International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-04, Issue-09, Dec 2018

# Test Effort Estimation Tool for Mobile Apps

**Anureet Kaur, Research Scholar, I.K. Gujral Punjab Technical University, Kapurthala, India, anumahal@gmail.com**

**Kulwant Kaur, Asst. Professor, School of IT, Apeejay Institute of Management Technical Campus Jalandhar, India, kulwantkaur@apjimtc.org**

**Abstract The growth of mobile apps in the last decade has taken over all aspects of human life. With this growth rate of mobile devices, the market demand for software running on these devices has also increased. Developers and testers are expected to deliver the mobile apps on time and within budget. Testing is preliminary to deliver the high-quality app. The estimation of testing plays a vital role. The objective of this paper is to present a regression model that estimate test effort for mobile apps considering test size, mobile app factors, and test factors. The Multiple Linear Regression (MLR) model is further analyzed and validated with a new mobile project data by taking a case study to verify its suitability in real-life. The proposed model is also implemented as a web tool using ASP.NET and C# to ease out the manual calculation process. The web tool is provided to the mobile app software industry for trial and analyzes the results. To analyze the actual effort with estimates MRE measure is used. For the case study, MRE obtained with the proposed regression model is 9.71% and 30.4% is obtained with expert estimation prevalent in the company which is very high as compared to the proposed model.**

***Keywords* —*Mobile applications, Software Engineering, Software Testing, Test Effort, Estimation, Web Tool***

## I. INTRODUCTION

The popularity of mobile devices and the apps running on these devices has been amplified from the former decade. Rendering to the statistics obtainable by [1], the figure for mobile apps downloads globally in 2018 is approximately 178.1 billion and it is predictable to nurture to 258.2 billion app downloads by 2022. Perceiving the mobile app strength over desktop software hints that applications are unmistakably setting down deep roots, particularly in the realms of business and tech. Mobile application developers' are obligatory to release software on within given time frame and cost. For this reason software estimation assumes an extremely crucial job in giving the most exact measuring figure for building trust in developers and customers relationship [2].

Test Estimation is the estimation of the testing size, testing effort, testing cost and testing schedule for a specified software testing project in a specified environment using defined methods, tools and techniques [2]. A very few investigations have been performed in the context of estimating the test effort of mobile apps [3], [4]. Estimation of testing effort based on size has been proposed in many studies for traditional software [5],[6],[7],[8],[9]. Functional Size Measurement (FSM) is concerned with measuring the functionality of the software attained from requirements. The functional size of the software has proved to affect effort and then the cost of software projects [10]. The software size measurement can be used for assessment and for prediction. The prediction measurement or estimation uses a mathematical model and prediction procedures [11]. In this paper, COSMIC FSM measure has been chosen for size measurement due to its suitability for a broader range of application domains such as like embedded software, real-time apps, mobile apps etc. [12]. Mobile apps are cogitated as a hybrid of real-time apps and business apps handled in COSMIC FSM [13]. In COSMIC FSM, the functional users can interact with the mobile software using touchscreens, voice, keypad etc. The persistent storage in mobile apps is either memory cards (external memory), or over the cloud storage. Figure 1 depicts data movements in mobile app software. Entry (E) and Exit (X) data movements represent data exchange between the mobile user and mobile application. Read (R) and Write (W) data movements occur between either mobile device memory and mobile app or cloud storage and mobile app. The details of calculating COSMIC FSM can be referred from [13].

The remaining paper is ordered in the following way. Section II provides a review of the literature. Section III delivers the proposed Multiple Regression Model. Section IV presents the implementation of the proposed model using a case study. Section V depicts the results of the proposed model from the case study. Finally, section VI represents the conclusion of the paper.

