Computing and Informatics, Vol. 37, 2018, 553–580, doi: 10.4149/cai_2018_3_553

EFFORT ESTIMATION FOR SERVICE-ORIENTED COMPUTING ENVIRONMENTS

Siba Mishra, Chiranjeev Kumar

Department of Computer Science and Engineering
Indian Institute of Technology (Indian School of Mines)
Dhanbad, 826004
Jharkhand, India
e-mail: sibamishracse@gmail.com, k_chiranjeev@yahoo.uk

Abstract. The concept of service in Service-Oriented Architecture (SOA) makes possible to introduce other ideas like service composition, governance and virtualization. Each of these ideas, when exercised to an enterprise level, provides benefits in terms of cost and performance. These ideas bring many new opportunities for the project managers in making the estimates of effort required to produce SOA systems. This is because the SOA systems are different from traditional software projects and there is a lack of efficient metrics and models for providing a high level of confidence in effort estimation. Thus, in this paper, an efficient estimation methodology has been presented based on analyzing the development phases of past SOA based software systems. The objective of this paper is twofold: first, to study and analyze the development phases of some past SOA based systems; second, to propose estimation metrics based on these analyzed parameters. The proposed methodology is facilitated from the use of four regression(s) based estimation models. The validation of the proposed methodology is cross checked by comparing the predictive accuracy, using some commonly used performance measurement indicators and box-plots evaluation. The evaluation results of the study (using industrial data collected from 10 SOA based software systems) show that the effort estimates obtained using the multiple linear regression model are more accurate and indicate an improvement in performance than the other used regression models.

Keywords: Effort estimation, orchestration, SOA, regression, web services

Mathematics Subject Classification 2010: 68N30

1 INTRODUCTION

Prevalent business and industrial organizations around the globe adheres SOA style for building business, commercial and financial software applications. This is because SOA provides a promising way for addressing many problems related to the integration of heterogeneous applications in a distributed environment [1]. SOA is an architectural approach for developing enterprise level business systems using *loosely coupled* interoperable services. Services – the core component of SOA is defined as a *logical encapsulation* of self-contained business functionality. Technically, the term *self-contained functionality* suggests that any changes to the available services could be incorporated without affecting other services of the system. Moreover, the use of services in SOA increases the overall *flexibility* and adds improved flow of *functionality*. Due to this implicit advantage, in the last decades, SOA emerged up quite rapidly and has numerous applications in the field of biotechnology, health care systems, communication networks, irrigation, mass-customizations and e-health support services [38, 39, 40, 41, 42, 43, 44, 45, 46].

From these broad applications and advantages of SOA, it is clear that the design and development activities are different from that of traditional programming paradigms [2, 3]. Further, the development of SOA systems introduces many new concepts, technological factors and architectural issues for building complex business applications. These *new concepts* include services, messages, property of orchestration, loose coupling and many more [3, 4]. Also, developing SOA systems for business, financial and banking sectors are much more complex and expensive specifically in terms of resources and schedules. In the context of SOA project management, these new concepts and principles add many complex issues that are different from traditional software development paradigms [6]. In this way, the development of SOA systems is different from traditional software development. Moreover, from having an efficient effort estimate, a valid conclusion about the SOA system implementation phase are drawn for some measurement dimensions.

Estimation of effort[1] is an essential component of software project management. It is also a prerequisite feature of any software process, whether it is the design, testing, development, usability or the application as a whole. Generally, estimation depicts the way things will happen in the future based on the present conditions. In fact, it is an approximation for which some outcome is expected instead covering the set of possible outcomes. Having an efficient effort estimation technique is widely perceived by the business analysts and project managers. This is because an efficient estimation methodology helps in utilizing the project resources conveniently and thus helpful in avoiding project overestimation and late delivery [15]. As above mentioned, the development activities of SOA systems are different from traditional softwares. Thus, the existing software size and effort estimation techniques are not

[1] In the field of Software Engineering, "effort" estimation is also known as "cost" estimation. In this section and throughout the paper, both the terms have been used interchangeably.

adequate to capture specific development features for influencing the development effort parameters in building of the SOA based software applications.

For this objective and the aforementioned issues, we adopt a similar classification framework proposed by Lowe et al. [7] and Mendes et al. [8] for predicting the design and authoring effort of web hypermedia and software applications. However, our contribution includes the following additional research, i.e., the usage of metrics is designed and proposed considering various SOA related artifacts like orchestration, services, principle of loose-coupling and messages on different scales for estimating the development effort. In general, the service design phase covers the modeling of total number of processes – that is tasks and other constituent elements (like looping, parallel flow and synchronization) required for building SOA systems. This suggests by analyzing the service design phase, different measures could have been obtained for the SOA systems and that is considered as a suitable predictor of effort.

In our work, we measured some interesting theories relevant to service design phase and proposed some associated cost drivers necessary for predicting the SOA systems development effort. The proposed metrics highlights the design related issues of SOA systems mainly supported from the environment configuration and total size. Besides, following are the highlights of this work:

- The proposed approach geared up from an initial study with a motivation of identifying some *design measures* related to SOA systems.
- Introduction of *novel metrics* for facilitating the estimation methodology for the identified parameters of the initial study.
- For evaluating the accuracy (in terms of predictive power) for the obtained results, rigorous experiments were carried out using some statistical significance tests, performance measurement indicators and box-plots evaluation.

The rest of the paper is organized as follows: Section 2 discusses related works. Section 3 presents the principles of methodology relevant to our work. The proposed work has been introduced in Section 4. Section 5 reports and analyzes the empirical results and discussions. Section 6 concludes the paper.

2 RELATED WORK

So far in the literature, adequate attempts have been made to solve the problem of effort estimation for *traditional softwares* [15, 16, 17, 18, 19, 20, 21, 22]. These techniques[2] are classified mainly into probabilistic and statistical, expert judgement, analogy, algorithmic and machine-learning based estimation techniques [16, 48].

Generally, *probabilistic models* use the Baye's theory and probabilistic method for predicting the development effort. The *statistical models* use the method of regression for estimating the software development effort for some past data. The

[2] In this section and throughout the paper, the term techniques and models have been used interchangeably.

expert judgement models [18] involve consultation with one or more local experts, having knowledge about the core design and development environment or application domain in context to software project management. *Analogy models* estimate the development effort of a target project as a function of known efforts from a set of similar historical projects [19, 20]. In *algorithmic models*, the development costs are analyzed using some mathematical formula linking the costs with metrics to produce an estimated output. Next, the formula is applied to a formal model arising from the analysis of historical data. The *machine learning techniques* use both supervised and unsupervised learning techniques, for the training purpose and the development effort are calculated using a set of historical datasets. Each and every above mentioned estimation techniques are used based on some certain conditions and requirements. Research has still been in progress for investigating the best prediction technique.

In the last decades, SOA approaches are used for developing software applications sourced as virtual hardware resources, including on-demand and utility computing [49]. SOA uses both *services* and *messages* to support the development of low-cost distributed applications [50]. Moreover, recently the service-oriented technologies gained the mainstream attention quite remarkably, as SOA addresses a promising way for creating the basis of agility using which the software industries deliver more flexible business processes [49]. Despite the wide practice of using SOA, plethora amount of research has already been devoted to service-orientation research road-maps, challenges, fundamental perspectives, evolution, re-usability, governance and composition [50, 51, 53, 54, 55]. However, we believe that the research on effort estimation of SOA systems is definitely a novice option with many interesting challenges and new opportunities in terms of future research. Also, the research on effort estimation of SOA systems are very scarce in the literature. In the literature of traditional software development effort estimation, most of the work focused on *algorithmic techniques*, whereas in SOA system effort estimation, the *algorithmic* along with *probabilistic* techniques covers more than half of the reported work. To the best of our knowledge, no *analogy*, *statistical* and *machine learning* based estimation techniques has been reported in literature. Nevertheless, some approaches [10, 11, 12, 13, 14, 23, 47] are worth mentioning facilitating an efficient estimation, without any consideration of *predictive accuracy* for the set of past project data.

For example, Liu et al. [13] proposed a *probabilistic approach* using the *Bayesian net model*. This technique focuses only on different *service governance* processes. This method [13] highlights the Bayesian approach for predicting the development effort and *more improvement needs to be incorporated* for providing a systematic and accurate prediction, as suggested for their future work. However, we believe that this objective may be fulfilled using some detailed indicators and mathematical models in the analysis procedure. O'Brien [12] from NICTA, Australia also introduced a *probabilistic based* framework [SMAT-AUS] for capturing various aspects of SOA projects. Furthermore, the proposed framework used for determining the *scope and development effort* by considering the *technical*, *social-cultural*, *maturity models* and

other *organizational* aspects. The SMAT-AUS framework is in its development stage and not yet fully developed. The complete framework may provide an efficient way for determining the scope and effort of SOA systems. The limitations of the above mentioned probabilistic approaches [12, 13], besides not being fully developed, are that it does not consider adequate cost drivers for balancing the trade-off between the estimation methodology and SOA systems.

