Scientific Research Publishing

Journal of Software Engineering and Applications, 2017, 10, 338-353
http://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Estimation Models for Software Functional Test Effort

Kamala Ramasubramani Jayakumar[1], Alain Abran[2]

[1]Amitysoft Technologies, Chennai, India
[2]École de technologie supérieure, University of Quebec, Montreal, Canada
Email: jayakumar@amitysoft.com

How to cite this paper: Jayakumar, K.R. and Abran, A. (2017) Estimation Models for Software Functional Test Effort. *Journal of Software Engineering and Applications*, **10**, 338-353.
https://doi.org/10.4236/jsea.2017.104020

Received: January 25, 2017
Accepted: April 24, 2017
Published: April 27, 2017

Abstract

The International Software Benchmarking and Standards Group (ISBSG) database was used to build estimation models for estimating software functional test effort. The analysis of the data revealed three test productivity patterns representing economies or diseconomies of scale and these patterns served as a basis for investigating the characteristics of the corresponding projects. Three groups of projects related to the three different productivity patterns, characterized by domain, team size, elapsed time and rigor of verification and validation carried out during development, were found to be statistically significant. Within each project group, the variations in test effort can be explained, in addition to functional size, by 1) the processes executed during development, and 2) the processes adopted for testing. Portfolios of estimation models were built using combinations of the three independent variables. Performance of the estimation models built using the function point method innovated by the Common Software Measurement International Consortium (COSMIC) known as COSMIC Function Points, and the one advocated by the International Function Point Users Group (IFPUG) known as IFPUG Function Points, were compared to evaluate the impact of these respective sizing methods on test effort estimation.

Keywords

COSMIC Function Points, Estimation, Functional Sizing, Performance Measurement, Software Testing

1. Introduction

This paper reports on a set of estimation models designed with data chosen from the ISBSG repository consisting of functional sizes reported both in IFPUG function points [1] and COSMIC function points. These estimation models were evaluated using criteria for measuring outputs from estimation models. The

models were compared to understand their performance based on the measure of their predictability.

The motivation for this research work arises from the fact that existing techniques for estimating test effort (such as judgment-based, work breakdown, factors & weights, and functional size based techniques) suffer from several limitations [2] [3], while other innovative approaches for estimating testing effort (such as fuzzy inference, artificial neural networks, and case-based reasoning as proposed in the literature) are yet to be adopted in the industry. There is a growing body of work on the use of the COSMIC function points [4] [5] for estimation and performance measurement of software development projects which can be adapted for estimating software test effort too.

The remainder of this paper is structured as follows. Section 2 presents the data preparation; Section 3 is data analysis; Section 4 is the estimation models and Section 5 is the conclusions.

2. Data Preparation

2.1. ISBSG Data

Release 12 of ISBSG data published in 2013 [6] consists of data related to parameters of software projects re-ported over the last two and half decades, providing industry and researchers with standardized data for benchmarking and estimation. The ISBSG dataset has been extensively reviewed for its applicability to building effort estimation models, including effects of outliers and missing values [7] [8].

The attributes of interest for test effort estimation models are:

a. Functional size data based on international measurement standards such as IFPUG and COSMIC function points.

b. Schedule, team size, work effort information, project elapsed time and breakdown of work effort by project phase (planning, specifications, design, build, test and install).

c. Project process related data based on software life cycle activities (e.g. planning, specifications, design, build, test) and adoption of practices from standards or models such as ISO 9001, CMMI, SPICE, PSP etc. used in developing the software.

d. Grouping attributes: industry sector, application group (e.g., business, real time etc.), and development type (new development, enhancement or re-development).

e. Development platform: PC, mid-range, main frame or multi-platform.

f. Architecture: whether the application is standalone, multi-tier, client/server or Web-based.

g. Language type: 3GL, 4GL, or application generators used in development.

h. Overall data quality rating assigned by the ISBSG: A, B, C or D indicating very good to unreliable.

i. Function points data quality rating assigned by the ISBSG: A, B, C or D ranging from very good to unreliable.

2.2. Data Preprocessing

A set of criteria was defined to ensure data quality, relevance to current industry needs, suitability to the testing context and adequacy for statistical analysis, as follows:

1) Data Quality

a. ISBSG quality rating:

Data quality ratings of A and B were selected to reduce risk and improve confidence in the results.

b. Function point size quality:

When IFPUG function points were used for the measurement of size, only the un-adjusted function point value was considered. Function point data quality ratings of C and D were excluded from the data.

2) Data Relevance

ISBSG data consist of projects reported since the early 90s. Data prior to 2000 and projects with an architecture type of "standalone" were removed while client/server or Web-based projects were considered for modelling.

