Analyzing Adaption of Size and Effort Estimation Approaches in Mobile Software

Anureet Kaur[1], Dr. Kulwant Kaur[2]
[1]*IKG Punjab Technical University, Kapurthala*
[2]*School of Information Technology, Apeejay Institute of Management Technical Campus, Jalandhar*
(E-mail: anumahal@gmail.com)

***Abstract*—**The remarkable progression of mobile technology has taken over all the important demeanors of human life. With the increase in number of end users connected via network all over the world, the demand for applications running on mobile devices will consequently upsurge. As mobile applications have its specific peculiarities and restrictions, there is no dedicated development and estimation method. Till date, the software developed for mobile devices are estimated with traditional methods ignoring specific characteristics of mobile software. There is a need to reconsider these existing methods of estimations before employing them to mobile applications. The objective of this paper is to highlight specific characteristics of mobile applications and adapting existing techniques for size and effort estimation in development for mobile apps. Statistical analysis is performed on ten different mobile applications using Function Point Analysis (FPA) and Use Case Point (UCP) methods for estimating size and effort of a mobile application software. The estimates are assessed against actual effort of mobile applications after development, and results are presented. To improve the results, a mobile size and effort estimation model is proposed which considers specific characteristics of mobile apps. The proposed method is believed to help developers to decrease the gap between estimated size and actual size of mobile applications.

***Keywords* —** *Effort Estimation, Function Point Analysis (FPA), Mobile Applications, Size, Software Engineering, Software Metric, Use Case Point (UCP).*

I. INTRODUCTION

In the last decade, because of convenience and usefulness, mobile devices have refurbished desktops and laptops. Humans of today have become slaves of mobile technology. There is not a lone domain left where mobile technology has not assumed control. They have swiftly gained reverence over the last 15 years. According to a forecast by International Data Corporation's (IDC) [1] the growth of mobile app installation on client devices will increase to 210 billion annually from 2016 to 2020 adding more than 55 billion dollars to the total revenue. In 2015 the mobile apps download was 156 billion. Also, conferring to global analyst firm Ovum, Global mobile app revenue will nurture by 2.2 times in succeeding five years, from US$36 billion in 2015 to US$79 billion in 2020[2]. With this growth, the applications running on mobile devices are developed in short time. If we can efficiently utilize resources needed for developing the mobile applications, then we can easily meet deadlines and ensures successful completion of projects on-time and within budget. For this purpose, estimation of size and effort involved in software progression plays a vital role. Size estimation [3] is used to estimate the size of a software application. Effort estimation [4] is defined as the process of predicting the person-months required to develop software. The software size estimation methodologies used for software development are Line of Code (LOC), Function Point Analysis (FPA), Feature Point metric, COCOMO models, Use Case Points, etc. [3, 4, 5]. The software development estimation methods used for developing desktop/laptop software are adapted to mobile application size and effort estimation [6, 7, 8]. But these methods do not consider the features specific to mobile software.

The objective of this research work is to cut the breach between estimated size and effort and actual size and effort for developing mobile applications by amending conventional estimation techniques. The remaining part of this paper is organized as follows; Section 2 studies prevailing research on traditional software size and effort estimation methods mainly focusing on FPA and UCP. Section 3 performs a case study on 10 mobile applications. In this case study Function Point Analysis (FPA) and Use Case Point (UCP) are used for estimating size and effort. Analysis of case study is done to compare the estimated and actual size effort. Section 4 proposes a model for size and effort estimation in developing mobile application. In section 5, the proposed method for size effort estimation is implemented on same case study done in section 3 and results are compared in section 5. The final section 6 concludes the work with future work followed by references.

II. LITERATURE REVIEW

FPA was presented by Allan Albrecht in 1979. FPA is used to predict size and effort related in software development. Nitze [8] proposes effort estimation based on empirical analogy and a component functional size of the mobile application. Wasserman [9] presents research issues emerging in software engineering for development of mobile applications. Jost [10] aims to examine if existing software estimation techniques are appropriate for measuring the mobile applications. The main verdict states that this technique cannot be applied to the initial phase of development as it depends on the complete source code to get an appropriate measure. Wadhwani [11] presents the fortes and flaws of the existing old practices of testing the software and its reliability. Also, a new framework is suggested for reliability and testing for mobile applications. De Souza et al. [12] show different characteristics of mobile applications and propose a model for estimating effort involved in mobile application development. Tunalı [13] conducts a case study for size estimation on mobile application using Function Point Analysis. The results are then compared with actual project size.

