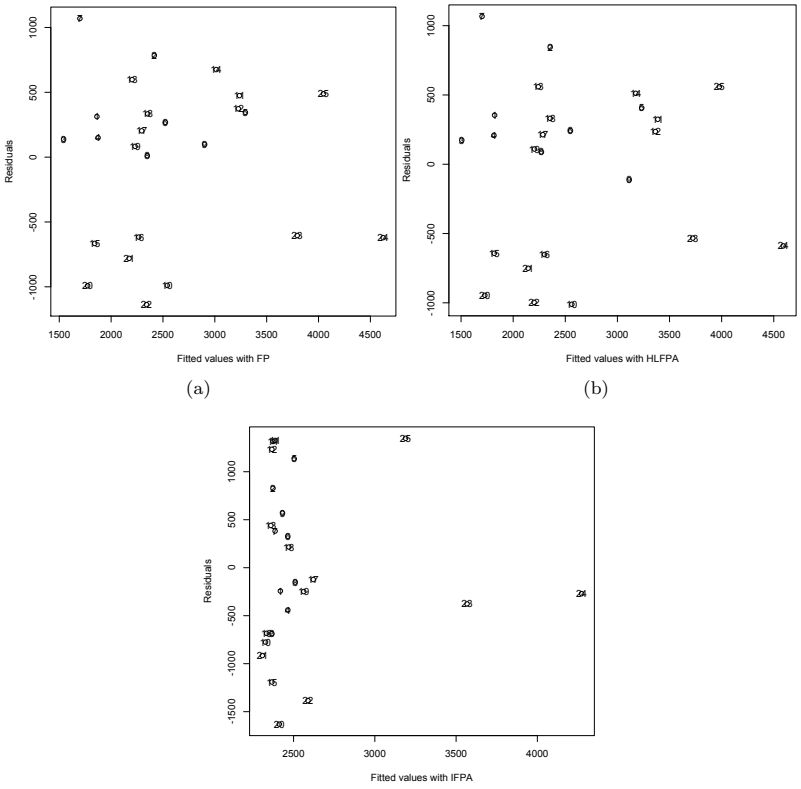
The above results suggest that a transformation of the data is not required before to apply linear regression to built an effort estimation model based on the COSMIC measure. The obtained model is:

$$EFF = 512.43 + 3.429 \times CFP \tag{A.11}$$

The other statistics about the models (standar error values of the intercept and coefficient as well as the results of performed t-statistics) are shown in Table A.19.

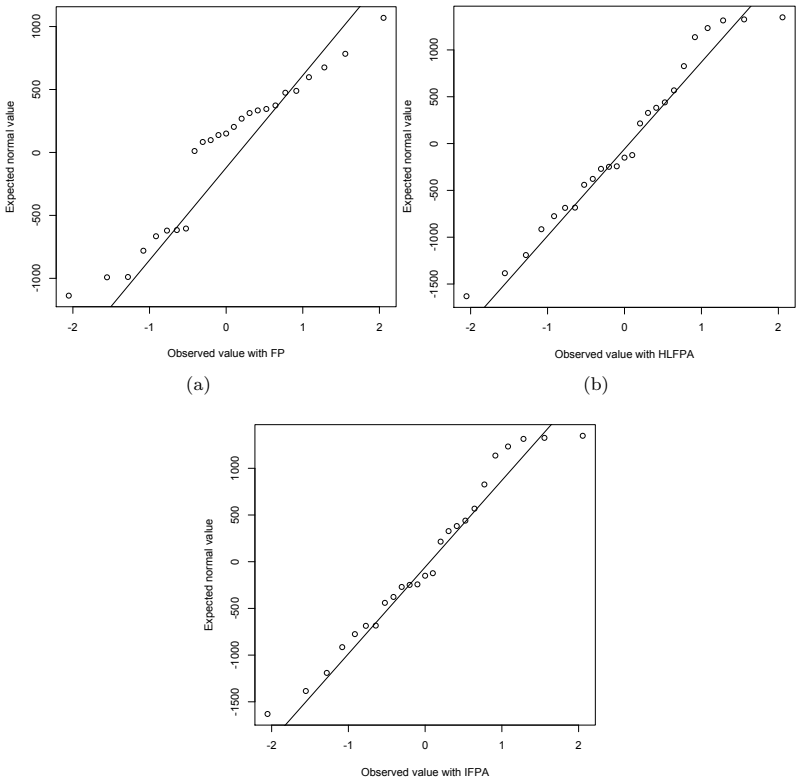
Table A.19: Results of linear regression for RQ4

