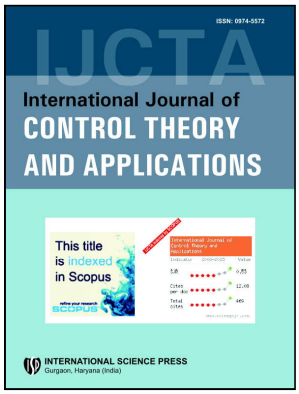


International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 19 • 2017

Evaluation of Software Project Estimation Methodology: AUCP

Archana Srivastava[1], S.K. Singh[1] and Syed Qamar Abbas[2]

[1] *Amity Institute of Information Technology Amity University, Lucknow, India, Emails: asahai@amity.edu , sksingh1@lko.amity.edu*

[2] *Ambalika Institute of Management & Technology, Lucknow, India, Email- qrat_abbas@yahoo.com*

Abstract: Organizational attitudes and demand regarding end-user development (EUD) have increased radically in the past 25 years and researchers have been describing end-user computing as a motivational component of the overall information resource in the organization. Software effort and cost estimation is a very important activity in any software development project. In most of the frequently used effort and cost estimation models EUD features are not considered as cost drivers or as a complexity enhancing feature. Use Cases are frequently used to describe the business process of object oriented projects. Advance Use Case Point Method (AUCP) is an extension of UCP. AUCP includes the additional effort required in incorporating end user development features in the software for overall project effort estimation. This paper provides a case study to demonstrate how End User Development features exceeds the development effort by using AUCP but increases End User Satisfaction to a greater extent.

Keywords: Use case point method (UCP); Advance use case point method (AUCP); End User Development (EUD); Human Computer Interaction (HCI); End User Computing (EUC); Technical Complexity Factors (TCF); Environmental Complexity Factors (ECF).

1. INTRODUCTION

Estimation of object oriented software cost and effort is an important and essential management activity which is full of uncertainty. To help the managers in this task, there are in the literature many estimation models that usually include two main metrics: Lines of Code (LOC) and Function Points (FP) [12], both of them with skills and limitations. LOC is dependent on the programming language, individual coding style and can only tell about the lexical complexity of the algorithm. The FP Analysis (FPA) is subjective and based on human decisions [13]. The most popular technique for object-oriented software cost estimation is Use Case Points (UCP) method [15].

The Use Case Points Method (UCP) is an effort estimation algorithm proposed by Gustav Karner (1993). This method was used to produce the estimate from the project's use cases. The UCP method examines the

project's use case, actors, scenarios and various technical and environmental factors and conceptualizes them into an equation. The UCP considers the Use Case (UC) model [21] to estimatethe size and effort of an object-oriented system.

The advantage to estimating with use case points is that the process can be automated. Some use case management tools will automatically count the number of use case points in a system. This can save the team a great deal of estimating time. Second advantage is that it should be possible to establish an organizational average implementation time per use case point. This would be very useful in forecasting future schedules. Though, this depends heavily on the assumption that all use cases are consistently written with the same level of detail.

A third advantage to use case points is that they are a very clean measure of size. Good estimation approaches allowsseparating estimating of size from deriving duration. Use case points succeed in this regard because the size of an application will be independent of the size, skill, and experience of the team that implements it [7][16].

If the software project estimation is done earlier in the software project development life cycle, it supports managers, developers, and software testers to plan earlier accordingly. The UCP method can produce an early estimate within 20 percent of the actual effort, and often, closer to the actual effort than expert judgment and other estimation methodologies [8][19]. The accurate estimation is a crucial & most important issue for the timely development & delivery of software within the expected budget. There are many models available for estimation still research community has yet to provide a viable model for End-User Development (EUD) environments. EUD essentially out-sources development effort to the end user. Hence one element of the size and effort is the additional design time consumed in end-user programming [2][17].