 DOI : 10.18231/2454-9150.2018.1202 

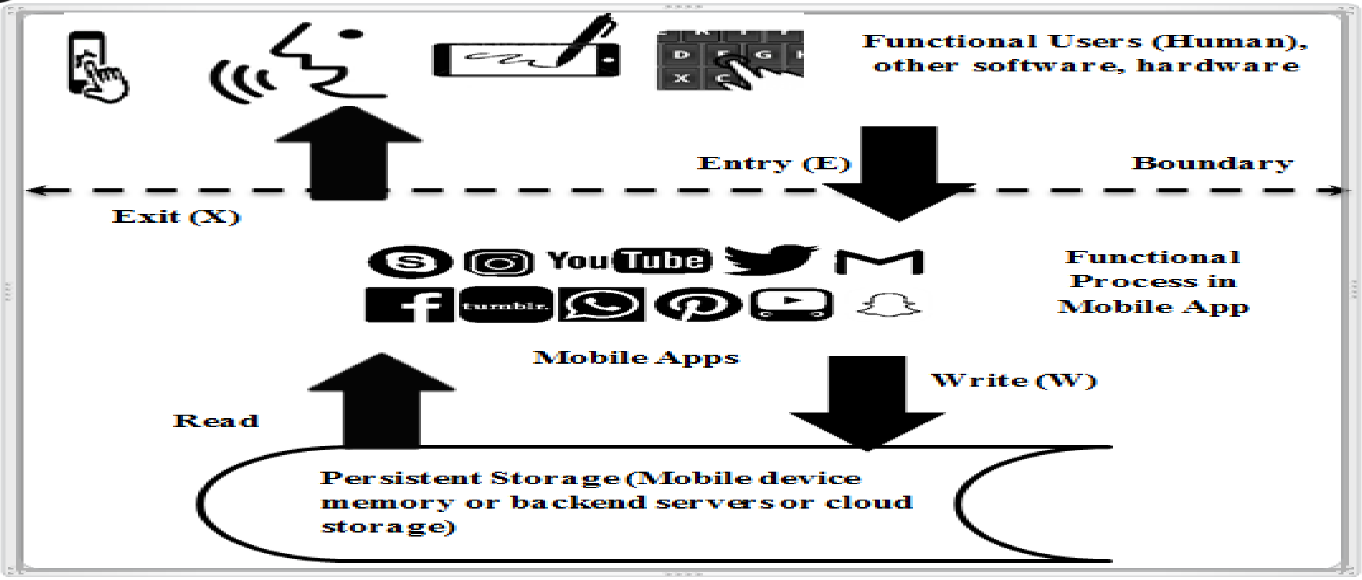



**Figure1 Data movements in mobile Application**

## II. LITERATURE REVIEW

The existing literature on test effort estimation exclusively for mobile apps is very limited. However, there is a number of studies published for estimation of test effort in traditional software. As the focus of this paper is on test estimation of mobile apps, thus the identified studies are presented therein.

Studies by [3], [14], [15] involve estimation of manual test execution effort investigating a risk and test factors as a major factor in estimation and used the mobile application case study for employment of the proposed model. The test estimation model custom the sources on the test specifications which are comprised of Controlled Natural Language (CNL). Execution points of test cases from test specification are measured for its size and complexity and then execution effort is estimated with test productivity and regression model. The authors [16] have also implemented the model in a tool that bears the measurement of test size and execution complexity, and lastly estimation of test execution effort.

A study by [4] proposed an architecture based estimation model aimed at reliability testing prediction of the mobile apps. Also, a case study is directed in two software industries. According to the authors, the notion of specification-based testing can be thought of as architecture-based testing, in which it is checked that whether the implementation is in accordance with the architectural specifications. The authors applied this framework on two mobile companies that are developing mobile applications and the results of architecture based testing helped the companies to expurgate their funds and lessen the time for software quality assurance.

A study by [17] enhanced the traditional estimation method i.e. use case point. The author considered efficiency and risk factor of testers along with environment and technical factors to estimate test effort in an agile environment. The author has applied the method on four real projects and out of the two are mobile application projects. According to the authors, the implementation of the proposed model presented more accurate results from the original use case point method for test estimation.

But none of the above techniques have considered mobile specific characteristics identified in [18].