Model	Variable	Value	Std.err	t-value	p-value
MCFP	CFP	3.429	0.279	12.302	< 0.001
	Intercept	512.43	183.137	2.798	0.01



(a) (b) (c)

Figure A.4: The scatter plot for residuals and predicted values for FP (a), HLFPA (b), and IFPA (c) resulting from the application of SLR



(a) (b)

(c)

Figure A.5: The Q-Q plot for residuals for FP (a) HLFPA (b), and IFPA (c) resulting from the application of SLR

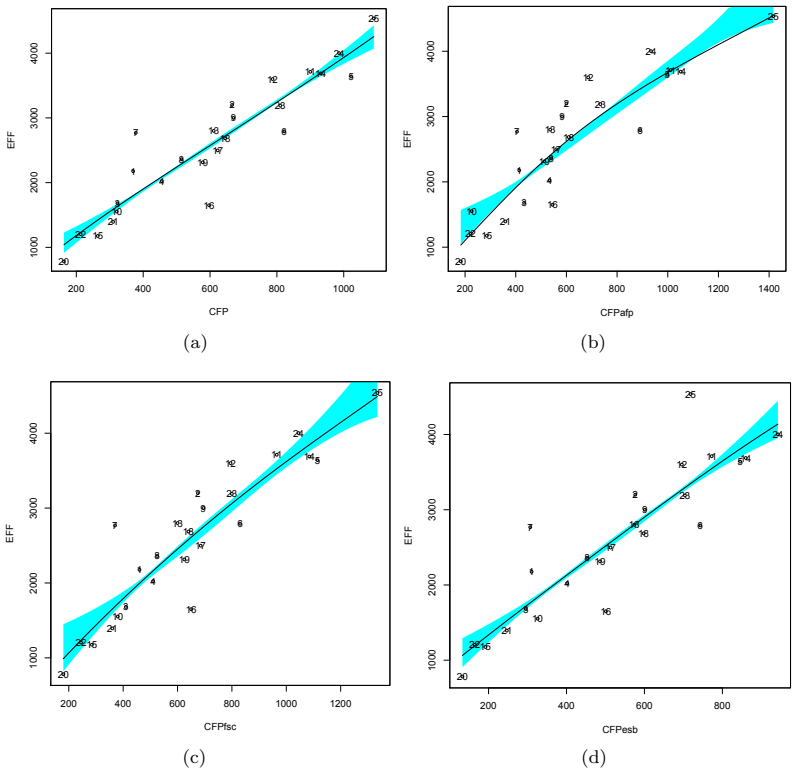
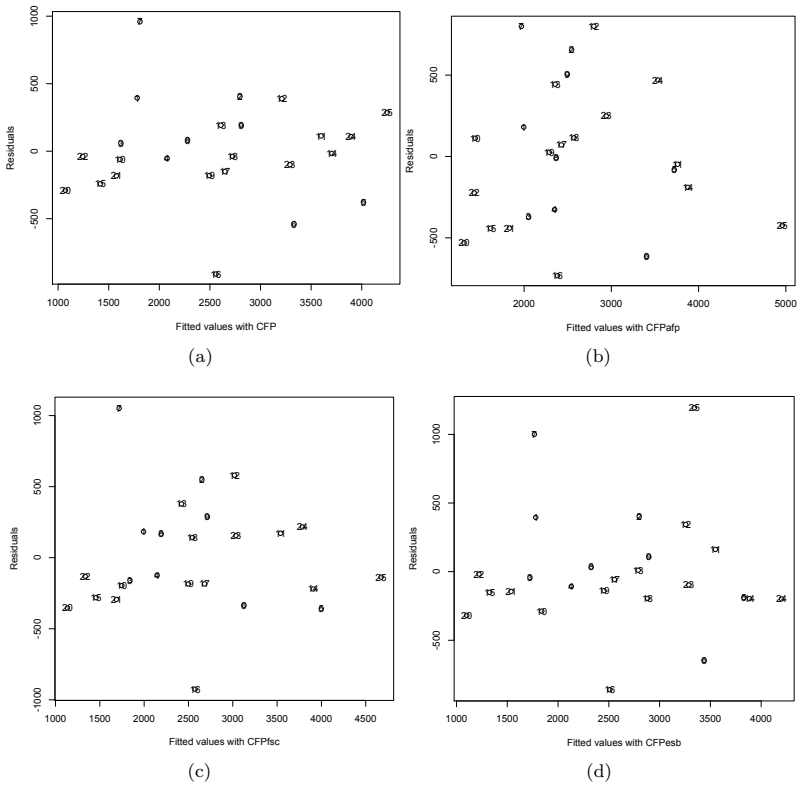