3) Data Suitability

To exclude trivial projects, the following filters were applied:

a. Total normalized work effort (full life cycle effort for project) equal to or greater than 80 hours.

b. Efforts reported for testing greater than or equal to 16 hours.

c. Types of testing other than functional testing were excluded.

4) Data Adequacy

a. Application group chosen: business.

b. Development type chosen: new development and re-development.

2.3. Generation of Datasets

Applying the filters related to the criteria for data selection and removal of outliers resulted in 142 data points, which were then grouped to form four datasets:

Dataset A: This dataset consists of all 142 data points including project functional size measures reported in IFPUG 4.1 or COSMIC FP. For this study, they were not differentiated within dataset A as they correlate well even though the relationship is not the same across all size ranges [9].

Dataset B: In the case of dataset A, projects with an architecture field value of "standalone" were eliminated from the original ISBSG data set, while "blanks" were retained. To be very specific about the architecture type, "blanks" were also eliminated from dataset A to arrive at dataset B, with 72 data points.

Dataset C: Data set C is made up of projects where functional size was reported in COSMIC function points. It is a subset of data set A and has 82 data points.

Dataset D: Dataset D includes only projects where functional size was reported in IFPUG function points. It is another subset of dataset A and contains 60 data points.

3. Data Analysis

3.1. Strategy

The following strategy was adopted for data analysis:

a) Identify data point subsets exhibiting different levels of testing productivity.

b) Analyze these subsets to identify the possible causes for the differences in productivity.

3.2. Identification of Test Productivity Levels

The scatter diagram in Figure 1 depicts a large dispersion between functional size and test effort, the independent and dependent variables, respectively. The pattern is closer to wedge-shaped and is typical of data from large repositories [10].

Within the dataset of Figure 1, there are candidate groups exhibiting both large economies of scale and large diseconomies of scale. The rate of increase of test effort is not the same for all similar functional sizes. Analyzing various slices of data brought out different testing productivity levels (Figure 2).

As economies and diseconomies of scale correspond to different productivity levels, a new term "test delivery rate" (TDR) was defined to describe project testing productivity. TDR is the rate at which software functionality is tested as a factor of the effort required, and is expressed as hours per functional size unit

Figure 1. Scatter diagram: size versus test effort.

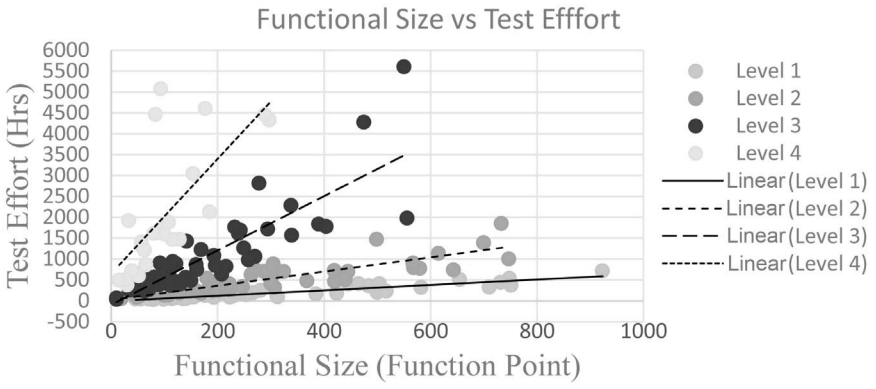

Figure 2. Multiple data groups representing different economies of scale.

(hr./FSU). Functional size unit (FSU) refers to either IFPUG or COSMIC function points, depending upon the sizing method used for measurement. The four varying levels of productivity are referred as "TDR levels". TDR being the effect, the characteristics of the projects falling into each level were then investigated to identify the underlying causes. Due to the highly dispersed nature of TDR level 4, only TDR levels 1 to 3 were taken up for further analysis and development of the estimation models.

3.3. Identification of Candidate Characteristics of Projects

Previous research work [11] [12] based on data from hundreds of software projects has indicated that team size and schedule (duration of the project) within a particular domain affect the productivity of development projects. As software testing is one of the phases of development, project attributes such as domain, team size and elapsed time are likely causes for test productivity, too. Testability of the software components, *i.e.*, quality of the software delivered for testing, is critical for reducing the testing cost [13] and hence the effort for testing. The quality of the software delivered for testing can be determined by the extent of verification and validation activities carried out during the development process.

From these previous studies, therefore, in choosing candidates of interest for our investigations we selected the following project characteristics:

Team size: We classified team size into three categories typically present in the industry: (a) 1 - 4 persons, (b) 5 - 8 persons and (c) more than 8 persons resenting small, medium and large team sizes, respectively.