Till now, many researchers have reported concerning Use Case Point (UCP) and several case studies have also been done using Use Case Point method. Use Case Point (UCP) method allows basis on use case model constructed on functional range of the software [14]. Robiolo, G. and Orosco R. [15] highlights drawbacks of function points listing need to know all the requirements well known in advance. Such limitations are improved using use case point but still with a limitation having implemented only on projects with common environmental circumstances. Schneider et al. [16] also compared use case point to be better effort estimation measure but concluded saying that it lacks certain standards for overall calculation. Damodaran M. and Washington A, [17] also showed that there is still lack of proper standards in Use case point and if these limitations are removed then UCP can be considered as an efficient method for size and effort estimation. Carroll, E. R. [18] changed the original UCP by modifying factors effecting the calculations and then applied the modified version on 200 projects. Ali Bou Nassif et al. [19] have proposed a new use case model based on regression paradigm by means of fuzzy inference system aimed at estimating effort. Ali Bou Nassif et al. [20] suggested using Artificial Neural Network (ANN) for estimating effort imposed on the Use Case Point (UCP) method. J. Smith [21] provided a technique for calculating Line of Code (LOC) along with use case diagram. Jonathan Lee et al. [22] proposed a modified use case point measure for calculating the unadjusted use case points based relations and goals along fuzzy membership functions.

III. CRITICAL ANALYSIS OF SIZE AND EFFORT ESTIMATION ON TEN MOBILE APPLICATIONS USING FPA AND UCP

For estimation of size and effort in developing mobile applications, the actual size of mobile application after development is calculated using Source Monitor Software tool [23] in terms of line of code and then that size is compared against estimated size, calculated using FPA and UCP.

For this purpose, ten mobile applications are taken as a case study. Table 1 shows effort calculated for 10 mobile applications using FPA and UCP. The procedure for calculation of FPA and UCP is given in [3] [14] respectively.

For assessing the accuracy of estimation, a common measure called Magnitude Relative Error (MRE). MRE used by Jeffery [24] and Conte [25].

MRE (%) = (|actual effort-estimated effort| /actual effort) *100

MRE calculated for 10 mobile apps is shown in table 1. It can be analyzed that estimated values are underestimated than the actual effort for most of the apps.

TABLE 1: EXPERIMENTAL ANALYSIS OF ACTUAL AND ESTIMATED SIZE AND EFFORT

Mobile Apps	Estimated Effort(hours/person) using FP	Estimated Effort(hours/person) using UCP	Actual Effort (hours/person)	MRE in Effort	
				FP	UCP
MyApp	28.32	30.2	23.46	20%	28%
LittleIndia Restaurant	214	230	244.9	12.6%	6%
AndyChat	40.72	53.2	61.33	33.6%	13.2
Student Result Automation	57.30	66.4	77.39	25.9%	14.2%
MyCalender	212.96	220.2	238.18	11.84	7.5%
audioRecorder	47.27	59.3	68.47	30%	13.39
Pronunciation MadeEasy	166.54	198.7	211.16	21.1%	5.9%
SafariTourist App	337.2	340.8	358.49	5.9%	4.9%
TheGotToRun	96.14	102.5	116.08	17.1%	11.69%
GradeSystem	56.21	63.2	74.3	24.3%	14.9%

IV. CHARACTERISTICS SPECIFIC TO MOBILE APPLICATIONS

De Souza et.al [6] presented different characteristics of mobile applications. Table 2 enlists these characteristics. By taking these characteristics in consideration, a new method is proposed in this paper.

TABLE 2: CHARACTERISTICS OF MOBILE APPLICATIONS [6]

Sr. No.	Characteristics	Description
1.	Restricted Energy	Apps should be developed with minimum resources requirement and less energy.
2.	Graphical User Interface	Mobile devices have a small screen, keeping this constraint in mind app should be developed.
3.	Input Interface Means	Input to mobile device can be through voice, touch, keypad, stylus, etc.
4.	Network Availability	Network availability varies, so apps should be developed to work in all different situations.
5.	Limited Performance	Limited processors, the mobile apps should be developed considering the constraints.
6.	Abridged Memory	Apps should be programmed to require less amount of memory.
7.	Diverse Connections	Different connections Wi-Fi, Bluetooth, 3G, 4G, NFC etc., app should upkeep with all.
8.	Intervention	Interruptions such as receiving a message, battery low, in between calls should be considered.
9.	H/w and S/w Portability	App developer should consider hardware software portability.
10.	Response time	The mobile app should very responsive to any means of input.
11.	Power	App developers must consider battery power of the device for long duration usage.

