

Webapp Effort Estimation using Cosmic Method

Ahmed Thair Ahmed
Software Engineering Department
College of Computer science and Mathematics
University of Mosul

Dujan B. Taha
Software Engineering Department
College of Computer science and Mathematics
University of Mosul

ABSTRACT

Effort estimation is the important task in project planning phase in software industry. The process is complex as the complexity and size of software to be built. In the conventional software, there are many classical techniques have been used for effort estimation purpose. These techniques are not suitable to be applied on Webapp development because of the difficulty of sizing software in Webapp and the huge amount of data. In this work we proposed an algorithm for estimating the effort of Webapp software using COSMIC function point combined with the conceptual model of the software which is data based driven from Webapp. The method focuses on the extensive of data movement in the software which may be elicited from the conceptual model. For this purpose 19 Webapps with their conceptual models are used for effort estimation. Effort estimation model was built using cubic regression model and accuracy indicators such as MRE, MMRE, MdMRE, and Pred(0.25) were used to evaluate the efficiency of the proposed method.

Keywords

Webapp, Effort estimation, COSMIC method, COSMIC function point(CFP), functional process

1. INTRODUCTION

Many organizations now have a website for providing business needs and information about themselves to the customers. Presently, there is a growing request for more sophisticated Webapps. To meet customer requirements, there are many challenges WebApp developers face in estimating the effort, time, and cost of projects. In Webapp development, estimating effort is one of the critical challenges because there is no specific approach to that end. In conventional software engineering, measurements of software size are well inspected and there are a number of methodologies that may be used for sizing and estimating effort in the conventional software development. Nevertheless, it differs with the Webapp because the Webapp development process differs from the Conventional software development process. The methods of estimating effort of conventional software are not convenient to grasp the features of software development which affect the Webapps size and effort. [4]

There are two techniques for sizing Webapps: lines of code technique (LOC) and functional size technique (FS). LOC relies on the programming language and it can be applied when the project is finished. Otherwise, functional size estimation is applicable during early stages of the development. Functional size measurement denotes to measure the size of software depending on functional user requirements [3]. The combination of conceptual models and FS technique provides a good tool for estimating effort during early phases of the development process. The conceptual model of Webapp is used to support the designing and implementing Webapps and to provide guidelines for verification of functional requirements. [5]

The Common Software Measurement International Consortium COSMIC method has a powerful capability for WebApp size estimation. It is stable and understandable. It has great acceptance by the estimation process staff as it is easy to implement and easy to deal with the documentations of modern software and its appropriate with modern software architectures. There is the ability for making improvements in the accuracy of estimation, essentially for the large projects. And there is ability to calculate software size based on functional requirements created by CASE tools. [2] [15]

The development of Webapp based on conceptual models has become widespread as the conceptual models can be tested and corrected before implementing applications. This enables developers to analyze and design projects in detail before writing the program source codes. [5]

Dynamic Webapp includes large quantity of data movements (among user, system and storage) , therefore , COSMIC method is suitable for measuring data movement of these applications [10], For applying measurement procedure , COSMIC method can be adaptive in accordance with UML-based Web Engineering (UWE) approach for constructing diagrams of conceptual model. MagicUWE tool supports UWE approach that plug-in implemented in Magic Draw tool. UWE was the first approach that created for modeling Webapp; UWE uses UML profiles, extended with the stereotypes constructing the conceptual models of Webapps.[8]

The proposed algorithm presented in this work is based on model-driven methodology for modeling application using UWE approach and COSMIC method for calculating data movements in Webapp basing on UWE model [7].

2. THE COSMIC METHOD

The COSMIC method was designed to measure the Functional User Requirements (FURs) of software. The method denotes that FURs are consisting of a set of functional processes. The user run the process or may be triggered by an actor (external component or user) which can be exist outside of the software boundaries being measured. This process specifies a set of data movements that can contain one or more attributes belonging to that group. The process will be complete when the software completed all requirements to response to the event. There are four types of data movements in COSMIC function method as below:

- *Entry* – the data moves from users to the process
- *Exit* – the data moves from the process to users
- *Write* – the data moves from the process to the storage.
- *Read* – the data moves from the storage to the process.