2. END USER DEVELOPMENT

End User Development is related to HCI fields of intelligent user interfaces, programming-by-demonstration, adaptive user interfaces and development tools. End User Development features if incorporated in a software or website it enhances the quality and increases end user satisfaction. End users concerned here are not professional developers but have sufficient knowledge of their respective domains and like to do coding or use various wizards to customize things as per their own requirements.

EUD research mainly emphases on methods for lowering the obstacle of entry to software development. Such methods cover a wide spectrum, from enhancing the macros and spreadsheets that millions use every day to sophisticated algorithms that create programs by example without ever exposing the user to textual code [11][20]. Additional EUD features in softwareincreases the development effort but ensures high quality that is measured by the fulfillment of end user requirements.

3. CASE STUDY

Use Case Point Method (UCP) are as follows:[1]

a) Calculating Unadjusted Use Case Weight (UUCW).

b) Calculating Unadjusted Actor Weight (UAW)

c) Calculating the Technical Complexity Factor (TCF)

d) Calculating Environment Complexity Factor (ECF)

e) Calculating Unadjusted Use Case Points (UUCP), where UUCP = UAW + UUCW.

f) Calculating Complexity Factor, where:

a. TCF = 0.6 + (0:01*TF)

b. ECF = 1.4 + (0:03 * EF)

g) Calculating the Use Case Point (UCP), where:

a. UCP = UUCP * TCF * ECF

The final step, is generating estimated effort by multiplying UCP and person-hours per UCP (PH per UCP) E=UCP * PH per UCP [1,9,10]. Karner originally proposed a ratio of 20 development hours per Use Case point. Kirsten Ribu (2001) reports that this effort can range from 15 to 30 hours per Use Case point.

In this case study we have taken data of five projects and computed UCP using above mentioned steps.

UUCW = ∑ (Use Case in each group * WF) for five different projects can be calculated as follows:

Table 1
Use Case Complexity Level

Use Case Complexity level	*Description*	*Weight*	*project 1*	*value*	*project 2*	*value*	*project 3*	*value*	*project 4*	*value*	*project 5*	*value*
Simple	Using 1 to 3 Transactions	5	7	35	4	20	3	15	10	50	5	25
Average	Using 4 to 7 Transactions	10	13	130	8	80	4	40	8	80	4	40
Complex	Using more than 7 Transactions	15	7	45	4	60	2	30	3	45	6	90
		UUCW	project 1	210	project 2	160	project 3	85	project 4	175	project 5	155

UUCW = ∑ (Use Case in each group * WF) WF)

Table 2
Actor Complexity Level

Actor Complexity level	*Description*	*Weight*	*project 1*	*value*	*project 2*	*value*	*project 3*	*value*	*project 4*	*value*	*project 5*	*value*
Simple	Interacts through API, as Command Prompt	1	0	0	3	3	8	8	5	5	8	8
Average	Interacts through Protocol as TCP/IP, HTP	2	4	8	0	0	7	14	5	10	5	10
Complex	Interacts through GUI or Web Page	3	3	9	8	24	0	0	5	15	7	21
		UAW	project 1	17	project 2	27	project 3	22	project 4	30	project 5	39

UAW = ∑ (Actors in each group * WF)

Calculating Unadjusted Use Case Points(UUCP)

UUCP = UUCW + UAW

UUCP	project 1	227	project 2	187	project 3	107	project 4	205	project 5	194

Table 3
Project Complexity Level (Technical Factors)