12.	Localization	Mobile apps can have sensors that respond when mobile device moves such features make good user experiences.
13.	Integration with other mobile Apps	Check if mobile app will be integrated with other apps on user's device or not.
14.	Type of Mobile App	Consideration on whether app will run in the device only or on web also, or its native or hybrid app.

A. *Proposed Methodology*

As indicated in section 3, effort calculation for the estimated values (FPA and UCP) is lower than the actual ones. This indicates that there is necessity to improve the estimation performance. The proposed approach is a combination of existing two methods i.e. FPA and UCP. In this model, the identified mobile application characteristics as shown in table 2 will be added to FPA and UCP. The mobile applications in the case study earlier; did not consider characteristics specific to mobile apps. A modified estimation model is proposed in figure 1 considering mobile application aspects. The steps followed in proposed methodology are as follows: -

1. Customer Requirements for Mobile Application development are documented in SRS (Software requirement specification).

2. From SRS document, the requirements are converted to unique functions types. Each identified function type is given complexity based on table 3. At the end of this step we will get the value for Unadjusted Function Point (UFP) [3].

3. Calculate Technical Complexity Factor (TCF): -13 technical factors as shown in table 4 can be assigned a degree of influence from 0(no influence) to 5 (very influential). The degree of influence rating of each characteristic is summed up to give Total Degree of Influence (TDI) [14].

$$\text{TDI}= \sum_{i=1}^{13} Fi \quad (1)$$

$$\text{TCF}= 0.65 + (0.01 * \text{TDI}) \quad (2)$$

4. Calculate Environment Complexity Factor (EF): - In this step, 8 Environmental factors as shown in table 5 are considered and assigned degree of influence in the range 0 (no experience) to 5 (expert)[14].

$$\text{ECF}= 1.4 + (-0.03 * \text{EF}) \quad (3)$$

where EF is total environmental factor.

5. Calculate Mobile Complexity Factor (MCF): - In this step, 14 Mobile factors shown in table 2 are considered and assigned degree of influence in the range from 0 (No Need) and 3 (Highly needed) as shown in table 6. Finally, all the influences for each characteristic are summed up to give Mobile Total Degree of Influence (MTDI).

$$\text{MTDI}= \sum_{i=1}^{14} Fi \quad (4)$$

$$\text{MCF}=0.65+ (0.01*\text{MTDI}) \quad (5)$$

where 0.65 and 0.01 are constants.

6. To find the Mobile Adjusted Function Points (MAFP), UFP is then multiplied by Technical Complexity Factor (TCF), Environmental Complexity Factor (ECF) and Mobile Complexity Factor (MCF) as follows:

$$\text{MAFP} = \text{UFP x TCF X ECF x MCF} \quad (6)$$

7. Convert Function Point (FP) to Lines of Code (LOC)

LOC=FP*Conversion factor [26] (7)

8. Apply productivity factor to determine required effort.

Effort=FP/Productivity factor in person month [27][28].(8)

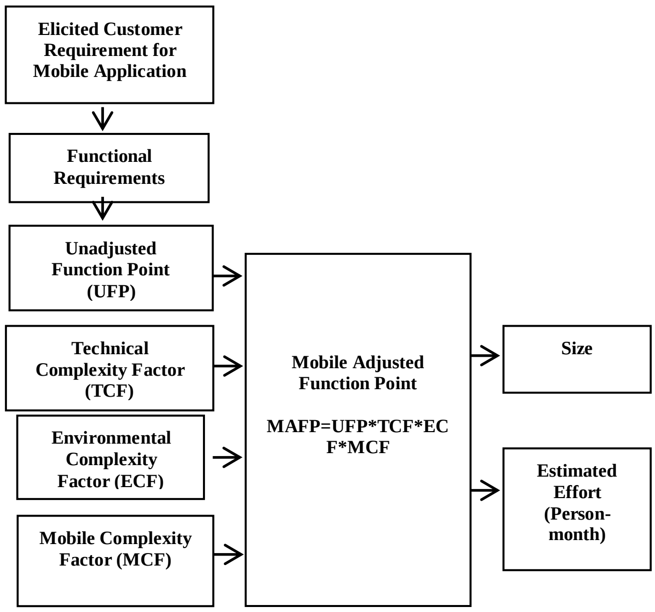

Figure 1. Proposed Mobile Size and Effort Estimation (MSEE) Model

TABLE 3: FUNCTION TYPE COMPLEXES [3]