Each of these of movement types represents a COSMIC function point CFP. Once completion of all functional processes the sum of CFPs will be measured to obtain the total size of the software [14]. Figure 1 demonstrates the movement

of data that can occur in the COSMIC method; the flow of data among users, processes and storage

Functional size of software can be identified through calculating CFP value for software. For a process i , the size is calculated by summation total data movements that can be occur in that process as show in the following equation:

$$\text{size(CFP}_i) = \sum \text{size(Entries)} + \sum \text{size(Exits)} + \sum \text{size(Reads)} + \sum \text{size(Writes)} \dots \dots \dots (1)$$

The size of software based on CFP method is the sum of processes sizes of the software as below [5]:

$$\text{Size (Software)} = \sum \text{size (CFPs)} \dots \dots \dots (2)$$

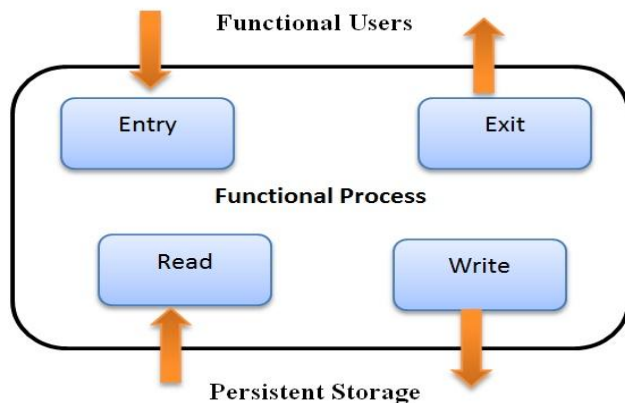


Figure 1 Types of data movement in the COSMIC method

3. RELATED WORK

Recently, many papers investigated Webapp effort estimation using various methods in the Preliminary phase of Webapp development. Costagliola et al. [4] used COSMIC function method for estimating effort of Webapp. They applied COSMIC function point method on analysis and design documentation. Di Martino et al. [11] used the modified COSMIC method for Webapps to calculate the size and the Ordinary Least Square Regression (OLSR) for predication model. They suggested that the COSMIC-FFP method is good for effort estimation of Webapp. Sworooop [13] et al. used Webapp components as measure unit for sizing Webapp and effort estimation of Webapp. Denis Ceke et al. [5] used the conceptual models (process structure, use case, content, and navigation) with COSMIC method for estimating effort. They

used simple linear regression analysis for predication model construction. Sergio Di Martinoa et al. [12] investigated that if COSMIC method can be more efficient than Functional Point Analysis for Webapp effort estimation. They concluded that the COSMIC method is more efficient than Functional Point Analysis and they suggested that more additional researches required for the purpose of smooth moving from Functional Point Analysis to COSMIC.

4. THE PROPOSED METHOD

In the proposed method, use case model (use case diagram), content model (class diagram), and process model (activity diagram) have been used for requirements analysis with special stereotypes for Webapps. COSMIC method has applied on activity diagram in order to counting the amount of data movement which represents COSMIC function point (CFP). Then, cubic regression has been used for construction of effort estimation equation using COSMIC function point and the actual effort of the projects in the dataset.

FURs are defined using use case model that can be created in UWE using use case diagram. FURs represent the functional requirements of user which COSMIC method depends on it. Although can be neglected if the Webapp is considerably simple.

Then, content model is created, which is consist of class diagram containing set of classes. Content model identifies the data groups that will be used in the Webapp. Each class in the class diagram includes information about one group of data in the Webapp. Attributes in each class have been used for data storing in database or for data either that triggered by users or that may be used for functional processes. Next, process model is created for each process class in the class diagram. Process model have been created as the activity diagram in UML but using UWE approach that can have some extensions. These extensions support in identifying of user process, system elements and persistent storage elements.

After the creation of conceptual model of the Webapp, COSMIC method has been applied to count the amount of data movements for all functional processes that have been determined using activity diagram in order to obtain the Webapp size represented as COSMIC function point (CFP). Figure 2 demonstrates the general steps of the proposed method.