Figure A.6: The scatter plot for EFF and CFP (a), EFF and CFPafp (b), EFF and CFPfsc (c), and EFF and CFPesb (d), resulting from the SLR



(a) (b) (c) (d)

Figure A.7: The scatter plot for residuals and predicted values for CFP (a), CFPafp (b), CFPfsc (c), and CFPesb (d) resulting from the application of SLR

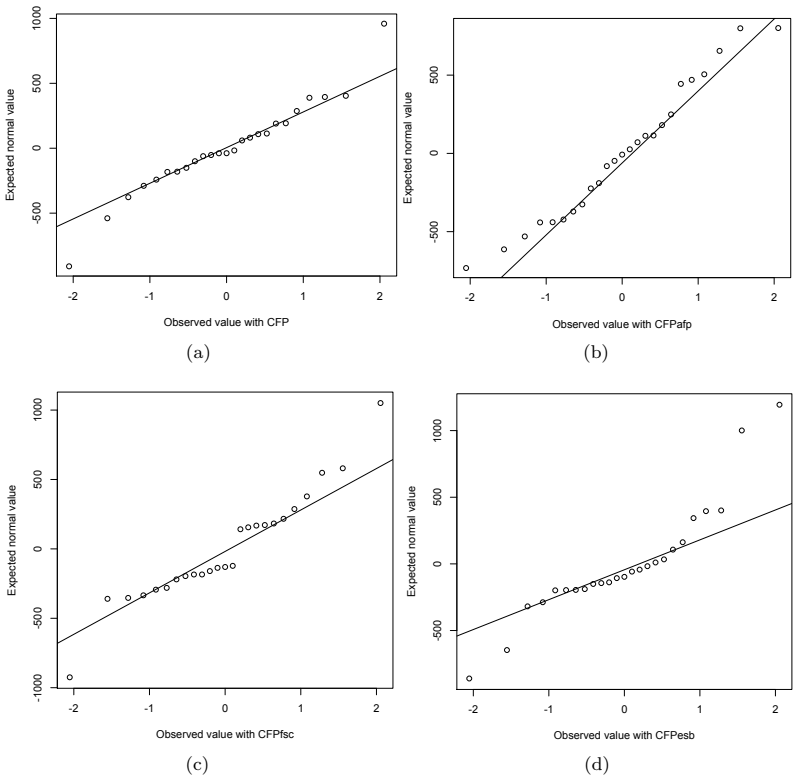


Figure A.8: The Q-Q plot for residuals for CFP (a) CFPafp (b), CFPfsc (c), and CFPesb (d) resulting from the application of SLR