Function Type	Low	Average	High	Total
External Input	Multiplied by 3	Multiplied by 4	Multiplied by 6	
External Output	Multiplied by 4	Multiplied by 5	Multiplied by 7	
Logical Internal File	Multiplied by 7	Multiplied by 10	Multiplied by 15	
External Interface File	Multiplied by 5	Multiplied by 7	Multiplied by 10	
External Inquiry	Multiplied by 3	Multiplied by 4	Multiplied by 6	
Total No of Unadjusted Function Points(UPF)				

TABLE 4: TECHNICAL COMPLEXITY FACTOR (TCF) WITH WEIGHTS [14]

ID	Technical Factor	Weight
1.	Distributed system	2
2.	Response	1
3.	End-user efficiency (online)	1
4.	Complex internal processing	1
5.	Reusable Code	1
6.	Install Ease	0.5
7.	Use Ease	0.5
8.	Portable	2
9.	Easy to change	1
10.	Concurrent	1
11.	Security features	1
12.	Third parties Access	1
13.	Requirement of user training facility	1

TABLE 5: ENVIRONMENTAL COMPLEXITY FACTOR (ECF) WITH WEIGHTS [14]

ID	Environmental Factor	Weight
1.	Familiarity with the Project	1.5
2.	Experience	0.5
3.	OO Programming Experience	1
4.	Lead Analyst Ability	0.5
5.	Motivation	1
6.	Stable Requirements	2
7.	Part Time Staff	-1
8.	Complex Programming Language	-1

TABLE 6: DEGREE OF INFLUENCE

Degree	Weight
Highly Needed	3
Moderate Need	2
Low Level Need	1
No Need	0

B. *Applying The Proposed Methodology on Ten Different Mobile Applications*

To estimate the total size and effort for 10 mobile applications, the results thus obtained by proposed method is shown in table 7.

TABLE 7: ESTIMATED SIZE AND EFFORT USING PROPOSED METHOD OF ESTIMATION

Mobile Apps	Mobile Adjusted Function Point (MAFP)	Estimated size using Proposed Method	Estimated Effort using Proposed Method
MyApp	2.87	152.11	22.96
LittleIndia Restaurant	29.96	1588	239.69
AndyChat	7.9	420	57.63
Student Result Automation	9.47	502	68.88
MyCalender	28.69	1521	229.58
audioRecorder	8.49	450	61.74
PronunciationMadeEasy	24.86	1318	208.45
SafariTourist App	43.58	2310	348.67
TheGotToRun	15.35	814	111.69
GradeSystem	9.47	502	68.88

V. ANALYSIS OF PROPOSED ESTIMATION MODEL WITH FPA ESTIMATION, UCP ESTIMATION AND ACTUAL EFFORT

Based on the function points obtained using proposed method, size and effort is calculated as shown in table 8. Table 9 shows the comparison of Magnitude in Relative Error (MRE) among effort estimated with FPA, UCP and with proposed methodology. The results obtained after applying the proposed method on the same case study are closer to actual effort incurred. So, chances of errors in estimating the effort involved in mobile application development are reduced after applying the proposed methodology. Figure 2 evidently describes that estimated effort using FPA and UCP are having more error rate as compared to effort estimation using proposed method.

TABLE 8: COMPARISON OF ACTUAL AND ESTIMATED SIZE AND EFFORT USING PROPOSED METHOD OF ESTIMATION

Mobile Apps	Estimated Effort using Proposed Method	Actual Effort
MyApp	22.96	23.46
LittleIndia Restaurant	239.69	244.9
AndyChat	57.63	61.33
Student Result Automation	68.88	77.39
MyCalender	229.58	238.18
AudioRecorder	61.74	68.47
PronunciationMadeEasy	208.45	211.16
SafariTourist App	348.67	358.49
TheGotToRun	111.69	116.08
GradeSystem	68.88	74.3

TABLE 9: COMPARISON OF ACTUAL AND ESTIMATED SIZE AND EFFORT USING PROPOSED METHOD OF ESTIMATION

Mobile Apps	MRE FP	MRE UCP	MRE Proposed Method
MyApp	20%	28%	2%
LittleIndia Restaurant	12.6%	6%	2.12%
AndyChat	33.6%	13.2%	6.03%
Student Result Automation	25.9%	14.2%	10.99%
MyCalender	11.84%	7.5%	3.61%
audioRecorder	30%	13.39%	9.82%
PronunciationMadeEasy	21.1%	5.9%	2.70%
SafariTourist App	5.9%	4.9%	2.73%
TheGotToRun	17.1%	11.69%	3.78%
GradeSystem	24.3%	14.9%	7.2%