## III. MODEL PROPOSAL

The proposed model for test effort estimation of mobile apps can be depicted in figure 2 starting with mobile application requirements as felicitated by the customer. The functional requirements are extracted, and the size of functional requirements is calculated using COSMIC function size measurement method. The functional size is considered as functional test size for input to the estimation model. The dataset of previous completed mobile applications is used to generate the regression test effort estimation, model. The dataset incorporates domain of the mobile applications along with COSMIC function size, test effort (Person-hour) for each mobile app, ratings against each mobile characteristics and test characteristic forming the basis for calculation of MobileFactor and for TestFactor. MobileFactor comprises of weight given to fifteen mobile app characteristics identified by [18]. TestFactor comprises of weight given to seven test factors from [7], [8], [19],[20],[21].

 DOI : 10.18231/2454-9150.2018.1202 

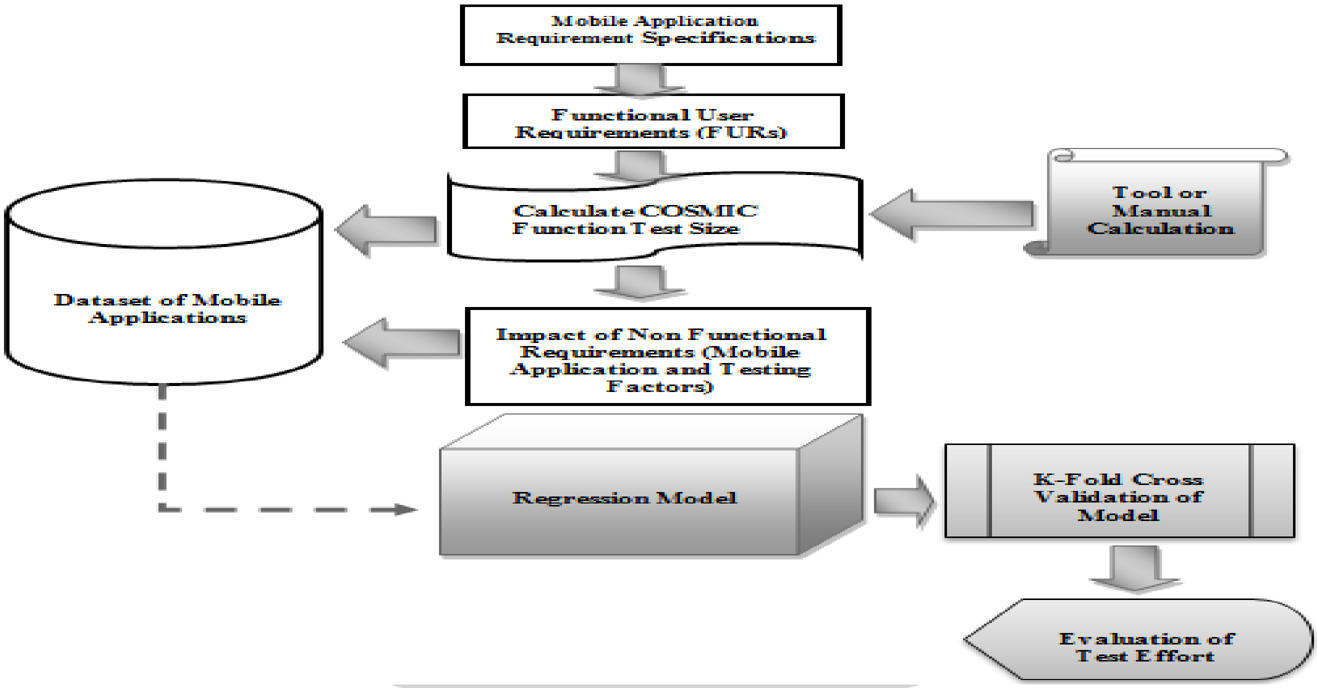



**Figure 2 High-Level view of Mobile App Test Effort Estimation Model**

An empirical study is conducted using dataset created by collecting mobile applications data from different industries and freelancers. Correlation between test effort and COSMIC measure as test size is first verified using nonparametric association statistics Spearman's rho test [22]. Multiple Linear Regression (MLR) is a model-based approach and is chosen because of its wide acceptance in many industrial contexts and researches for effort prediction based on size [23], [24]. In Multiple Linear Regression (MLR), a dependent variable is a function of independent variables. MLR model is defined as in below Eq. (1):

$$Y = a_1 + a_2 x_1 + a_3 x_2 + a_4 x_3 + ......... + a_n x_n \qquad (1)$$

In this study, actual test effort (Person-Hour) is chosen as a dependent variable that will be predicted and CFP is independent variable contributing to functional size and thereafter test size. Other two independent variables are MobileFactor and TestFactor. The model obtained takes the format of Eq. (2).