The authors [11] introduced an *algorithmic* framework based on Divide and Conquer (D & C) technique for estimating the cost of building SOA softwares. The *novelty* of this approach is that the estimation mechanism is employed by focusing only the different types of services. However, this approach besides being incorporated as an efficient framework is also limited for not providing proper validation for the set of past project data. Additionally, Gomes [14] – A SOA architect of Unimix, introduced an *algorithmic* approach for estimating and counting SOA projects using service candidate descriptions, Web Services Description Language (WSDL) and XML Schema Definition (XSD). The proposed method is presumed as an *algorithmic approach* because of the use of *function points*. The proposed technique is suitable only for *small* sized projects and fails for large size and complex SOA systems. In addition, authors in [10] proposed a qualitative *expert judgement based approach* to *judge* the effort of different SOA styled project proposals before implementing the Web Services Compositions (WSCs). The authors borrowed D & C approach [11] to narrow down the problem of effort judgement of an entire SOA implementation rather individual Web Services. Moreover, the authors introduced a novel approach for determining the *effort factors* of WSCs, using *classification matrix* and *hypothesis*. This approach neither considers any case study evaluation nor provides any proper validation for the past project data.

Therefore, owing to the above reasons, some new metrics relevant to the design phase of SOA system are introduced in this paper. The estimation methodology comprises proposed metrics and statistic based regression techniques has been found as a suitable candidate for solving the problem of SOA system effort estimation. Furthermore, proper validation (predictive accuracy) has been incorporated into the calculated predicted values using some commonly used *performance measurement indicators* and *box-plots evaluation* for the data collected from multiple sources Indian software organization.

3 PRINCIPLES OF METHODOLOGY

In this section, we provide an overview about the principles of methodology and some essential background relevant to our work. We have used four statistic based regression techniques for calculating the predicted values. From the generated models, the development effort is computed based on some contextualized design related issues of SOA systems.

We have used regression techniques for predicting the development effort of SOA systems. "Regression" is one of the most popular statistical technique used

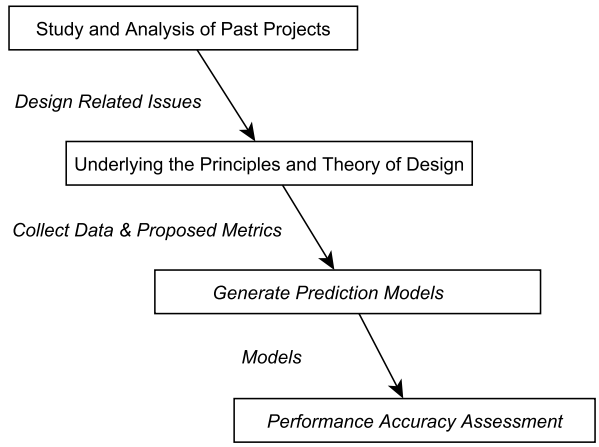

Figure 1. Flow diagram of our proposed work

commonly in the field of estimation. Typically, regression models represent the relationship between independent and dependent variables of a used dataset. Moreover, the most important identified parameters from the design phase of SOA systems are *initial heads*, *configuration environment*, *definition* and *length*. From these identified parameters, a set of metrics for the SOA system is proposed with the notion that these metrics conceived to have some significant impact on the total size of the application[3].

Figure 1 shows the basic flow diagram of our proposed approach. The different stages of the flow diagram are described below.

- The *objective* of the *first stage* is to study and analyze the design and development related issues of SOA systems. The *output* of this phase classifies the identified contextualized *design related issues* relevant to SOA systems.
- The *second stage* focuses on grasping some basic theories and principles relevant to design phase of SOA systems. Here, the main *objective* is to design some *key metrics*, *cost drivers* and *other theories* relevant to service design phase. One more *aim* of this phase is to collect data for these identified issues. The *output* of this stage provides *data* that are to be used for generating regression models. This stage also facilitates some key design related issues based upon which metrics are designed and proposed.
- The *third stage* emphasizes the generation of different regression models for some past project data. The stage wraps up soon after the generation of prediction models. This stage gets final completion of two defined *objectives*:

[3] An application is a process or a task implemented as a web service or a scripting language like Java Script or an orchestrated task or a fully integrated application.

  - *Validation of data values for the used dataset*: The objective of this sub-stage is to identify the *missing and influential* data-points for the used dataset and normalize the collected data values.
  - *Selection of appropriate variables and regression model*: This sub-stage assists in selecting some appropriate *dependent and independent variables* from the used dataset and choosing an appropriate regression model.

- The *final stage* illustrates performance assessment (predictive accuracy) of the generated estimation models. This is achieved with the help of some commonly used performance measurement indicators and statistical significance tests. The *objective* of this stage is to ensure the *accuracy level* of the calculated values.

The data of the used dataset constitute 10 different SOA styled applications. The dataset aimed at the following objectives.

1. Design and development of processes.
2. Implementing tasks as loosely coupled web services for the processes.
3. Development of service-oriented orchestrations using X-Path queries.
4. Development of parallel loops (⟨while⟩, ⟨repeatUntil⟩ and ⟨forEach⟩), concurrency elements (⟨scope⟩) and synchronization mechanism associated with the processes.

All the analytical and empirical results presented in this paper have been carried out using the data collected from the design related issues of past SOA styled software applications. The data corresponding to the used dataset are provided by an *Indian software organization*. The projects of the used dataset were developed between the years 2009 to 2013. Moreover, the projects corresponding to the used dataset include integrated SOA applications for *universal banking*, *public retail* and *health care solution systems*. The used dataset constitutes 10 different SOA applications with 30 data points.

Variable Name	Variable Description	N	Missing	Mean	Median	Std. Dev.	Min	Max
Actual_Effort (in PH)	Total development effort (PH)	10	0	7 014.91	3 363.47	7 931.52	377.8	22 479.3
No. of Processes	Total number of processes for an application	10	0	2.9	2.5	1.96	1	7
No. of Tasks	Total number of tasks for processes	10	0	6.9	5.5	4.58	2	15
TCC	Total size based on the definition of processes and tasks of an application	10	0	975.7	539	1 027.62	74	2 890
No of partnerLinks	Total number of partnerLink Elements	10	0	11.9	10.5	7.56	3	23
Task Variables	Total number of used input and output variables	10	0	72.6	71.5	38.25	22	130
Event Variables	Total number of receive and reply start events	10	0	6.2	5.5	4.10	2	13
Elements	Total number of variables and message definitions	10	0	14.9	13.5	7.15	7	28
XScript	Total number of X-path queries	10	0	7.4	4.5	6.65	1	20

Table 1. Descriptive statistics of some numerical variables of the used dataset

The analysis of our proposed methodology is aimed at measuring the metrics used as arguments for the generation of regression models. More concise form of

the proposed metrics is described in Section 4. We have presented the descriptive statistics of the used dataset in Table 1. The statistical summary is presented only considering *some numerical* variables of the dataset. In Table 1, the variable "TCC" denotes the total code size and "N" signifies total number of projects in the used dataset. The variable Actual_Effort (in PH) denotes the actual effort needed for developing the final application. The descriptive statistics are essential for carrying out the empirical study because it presents the data in more meaningful way and facilitates simpler interpretation of data.

4 PROPOSED WORK

This section illustrates the proposed methodology. After rigorous in-depth study and analysis, the metrics are proposed and presented in the first subsection. The process of generation of regression based estimation models using the proposed metrics is highlighted in the next subsection.

Items	Type	Description
Output	Integrated Application	Developed SOA styled integrated application
	Process and Element Definition	Process, abstract process and flow elements
	Tasks	Processes tasks implemented as web services
	SOA-Orchestration	X-path Queries
	Scripting Languages	Java Script or VB Script
	Message Start Events	Receive or reply events
Software	GUI tools	IBM Web Sphere or BPMN Modeler
People	Designers	Involved in design of processes
	Developers	Persons engaged in development of the application
Technique	Application Design	Exercise carried out in the design of application
	Integration	Exercise carried out for integrating the web application
	Task	Exercise carried out for developing the tasks of processes

Table 2. Initial items for the case study

4.1 Proposed Metrics

The proposed metrics are aimed to measure the different types of items listed in Table 2. For each category of items, there exists set of measuring metrics, that we classified into 4 different categories. They are: environment configuration and

re-usability, length and size, effort, and perplexing factors. Each category of items defined in Table 2 plays a vital role in the estimation process. The last categorical variables consisting the perplexing factors also play an influential role in the overall estimation process.