Elapsed Time: Elapsed time in calendar months was derived from the "project elapsed time". Based on this attribute, projects were classified into three groups: (a) 1 - 3 months, (b) 4 - 6 months and (c) greater than 6 months, referred to as small, medium, and large, respectively.

V & V rigor: This attribute was derived from the data fields related to the "Documents & Techniques" category in ISBSG, which indicates the degree of rigor applied during verification and validation. Two ratings are proposed for V&V rigor (Table 1).

Application domain: This attribute was derived from the ISBSG data field "Industry Sector". Considering the number of data points available for different industry sectors, the application domain was classified into three categories, namely: (a) banking, financial services and insurance (BFSI) (b) education and (c) government (govt).

Table 1. V&V rigor rating scheme.

V&V Rigor Rating	Description
Low	Little or no evidence of reviews/inspection
High	Reviews/inspection reported for at least one of the specification, design and build phases

The group of projects contributing to each TDR level was termed the project group. Accordingly, project group 1 (PG1), project group 2 (PG2) and project group 3 (PG 3) refer to TDR levels 1, 2 and 3 discussed in Section 3.2. Based on the percentage of projects falling into each of the project groups for the four attributes of interest (Table 2), we were able to characterize the project groups.

Close to half of the BFSI projects (46%) fell into PG3 followed by a third in PG2. All education projects fell into PG1 while slightly more than half of the government projects fell in PG2. Close to two thirds of projects with a small team size fell into PG1, while 82% (46% + 36%) those with a medium team size were distributed between PG1 and PG2. Slightly less than 50% of the large team size projects fell into PG3. Close to two thirds of projects with a small elapsed time went into PG1, while 71% (38% + 33%) of those with a medium elapsed time were spread between PG2 and PG3. Similarly, PG2 and PG3 shared 72% (40% + 32%) of the projects with a large elapsed time. Projects with higher V&V rigor had a two thirds presence in PG1 while 77% (38% + 39%) of the lower V&V rigor projects were divided more or less evenly between PG2 and PG3.

The above analysis reveals that the three project groups have certain distinctions with respect to the domain, team size, elapsed time and V&V rigor besides test productivity. To establish statistical significance, a test of hypothesis was performed and the p value computed. The significance test conducted on the three project groups with respect to these attributes resulted in a p value of less than 0.001 for V&V rigor and domain and less than 0.1 for team size and elapsed time. This further establishes that the variations across the three project groups are reasonably significant and the attributes identified are potential contributors to test productivity.

Table 2. Analysis of project characteristics—Dataset A.

Domain	No. %	PG1	PG2	PG3	Team Size	No. %	PG1	PG2	PG3
BFSI	No.	14	23	32	**Small**	No.	10	4	2
	%	20	33	46		%	63	25	13
Education	No.	11	0	0	**Medium**	No.	18	14	7
	%	100	0	0		%	46	36	18
Govt.	No.	6	10	2	**Large**	No.	5	4	8
	%	33	56	11		%	29	24	47
Elapsed Time	No. %	PG1	PG2	PG3	**V & V Rigor**	No. %	PG1	PG2	PG3
Small	No.	18	7	5	**Low**	No.	25	42	43
	%	60	23	17		%	23	38	39
Medium	No.	6	8	7	**High**	No.	21	7	4
	%	29	38	33		%	66	22	13
Large	No.	14	20	16					
	%	28	40	32					

The results of the analysis of the project characteristics in the three datasets A to C, excluding dataset D, demonstrate similar behavior. Characteristics of project groups PG1 to PG3 based on these attributes are summarized in Table 3.

3.4. Identification of the Independent Variables

1) Size

It has been observed that functional size is the most accepted approach for measuring size, as sensitivity to changes in functional size has a greater impact on project effort [14] [15]. Here, correlation coefficients computed using dataset A, between size and test effort values of 0.9035 for PG1, 0.8572 for PG2 and 0.8572 for PG3, indicate good correlation of functional size with effort. Size was therefore chosen as the primary independent variable.

2) Non-Size Variables

Size being the main independent variable, other independent variables were next examined for significance of incorporating them into estimation models. It has been observed [13] that "testability of software components", meaning the quality of the software delivered for testing and testing processes followed while testing, are critical factors for reducing testing effort and improving software quality. To accommodate these process factors two new variables representing development process quality and testing process quality were defined and investigated as follows.

a) Development Process Quality Rating (DevQ)

The process followed during development was rated by considering the nature of the development life cycle followed and the artefacts produced, based on the following project attributes:

- Standards followed.
- Distinct development life cycle phases followed.
- Verification activities carried out during development.