			project complexity								
Ti	*Technical factors*	*Weight*	*project 1*	*value*	*project 2*	*value*	*project 3*	*value*	*project 4*	*value*	*project 5*	*value*
T1	Distributed System Required	2	1	2	2	4	1	2	3	6	5	10
T2	Response Time Is Important	1	3	3	4	4	1	1	4	4	5	5
T3	End User Efficiency	1	3	3	2	2	2	2	5	5	5	5
T4	Complex Internal Processing Required	1	3	3	5	5	1	1	3	3	3	3
T5	Reusable Code Must Be A Focus	1	0	0	3	3	2	2	3	3	3	3
T6	Installation Easy	0.5	0	0	3	1.5	1	0.5	2	1	5	2.5
T7	Usability	0.5	5	2.5	4	2	1	0.5	4	2	5	2.5
T8	Cross-Platform Support	2	0	0	2	4	0	0	3	6	5	10
T9	Easy To Change	1	3	3	3	3	1	1	3	3	4	4
T10	Highly Concurrent	1	0	0	3	3	2	2	3	3	4	4
T11	Custom Security	1	0	0	3	3	2	2	2	2	5	5
T12	Dependence On Third-Part Code	1	3	3	4	4	1	1	3	3	3	3
T13	User Training	1	0	0	3	3	2	2	3	3	4	4
		TF	project1	19.5	project2	41.5	project3	17	project 4	44	project 5	61
TCF = 0.6 + (0.01 * TF)		TCF	project1	0.795	project2	1.015	project3	0.77	project 4	1.04	project 5	1.21

Table 4
Project Complexity Level (Environmental factors)

			project complexity								
Fi	*Environmental factors*	*Weight*	*project 1*	*value*	*project 2*	*value*	*project 3*	*value*	*project 4*	*value*	*project 5*	*value*
F1	Familiar with Objectory	1.5	5	7.5	4	6	1	1.5	3	4.5	4	6
F2	Part time workers	-1	0	0	1	-1	1	-1	3	-3	3	-3
F3	Analyst capability	0.5	5	2.5	4	2	2	1	4	2	5	2.5
F4	Application experience	0.5	0	0	2	1	4	2	3	1.5	5	2.5
F5	Object oriented experience	1	5	5	3	3	4	4	3	3	5	5
F6	Motivation	1	5	5	3	3	4	4	4	4	5	5
F7	Difficult programming language	-1	0	0	2	-2	4	-4	3	-3	3	-3
F8	Stable requirements	2	3	6	4	8	3	6	3	6	5	10
		EF	project1	26	project2	20	project3	13.5	project 4	15	project 5	25
ECF = 1.4 + (-0.03 * EF)		ECF	project1	0.62	project2	0.8	project3	0.995	project 4	0.95	project 5	0.65

4. ADVANCE USE CASE POINT

End user takes some effort in programming as to satisfy their requirements. Additional Technicaland environmental factors are provided to the end-user for development comfort. The additionaltechnical and environmental cost drivers considered while providing end user developmentfeatures in software incur extra cost in development [4]. These additional technical and environmental cost drivers are assigned weights as per the effort required in implementing them. The AUCP method is proposed to be implemented as given below.

method is as follows:

a) Calculate UCP

b) Total seventeen EUD_Technical factors (EUD_TF) are identified and assigned value 0 or 1, depending on whether that feature is required or not required. If the feature is required it's rated as 1 else 0, and multiply it with weights of EUD_TF. Take the summation of all factors.

c) Total eight EUD_Environmental factors (EUD_EF) are identified and assigned value 0 or 1, depending on whether that feature is required or not required. If the feature is required it's rated as 1 else 0, multiply it with weights of EUD_EF. Take the summation of all factors.

d) Calculate EUD Technical Complexity Factor, EUD_TCF = 0.6 + (0.01 * EUD_TF)

e) Calculate EUD Environmental Complexity Factor EUD_ECF = 1.4 + (0.03 * EUD_EF)

f) AUCP = UCP X (EUD_TCF X EUD_ECF)

Advance UCP is now calculated by taking the product of Use Case Point, End User Development Technical Complexity Factors and End User Development Environmental Complexity Factors.