Before moving to the metrics some essential concepts, parameters and cost drivers are recalled in this section. Let us consider the item type "Process and Element Definition" defined in Table 2. In the context of service design phase, for the associated processes, all the composite links to the services using which the process interact are known as the *partnerLink* elements. These elements serve as a reference to the actual implementation, using which the processes interacts with external services. Moreover, the tasks of processes are implemented as a loosely coupled web service which *defines* the participant of web services, and the *properties* of the participant are linked to the partnerLink elements. Furthermore, the *partnerLink elements* are defined "how two individual service partner interact with each other and what each of the partner has to offer". As the partnerLink element is defined and included in each and every service involved in the process design phase, it is considered as an important parameter (cost driver) in context of SOA system development. Similarly, the XML Path Expression (XPath Expression) is used to check the data constraint of the service offered by the client. Generally, XPath queries are available to access the Domain Value Maps (DVMs) which are responsible for SOA orchestration. The SOA *orchestration* allows the work-flow definition between two different services. This is the reason to use X-Path queries as a critical parameter, as it facilitates the SOA system orchestration using the mapping process. Therefore, the *partnerLink elements* and *XPath queries* are included in the SOA system effort estimation as an essential cost driver.

Items	Metrics	Description
Process and Elements Definition	Re-used Process Count	Total number of re-used processes
	Re-used Task Count	Total number of re-used tasks
	Re-used Participant Count	Total number of re-used participants of web services corresponding to the tasks
	Re-used Space allotment Count	Total space (in Kilo Bytes) of the re-used application
	Re-used partnerLink Element	Total number of re-used invoked and client partnerLink elements
Integrated Application	Re-used Code Count	Total lines of code for all the re-used processes

Table 3. Environment configuration and re-usability metrics

Items	Metrics	Description
Process and Elements Definition	Total Process	Number of processes
	Total Abstract Process	Number of abstract processes
	Total partnerLink Elements	Number of invoked and client partnerLink elements
	Total Process Mapping	Number of variables and message definitions
	Total Parallel Flow	Number of links for a process
	Total Code Length	Total lines of code of a process
	Total Participants	Number of participants subjected to type of configuration
	Total Interfaces	Number of interfaces
	Total Scripts	Number of lines of the scripting languages and number of fault handlers
Tasks	Total Tasks	Number of tasks implemented as web services
	Total Operations	Number of designed operations
	Total Variables	Number of used input and output variables
SOA-Orchestration	Orchestration Count	Total number of X-path queries
Message Start Events	Total Receive Events Count	Number of receive start events
	Total Reply Events Count	Number of reply start events
	Total Confirmed and Submit Count	Total number of submit and confirmed events
Integrated Application	Total Code Count	Total lines of code for an application
	Comment Count*	Number of comment lines
	Space Count	Size of application (in Kilo Bytes)

* In our empirical study, the Comment Count metric is not used. This is because the dataset used does not match these requirements.

Table 4. Length and size metrics

Items	Metrics	Description
Process and Elements Definition	Process Effort	Estimated time for designing all the processes, interfaces and abstract processes of an application
Tasks	Task Effort	Estimated time for developing all the implemented tasks (partnerLink elements and parallel activities) of processes
SOA-Orchestration	Orchestration Count	Estimated time for building SOA Orchestration
Message Start Events	Event Effort	Estimated time for designing all types of events of an application
Integrated Application	Total Effort	(Process + Task + Orchestration + Event) Effort

Table 5. Effort metrics

Items	Metrics	Description
People	Skill	Design experience of subject on a scale of 0 to 5
Tool	Type*	Types of tool (GUIs) used in the design process of the application

* In our empirical study, the Type metric is not used. This is because the dataset used does not match these requirements.

Table 6. Perplexing factors

Similarly, the perplexing factors presented in Table 6 are the parameters conceived to have an effect on the estimated (dependent) variable, but were considered in the experimental (independent) variables unlike the confounding factors used in statistics [5]. Additionally, Tables 3 to 6 exemplifies metrics for different categories of items based on the environment configuration, size and effort related constraints. Furthermore, Table 3 presents some re-usable aspect of the packaged solution of SOA systems. The primary focus is on the interaction and dependency among the service groups like composite services which enables the middle-ware and platform technologies [6]. The re-usable metrics focuses only on the parameters that are being re-used[4]. Table 4 imitates the analyzed size and length related metrics. Table 5

[4] The dataset used in this empirical study does not constitute any re-used artifacts. Thus, we have not used the Environment Configuration and Re-usability Metrics, while generating the regression models.

presents different types of effort related metrics. The *total effort* of an application is calculated by *adding* all the calculated effort for the classified items.

4.2 Estimation Methodology

The main aim of any regression model is to analyze the relationship between different variables. This analysis is carried out with the implication of some general purpose regression models through the estimation of the relationship. These regression models are constructed with the help of some appropriate variables. The selection of variables from the proposed metrics is a preliminary activity for carrying out the further process. The general form of the statistic based regression model is defined in Equation (1).

$$y = f(x_1, x_2, \ldots, x_k) \tag{1}$$

where y is the dependent variable and x_1, x_2, ..., x_k are the independent variables.

The empirical and simulation results calculated using the regression models serve the following two purposes:

1. How is the predictor or dependent variables (y) affected with some changes in each of the response variables of x (x_1, x_2, ..., x_k), and
2. to predict the value of y using the values of x.

The data collected from multiple sources of an Indian software organization are used to generate statistic based regression models on the set of proposed metrics. The various techniques included in our proposed work are: simple linear, multiple linear, stepwise and ordinary least square regression models. We generate the estimation models for each category of *items* defined in Table 2. The estimated variables namely (Total Effort) is *computed and summed* with respect to each classified item.

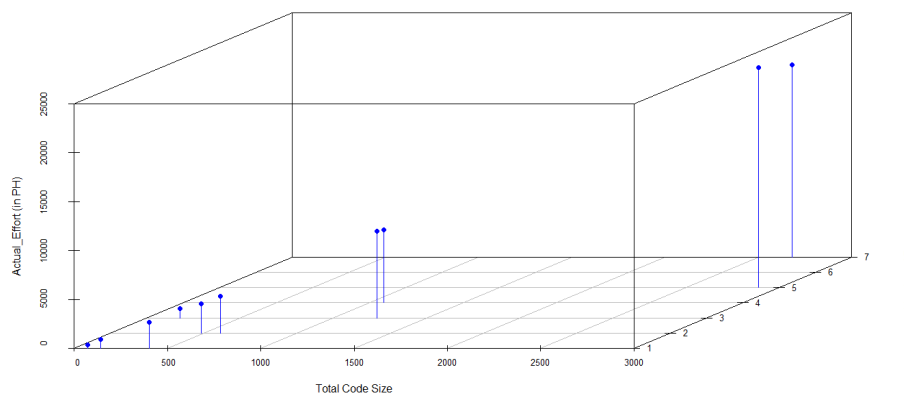

Figure 2. 3D scatter plot for the numerical variables of used dataset

Figure 2 shows the 3D scatter plot plotted for some numerical variables, that are used for generating the regression based estimation models. These numerical

variables are: the actual development effort (Actual_Effort (in PH)), the total size (Total Code Size) and the total number of processes (Number of Processes) implemented as services for an individual application. A *Scatter plot* is a mathematical diagram used to represent the displayed data as a collection of points using the value and position of used variables. A scatter plot also depicts a different kind of *correlation* that exists between certain variables with a confidence interval of the dataset. For a scatter plot, if the pattern of dots slopes from lower left corner to the upper right corner, it suggests that a *positive correlation* exists between the set of pair variables. A 3D scatter plot allows better visualization of multivariate data for multiple scalar variables and displays them on the different axes in space. Figure 2 is also useful for discovering the relationship between three variables simultaneously. The plot of Figure 2 suggests that the three variables are positively *correlated* and *associated*, since the variable (Actual_Effort (in PH)) increases linearly.

For generating regression models, we need some dependent and independent variables. The response (dependent) and predictor (independent) variables used for the generation of regression models are listed in Table 7. The variable (Total Effort) corresponding to the item "Integrated Application" is responsible for generating the prediction models. Further, it helps in computing the predicted results through summing all the calculated effort values for different classified items. (See Section 4 for more details).

5 RESULTS AND DISCUSSIONS

The predicted values have been calculated from the generated (simple linear, multiple linear, stepwise and ordinary least square regression) models using R 3.0.2 for Windows. Furthermore, this section discusses the following.