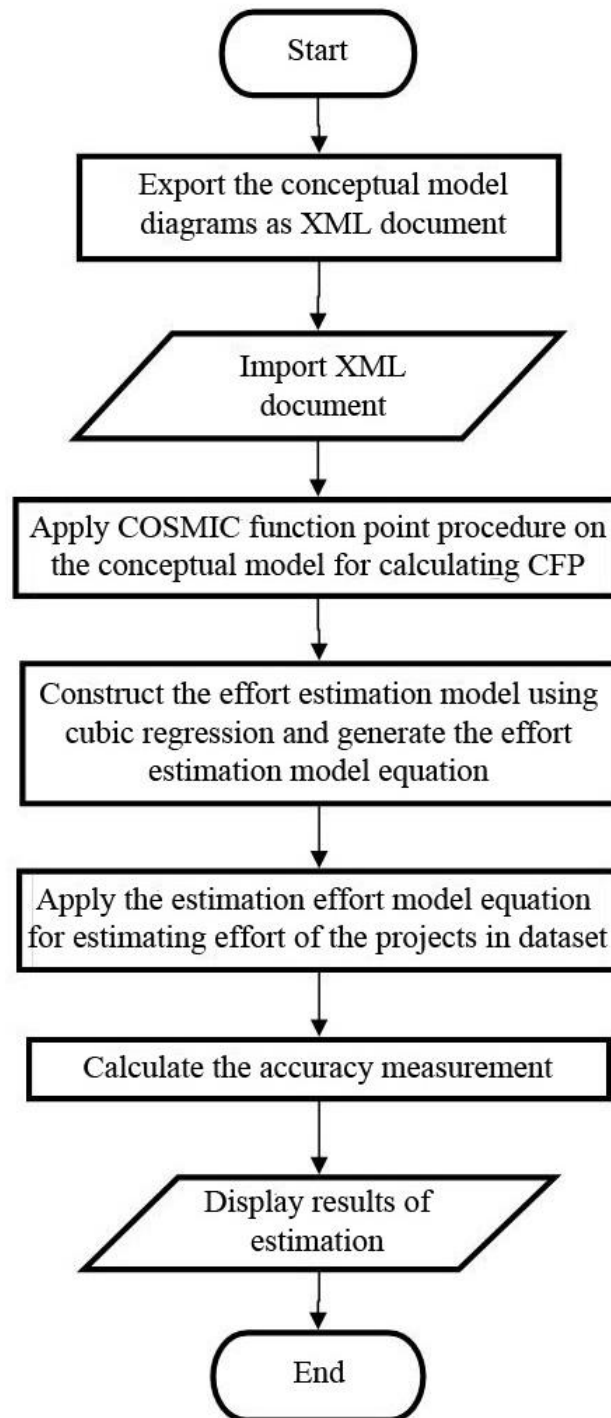


Figure 2 the proposed method steps

UWE diagrams have been exported as an XML document to be used as input to the method. The COSMIC size for Webapp measures the functional processes that were presented using activity diagram which describes all

functional processes. Implementing the COSMIC method process to all parts of the software will result the functional size of FURs in term of COSMIC function points (CFP). Figure 3 illustrates the steps for calculating CFP.