$$TestEffort = a_1 + a_2 * (CFP) + a_3 * (MobileFactor) + a_4 * (TestFactor) \qquad (2)$$

Here '$a_1$' is an intercept, '$a_2$' is a coefficient of *CFP*, '$a_3$' is a coefficient of *MobileFactor* and '$a_4$' is coefficient of *TestFactor*.

A testing effort estimation model is built using a Multiple Linear Regression (MLR) technique for predicting test effort as the outcome variable and COSMIC test size as a predictor variable. All the assumptions for linear regression are satisfied. The model presented is further validated and evaluated for their effectiveness by using k-fold cross-validation method. MRE, MMRE, MdMRE, and PRED (25) indices are used for measuring the accuracy of the model. The final MLR equation takes the following format as shown in Eq. (3):-

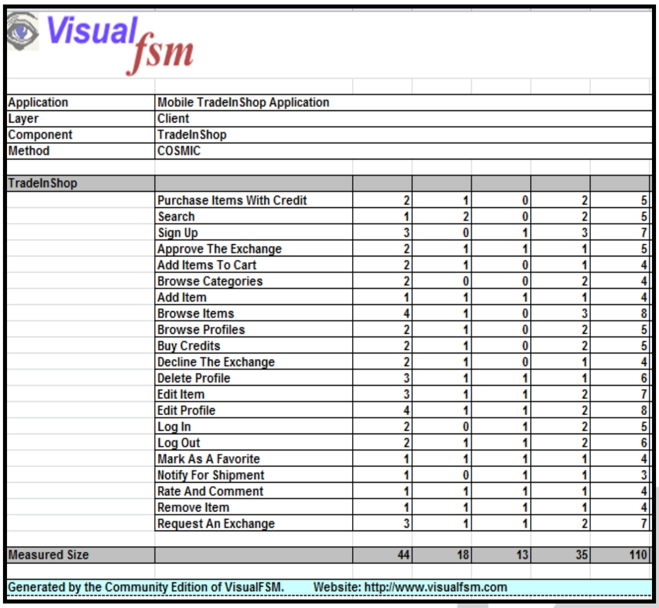
$$TestEffort = 0.309 * CFP^{1.103} * MobileFactor^{0.164} * TestFactor^{0.199} \qquad (3)$$

## IV. IMPLEMENTATION OF MODEL

The model is implemented in a case study using the proposed regression model which is implemented as a web tool. The goal of directing case study is to validate the applicability of the proposed model in real mobile app test effort estimation. A case study is conducted on US-based software industry engaged in the development and testing of web and mobile applications. A mobile app named TradeInShop is to be developed and tested. TradeInShop is a mobile application with a web service in order to get shopping items or services. The mobile application will work on mobile Android devices. It will have functions as data managing, web-based searching, items collecting (selling or favorites), messaging with users. When users run the application, they can use the functionalities the device. All information will be kept on a database which can be accessed by users with or without login. The Functional User Requirements (FURs) of the app are used for counting COSMIC size. Then COSMIC size acts as a test size for the app. Figure 3 presents the summarized report for CFP count from VisualFSM tool [25]. VisualFSM COSMIC Quick Start tool provided by Director of Pentad-SE Ltd can be used for calculating CFP and available online for academic/industrial use. Figure 4 shows the input of CFP test size along with mobile app name and description into the web tool developed by authors for further calculation.