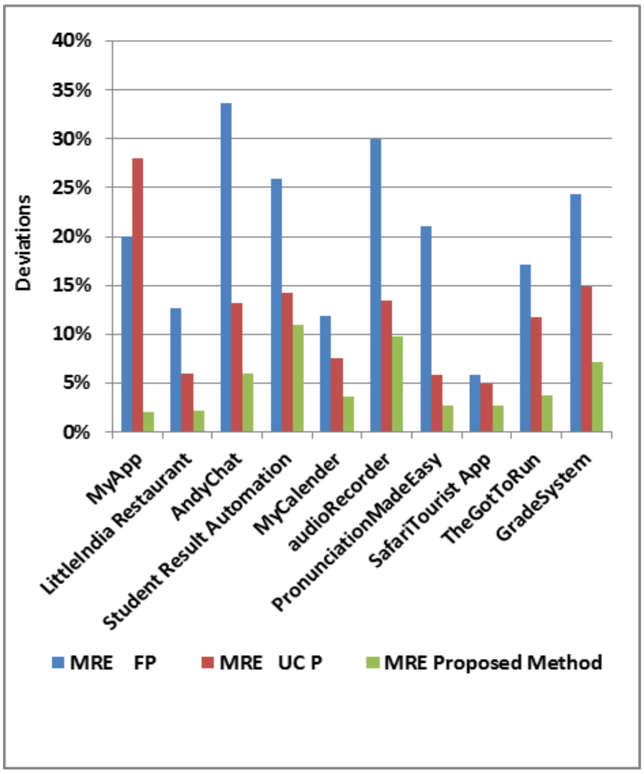

Figure 2. Comparison of Magnitude of Relative Error (MRE) in estimating effort using FPA,UCP and Proposed method

VI. CONCLUSION AND FUTURE WORK

In this paper a case study is conducted for investigating how to estimate size and effort for mobile domain using existing estimation methods i.e. FPA and UCP. Actual and estimated results are then analyzed and concluded that the relative error using FPA for estimated effort and actual effort is ranging from 5% to 33% and for UCP range is from 4% to 14% for 10 different mobile applications. A new method is proposed for improvisation. This method is smeared on same case study and there is a decrement in relative error and ranges from 2% to 10%. This comparative analysis is very important to be able to know if it is relevant to adapt the existing estimation models to mobile region. Mobile Application developers can use this piece of work as a reference to estimate effort for developing mobile apps and form basis for planning the entire schedule of the development process. In this paper, Function Point Analysis (FPA) and Use Case Point (UCP) are considered for estimation, for future work the COCOMO and other estimation models can be adapted to mobile domain for effort estimation and compare the results with proposed method.

VII. REFERENCES

[1] https://www.streetinsider.com (As accessed on 22/10/2017)

[2] https://www.ovum.com (As accessed on 22/10/2017)

[3] Yinhuan Zheng, Yilong Zheng, Beizhan Wang, Liang Shi," Estimation of software projects effort based on function point", 4th International Conference on Computer Science and Education, 2009.

[4] Pichai Jodpimai, Peraphon Sophatsathit, and Chidchanok Lursin- sap," Analysis of Effort Estimation based on Software Project Models", IEEE, 2009

[5] Nasir, M. (2006), "A Survey of Software Estimation Techniques and Project Planning Practices", proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06), IEEE, pp. 305 - 310 Las Vegas, NV.

[6] L. S. de Souza and G. S. de Aquino Jr., "The applicability of present estimation models to the context of mobile applications," in ENASE, 2014, p. 9

[7] L. S. De Souza and G.S. De Aquino, Jr., "Estimating the Effort of Mobile Application Development". Proceedings of Second International Conference on Computational Science and Engineering, 45–63, 2014.

[8] Nitze, A., Schmietendorf, A., Dumke, R. (2014)." An Analogy-Based Effort Estimation Approach for Mobile Application Development Projects.", Proc. of the IWSM/Mensura,Rotteerdam, Netherlands, CPS Publishing Service, Session New Approaches (online), 2014

[9] Wasserman A.I., (2010),"Software engineering issues for mobile application development", Proceeding FoSER '10 Proceedings of the FSE/SDP workshop on Future of software engineering research, pp. 397-400 ACM New York, NY, USA

[10] Jošt, G., Huber, J., & Heričko, M. "Using Object Oriented Software Metrics for Mobile Application Development". Second Workshop on Software Quality Analysis, Monitoring, Improvement and Applications,2013

[11] Wadhwani V., Memon F. and Hameed M.M., (2009),"Architecture based reliability and Testing estimation for mobile applications," in Wireless Networks, Information Processing and Systems, D. M. Akbar Hussain, Ed. Springer, pp. 64-75.