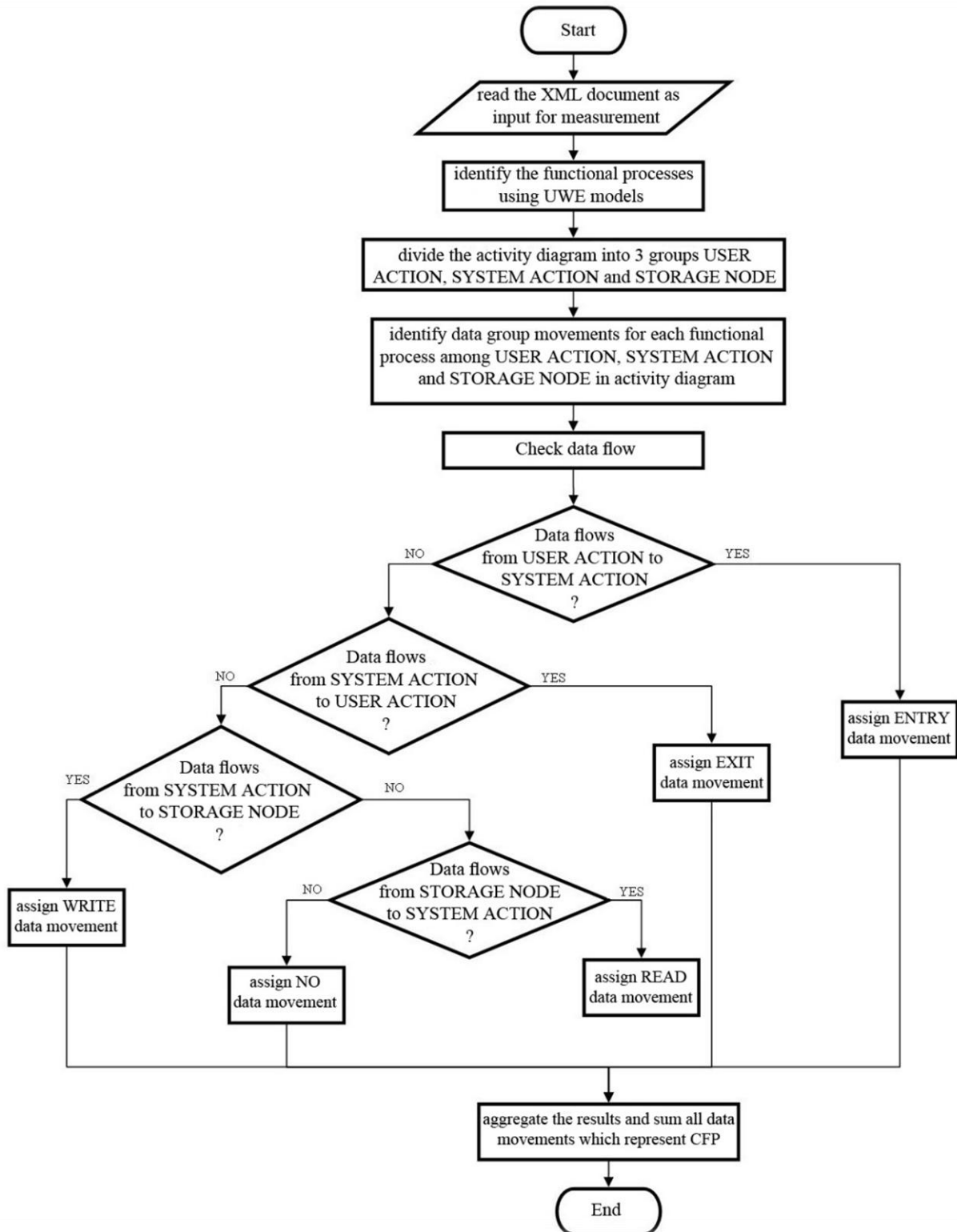


Figure 3 the steps for counting CFP

FURs have been determined using use case diagram. Use case diagram can be used for basic requirement stage and activity diagram can be used for detailed identification and analysis of all functional processes. Use case diagram cannot be capable to map all concepts of the functional processes.

Using UWE model, all functional user requirements in detailed analysis is presented with a class diagram. The class diagram includes process classes which contain the functionality that need to be provided for the user as methods in these classes. Process classes are mapped into activity dia-

Table 1: Data movement types between source and destination elements in activity diagram

| Source | Destination | Description |
|----------------|----------------|---|
| USER ACTION | SYSTEM ACTION | This is considered as Entry. |
| STORAGE ACTION | SYSTEM ACTION | This is considered as Read. |
| SYSTEM ACTION | USER ACTION | This is considered as Exit. |
| SYSTEM ACTION | STORAGE ACTION | This is considered as Write. |
| SYSTEM ACTION | SYSTEM ACTION | This is not data movement. |
| USER ACTION | USER ACTION | This is not data movement. |
| USER ACTION | STORAGE ACTION | Communication between USER ACTION and STORAGE ACTION is always performed through SYSTEM ACTION. |
| STORAGE ACTION | USER ACTION | |

-gram which perform some specific functions needed by a user. COSMIC functional size measurement is performed by using activity diagram through dividing the diagram into three partitions, USER ACTION, SYSTEM ACTION and STORAGE NODE and then identifying data group movement among them.

If there is data flow between USER ACTION and SYSTEM ACTION or SYSTEM ACTION and STORAGE NODE, that means there is a type of data movement. The direction of data movement is identified by specifying elements of source and destination in the activity diagram. The process flow arrow between two stereotypes in same or different swimlanes represents the sequence of command execution inside the functional process. Identification and counting of data movement (*Entry, Exit, Read, and Write*) can be performed in the activity diagram with the support of arrows of data flow. Table 1 shows the data flow direction among elements inside the activity diagram.

When the decision node forks into two same stereotypes, there is one Entry or one Exit data movement. If the decision node forks into two different stereotypes then two data movements are considered. If a decision node is in the destination side of the data flow arrow, there is no data movement. If there is data flow arrow between two stereotypes of the same type, there is no data movement.