 DOI : 10.18231/2454-9150.2018.1202 

Figure 5 presents input for mobile app characteristics impact on test effort. Similarly, figure 6 depicts test factor impact on test estimation. Finally, figure 7 shows the result as the estimated effort required for testing the mobile app.


Visual fsm

| | | | | | | |
|---|---|---|---|---|---|---|
| Application | Mobile TradeInShop Application | | | | | |
| Layer | Client | | | | | |
| Component | TradeInShop | | | | | |
| Method | COSMIC | | | | | |
| TradeInShop | | | | | | |
| | Purchase Items With Credit | 2 | 1 | 0 | 2 | 5 |
| | Search | 1 | 2 | 0 | 2 | 5 |
| | Sign Up | 3 | 0 | 1 | 3 | 7 |
| | Approve The Exchange | 2 | 1 | 1 | 1 | 5 |
| | Add Items To Cart | 2 | 1 | 0 | 1 | 4 |
| | Browse Categories | 2 | 0 | 0 | 2 | 4 |
| | Add Item | 1 | 1 | 1 | 1 | 4 |
| | Browse Items | 4 | 1 | 0 | 3 | 8 |
| | Browse Profiles | 2 | 1 | 0 | 2 | 5 |
| | Buy Credits | 2 | 1 | 0 | 2 | 5 |
| | Decline The Exchange | 2 | 1 | 0 | 1 | 4 |
| | Delete Profile | 3 | 1 | 1 | 1 | 6 |
| | Edit Item | 3 | 1 | 1 | 2 | 7 |
| | Edit Profile | 4 | 1 | 1 | 2 | 8 |
| | Log In | 2 | 0 | 1 | 2 | 5 |
| | Log Out | 2 | 1 | 1 | 2 | 6 |
| | Mark As A Favorite | 1 | 1 | 1 | 1 | 4 |
| | Notify For Shipment | 1 | 0 | 1 | 1 | 3 |
| | Rate And Comment | 1 | 1 | 1 | 1 | 4 |
| | Remove Item | 1 | 1 | 1 | 1 | 4 |
| | Request An Exchange | 3 | 1 | 1 | 2 | 7 |
| Measured Size | | 44 | 18 | 13 | 35 | 110 |

Generated by the Community Edition of VisualFSM. Website: http://www.visualfsm.com