The ISBSG data field "software process" has one of the values—CMMI, ISO, SPICE, PSP or any such standard followed during development. A set of fields representing "Documents and Techniques" exists in the ISBSG data providing information on the life cycle phases adopted and verification activities carried out during development. Based on these, a rating for DevQ was developed, as shown in Table 4.

b) Test Process Quality Rating (TestQ)

While reviewing, the data related to the testing process followed, it was found

Table 3. Characteristics of project groups.

Attribute	PG1	PG2	PG3
Domain	Educational	Government	BFSI
Team Size	Small/Medium	Small/ Medium	Large
Elapsed Time	Small	Medium/Large	Medium/Large
V&V Rigour	High	Low	Low

Table 4. Rating for development process (DevQ).

Software Process	Documents & Techniques	DevQ Rating
Not reported	Very little reporting to infer	0
Reported	Very little reporting to infer	1
Not reported	One or more phases has values	1
Reported	One or more phases has values	2

that there were not enough fields in the ISBSG data to capture the details of the testing process, such as testing techniques adopted, levels of testing executed, test artefacts produced, reviews of test cases etc., to gauge the extent of testing. This notwithstanding, it was possible to classify the test process rating broadly into two categories (Table 5).

3.5. Analysis of DevQ and TestQ

Projects in data set A were analyzed in terms of DevQ and TestQ:

- 36%, 49%, and 15% of the projects were found to be in DevQ with ratings 0, 1 and 2, respectively.
- 80% and 20% of the projects had TestQ ratings 0 and 1 respectively.

To further justify the inclusion of these variables, two statistical tests were carried out to quantify their significance (Table 6):

- the Kruskal-Wallis Test for DevQ as it involved three categories, and
- the Mann Whitney Test was applied for TestQ.

The p value indicated that size, DevQ and TestQ were statistically significant.

4. Estimation Models

4.1. Portfolio of Models

The linear form of relationship between input and output variables was chosen to build models for effort estimation. Linear regression analysis, a well-known and well understood algorithm in statistics and machine learning, does not require much training data, and is easily interpreted by project managers. Parametric models are objective, repeatable, fast and easy to use, and can be used early in the life cycle if they are properly calibrated and validated [16]. A set of 24 models under four portfolios were generated (Table 7) using datasets A to D.

Portfolio A models based on dataset A:

Models 1, 2 and 3 are for each project group using size as the independent variable.

Models 4, 5 and 6 use both size and DevQ as independent variables and relate to project groups 1, 2 and 3 respectively.

Models 7, 8 and 9 use size, DevQ and TestQ as independent variables and represent project groups 1, 2 and 3 respectively.

Portfolio B models based on dataset B:

Models 10, 11 and 12 relate to project groups 1, 2 and 3 respectively using size as independent variable.

Table 5. Rating for test process (TestQ).

Test Process Criteria	Test Process Rating (TestQ)
No evidence of Test Artefacts	0
Evidence of Test Artefacts	1

Table 6. Test of significance for independent variables.

Statistical Test	Variable	*p* Value
Chi Square	Size	< 0.001
Kruskal-Wallis Test	DevQ	0.005
Mann-Whitney Test	TestQ	0.003

Table 7. Portfolio of estimation models.

Portfolio	ID	PG	Model Coefficients							
			A	B	1D		2D		1T	2T
					DevQ = 0	DevQ = 1	DevQ = 0	DevQ = 1	TestQ = 0	TestQ = 0
A	1	1	1.617	0.604						
	2	2	20.69	1.705						
	3	3	98.13	4.801						
	4	1	16.12	0.485	19.347	−39.375	−0.23	0.214		
	5	2	20.57	1.56	−94.1	34.077	0.562	−0.009		
	6	3	38.85	3.734	−55.913	92.609	2.14	0.852		
	7	1	−9.62	0.65	6.967	−41.78	0.003	0.193	38.124	−0.191
	8	2	30.74	1.541	−19.755	62.481	−0.039	−0.338	−84.511	0.62
	9	3	38.85	3.734	−55.913	92.609	2.14	0.852	0	0
B	10	1	−8.3448	0.61						
	11	2	−30.569	1.929						
	12	3	−157.62	6.126						
	13	1	16.124	0.485	46.572	−52.672	−0.201	0.222		
	14	2	20.57	1.56	−180.84	−60.58	0.973	0.313		
	15	3	38.847	3.734	−375.38	5.027	3.881	1.449		
	16	1	−12.583	0.68	58.608	−43.272	−0.208	0.171	16.67	−0.188
	17	2	56.462	1.492	2.443	129.634	−0.354	−1.025	−219.18	1.395
	18	3	38.847	3.734	−375.38	5.027	3.881	1.449	0	0
C	19	1	−20.142	0.693						
	20	2	47.999	1.59						
	21	3	136.267	4.481						
D	22	1	37.588	0.455						
	23	2	−29.939	1.917						
	24	3	77.585	6.087						

Models 13, 14 and 15 belong to project groups 1, 2 and 3 respectively using size and DevQ as independent variables.