After the identification of all data movements inside each activity diagram, aggregation of measurement results can be performed by counting the summation of total data movements in the overall functional processes inside each class. Functional size of each class represents the summation of all data movements in that class. Effort estimation model can be constructed using cubic regression. Cubic regression can be applied using two variables, COSMIC function point and the actual effort. COSMIC function point represented as CFP and the actual effort defined as the total effort used for developing the Webapp which present in person/hour *p/h*.

According to Mendes stability [9], the estimating effort model should be tested so that the all data in the dataset which have been used must be tested to confirm symptoms of skewness and outliers. The confirmation can be through applying One-sample Kolmogorov–Smirnov (K–S) and Shapiro–Wilk (S–W) statistical tests. If these symptoms exist, this means that variables require to be normalized to look alike a normal distribution. The variables can be normalized through data transformation using the natural log transformation.

Some standard measures have been used for conducting evaluation of the effort estimation model. These measures are

magnitude of relative error MRE, mean magnitude of relative error MMRE, median magnitude of relative error MdMRE, and Pred(0.25). These are the common measures that can be used for estimation models evaluation [1] [6].

5. CASE STUDY

The proposed method applied on 19 Webapps as the case study for the method. Case study is the same that have been used by Denis Ceke and Boris Milasinovic [5]. In the proposed method, cubic regression model was used for creating the predication model of estimating effort. The independent variable that have been used for constructing the predication model was the COSMIC function point (CFP), and the dependent variable was the actual effort (ActualEffort). Table 2 shows the 19 projects data that have been used for effort estimation model, and Table 3 represents the statistics for variables that have been used in the study.

Table 2 :The projects data

| Project | CFP | ActualEffort |
|---------|-----|--------------|
| Pr1 | 168 | 136 |
| Pr2 | 75 | 115 |
| Pr3 | 63 | 107 |
| Pr4 | 86 | 89 |
| Pr5 | 72 | 97 |
| Pr6 | 69 | 90 |
| Pr7 | 39 | 50 |
| Pr8 | 202 | 365 |
| Pr9 | 96 | 215 |
| Pr10 | 79 | 190 |
| Pr11 | 72 | 130 |
| Pr12 | 70 | 120 |
| Pr13 | 42 | 25 |
| Pr14 | 123 | 110 |
| Pr15 | 45 | 90 |
| Pr16 | 60 | 110 |
| Pr17 | 30 | 16 |
| Pr18 | 30 | 32 |
| Pr19 | 65 | 85 |

Table 3 Variables statistics

| Variable | N | Missing | Minimum | Maximum | Mean | StDev |
|--------------|----|---------|---------|---------|--------|--------|
| CFP | 19 | 0 | 30 | 202 | 78.21 | 44.314 |
| ActualEffort | 19 | 0 | 16 | 365 | 114.31 | 78.558 |

The cubic regression assumptions were tested to specify whether the collected data could be applied for constructing model using this method or not. For stability, the variables have been analyzed to ensure that there were no variables containing missing values more than 60% [9]. Table 3 denotes that no variables exist with missing values.

Statistical tests have been applied to confirm symptoms of skewness and outliers. One sample K-S and S-W tests are the most popular statistical tests that have been used. The tests compare an observed distribution to a theoretical distribution. Significance values that are equal to or less than 0.05 ($p \leq 0.05$) denotes that the observed distribution is different from the theoretical distribution (see Table 4).

Table 4: Normality tests

| Variables | Kolmogorov–Smirnov (K–S) | | | Shapiro–Wilk (S–W) | | |
|--------------|--------------------------|----|---------|--------------------|----|---------|
| | Statics | N | P_value | Statics | N | P_value |
| CFP | 0.230 | 19 | 0.009 | 0.825 | 19 | 0.003 |
| ActualEffort | 0.233 | 19 | 0.008 | 0.824 | 19 | 0.003 |

K–S and S–W tests denoted that the distributions of variables are not corresponded with the normal distribution. Thus the variables must be transformed. The natural log transformation has been used for transforming all numerical variables. The transformation was applied for normalizing each of CFP and actual effort variables in order to produce transformed variables $\log(\text{CFP})$ LCFP and $\log(\text{ActualEffort})$ LActualEffort as shown in Table 5.