EUD_TECHNICAL FACTORS

Table 5
EUD_Technical Factors

EUD_ Ti	*EUD_TECHNICAL FACTORS*	*Weight*	*project 1*	*Value*	*project 2*	*Value*	*project 3*	*Value*	*project 4*	*value*	*project 5*	*value*
T1	Creating throw away codes	0.5	1	0.5	1	0.5	0	0	0	0	0	0
T2	Creating reusable codes	1.2	1	1.2	1	1.2	0	0	1	1.2	1	1.2
T3	Sharing reusable code	1.4	0	0	1	1.4	1	1.4	1	1.4	1	1.4
T4	Easy &understandable codes	1	1	1	1	1	0	0	1	1	1	1
T5	Security features in codes for more control by end users	1.3	1	1.3	1	1.3	1	1.3	1	1.3	1	1.3
T6	Authentication features	1.12	1	1.12	1	1.12	0	0	1	1.12	0	0
T7	Inbuilt feedback about the correctness	1.3	0	0	1	1.3	1	1.3	1	1.3	1	1.3
T8	Testable codes	1.2	1	1.2	1	1.2	1	1.2	1	1.2	0	0
T9	Tools for analyzing by debugging	1.4	0	0	1	1.4	0	0	0	0	1	1.4

(*contd...Table 5*)

EUD_ Ti	*EUD_TECHNICAL FACTORS*	*Weight*	*project 1*	*Value*	*project 2*	*Value*	*project 3*	*Value*	*project 4*	*value*	*project 5*	*value*
T10	Error detection tools	1.2	0	0	1	1.2	1	1.2	0	0	1	1.2
T11	online help availability	1.3	1	1.3	1	1.3	0	0	1	1.3	1	1.3
T12	Self – efficiency (High sense of control over the environment)	1.11	1	1.11	1	1.11	1	1.11	1	1.11	1	1.11
T13	Perceived ease of use: Apart from extrinsic motivation intrinsic motivation (enjoyment) should be present.	1.20	1	1.2	1	1.2	0	0	1	1.2	1	1.2
T14	Perceived usefulness	1	1	1	1	1	1	1	1	1	1	1
T15	Flexible codes	1.2	0	0	1	1.2	0	0	1	1.2	0	0
T16	Scalability features	1.25	1	1.25	1	1.25	1	1.25	1	1.25	0	0
T17	Ease of Maintenance	1.2	0	0	1	1.2	0	0	1	1.2	1	1.2

EUD_ENVIRONMENTAL FACTORS

Table 6
EUD_Environmental factors

Fi	*EUD_Environ-mental factors*	*Weight*	*project 1*	*Value*	*project 2*	*Value*	*project 3*	*Value*	*project 4*	*value*	*project 5*	*value*
F1	Content Level of EUP	1.4	0	0	1	1.4	1	1.4	1	1.4	0	0
F2	End User Computing Capability	0.25	1	0.25	1	0.25	0	0	1	0.25	0	0
F3	Ease of Use & Feedback	1.2	1	1.2	1	1.2	1	1.2	1	1.2	0	0
F4	Inbuilt System Assistance for EUP	1.25	1	1.25	1	1.25	0	0	0	0	1	1.25
F5	Training & learning Time Constraint for end user	1.12	1	1.12	1	1.12	1	1.12	0	0	1	1.12
F6	Reliability of End User Code	1.2	0	0	1	1.2	1	1.2	1	1.2	1	1.2
F7	End User Storage Constraint	1.02	1	1.02	1	1.02	0	0	0	0	0	0
F8	Risk Factors	1.12	0	0	1	1.12	0	0	0	0	1	1.12
		EUD_TF		12.18		19.88		9.76		16.78		14.61
		EUD_EF		4.84		8.56		4.92		4.05		4.69
EUD_TCF = 0.6 + (0.01 * EUD_TF)				0.7218		0.7988		0.6976		0.7678		0.7461
EUD_ECF = 1.4 + (0.03 * EUD_EF)				1.5452		1.6568		1.5476		1.5215		1.5407
AUCP = UCP X (EUD_TCF X EUD_ECF)				124.80		201		88.50		236.61		175.39