- Calculation of the predicted values using the proposed metrics and from generating 4 regression based estimation models.
- The obtained predicted values are further examined for investigating some statistical properties. These properties included (the linearity, normality and symmetry) and were tested on some commonly used statistical significance tests such as Shapiro-Wilk test, Kolmogorov-Smirnov test, Box-Cox transformation, correlation coefficient (r) and skewness distribution values.
- A comparative analysis of the generated regression models in terms of predictive accuracy is discussed and presented using some commonly used performance measurement indicators and box-plots evaluation.
- Some research threats to validity relevant to our work are also identified and discussed in this section.

For each generated regression model, a set of different plots are presented for assessing the statistical properties of the data variables of the used dataset. These different plots are constructed using some essential statistical artifacts like *residuals*, *fitted values*, *standardized residuals*, *theoretical quantiles*, *leverages*, *scale-location*,

Type	Items	Variables
Response (Dependent) Variables	Process and Elements Definition	Process Effort
	Tasks	Task Effort
	SOA-Orchestration	Orchestration Count
	Message Start Events	Event Effort
	Integrated Application	Total Effort
Predictor (Independent) Variables	Process and Elements Definition	Total Process
		Total Abstract Process
		Total partnerLink Elements
		Total Process Mapping
		Total Parallel Flow
		Total Code length
		Total Participants
		Total Interfaces
		Total Scripts
		Re-used Process Count*
		Re-used Task Count*
		Re-used Participant Count*
		Re-used Space allotment Count*
		Re-used partnerLink Element*
	Tasks	Total Tasks
		Total Operations
		Total Variables
	SOA-Orchestration	Orchestration Count
	Message Start Events	Total Receive Events Count
		Total Reply Events Count
		Total confirmed and submit Count
	Integrated Application	Re-used Code Count*
		Total Code Count
		Space Count

* Note: These predictor variables are useful only for the re-used artifacts.

Table 7. Selection of the variables

standard deviance residuals and *correlation*. Figure 3 flourishes box-plot considering some important numerical variables of the used dataset. The variables used in the box-plot are: the total number of processes, tasks, code size and the actual development effort *versus* the total number of projects (transformed into logarithmic scale) for the used dataset. The variables *Code_Size* and *Actual_Effort* correspond to the total length and effort for all the items and are listed in Table 2. Typically, box-plot is a graphical tool used to check the existence of outliers. Figure 3 depicts that there exist *no outliers* for the used data points of the used dataset.

Thus, there is no need of creating any *new variables* for the set of data variables, as it satisfies the property of *normality* and *linearity*. Additionally, some statistical

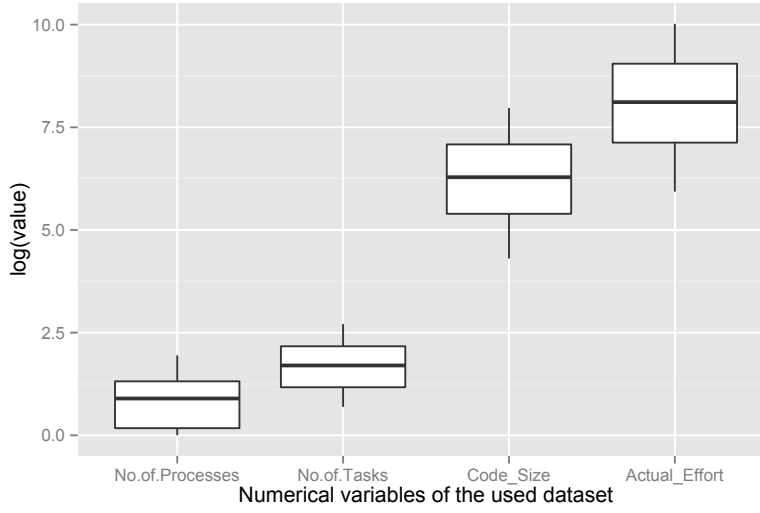

Figure 3. Box-plots for the numerical variables of the used dataset

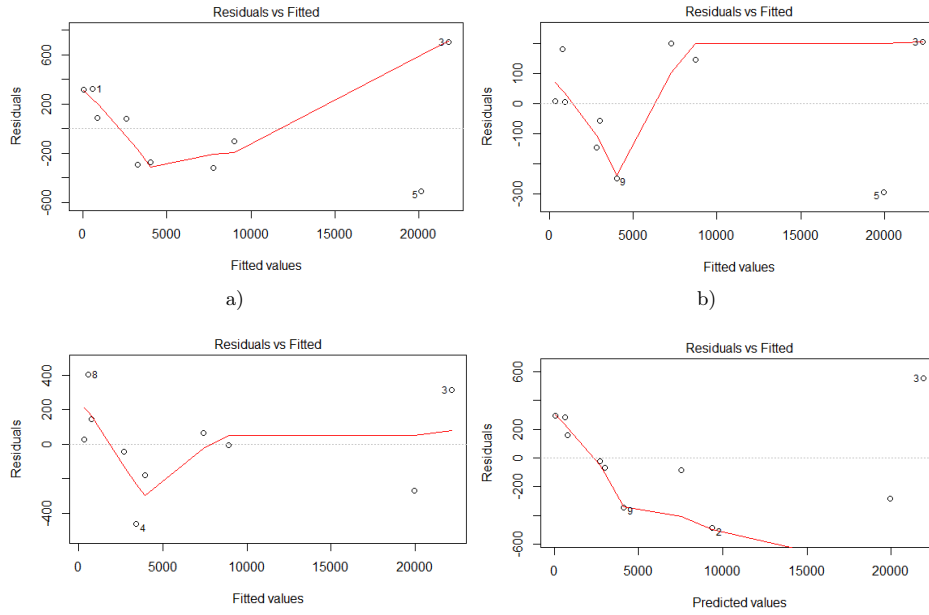

Figure 4. Residual Plot of different generated Regression Models. a) Simple Linear Regression. b) Multiple Linear Regression. c) Step-wise Regression. d) Ordinary Least Square Regression.

tests are performed for scrutinizing the linearity, normality and symmetry property of the used dataset. Figure 4 shows the graphical comparison between *residuals* (actual effort − estimated effort) in the Y-axis and *fitted values* (estimated effort) treated same as the predicted values in X-axis for the generated regression models. The different plots of Figures 4 a), b), c) and d) are known as Residual plot[5]. Figure 4 also indicates that the residuals and predicted values calculated from the generated regression models are not correlated and *equally spread*. Also, there exist no *non-linear* and *non-constant* variances for the used data points.

Additionally, Figure 5 reinforces the normal Quantile-Quantile (Q-Q plot)[6] comparing the randomly generated independent standard normal quantiles data (sample standardized residuals) on the vertical axis and the standard normal population (theoretical quantiles) on the horizontal axis for the generated regression models. Almost, all the points of Q-Q plot lie approximately on a straight line, but not necessarily on the line $y = x$. This is also marked as the condition of *linearity*, in-spite of some points do not lie on the line $y = x$. Moreover, the different plots of Figures 5 a), b), c) and d) are commonly used for scrutinizing the property of *skewness* and *normality*. The simple and multiple linear regression models generated using the proposed metrics for the used dataset yields an adjusted R^2 value as 0.997 and 0.999 respectively. Thus, it indicates 99 % of variation to the used dependent variables (proposed effort metrics). Moreover, none of the projects corresponding to the dataset denotes distance greater than the cook's distance $[3 * (4/10)]$.

After the implication of simple linear and multiple linear regression models, the stepwise regression model has been generated. We have generated the stepwise regression model using both forward and backward procedures as the mode of variable selection. The evaluation criterion for this model is characterized by Akaike Information Criterion (AIC)[7]. Allegedly, AIC provides an efficient mean of model selection because AIC deals with the association between *goodness of fit* and the *complexity* of model [33]. The output values induced by this criterion offer a relative estimate of data loss, when a regression model is used to represent the dependent variables of the used dataset.

Let us consider, a set of regression based candidate models having AIC values as: AIC_1, AIC_2, AIC_3, ..., AIC_n, respectively. The AIC value for the generated regression model is always chosen from the candidate models having the *minimum* AIC value. In this way, the AIC value provides the *relative estimate* of the data loss. Let AIC_{min} denote minimum AIC values and AIC_i depict other values for the set of candidate models. The expression is interpreted as the relative probability and

[5] The *Residual plot* is a graphical plot commonly used in statistics for showing the relationship between the fitted (estimated) values and residuals.

[6] Q-Q plot is basically a probability plot used for comparing two different probability distributions by plotting the quantiles with each other.

[7] The AIC measures the relative quality of the generated regression model for a given set of data values corresponding to the dataset.