[12] De Souza, Silva L., and de Aquino G.S., (2014), "Mobile Application Development: How to Estimate the Effort?", Computational Science and Its Applications–ICCSA 2014.Springer International Publishing, pp.63-72.

[13] Tunali, V. 2014. Software Size Estimation Using Function Point Analysis – A Case Study for a Mobile Application. Mühendislik ve Teknoloji Sempozyumu (MTS7), 73-76

[14] G. Schneider and J. P. Winters: Applying Use Cases, Second Edition, Addison Wesley (2001).

[15] Robiolo, G., and Orosco, R. "Employing use cases to early estimate effort with simpler metrics". Innovations in Systems and Software Engineering, Vol. 4 (1), 31-43,2008.

[16] Schneider, G. and winters, J. P. Applied use Cases, Second Edition, A Practical Guide. Addison-Wesley, 2001.

[17] Damodaran, M., and Washington, A. "Estimation using use case points". Computer Science Program. Texas–Victoria: University of Houston. Sd, 2002.

[18] Carroll, E. R." Estimating software based on use case points". In Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (pp. 257-265). ACM,2005

[19] Ali Bou Nassif, Luiz Fernando Capretz, Danny Ho "Estimating Software Effort Based on Use Case Point Model Using Sugeno Fuzzy Inference System", International Conference on Tools with Artificial Intelligence, IEEE, pp. 393-398,2011.

[20] Ali Bou Nassif, Luiz Fernando Capretz, Danny Ho, "Estimating Software Effort Using an ANN Model Based on Use Case Points", International Conference on Machine Learning and Applications, IEEE, pp. 42-46,2012.

[21] J. Smith: "The Estimation of Effort Based on Use Cases", Rational Software white paper, (1999).

[22] Jonathan Lee, Hen-Tin Lee, Jong-YihKuo, "Fuzzy Logic as a Basic for Use Case Point estimation", International Conference on Fuzzy Systems, IEEE, pp.2702-2707, 2011.

[23] http://www.campwoodsw.com/sourcemonitor.html/ (As accessed on 22/10/2017)

[24] R. Jeffery, M. Ruhe and I. Wieczorek, "Using Public Domain Metrics to Estimate Software Development Effort", in Proceedings. Seventh International Software Metrics Symposium, 2001. METRICS 2001, p.16-27.

[25] S.D. Conte, H.E. Dunsmore, V.Y. Shen, Software Engineering Metrics and Models, The Benjamin/Cummings Publishing Company, Inc., 1986.

[26] Albrecth and Gaffney,"Software function, source lines of code, and development effort prediction: a software science validation", IEEE Trans. on Softw.Eng. 9(6),pp 639-648, 1983.

[27] Capers Jones - Programming Languages Table – 1996

[28] ISBSG – Repository Data - Release 9 – 2004

Anureet Kaur pursed Bachelor of Science in Information Technology from Guru Nanak Dev University, Amritsar, India in 2004 and Master of Computer Application from Panjab University, Chandigarh, India in 2010. She is currently pursuing Ph.D. and currently working as Assistant Professor in Department of Computer Science and Applications, Khalsa College, Amritsar, India since 2016. She has published more than 15 research papers. Her main research work focuses on Mobile Application Testing estimation. She has 7 years of teaching experience and 3 years of Research Experience.

Kulwant Kaur is a Dean, School of Information Technology, Apeejay Institute of Management Technical Campus, Jalandhar. She received master's degree from M.B.M. Engineering College, Jai Narian Vyas University, Jodhpur. She received her Ph.D. in Software Engineering in the Faculty of Engineering & Technology, Guru Nanak Dev University, Amritsar, India. Her career spans about two decades' of research guidance, and teaching in field of Computer Science/ Applications at Bachelor and Master level courses. Her expertise areas include Artificial Intelligence and Software Engineering. As ingenious researcher, she has presented several research papers in national conferences and seminars. She has also organized national conferences and seminars. Ms. Kaur has edited two books and has contributed numerous papers to several journals and chapters to edited books. She is Life Member of Computer Society of India and Punjab Academy Sciences, Patiala.