**Figure 3 Final Reports with CFP Count**

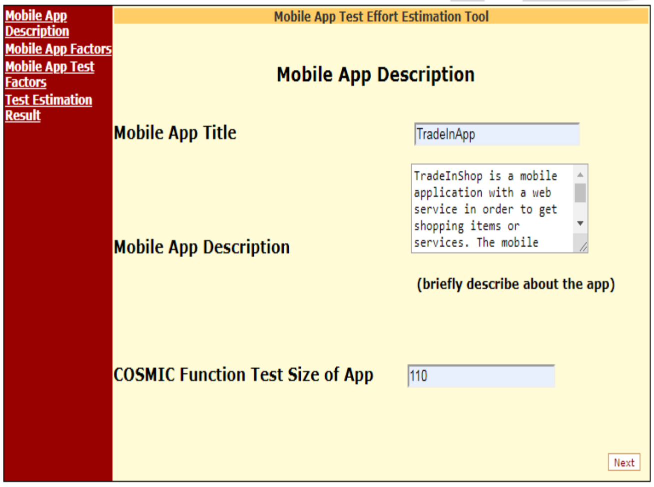



**Figure 4 Input to Mobile App Test Effort Estimation Tool**

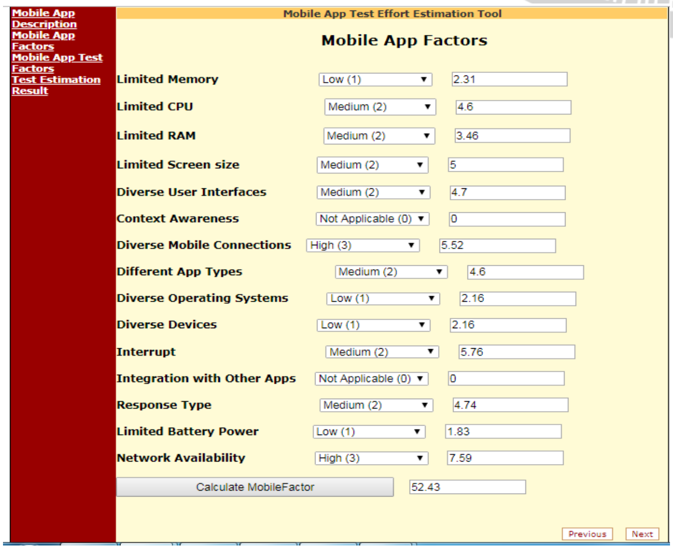



**Figure 5 MobileFactor Calculation**

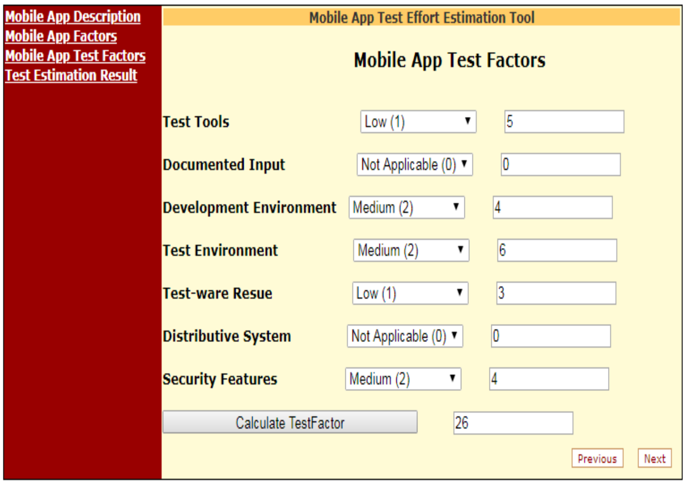



**Figure 6 TestFactor Calculation**

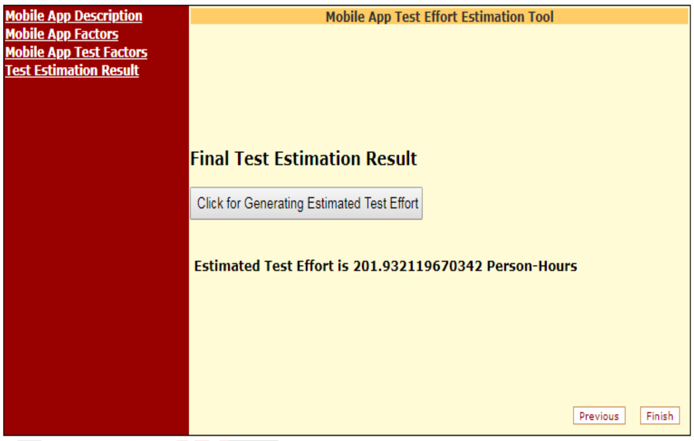



**Figure 7 Final Report on test estimation of TradeInShop**

## V. RESULTS

In this case study, different experiments are performed to compare proposed model against a traditional approach where the test effort estimation is generated by a human expert is asked to estimate the new project's effort based on personal experience and knowledge. This informal expert-based estimation method [26] used within the organization is far less near to the actual effort incurred. Expert judgment effort estimation techniques are based on the person's experience and intuition. The evaluation indices MRE is calculated for the proposed model and expert estimation method prevalent in Company indicates that the proposed model performs better than expert estimation. Table 1 and figure 8 shows the comparison of the actual test effort with the proposed model and expert judgment method.