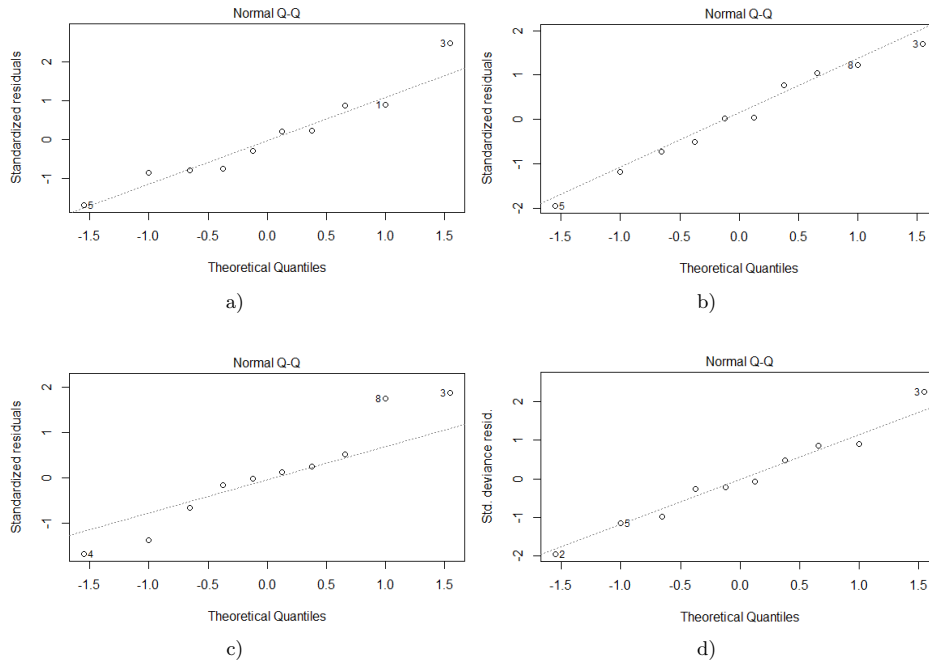

Figure 5. Normal Q-Q plot of different generated Regression Models. a) Simple Linear Regression. b) Multiple Linear Regression. c) Step-wise Regression. d) Ordinary Least Square Regression.

the i^{th} model minimizes the estimated data loss. Equation (2) specifies the relative likelihood of the i^{th} model.

$$e^{(AIC_{min}-AIC_i)/2}. \tag{2}$$

While generating the stepwise regression model, a set of four candidate models are generated having AIC values as $180.52, 121.39, 118.75$ and 118.27, respectively. The chosen AIC value for the stepwise regression model is 118.27. The candidate model having AIC value 118.27 omits all other generated candidate models for *minimizing* the overall data loss. The generated stepwise regression model using the proposed metrics for the used dataset induces adjusted R^2 value as 0.998. It indicates 99 % of variation to the used dependent variables. Again, none of the projects corresponding to the dataset denotes distances greater than the cook's distance for both forward and backward variable selection procedure modes.

Lastly, the Ordinary Least Square (OLS) regression technique have been generated using the proposed metrics. It is a linear approach to the multiple regression technique which results in *eliminating* the error terms. The OLS regression model is a linear approach but many times it works efficiently for the non-linear data. In

our work, this regression model have been generated considering the *family type* as "Gaussian". The density function of the Gaussian family is defined in Equation (3).

$$p(y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{(y-\mu)^2}{2\sigma^2}\right] \tag{3}$$

where μ is the mean, σ^2 is the variance and y is the response variable. The deduced AIC value for the OLS regression model is computed as 151.

Table 8 shows the values of Multiple R^2, Adjusted R^2 and AIC values, which are calculated using four regression models for all the items defined in Table 2. Typically, R^2 is a statistical value used for determining the goodness of fit of a regression model. R^2 is mostly used for determining the coefficient of the regression model. The AIC value helps to choose the minimum value from a set of candidate models generated for the stepwise and OLS regression techniques. Table 8 illustrates the coefficient values based upon the best criteria[8] for the used regression models. In the OLS regression model, the *dispersion parameter* of Gaussian family, using the density function is computed as 135 740.4.

Regression Models	Multiple R^2	Adjusted R^2	AIC Value
Simple Linear Regression	0.9978	0.9975	–
Multiple Linear Regression	0.9995	0.9999	–
Stepwise Regression	0.9989	0.9984	118.27
Ordinary Least Square (OLS) Regression	–	–	151

Table 8. Coefficient of determination for different generated regression models

5.1 Examining the Statistical Properties

In our work, an effort was made for examining the property of *normality*, *linearity* and *symmetry*. Although the property of *normality* and *linearity* could have been reviewed from the stability, normal Q-Q and residual plot. But for the sake of completeness, a few tests are required to investigate the statistical properties of the computed prediction results. Moreover, we have generated the regression based estimation models following the existing practices and rules [31].

The Shapiro-Wilk test has been introduced to the obtained absolute residual values using the simple and multiple linear regression models, for investigating the property of *normality* and *linearity*. Similarly, box-cox transformation have been employed in the stepwise regression model. In general, the most popular and widely used test for scrutinizing the property of *normality* is Shapiro-Wilk test. Some researchers also used Kolmogorov-Smirnov test as an alternative test for investigating the property of normality [35]. The Kolmogorov-Smirnov test is used to compare

[8] The criterion (AIC) is applicable only for those regression models, which can enable to generate multiple candidate models.

an observed cumulative distribution function (cdf) to an estimated cumulative distribution function. Moreover, the Kolmogorov-Smirnov test is an effective method for comparing the shape of two different cumulative distribution function samples for a small size dataset. For large real-time dataset, the calculated values comprised biases because the sample mean and standard deviation are used to estimate the *population* mean and standard deviation. Thus, Shapiro-Wilk test is presumed to be a *better* approach for testing the property of *normality* over Kolmogorov-Smirnov test. For the generated regression models, the probability-value (p-value) of the absolute residuals are calculated as follows: 0.16, 0.50, 0.33 and 0.92, respectively. Generally, lower the p-value, the *lesser* is the chance of *normality*. Furthermore, many statisticians used p-value 0.05 as the cut-off, the p-value lower than 0.05 depicts that the sample *deviates* from *normality*. For the generated regression models, the absolute residual values are *normally distributed* and satisfy the property of *normality*, as p-value is *greater* than the defined cutoff (> 0.05) value. The *absolute residuals* represent the difference between the actual effort and predicted effort values, and the variable actual effort is used as the dependent variable for generating the regression models. So, based upon this criterion (choosing absolute residual values), it is double checked that the used data and the predicted values obtained using the regression models are normally distributed [34].

We have also investigated the property of *symmetry* for the absolute residual values calculated for the generated regression models. The property of *symmetry* is validated from calculating the *skewness distribution* values. Thus, for the absolute residual values (x) corresponding to the generated regression models, the skewness distribution values are calculated as: 0.62, -0.22, 0.35 and 0.13, respectively. The computed skewness distribution values are skewed both towards right and left. This is because as a rule the *negative skewness* indicates that the mean of absolute residuals for different regression models is less than the median and data distribution is therefore *left-skewed*. Similarly, *positive skewness* indicates that the mean of absolute residuals is larger than the median and data distribution is *right-skewed*. So, for the multiple linear regression model, the data distribution values for the absolute residuals are left-skewed.

Some important guidelines for generating the regression models are "the residual values have not been correlated" and the "used independent variables should not be *linearly dependent* [31]". For validating the property of *linearity*, we have calculated the correlation coefficient (r) between the dependent variables (Actual_Effort (in PH)) and the independent variables (predicted effort values) for each of the generated regression models. If the value of "r" for the two variables is close to 1, then the variables are *linearly positively* related [34]. The calculated correlation coefficient (r) for the different generated regression models are computed as 0.998, 0.999, 0.999, 0.999, respectively. Since the calculated "r" value for all the generated regression models are close to 1. It is concluded that both actual and predicted effort values are *linearly positively* related.