Table 5: Normalized data of CFP and ActualEffort

| Project | LCFP | LActualEffort |
|---------|------|---------------|
| Pr1 | 5.12 | 4.91 |
| Pr2 | 4.32 | 4.74 |
| Pr3 | 4.14 | 4.67 |
| Pr4 | 4.45 | 4.49 |
| Pr5 | 4.28 | 4.57 |
| Pr6 | 4.23 | 4.50 |
| Pr7 | 3.66 | 3.91 |
| Pr8 | 5.31 | 5.90 |
| Pr9 | 4.56 | 5.37 |
| Pr10 | 4.37 | 5.25 |
| Pr11 | 4.28 | 4.87 |
| Pr12 | 4.25 | 4.79 |
| Pr13 | 3.74 | 3.22 |
| Pr14 | 4.81 | 4.70 |
| Pr15 | 3.81 | 4.50 |

| | | |
|------|------|------|
| Pr16 | 4.09 | 4.70 |
| Pr17 | 3.40 | 2.77 |
| Pr18 | 3.40 | 3.47 |
| Pr19 | 4.17 | 4.44 |

Once the stability of estimation model has been checked, the effort estimation model equation have constructed with the support of cubic regression as below:

$$LActualEffort = -76.68 + 52.82 * LCFP - 11.51 * LCFP^2 + 0.8437 * LCFP^3 \dots (3)$$

As we mentioned, we used standard measures for conducting evaluation of the effort estimation model such as mean magnitude of relative error (MMRE), median of MRE (MdmRE), and Pred(0.25). These measures are based on the evaluation of the magnitude of relative error (MRE) which is defined as:

$$MRE = \frac{|ActualEffort - EstimatedEffort|}{ActualEffort} \dots \dots \dots (4)$$

Pred(k) was used at the level k is as follow :

$$Pred(k) = p/m \dots \dots \dots (5)$$

Where p is the projects number and MRE is less than or equal to k , and m is the total number of projects in the dataset. Level k is often being 0.25. Pred(0.25) is an amount of the estimating values where MRE is less than or equal to 0.25.

As mentioned by Abrahão et al 2010 [1], to obtain a good effort estimation model, the MMRE must be less than or equal to (0.25), $MMRE \leq 0.25$ and Pred(0.25) is greater than or equal to (0.75), $Pred(0.25) \geq 0.75$. That mean 75% of estimated values should fall within 25% of the actual values.

Table 6: The results of datasets that used for estimation effort

| Data set | LCFP | LActualEffort | estimatedEffort | MRE |
|----------|------|---------------|-----------------|-------|
| Set1 | 5.12 | 4.91 | 5.270 | 0.073 |
| Set2 | 4.32 | 4.74 | 4.718 | 0.004 |
| Set3 | 4.14 | 4.67 | 4.585 | 0.018 |
| Set4 | 4.45 | 4.49 | 4.790 | 0.066 |
| Set5 | 4.28 | 4.57 | 4.693 | 0.026 |
| Set6 | 4.23 | 4.50 | 4.658 | 0.035 |
| Set7 | 3.66 | 3.91 | 3.822 | 0.022 |
| Set8 | 5.31 | 5.90 | 5.576 | 0.054 |
| Set9 | 4.56 | 5.37 | 4.843 | 0.098 |
| Set10 | 4.37 | 5.25 | 4.747 | 0.095 |
| Set11 | 4.28 | 4.87 | 4.693 | 0.036 |
| Set12 | 4.25 | 4.79 | 4.672 | 0.024 |
| Set13 | 3.74 | 3.22 | 4.006 | 0.244 |
| Set14 | 4.81 | 4.70 | 4.978 | 0.059 |
| Set15 | 3.81 | 4.50 | 4.145 | 0.078 |
| Set16 | 4.09 | 4.70 | 4.537 | 0.034 |
| Set17 | 3.40 | 2.77 | 3.013 | 0.087 |
| Set18 | 3.40 | 3.47 | 3.013 | 0.131 |
| Set19 | 4.17 | 4.44 | 4.611 | 0.038 |
| MMRE | | | | 0.060 |
| MdMRE | | | | 0.054 |
| Pred(25) | | | | 100% |

As we mentioned earlier, mean magnitude of relative error (MMRE), median magnitude of relative error (MdMRE), and Pred(0.25) statistical methods were used for the measurement of the prediction accuracy. Results obtained for these measures are shown in Table.6. Data in the table denote that the mean magnitude of relative error MMRE value of developed model less than 0.25 which have been suggested for good predication level of the model. And the median magnitude of relative error MdMRE value of all MREs is 0.054 which states that half of projects are equal or above, and the other half are equal or less than the value of 0.054. This means that we obtained a satisfactory estimation accuracy for the model.