Models 16, 17 and 18 refer to project groups 1, 2 and 3 using size, DevQ and TestQ as independent variables.

Portfolio C models based on dataset C (COSMIC FP Projects):

Model 19, 20 and 21 relate to project groups 1, 2 and 3 respectively using size as independent variable.

Portfolio D models based on dataset D (IFPUG FP Projects):

Model 22, 23 and 24 relate to project groups 1, 2 and 3 respectively using size as independent variable. Depending upon the number of independent variables, model equations have coefficients A, B, D1, D2, T1 and T2 (Table 7), which were used to estimate the value for test effort for specific values of size, DevQ and TestQ as explained next.

Using estimation models based on size:

Test effort for a particular functional size can be estimated from models using the following equation representing size based estimation models:

$$\text{Test Effort} = \text{A} + \left(\text{B} \times \left(\text{Size}\right)\right) \tag{1}$$

Test effort for a particular functional size can be computed by using the values of A and B from Table 7 and substituting functional size for "size" in Equation (1).

Using estimation models based on size and DevQ:

Test effort for a particular value of functional size and DevQ can be estimated from models using the following equation:

$$\text{Test Effort} = \text{A} + \left(\text{B} \times \text{Size}\right) + \text{D1} + \left(\text{D2} \times \text{Size}\right) \tag{2}$$

Test effort for a particular functional size where rating for DevQ is available can be computed using Equation (2). D1 and D2 have different values based on the value of DevQ. Appropriate values from Table 7 are to be chosen depending on whether DevQ = 0 or Dev Q = 1. For DevQ = 2, the value is 0, the base value considered while modelling.

Using estimation models based on size, DevQ and TestQ:

The equation for estimating Test Effort for particular values of size, DevQ and TestQ from the model has the form:

$$\text{Test Effort} = \text{A} + \left(\text{B} \times \text{Size}\right) + \text{D1} + \left(\text{D2} \times \text{Size}\right) + \text{T1} + \left(\text{T2} \times \text{Size}\right) \tag{3}$$

Equation (3) can be used for computing test effort estimate for a particular functional size when ratings for both DevQ and TestQ are available. Values for D1 and D2 are to be chosen from Table 7 depending upon the input value of DevQ, is either 0 or 1. Values for T1 and T2 are provided for TestQ = 0. Values for DevQ = 2 and Test Q = 1 are zero, as they were the baseline for the modelling.

An estimator chooses the project group by mapping the characteristics of the project to be estimated to the attributes of project group and selects the related data set in order to choose the closest model for estimation.

4.2. Evaluation of Estimation Models

The quality of estimation models was evaluated using criteria such as coefficient of determination (R^2), Adj R^2, magnitude of relative error (MRE), median magnitude of relative error (MedMRE) [10] (Table 8).

The value of R^2 for portfolio A ranged between 0.74 and 0.86, and that of Adj R^2 ranged between 0.73 and 0.83 indicating a strong relationship between the independent variables-size, DevQ and TestQ with the dependent variable test effort in all models.

The value of MedMRE ranging between 0.22 and 0.28 shows that the error levels between the estimate and actual are within the range of 22% to 28% for 50% or less of the samples, which is practical considering the multi-organizational data used for building the models.

Similar observations can be made for rest of the models.

Table 8. Quality of estimation models.

Portfolio	Model id	No. of projects	R^2	Adj R^2	MedMRE
A (N = 142)	1	46	0.82	0.81	0.24
	2	49	0.74	0.73	0.27
	3	47	0.77	0.79	0.25
	4	46	0.85	0.83	0.24
	5	49	0.75	0.73	0.28
	6	47	0.79	0.77	0.22
	7	46	0.86	0.83	0.23
	8	49	0.78	0.74	0.24
	9	47	0.79	0.77	0.22
B (N = 72)	10	32	0.80	0.8	0.24
	11	24	0.67	0.66	0.26
	12	16	0.83	0.82	0.25
	13	32	0.84	0.81	0.22
	14	24	0.70	0.62	0.25
	15	16	0.91	0.86	0.10
	16	32	0.87	0.83	0.20
	17	24	0.70	0.57	0.25
	18	16	0.91	0.86	0.10
C (N = 82)	19	27	0.87	0.86	0.19
	20	26	0.73	0.71	0.30
	21	29	0.82	0.82	0.23
D (N = 60)	22	19	0.78	0.77	0.25
	23	23	0.76	0.75	0.26
	24	18	0.70	0.68	0.33

4.3. Comparison of Model Performance

4.3.1. Predictive Performance of Models

The criterion used to evaluate the predictive quality of an estimation model was PRED (l) = k/n, where k is the number of projects in a specific sample of size n for which MRE <= l. In the software engineering literature, an estimation model is considered good when PRED (0.25) = 0.75 [17] or PRED (0.30) = 0.70 and PRED (0.20) = 0.80 [10]. PRED (0.25) = 0.75 means 75% of the samples should have MRE values less than or equal to 0.25. While an MRE error level in 75% of the population less than 0.25 is the expectation of this criterion, multi-organizational data such as in ISBSG data exhibit large MRE for 75% of the population.