Similarly, for the stepwise regression model the Box-Cox transformation is used for examining the statistical properties. The Box-Cox transformation is best suited

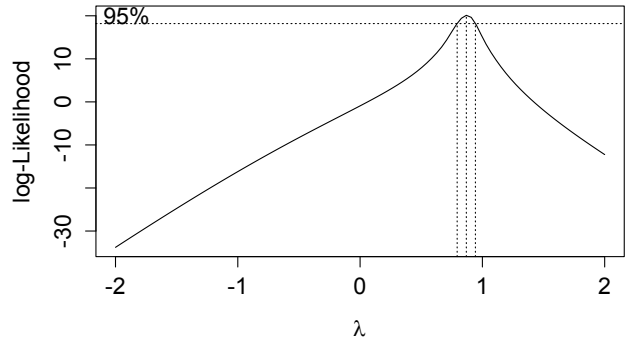

Figure 6. 2D Box-Cox transformation plot for the step-wise regression model

for examining the statistical properties of step-wise regression models and it computes and optimally plots a 2-Dimensional (2D) curve comprising log-likelihood value versus *lambda* (λ). Typically, lambda is the vector values for the chosen parameters. By default, the range of lambda varies within -2 to 2. Figure 6 shows 2D Box-Cox transformation plot for the generated step-wise regression model. In Figure 6, X-axis denotes the vector series (lambda) and the Y-axis represents the log-likelihood values of a particular variable or the parameter for the step-wise regression model. The Box-Cox linearity plot provides an efficient way for finding the suitable transformation mechanism without engaging in a lot of hits and trial fitted models [9]. After generating the step-wise regression model, the Box-Cox transformation have been employed for scrutinizing the statistical properties of the model. The value of lambda (λ) for the generated step-wise regression model is calculated as 0.86. This is an essential data transformation technique used to stabilize the variance and make the data normally distributed, for improving the validity of the associated measures. Table 9 presents the summary of the statistical significant results for the four generated regression based estimation models.

Property	Techniques	Simple Linear Regression	Multiple Linear Regression	Step-Wise Regression	Ordinary Least Square Regression
Normality	Shapiro-Wilk test (p-Value)	0.16	0.50	0.33	0.92
Linearity	Correlation Coefficient (r)	0.998	0.999	0.999	0.999
Symmetry	Skewness Value (s)	0.62	-0.22	0.35	0.13

Table 9. Results of statistical significance tests

5.2 Measuring the Results in Terms of Predictive Accuracy

Each and every used regression model has been iterated for 10 iteration and the average results are computed and presented for calculating the generalization error. Firstly, the used dataset is partitioned into two sets, i.e. the training and testing set. We have used this validation method because the training set of the used dataset have been partitioned randomly (a variant of k-fold cross validation). The training set is defined by $k-1$ samples, and the testing set is defined by k^{th} subset. The process is performed k times and for each iteration and it uses a different project of the used dataset as the testing set[9].

Moreover, we have evaluated the predictive power of the generated regression models using some commonly used performance measurement indicators like Mean Magnitude of Relative Error (MMRE) and Root Mean Square Error (RMSE) [24, 25, 26]. In our work, we have calculated the performance measurement indicator values in terms of *percentage*. This is because for any regression based empirical measurement, there is always a need for combining both the response and predictor variables, for measuring the accuracy. In general, the values of measurement indicators are not *exact*, thus calculating the percentage value allows comparing the predicted (estimated) values to an exact (actual) values. The *percentage value* for any measurement indicator like (MMRE, RMSE) gives the difference between the estimated and exact values in terms of the percentage of exact values. It is helpful in concluding how close the estimated values are with the actual values. The lower values of measurement indicators assure better prediction model. The used performance measurement indicators are described below.

1. **MMRE:** In the era of software effort estimation, MMRE is the most commonly used performance measurement indicators and is used in all types of estimation techniques [25, 26]. The basic metric of MMRE is the Magnitude of Relative Error (MRE) and is defined inside the braces of Equation (4). After calculating MRE, MMRE is obtained from the mean value of MRE.

$$MMRE = \frac{\sum_{i=1}^{n} \left(\frac{|Eact_i - Eest_i|}{Eact_i} \right)}{n} \times 100 \tag{4}$$

   where:

   - $Eest_i$: is the total estimated effort for i number of projects of a dataset.
   - $Eact_i$: is the actual effort for i number of projects of a dataset.
   - n: is the total number of applications or projects of a dataset.

2. **RMSE:** RMSE measures the difference between the estimated value ($Eest_i$) and the actual value ($Eact_i$), for i number of projects of the dataset. In RMSE, the

[9] For conducting the experiment, the value of k is 10, same as the size of the used dataset.

basic metric for computing the error is Mean Square Error (MSE). Taking the square root of MSE yields root mean square error having the same units as the quantity estimated for an unbiased estimator [32]. The RMSE metric is defined in Equation (5).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Eact_i - Eest_i)^2} \times 100. \tag{5}$$

Generated Models	MMRE (in %)	RMSE (in %)
Simple Linear Regression	15.07	3.54
Multiple Linear Regression	**10.80**	**1.75**
Stepwise Regression	14.10	2.47
Ordinary Least Square (OLS) Regression	19.30	3.08

Table 10. Comparison of the generated regression models in terms of predictive accuracy

Table 10 illustrates the comparative results of the four generated regression models in terms of predictive accuracy. The major observation from Table 10 is that the technique of *multiple linear regression* is treated as the *best* estimation model, as it induces lower MMRE and RMSE percentage values, compared to other generated models.

5.3 Comparison of Results Using Boxplots

For any statistical techniques, it is important to investigate the values of *absolute residuals*. An accurate measure is relatively dependent on how much the values of residuals are. Lower values of absolute residuals denote the predicted and actual effort to be similar. Figure 7 shows the box-plot evaluation of the four generated regression models for the values of absolute residual. The box-plot evaluation verifies the results obtained from the performance measurement indicators.

The computed absolute residuals for the generated regression models are shown in the vertical side of each box-plot in Figure 7. Moreover, Figure 7 suggests that the generated *multiple linear regression* model is having the *minimum* values, of absolute residuals when compared against other generated regression models. Intuitively, the box-plots signifies the spread distribution much wider when the absolute residuals are compared with each other for the different generated regression models.

Thus, from the box-plot evaluation and from the implication of performance measurement indicators, it is double-checked that the predicted values obtained using the *multiple linear regression* model furnishes best results in terms of inducing lower residual values for the used dataset. Furthermore, the results (effort values) computed using the multiple linear regression model outperforms all other employed regression models in terms of predictive accuracy.

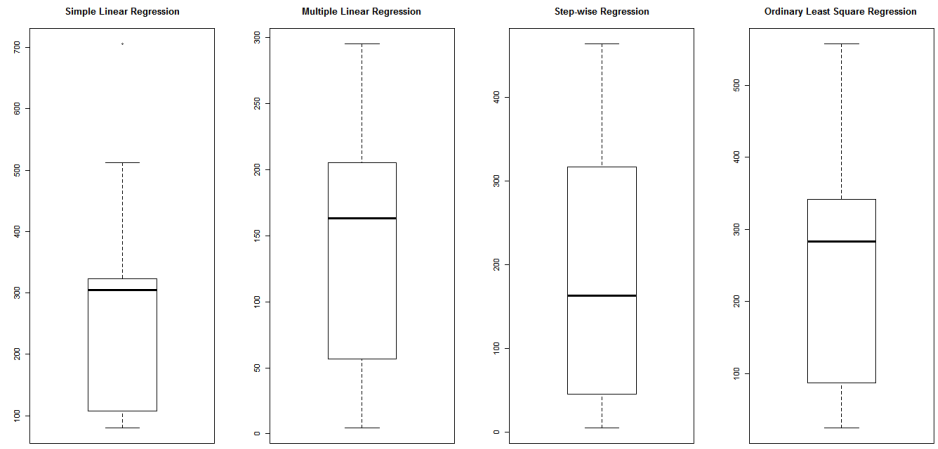

Figure 7. Box-plots of the absolute residuals for different generated regression models

5.4 Threats to the Validity

When conducting an experiment or empirical study, there are always threats to the validity of results. This subsection discusses the validity threats associated with our empirical results on the basis of list of threats by Cook and Campbell [56]. This is conceived as an effective step for concluding the results procured using the statistical methods.

The major factor that may act as an *internal threat* to our simulated results is the ability to draw conclusions about the connections between the chosen independent and dependent variables in the model generation process [56]. This part might also subject to errors and bias. To reduce this threat, manual cross verification of the obtained results was undertaken between two researchers.

Threats to *external validity* associated with the calculated results may be the *size and structure* of the used dataset. However, we conceive that this does not affect the validity of the results, since statistical significant results have been calculated and obtained. Moreover, in the literature of traditional software and web development effort estimation, the prediction results have been calculated using smaller size dataset of 12 and 15 projects respectively [27, 36, 37]. Furthermore, in regard to the external validity, the used dataset overcomes this threat, as the dataset used in our experimental study has been collected from *multiple sources* Indian software organization. On the other hand, we believe that for enhancing more accurate validation, it is essential to collect data from the multiple industrial organization.

Threats to conclusion validity refers to the degree of which the conclusions reached and their relationships using our data are reasonable [56]. To address this threat, the obtained results are examined using some commonly used statistical sig-

nificance tests. Moreover, the violated assumptions of statistical tests were reduced from the advent of some important test measures for the data variables.