Referring to Table 6 again, there is no project had MRE greater than 0.25. In accordance with Pred(0.25), the results show that 100% of the dataset in the validation set gave effort estimation fallen within 0.25 of MRE, which was a good result. That means that the proposed model has a 100% probability of predicting the estimated effort to be within 25% of MRE with actual effort.

6. CONCLUSION

The proposed method presented in this work predicts Webapp development effort before application is really implemented. This method was based on the conceptual model of Webapp rather than the application source code. The proposed method

composed of Webapp effort estimation procedures depending on the conceptual model that have been constructed using UWE approach, then measuring the functional size of software using COSMIC method. The effort estimation model was created using cubic regression equation.

In this work, all functional user requirements have been captured as much as possible by using conceptual model created by UWE approach in order to apply the measurement procedures of COSMIC method for reaching good results. Case study showed that the method which have used is suitable and effective for effort estimating. Using cubic regression analysis gave more accurate estimation results than the linear regression analysis that have been used in many other previous works.

Results of this work are encouraging. Using COSMIC method with UWE models for calculating data movements can be useful for estimating effort of Webapp.. The UWE models are schema of the Webapp, so they are suitable elements for applying the COSMIC measurement procedure.

7. REFERENCES

- [1] Abrahão, S., Gómez, J., Insfran, E., "Validating a size measure for effort estimation in model-driven Web development", Inform. Sci. 2010. 180, 3932–3954, doi:10.1016/j.ins.2010.05.031.

- [2] Advantages of the COSMIC Method, COSMIC, 2014, URL <http://www.cosmicon.com/advantagesV3.asp>.
- [3] Candido, E.J.D., Sanches, R., "Estimating the size of Web applications by using a simplified function point method", IEEE Computer Society, Washington, DC, USA, pp.2004, 98–105, 2004, doi:10.1109/WBMMED,1348154.
- [4] Costagliola, G., Di Martino, S., Ferrucci, F., Gravino, C., Tortora, G., Vitiello, G., "A COSMIC-FFP approach to predict web application development effort". J. Web Eng., 2006, 5, 93–120.
- [5] Denis Ceke, Boris Milašinovic, "Early effort estimation in web application development", The Journal of Systems and Software, 2015, 103 (2015) 219-237.
- [6] Ferrucci, F., Gravino, C., Oliveto, R., Sarro, F., Mendes, E., "Investigating Tabu Search for Web Effort Estimation", the 36th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2010, pp. 350–357, 2010. doi:10.1109/SEAA.
- [7] Knapp, A., Koch, N., Zhang, G., Hassler, H.-M., "Modeling business processes in Web applications with Argo UWE", Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp, 2004, 69–83.
- [8] MagicUWE, Version 1.3.6, UWE – UML based Web Engineering. URL <http://uwe.pst.ifi.lmu.de/toolMagicUWE.html>, 2013
- [9] Mendes, E., "Cost Estimation Techniques for Web Projects", IGI Global, 2007.
- [10] Rollo, T., "Sizing e-commerce", Proceedings of the Australian Conference on Software. Sydney, Australia, 2000.
- [11] S. Di Martino, C. Gravino, " ESTIMATING WEB APPLICATION DEVELOPMENT EFFORT USING COSMIC-FFP METHOD", International Journal of Computers and Applications, Vol. 31, No. 3, 2009.
- [12] Sergio Di Martinoa, Filomena Ferruccib, Carmine Gravinob, Federica Sarroc, " Web Effort Estimation: Function Point Analysis vs. COSMIC", Information and Software Technology, 2016, 72 (2016) 90–109.
- [13] Sworooop Suhanjan Das, Kumar Devadutta, Santosh Kumar Swain, Sanjay Kumar, "Web Components as a measure for estimating Effort and Size of Web Applications", International Journal of Computer Science and Information Technologies (IJCSIT),2011, Vol. 2 (3), 1137-1143.
- [14] The COSMIC Functional Size Measurement Method Version 3.0.1 Measurement Manual, 2009. COSMIC ON URL www.cosmicon.com/dl_goto.asp?id=73
- [15] The COSMIC Functional Size Measurement Method Version 4.0, Introduction to the COSMIC Method of Measuring Software, 2014. URL www.cosmicon.com/dl_goto.asp?id=471