To compare the performance of models, MRE values for 50% and 25% of the population in addition to 75% were taken into consideration. Each vertical bar in the charts (Figures 3-7) depicts the MRE value for 50% in the middle with either extremes showing MRE values for 25% and 75% of the population for each of the identified model. The middle points of each bar (MRE for 50% of population) are connected by a line to visualize the difference between successive models. This point is referred simply as MRE in the following discussions and was used to compare the performance of the models.

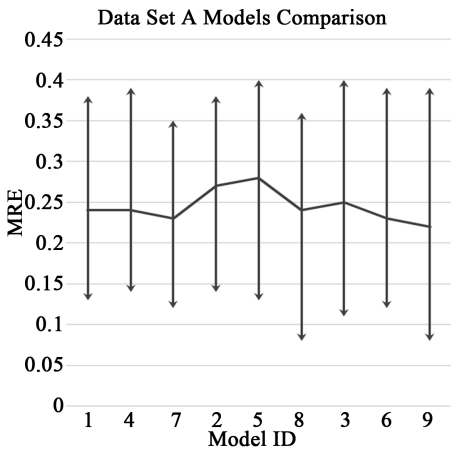

Figure 3. Performance of data set A models.

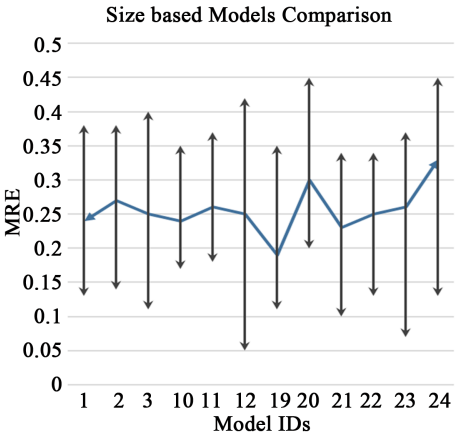

Figure 4. Performance of size based models.

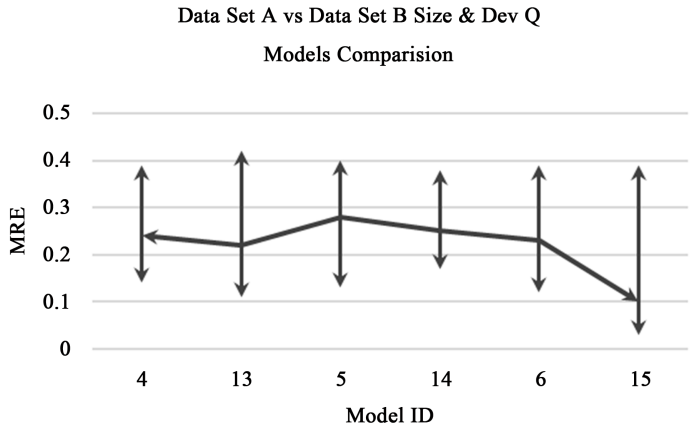

Figure 5. Size & DevQ models in data set A and B.

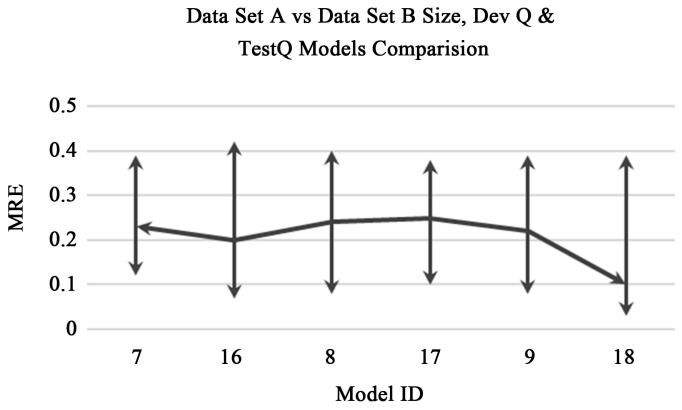

Figure 6. Size, DevQ & TestQ models in data set A and B.