6 CONCLUDING REMARKS

The main aim of this paper was to accord an efficient technique for estimating the SOA systems development effort along with a proper validation. For this, we presented a methodology based on analyzing some initial items associated with the service development life cycle. The measuring metrics are proposed for these identified items. To the best of our knowledge, this was the first time that someone tried to create mapping rules between the service design phase and regression models for generating effort estimation models for SOA systems with their support. More explicitly, none of the previous works used statistics based approach to solve the aforesaid problem along with proper validation for some past project data.

We believe that this approach would definitely add an ease for the readers, analysts and project managers practicing SOA system effort estimation. Our approach of estimating the development effort, builds from the generation of four regression models using the proposed metrics listed in Tables 3, 4, 5, 6. These metrics are proposed based on the different classified parameters like environment configuration, length, size, reused services, effort and perplexing factors for a set of initial items defined in Table 3. Considering the *interest of practitioners,* our proposed technique serves as helpful in dealing with many new complex challenges that project managers encounter with the large size business process SOA systems. In this regard, our proposed metrics goes well beyond the typical capabilities offered by the traditional software estimation techniques.

The use of SOA in developing business process solutions provides better customer services through increased transparency and better consolidation of data and functionality. Some important *statistical properties* have been scrutinized for enhancing the accuracy of the calculated predicted results. The predictive accuracy of the generated regression models has also been demonstrated for some past industrial data using some commonly used performance measurement indicators and boxplots evaluation. The predicted values computed using the multiple linear regression model outperforms every other generated (simple linear, stepwise and ordinary least squares) regression models.

In addition, there is a persuasive need of an efficient effort estimation technique for SOA systems, as the implication of some new features increases the overall complexity of the system. Thus, having an efficient effort estimation technique could contribute in reduction of cost and time implied for developing future SOA systems. As a future work, there is some interesting challenge to perform a replicated study by judging the use of micro services in SOA systems. It will also intrigue to analyze the use of analogy and machine learning approaches in SOA system effort estimation.

REFERENCES

[1] Mukhi, N. K.—Konuru, R.—Curbera, F.: Cooperative Middleware Specialization for Service-Oriented Architectures. 13th International World Wide Web Conference on Alternate Track Papers and Posters, ACM, New York, USA, 2004, pp. 206–215, doi: 10.1145/1013367.1013401.

[2] Erl, T.: SOA: Principles of Service Design. Prentice Hall, Upper Saddle River, NJ, USA, 2008.

[3] Josuttis, N. M.: SOA in Practice. O'Reilly Media, Inc., Sebastopol, CA, USA, 2007.

[4] Vasiliev, Y.: SOA and WS-BPEL. Packt Publishing Ltd., Birmingham, 2007.

[5] Montgomery, D. C.: Design and Analysis of Experiments. Wiley, New York, 1984.

[6] Bell, M.: Service-Oriented Modeling : Service Analysis, Design, and Architecture, John Wiley and Sons, Inc., Hoboken, New Jersey, 2008.

[7] Lowe, D.—Hall, W.: Hypertext and the Web – An Engineering Approach. John Wiley and Sons, New York, 1998.

[8] Mendes, E.—Mosley, N.—Counsell, S.: Web Metrics – Estimating Design and Authoring Effort. IEEE MultiMedia, Vol. 8, 2001, No. 1, pp. 50–57, doi: 10.1109/93.923953.

[9] Kutner, M. H.—Nachtsheim, C.—Neter, J.—Li, W.: Applied Linear Statistical Models. McGraw-Hill/Irwin, Blacklick, Ohio, USA, 2005.

[10] Li, Z.—O'Brien, L.: A Qualitative Approach to Effort Judgment for Web Service Composition Based SOA Implementations. 25th International Conference on Advanced Information Networking and Applications (AINA), IEEE, Biopolis, Singapore, 2011, pp. 586–593.

[11] Li, Z.—Keung, J.: Software Cost Estimation Framework for Service-Oriented Architecture Systems Using Divide-and-Conquer Approach. Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), IEEE Press, Nanjing, China, 2010, pp. 47–54, doi: 10.1109/SOSE.2010.29.

[12] O'Brien, L.: A Framework for Scope, Cost and Effort Estimation for Service-Oriented Architecture (SOA) Projects. Australian Software Engineering Conference (ASWEC'09), IEEE, Gold Coast, Queensland, Australia, 2009, pp. 101–110, doi: 10.1109/ASWEC.2009.35.

[13] Liu, J.—Qiao, J.—Lin, S.—Li, Q.: A Bayesian Net Based Effort Estimation Model for Service Governance Processes. Second International Conference on Information and Computing Science, Manchester, England, UK, IEEE, 2009, pp. 83–86, doi: 10.1109/ICIC.2009.129.

[14] Gomes, Y. M. P.: Functional Size, Effort and Cost of SOA Projects with Function Points. Service Technology Magazine, Issue LXVIII, 2012.

[15] Jorgensen, M.—Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies. IEEE Transactions on Software Engineering, Vol. 33, 2007, No. 1, pp. 33–53, doi: 10.1109/TSE.2007.256943.

[16] Boehm, B. W.: Software Engineering Economics. IEEE Transactions on Software Engineering, Vol. SE-10, 1984, No. 1, pp. 4–21.

[17] Heemstra, F. J.: Software Cost Estimation. Information and Software Technology, Vol. 34, 1992, No. 10, pp. 627–639, doi: 10.1016/0950-5849(92)90068-Z.

[18] Hughes, R. T.: Expert Judgement as an Estimating Method. Information and Software Technology, Vol. 38, 1996, No. 2, pp. 67–75, doi: 10.1016/0950-5849(95)01045-9.

[19] Shepperd, M.—Schofield, C.—Kitchenham, B.: Effort Estimation Using Analogy. 18th International Conference on Software Engineering, IEEE, Berlin, Germany, 1996, pp. 170–178, doi: 10.1109/ICSE.1996.493413.

[20] Shepperd, M.—Schofield, C.: Estimating Software Project Effort Using Analogies. IEEE Transactions on Software Engineering, Vol. 23, 1997, No. 11, pp. 736–743, doi: 10.1109/32.637387.

[21] Li, J.—Ruhe, G.—Al-Emran, A.—Richter, M. M.: A Flexible Method for Software Effort Estimation by Analogy. Empirical Software Engineering, Vol. 12, 2007, No. 1, pp. 65–106.

[22] Nagpal, G.—Uddin, M.—Kaur, A.: Analyzing Software Effort Estimation Using K Means Clustered Regression Approach. ACM SIGSOFT Software Engineering Notes, Vol. 38, 2013, No. 1, pp. 1–9.

[23] Tansey, B.—Stroulia, E.: Valuating Software Service Development: Integrating COCOMO II and Real Options Theory. First International Workshop on the Economics of Software and Computation (ESC'07), IEEE, Minneapolis, MN, 2007, pp. 8–8.

[24] Conte, S. D.—Dunsmore, H. E.—Shen, V. Y.: Software Engineering Metrics and Models. Benjamin-Cummings Publishing Co. Inc., Redwood City, CA, USA, 1986.

[25] Kitchenham, B. A.—Pickard, L. M.—MacDonell, S. G.—Shepperd, M. J.: What Accuracy Statistics Really Measure: Software Estimation. IEE Proceedings – Software, Vol. 148, 2001, No. 3, pp. 81–85.

[26] Lo, B.—Gao, X.: Assessing Software Cost Estimation Models: Criteria for Accuracy, Consistency and Regression. Australasian Journal of Information Systems, Vol. 5, 1997, No. 1, pp. 30–44.

[27] Di Martino, S.—Ferrucci, F.—Gravino, C.—Mendes, E.: Comparing Size Measures for Predicting Web Application Development Effort: A Case Study. First International Symposium on Empirical Software Engineering and Measurement (ESEM'07), IEEE, Madrid, Spain, 2007, pp. 324–333, doi: 10.1109/ESEM.2007.20.

[28] Ferrucci, F.—Gravino, C.—Di Martino, S.: A Case Study Using Web Objects and COSMIC for Effort Estimation of Web Applications. 34th Euromicro Conference Software Engineering and Advanced Applications (SEAA'08), IEEE, Parma, 2008, pp. 441–448, doi: 10.1109/SEAA.2008.60.

[29] Mendes, E.: Cost Estimation Techniques for Web Projects. IGI Global, Hershey, PA, USA, 2007.

[30] Mendes, E.—Mosley, N.—Counsell, S.: Investigating Web Size Metrics for Early Web Cost Estimation. Journal of Systems and Software, Vol. 77, 2005, No. 2, pp. 157–172, doi: 10.1016/j.jss.2004.08.034.

[31] Maxwell, K. D.: Applied Statistics for Software Managers. Prentice-Hall, Software Quality Institute Series, Harlow, United Kingdom, 2005.