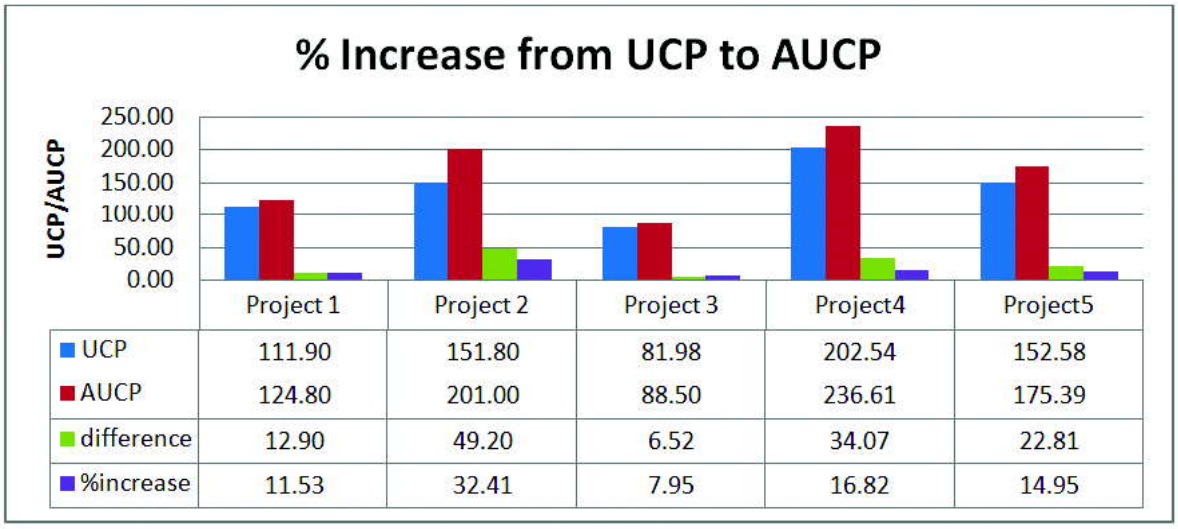
5. RESULT ANALYSIS

On preliminary analysis of the results of UCP values and AUCP values of the three different projects we observe that the difference between AUCP and UCP is as follows:

Table 7
Result Analysis

	UCP	*AUCP*	*difference*	*%increase*
Project 1	111.90	124.80	12.90	11.53
Project 2	151.80	201.00	49.20	32.41
Project 3	81.98	88.50	6.52	7.95
Project4	202.54	236.61	34.07	16.82
Project5	152.58	175.39	22.81	14.95

This result analysis shows minor increase in the effort required for development incorporating EUD features. Average increase will be approximately 5%-12%. In extreme cases it may be around 20%-33%. End User computingenriches end user satisfaction which is the final goal of Information system. One promising approach is end-user development (EUD), the practice of users creating, modifying, or extending programs for personal use [6, 5]. This approach has two main benefits. One, it puts systems design in the hands of the domain experts who are most familiar with what requirements must be built. Two, it scales with both a rapid increase in users and the increasing rate of change of many business processes [18].



6. CONCLUSION

User satisfaction refers to the quality product, system or tool that is able to satisfy specific requirements of the end user. User satisfaction with an application has been defined as 'the affective attitude towards a particular computer application by an end user who interacts with the application directly' [4]. AUCP calculates the effort considering the additional End User Development features that are incorporated in the software considering the development requirements of the end user. When EUD features are incorporated in the system or tools, it can be easily justified that the cost incurred in providing additional tools required for end user computing will be of less significance in cost compared to the benefits that end user will get in return[3]. This case study justifies that EUD features increase a little development effort but enhance End User Satisfaction exponentially hence the additional effort is justified.

Thus End User Development enhances the End user satisfaction level as users are involved throughout the development process starting from the requirement gathering to designing phase. These results are based on the preliminary analysis and will be further evaluated for accuracy using statistical tools.