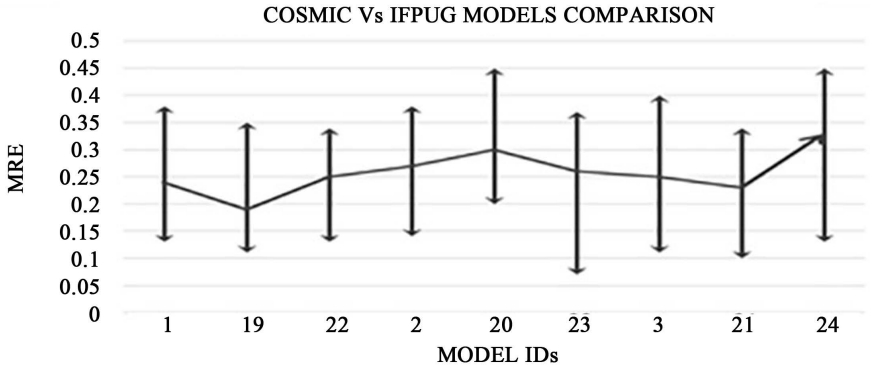

Figure 7. Performance of COSMIC vs. IFPUG models.

4.3.2. Dataset A Models in Portfolio A

There are 9 models in portfolio A. A comparison of these nine models reveals how predictability varies between project groups and while using different independent variables. Figure 3 depicts the MRE levels of models corresponding to PG1 (the leftmost three bars), PG2 (the next three bars) and PG3 (the rightmost three bars).

Within PG1, the model with size, DevQ and TestQ as independent variables (model 7) demonstrate lower MRE for 50% of the population compared to the

model with size, Dev Q (model 4) which is lower than the model with size alone (model 1).

A similar pattern is observed in PG3 for models 3, 6 and 9. In the case of PG2, the model using size, DevQ and TestQ (model 5) exhibits higher MRE compared to other models in PG2 (models 2 & 8) as well as for all the models in dataset A.

4.3.3. Size-Based Models

A comparison of all models using only size as an independent variable across all portfolios shows under which context size-based models provide better predictability. Figure 4 illustrates size-based models from each portfolio, the first three bars corresponding to each project group in portfolio A, the next three corresponding to each project group in portfolio B and so on.

In summary:

- Size-based models for project groups PG1 and PG3 are better than PG2 with the exception of portfolio D.
- PG3 size models, in general perform better than PG1 and PG2 except for the last model (Model ID 24).

4.3.4. Models in Portfolios A and B

Portfolio B models were developed using a subset of data used for portfolio A. Portfolio B models were more specific to web or client/server architecture, unlike portfolio A models where there was an approximation due to differences in architecture. A comparison between the models across portfolios A and B using the independent variables DevQ and TestQ along with size helped to make certain observations. Figure 5 depicts size & DevQ models for PG1, PG2 and PG3 for dataset A and B, while Figure 6 illustrates size, DevQ and TestQ models for PG1, PG2 and PG3 for data set A and B.

Examination of Figure 5 reveals that models in portfolio B (models 13, 14,15,16 & 18) performed much better than models in portfolio A (models 4 to 9) with model 17 being an exception.

4.3.5. COSMIC and IFPUG Models

The performance of COSMIC (dataset C) and IFPUG (dataset D) models was compared next using size-based models from portfolio A as the reference. Both COSMIC and IFPUG data are subsets of dataset A consisting of projects measured using the corresponding sizing method. This comparison can help to evaluate prediction accuracy of COSMIC-based models versus IFPUG-based models. Figure 7 depicts PG1 models for dataset A (model 1), COSMIC (model 19), IFPUG (model 22), PG2 models for data set A (model 2), COSMIC (model 20), IFPUG (model 23) and PG3 (model 3), COSMIC (model 21) and IFPUG (model 24).

COSMIC-based estimation models using dataset C had better performance than IFPUG-based estimation models using data set D, with the exception of PG2 (model 20). COSMIC-based PG3 model demonstrated the best predictability. Furthermore, the R^2 values for COSMIC-based models ranged from 0.73 to

0.87 while that of IFPUG-based models ranged from 0.70 to 0.78 (Table 8). The MedMRE value for COSMIC-based models ranged between 0.19 and 0.30 compared to IFPUG based models ranging between 0.25 and 0.33 (Table 8) demonstrating better accuracy of COSMIC based models.

5. Conclusions

This research work explored software testing from the perspective of estimation of efforts for functional testing. The ISBSG database, with its wealth of project data from around the globe, was used for the first time in building effort models for functional testing. The analysis of the data revealed three test productivity patterns representing economies and diseconomies of scale, based on which characteristics of the corresponding projects were investigated. Three project groups, characterized by domain, team size, elapsed time and rigor of verification and validation, and related to three productivity patterns were found to be statistically significant. Within each project group, the variations in test effort could be explained, apart from the functional size, by 1) the processes executed during the development, and 2) the processes adopted for testing.