[32] Wackerly, D. D.—Mendenhall III, W.—Scheaffer, R. L.: Mathematical Statistics with Applications. Brooks/Cole CENGAGE Learning, Seventh Edition, 2007.

[33] Burnham, K. P.—Anderson, D. R.: Multimodel Inference Understanding AIC and BIC in Model Selection. Sociological Methods and Research, Vol. 33, 2004, No. 2, pp. 261–304, doi: 10.1177/0049124104268644.

[34] Gentle, J. E.: Computational Statistics. Springer, New York, 2009, doi: 10.1007/978-0-387-98144-4.

[35] Abrahão, S.—Gómez, J.—Insfran, E.: Validating a Size Measure for Effort Estimation in Model-Driven Web Development. Information Sciences, Vol. 180, 2010, No. 20, pp. 3932–3954.

[36] Di Martino, S.—Ferrucci, F.—Gravino, C.—Sarro, F.: Using Web Objects for Development Effort Estimation of Web Applications: A Replicated Study. In: Caivano, D., Oivo, M., Baldassarre, M. T., Visaggio, G. (Eds.): Product-Focused Software Process Improvement (PROFES 2011). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6759, 2011, pp. 186–201.

[37] Ruhe, M.—Jeffery, R.—Wieczorek, I.: Using Web Objects for Estimating Software Development Effort for Web Applications. Proceedings of Ninth International Software Metrics Symposium, IEEE, Sydney, NSW, Australia 2003, pp. 30–37, doi: 10.1109/METRIC.2003.1232453.

[38] Lund, K.—Eggen, A.—Hadzic, D.—Hafsoe, T.—Johnsen, F. T.: Using Web Services to Realize Service Oriented Architecture in Military Communication Networks. IEEE Communications Magazine, Vol. 45, 2007, No. 10, pp. 47–53.

[39] Omar, W. M.—Taleb-Bendiab, A.: E-Health Support Services based on Service-Oriented Architecture. IT Professional, Vol. 8, 2006, No. 2, pp. 35–41.

[40] Barker, A.—Weissman, J. B.—Van Hemert, J. I.: Reducing Data Transfer in Service-Oriented Architectures: The Circulate Approach. IEEE Transactions on Services Computing, Vol. 5, 2012, No. 3, pp. 437–449, doi: 10.1109/TSC.2011.23.

[41] Girbea, A.—Suciu, C.—Nechifor, S.—Sisak, F.: Design and Implementation of a Service-Oriented Architecture for the Optimization of Industrial Applications. IEEE Transactions on Industrial Informatics, Vol. 10, 2014, No. 1, pp. 185–196, doi: 10.1109/TII.2013.2253112.

[42] Kim, T.-W.—Kim, H.-C.: Service-Oriented Architecture Structure for Healthcare Systems Utilising Vital Signs. IET Communications, Vol. 6, 2012, No. 18, pp. 3238–3247.

[43] Dietrich, A. J.—Kirn, S.—Sugumaran, V.: A Service-Oriented Architecture for Mass Customization – A Shoe Industry Case Study. IEEE Transactions on Engineering Management, Vol. 54, 2007, No. 1, pp. 190–204, doi: 10.1109/TEM.2006.889076.

[44] Melament, A.—Peres, Y.—Vitkin, E.—Kostirev, I.—Shmueli, N.—Sangiorgi, L.—Mordenti, M.—D'Ascia, S.: BioMIMS – SOA Platform for Research of Rare Hereditary Diseases. Annual SRII Global Conference, IEEE, San Jose, CA, 2011, pp. 83–90, doi: 10.1109/SRII.2011.19.

[45] Xu, L.—Chen, L.—Chen, T.—Gao, Y.: SOA-Based Precision Irrigation Decision Support System. Mathematical and Computer Modelling, Vol. 54, 2011, No. 3-4, pp. 944–949.

[46] Cucinotta, T.—Mancina, A.—Anastasi, G. F.—Lipari, G.—Mangeruca, L.—Checcozzo, R.—Rusina, F.: A Real-Time Service-Oriented Architecture for Industrial Automation. IEEE Transactions on Industrial Informatics, Vol. 5, 2009, No. 3, pp. 267–277, doi: 10.1109/TII.2009.2027013.

[47] Verlaine, B.—Jureta, I. J.—Faulkner, S.: A Requirements-Based Model for Effort Estimation in Service-Oriented Systems. In: Lomuscio, A. R., Nepal, S., Patrizi, F., Benatallah, B., Brandić, I. (Eds.): Service-Oriented Computing – ICSOC 2013 Workshops. Springer International Publishing, Lecture Notes in Computer Science, Vol. 8377, 2014, pp. 82–94.

[48] Boehm, B. W.—Abts, C.—Chulani, S.: Software Development Cost Estimation Approaches – A Survey. Annals of Software Engineering, Vol. 10, 2000, No. 1-4, pp. 177–205.

[49] Demirkan, H.—Kauffman, R. J.—Vayghan, J. A.—Fill, H.-G.—Karagiannis, D.—Maglio, P. P.: Service-Oriented Technology and Management: Perspectives on Research and Practice for the Coming Decade. Electronic Commerce Research and Applications, Vol. 7, 2008, No. 4, pp. 356–376, doi: 10.1016/j.elerap.2008.07.002.

[50] Papazoglou, M. P.—Traverso, P.—Dustdar, S.—Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenges. Computer, Vol. 40, 2007, No. 11, pp. 38–45, doi: 10.1109/MC.2007.400.

[51] Brzoza-Woch, R.—Czekierda, Ł.—Długopolski, J.—Nawrocki, P.—Psiuk, M.—Szydło, T.—Zaborowski, W.—Zieliński, K.—Żmuda, D.: Implementation, Deployment and Governance of SOA Adaptive Systems. In: Ambroszkiewicz, S., Brzeziński, J., Cellary, W., Grzech, A., Zieliński, K. (Eds.): Advanced SOA Tools and Applications. Springer, Berlin, Heidelberg, Studies in Computational Intelligence, Vol. 499, 2014, pp. 261–323.

[52] Papazoglou, M. P.—Traverso, P.—Dustdar, S.—Leymann, F.: Service-Oriented Computing: A Research Roadmap. International Journal of Cooperative Information Systems, Vol. 17, 2008, No. 2, pp. 223–255, doi: 10.1142/S0218843008001816.

[53] Joachim, N.—Beimborn, D.—Weitzel, T.: The Influence of SOA Governance Mechanisms on IT Flexibility and Service Reuse. The Journal of Strategic Information Systems, Vol. 22, 2013, No. 1, pp. 86–101, doi: 10.1016/j.jsis.2012.10.003.

[54] Fiadeiro, J.—Lopes, A.—Abreu, J.: A Formal Model for Service-Oriented Interactions. Science of Computer Programming, Vol. 77, 2012, No. 5, pp. 577–608, doi: 10.1016/j.scico.2011.12.003.

[55] Vassiliadis, B.—Stefani, A.—Tsaknakis, J.—Tsakalidis, A.: From Application Service Provision to Service-Oriented Computing: A Study of the IT Outsourcing Evolution. Telematics and Informatics, Vol. 23, 2006, No. 4, pp. 271–293, doi: 10.1016/j.tele.2005.09.001.

[56] Cook, T. D.—Campbell, D. T.: Quasi-Experimentation: Design and Analysis Issues for Field Settings. Houghton Mifflin, Boston, Massachusetts, United States, 1979.

Siba Mishra received his Bachelor of Technology (B. Tech.) degree in computer science and engineering from Biju Patnaik University of Technology, Rourkela, India in 2009 and the Master of Technology (M. Tech.) degree in computer science and engineering from Kiit University, Bhubaneswar, India in 2012. He is currently working towards his Ph.D. degree in computer science and engineering at the Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand, India. His main research interests include software effort estimation, service-oriented architecture, aspect-oriented programming and program slicing. He is a student member of the IEEE and ACM.

Chiranjeev Kumar is working as Full Professor at the Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand, India. He received his Ph.D. degree in computer science and engineering from Allahabad University, India in 2006. He was the gold medalist of his M. Eng. batch at the Department of Computer Science and Engineering, Motilal Nehru National Institute of Technology (MNNIT), Allahabad, Uttar Pradesh, India in 2001. In 1998, he was felicitated upon certified novell administrator (CNA) and certified novell engineer (CNE). In his about 18 years of teaching and research carrier, he has contributed for several research papers in leading refereed journals and conference proceedings of the national and international repute. His main research interests include mobility management in wireless networks, ad hoc networks and software engineering. He is an IEEE member since 2006, and a fellow of the Inventive Research Organization (IRO) since 2016.