ACKNOWLEDGEMENT

We are extremely thankful to the Software Development & Consultancy QRAT, Lucknow, India, for providing the necessary data and information for conducting this case study.

REFERENCES

[1] Resource Estimation for Objectory Projects Gustav Karner Objective Systems SF AB Torshamnsgatan 39, Box 1128, 164 22 Kista, September 17, 1993.

[2] Srivastava, Archana, Syed Qamar Abbas, and S. K. Singh. "ENHANCEMENT IN FUNCTION POINT ANALYSIS."airccse.org/journal/ijsea/papers/3612ijsea10.pdf

[3] International Journal of Software Engineering & Applications (IJSEA), Vol.6, No.2, March 2015

[4] Srivastava, A., Singh, S.K. and Abbas, S.Q., ADVANCEMENT OF UCP WITH END USER DEVELOPMENT FACTOR: AUCP, 2015, airccse.org/journal/ijsea/papers/6215ijsea01.pdf

[5] Gelderman, M. (1998). The relation between user satisfaction, usage of information systems and performance. Information & Management, 34, 11-18.

[6] Lieberman, H., Paterno, F., Klann, M. and Wulf, V. 2006. End-user development: An emerging paradigm. *End User Development*. (2006), 1–8.

[7] Ko, A.J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwing, M., Scaffidi, C., Lawrance, J., Lieberman, H., Myers, B., Rosson, M.B., Rothermel, G., Shaw, M. and Wiedenbeck, S. 2010. The state of the art in end-user software engineering. *ACM Computing Surveys*. (2010).

[8] Estimating With Use Case Points, Mike Cohn, Mountain Goat Software, www.mountaingoatsoftware.com

[9] Carroll, Edward R. "Estimating Software Based on Use Case Points." 2005 Object-Oriented, Programming, Systems, Languages, and Applications (OOPSLA) Conference, San Diego, CA, 2005.

[10] B. Anda," Comparing Effort Estimates Based on Use Case Points with Expert Estimates," 2007.

[11] C. Gence, L. Buglione, O. Demirors, P. Efe," A Case Study on the Evaluation of COSMIC-FFP and Use Case Points," 2006.

[12] Lieberman, H., Paterno, F., Klann, M. and Wulf, V. 2006. End-user development: An emerging paradigm. *End User Development*. (2006), 1–8.

[13] Albrecht, A. (1979). Measuring Application Development Productivity. *In*: Proc. of IBM Applications Development Symposium,pg. 83–92, October.

[14] Marcio Rodrigo Braz, Silvia Regina Vergilio. (2006). Software Effort Estimation Based on Use Cases, *In*: Proceedings of the30th Annual International Computer Software and Applications Conference (COMPSAC'06). IEEE.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[15] Iraji, Mohammad Saber, and Homayun Motameni. "Object Oriented Software Effort Estimate with Adaptive Neuro Fuzzy use Case Size Point (ANFUSP)", International Journal of Intelligent Systems and Applications, 2012.

[16] www.ukessays.com

[17] www.martinig.ch

[18] ucersti.ieis.tue.nl

[19] PRABHAKARARAO S., COOK C., RUTHRUFF J., CRESWICK E., MAIN M., DURHAM,M., AND BURNETT M. 2003. Strategies and behaviors of end-user programmers with interactive fault localization. IEEE Symposium on Human-Centric Computing Languages and Environments, Auckland, New Zealand, September, 15-22.

[20] C. Schroth and O. Christ, "Brave new web: Emerging design principles and technologies as enablers of a global soa," in Proceedings of the IEEE International Conference on Services Computing, 2007. SCC 2007. Los Alamitos, CA, USA: IEEE Computer Society Press, 2007, pp. 597–604.

[21] I. Jacobson, M. Christerson, P., and G. vergaard. Object OrientedSoftware Engineering: A Use-Case Driven Approach.Addison-Wesley.