MRE (Magnitude of Relative Error) [27] gives a normalized measure of the difference between the actual test effort and the estimated test effort.

$$MRE_i = \frac{|\,ActualTestEffort_i - PredictedTestEffort_i\,|}{ActualTestEffort_i}$$

**Table1 Comparison of prevalent Test Effort Estimation technique with proposed model**

 DOI : 10.18231/2454-9150.2018.1202 

| | Actual Test Effort | Estimated Test Effort by Proposed Model | Estimated Test Effort by Expert Estimation | MRE Proposed Model | MRE Expert Estimation |
|---|---|---|---|---|---|
| **TradeInShop Mobile App** | 230 | 207.655 | 300 | 9.71% | 30.4% |

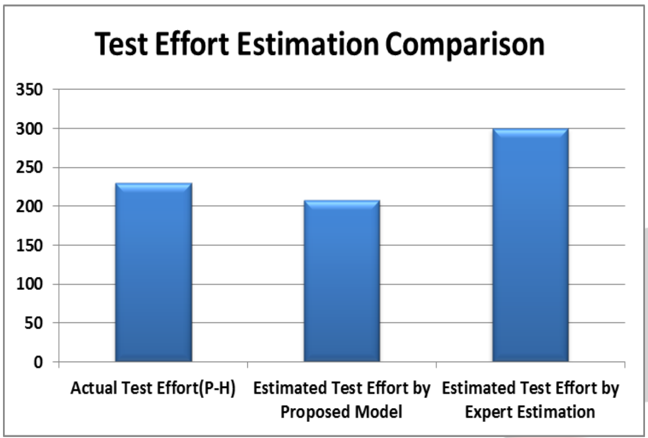



**Figure8 Comparison of Proposed Model with Expert Estimation Technique and Actual Test Effort**

From table 1, it can be analyzed that MRE % for the proposed estimation model is far than MRE% for Expert Estimation This indicates that the proposed model gives a better prediction for test estimation of mobile apps. The mobile app characteristics and test factor along with COSMIC size do play a major role in test effort prediction.

## VI. CONCLUSION

Mobile apps are captivating all miens of human ventures. The developers and testers of mobile apps are expected to deliver apps on time and of high quality. Estimation of testing helps in achieving the same by the coordinating the testing resources in the fine timeline. To fulfill this motive, this research paper proposes a for test effort estimation model for mobile apps. The model is based on COSMIC FSM size considered as test size along with mobile app characteristics and test characteristics. The proposed model is validated on real mobile app test estimation by conducting an industrial case study. The results obtained are compared against the expert judgment techniques followed in the software companies. The performance of test estimation models is evaluated. The estimated test results reveal that the proposed test estimation model gives more precise accuracy than the expert judgment technique. A web test effort estimation tool is also developed to reduce the manual calculation task. From the results obtained after implementing the proposed model on the case study, it can be concluded that the proposed model gives better prediction considering test size, mobile app factors, and test factors. For future work, the proposed model can be further be optimized using soft computing techniques.

## ACKNOWLEDGMENT

Authors are highly thankful to IK Gujral Punjab Technical University, Kapurthala, Punjab, India for providing the opportunity to conduct this research work. Also, authors are thankful to SK technology Software Company, US for the conduct of case study.

## REFERENCES

[1] "Statista," 2018. [Online]. Available: https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/.

[2] Y. Soares and R. Fagundes, "Software time estimation using regression methods," in *IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2017, pp. 1–6.

[3] E. Aranha and P. Borba, "Estimating Manual Test Execution Effort and Capacity Based On Execution Points," *Int. J. Comput. Appl.*, vol. 31, no. 3, pp. 167–172, Jan. 2009.

[4] V. Wadhwani, F. Memon, and M. M. Hameed, "Architecture based reliability and testing estimation for mobile applications," in *Communications in Computer and Information Science*, 2008, pp. 64–75.

[5] C. Arumugam and C. Babu, "Test Size Estimation for Object Oriented Software Based on Analysis Model," *J. Softw.*, vol. 10, no. 6, pp. 713–729, 2015.

[6] D. G. e Silva, B. T. de Abreu, and M. Jino, "A Simple Approach for Estimation of Execution Effort of Functional Test Cases," in *2009 International Conference on Software Testing Verification and Validation.IEEE*, 2009, pp. 289–298.

[7] É. R. C. de Almeida, B. T. de Abreu, and R. Moraes, "An Alternative Approach to Test Effort Estimation Based on Use Cases," in *2009 International Conference on Software Testing Verification and Validation*, 2009, pp. 279–288.