Two new independent variables, DevQ and TestQ were identified as influential in the estimation of effort. A total of 24 models were built, using combinations of the three independent variables. The quality of each model was evaluated using established criteria such as R^2, Adj R^2, MRE and MedMRE. As these models were built from ISBSG data, they could serve as an industry benchmark for functional test efforts. Test estimation models using projects measured in COSMIC function point exhibited better quality and resulted in more accurate estimates compared to projects measured in IFPUG function points.

The models are applicable only for the ranges of size in the data set and for testing of business applications. The models generated are not applicable for enhancement projects. These limitations can be overcome by generating specific models for enhancements or real-time projects, using an approach like the one followed in this work. This may require identification of additional project characteristics, as well as other variables influencing testing effort. PG4—the fourth group of project data points remains to be analyzed.

The process factors used for rating DevQ and TestQ can be further refined within organizational context. There could be other variables that influence test efforts in specific contexts, which would require further study and analysis. The estimation models designed can be further refined by considering testing techniques adopted as a parameter to evaluate their impact and then used to build estimation models.

Acknowledgements

The authors would like to acknowledge the support provided by COSMIC consultant Srikanth Arvamudhan and statistician Sriram Ramachandran during this work.

References

[1] ISO/IEC 20926:2009 (2009) Software and Systems Engineering—Software Measurement—IFPUG Functional Size Measurement Method. International Organization for Standardization (ISO), Geneva.

[2] Jayakumar, K.R. and Abran, A. (2013) A Survey of Software Test Estimation Techniques. *Journal of Software Engineering and Applications*, **6**, 47-52. https://doi.org/10.4236/jsea.2013.610A006

[3] Abran, A. (2010) Software Metrics and Software Metrology. Wiley & IEEE Computer Society Press, New Jersey. https://doi.org/10.1002/9780470606834

[4] ISO/IEC 19761 (2011) Software Engineering—COSMIC—A Functional Size Measurement Method. International Organization for Standardization (ISO), Geneva.

[5] COSMIC (2015) The COSMIC Functional Size Measurement Method, Version 4.0.1, Measurement Manual, The COSMIC Implementation Guide for ISO/IEC 19761:2011. Common Software Measurements International Consortium, Canada, April 2015. www.cosmic-sizing.org

[6] ISBSG (2013) Repository Data Release 12—Field Descriptions, "e.Field Descriptions—Data Release 12. Pdf" Document Provided as a Part of Data Set. International Software Benchmarking and Standards Group.

[7] Bala, A. (2013) Impact Analysis of a Multiple Imputation Technique for Handling Missing Value in the ISBSG Repository of Software Project. PhD Thesis, Ecole de technologie superieure, University of Quebec, Montreal (Canada), 17 October.

[8] Bala, A. and Abran, A. (2016) Use of the Multiple Imputation Strategy to Deal with Missing Data in the ISBSG Repository. *Journal of Information Technology & Software Engineering*, **6**, 171.

[9] Dumke, R. and Abran, A., Eds., (2011) COSMIC Function Points, Theory and Advanced Practices, Chapter 3.5: Measurement Convertibility—From Function Points to COSMIC FFP. CRC Press, New York.

[10] Abran, A. (2015) Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers. John Wiley & Sons & IEEE Computer Society, New Jersey. https://doi.org/10.1002/9781118959312

[11] Putnam, D. (2005) Team Size Can Be the Key to a Successful Software Project. Quantitative Software Management Inc., USA. www.qsm.com

[12] Armel, K. (2012) Top Performing Projects Use Small Teams Deliver Lower Cost, Higher Quality, Blog Posting. Quantitative Software Management, Inc., USA.

[13] Taipale, O. (2007) Observations on Software Testing Practice. PhD Thesis, Lappeenranta University of Technology, Finland.

[14] Bharadwaj, M. and Rana, A. (2015) Estimation of Testing and Rework Efforts for software Development Projects. *Asian Journal of Computer Science and Information Technology*, **5**.

[15] Hill, P., ISBSG (2010) Practical Software Project Estimation: A Toolkit for Estimating Software Development Effort and Duration. McGraw-Hill, New York.

[16] Galorath, D. (2015) Why Can't People Estimate: Estimation Bias and Mitigation. *Conference presentation at IT Confidence Conference*, Rio de Janerio, October 2015.

[17] Conte, S.D., Dunsmore, D.E. and Shen, V.Y. (1986) Software Engineering Metrics and Models. The Benjamin/Cummings Publishing Company, Inc., Menlo Park.