[8] S. Nageswaran, "Test Effort Estimation Using Use Case Points," in *In: Proceedings of 14th International Internet Software Quality Week.*, 2001.

[9] A. Abran, J. Garbajosa, and L. Cheikhi, "Estimating

 DOI : 10.18231/2454-9150.2018.1202 

the Test Volume and Effort for Testing and Verification & Validation," in *IWSM-MENSURA Conference ...*, 2007, pp. 216–234.

[10] H. Huijgens, M. Bruntink, A. van Deursen, T. van der Storm, and F. Vogelezang, "An exploratory study on functional size measurement based on code," in *Proceedings of the International Workshop on Software and Systems Process - ICSSP '16*, 2016, pp. 56–65.

[11] C. Gencel, "An Architectural Dimensions Based Software Functional Size Measurement Method.," Middle East Technical University, Ankara, Turkey., 2005.

[12] A. Abran *et al.*, "The COSMIC Functional Size Measurement Method Measurement Manual, version4.0.1." 2015.

[13] F. Vogelezang, J. Ramasubramani, and S. Rvamudhan, *Modern Software Engineering Methodologies for Mobile and Cloud Environments*, no. January. IGI Global, 2016.

[14] E. Aranha and P. Borba, "An Estimation Model for Test Execution Effort," in *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, 2007, pp. 107–116.

[15] E. Aranha and P. Borba, "Empirical studies of test execution effort estimation based on test characteristics and risk factors," in *Doctoral Symposium on Empirical Software Engineering (IDoESE 2007).*, 2007, no. June.

[16] E. Aranha, F. de Almeida, T. Diniz, V. Fontes, and P. Borba, "Automated test execution effort estimation based on functional test specifications.," in *22nd SBES –15th Tools Session*, 2008, pp. 1–6.

[17] A. W. M. M. Parvez, "Efficiency factor and risk factor based user case point test effort estimation model compatible with agile software development," in *Proceedings - 2013 International Conference on Information Technology and Electrical Engineering: "Intelligent and Green Technologies for Sustainable Development", ICITEE 2013*, 2013, pp. 113–118.

[18] A. Kaur and K. Kaur, "Systematic Literature Review of Mobile Application Development and Testing Effort Estimation," *J. King Saud Univ. - Comput. Inf. Sci.*, 2018.

[19] Abhilasha and A. Sharma, "Test effort estimation in regression testing," in *Innovation and Technology in Education (MITE), 2013 IEEE International Conference in MOOC*, 2013, pp. 343–348.

[20] P. R. Srivastava, A. Bidwai, A. Khan, K. Rathore, R. Sharma, and X. S. Yang, "An empirical study of test effort estimation based on bat algorithm," *Int. J. Bio-Inspired Comput.*, vol. 6, no. 1, p. 57, 2014.

[21] P. R. Srivastava, A. Varshney, P. Nama, and X. S. Yang, "Software test effort estimation: a model based on cuckoo search," *Int. J. Bio-Inspired Comput.*, vol. 4, no. 5, p. 278, 2012.

[22] W. J. Conover, *Practical Nonparametric Statistics*, 3rd Editio. Wiley, 1999.

[23] E. Mendes, "A Comparative Study of Cost Estimation Models for Web Hypermedia Applications," *Empir. Softw. Eng.*, vol. 8, no. 2, pp. 163–196, 2003.

[24] J. . Dolado, "On the problem of the software cost function," *Inf. Softw. Technol.*, vol. 43, no. 1, pp. 61–72, Jan. 2001.

[25] Pentad-SE, "VisualFsm The Software Measurement Tool," 2014. [Online]. Available: http://www.visualfsm.com/. [Accessed: 03-Dec-2018].

[26] B. Hauptmann, M. Junker, S. Eder, C. Amann, and R. Vaas, "An expert-based cost estimation model for system test execution," in *ACM International Conference Proceeding Series*, 2014, pp. 159–163.

[27] R. J. Freund, D. Mohr, and W. J. Wilson, *Statistical Methods*. Elsevier, 2010.

 DOI : 10.18231/2454-9150.2018.1202 