ADVANCEMENT OF UCP WITH END USER DEVELOPMENT FACTOR: AUCP

Archana Srivastava[1], Dr.S.K.Singh[2] and Dr.Syed Qamar Abbas[3]

[1]PhD Student, Amity University, Lucknow, India
[2]Professor, Amity University, Lucknow, India
[3]Director, Ambalika Institute of Management &Technology, Lucknow, India

ABSTRACT

Computer literacy and competitive pressures among end users is increasing day by day due to whichthe need for End-User Programming in software packages is also increasing for rapid, flexible, and user driven information processing solutions. End User Development out-sources development effort to the end user by enabling softwaredevelopers to create information systems that can even be adapted by technically inexperienced endusers and hence are in great demand. If end user decides to pay the price and add significant programmability to their system, there are additional costs to consider before end user can start to enjoy the payoff. It is important to calculate accurateand early estimation of software size forcalculating effort and cost estimation of software systems incorporating EUD features. With the evolution of object orientation, use cases emerged as a dominant method for structuring requirements. Use cases were integrated into the Unified Modeling Language (UML) and Unified Process and became the standard for Software Engineering requirements modelling. The Use Case Point (UCP)methodestimates project size by assigning points to use cases in the same way that Function Point Analysis (FPA) assigns points to functions. This paper discusses the concept of end-user programming and Advancement of UCP by adding end-user development/programming as an additional Effort Estimation Factor (EEF).

***KEYWORDS**:*

End User Development (EUD), Unified Modeling Language (UML), Use Case Point (UCP), Function Point Analysis (FPA), Effort Estimation Factor (EEF).

1. INTRODUCTION

EUD is defined as "a set of methods, techniques and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify, or extend a software artifact" (Lieberman et al 2006). EUD overlaps with two concepts first areend-user programming (EUP) and the other one is end-user software engineering. End-user programming (EUP) enables end users to create their own programs [2]. The term end-user programming covers the area of programming which aims for end-users whose primary jobs are other than programming. Furthermore, programming in this context is not only meant as traditional semantic programming with general purpose programming languages like C, C#, Java etc., but can be as simple as altering predefined application preferences. The key purpose with

end-user programming is to empower end-users withthe ability to customize an application or solve computable tasks through programming, and thereby, obtaining better results or solve problems more effectively. Gartner Research forecasts that by 2014, end-user developers will deliver 25% of the new business applications. The same study warns IT organizations that they need to capitalize on the opportunities that this new emerging class of developers presents, in order to respond to rapidly changing market forces and customer preferences.

The other related concept overlapping with EUD is end-user software engineering (EUSE). In EUSE the emphasis is on the quality of the software end users create, modify, or extend; thus its research focuses on methods, techniques, and tools that promote the quality of such software. This area has arisen because of the ample evidence that the programs end users create are filled with expensive errors (Panko 1998; Burnett 2010; Ko et al 2011) **[2]**.

Boehm estimated that end users in American workplaces would number 55 million in 2005 **[3]**, but in fact the end user population already exceeded 64 million in 1997 and continues to grow **[4]**. Using survey results and projections from the Bureau of Labor Statistics (BLS), Scaffidi, C., Shaw, M., and Myers estimated that over 90 million Americans will use a computer at work in 2012, including over 55 million spreadsheet and database users and 13 million self-reported programmers, compared to fewer than 3 million professional programmers **[4]**. Thus the potential number of end user programmers will significantly exceed the population of professional programmer's incoming future as EUD will result in enhancement in end user satisfaction. The strongest advantage the end-user developer has is their domain expertise. Because they are domain experts, they see no need for requirements gathering. Adjustments in functionality are more readily resolved due to the extensive domain knowledge and first-hand observations of systems limitations. This allows changes to be made faster and to be more readily accepted. Another advantage is that previous solutions can be leveraged to initiate or add needed functionality for new solutions.

2. EUD AN ADDITIONAL COST FACTOR

EUD essentially out-sources development effort to the end user. One element of the cost is the additional time requirement for designing. Another cost is learning. This is a critical cost and difficult task in EUD because end users are not professional programmers and programming is not their primary task. They only prefer development activity to achieve the end result as per their requirement for example creating a simulation, experimenting with a design, building a prototype.

Learning to use an EUD environment includes cost that has to be motivated with a perceived reward in the form of improved efficiency or empowered work practice. Cost of errors is a significant penalty for EUD users both in operation and learning. Cost of EUD to the user can be assessed in terms of the following:

- The durationwill be required to learn the End User Development product, its features and possibly its language to some extent.
- The requirements or specification effort required in refining general ideas into specific instructions.
- The effort required in programming by end users.
- The durationrequired for testing and correcting EUD productsfrom errors.

- End User Involvement in overall development and modification based on their feedback and evaluation.

Cost models like COCOMO and sizing methods like Function Point Analysis (FPA) are most commonly used in software engineering. But these approaches have some serious limitations. The COCOMO model uses lines of code(LOC) as input, which is an ambiguous measure. LOC doesn't give the algorithmic complexity of the algorithm it only gives the lexical complexity. None of these approaches are suited for sizing object-oriented or real-time software.

Object-oriented analysis and design (OOAD) applies the Unified ModelingLanguage (UML) to model the future system, and use cases to describethe functional requirements.

The use case model serves as the early requirements specification, defining the size of the future product. EUD is currently very much in demand and to include EUD features need extended designing and additional cost but in all the previously existing software cost models EUD is not included as an additional cost driver.

3. REVIEW OF USE CASE POINT (UCP) METHOD

End user expects advanced interactive user friendly applications that are easy to learn and easy to use. As user interface become easier to use they become complex to create. The Use Case Point (UCP) model [20] is based on mapping a use case diagram to a sizemetric called use-case points. A use case diagram shows how users interact with thesystem. A use case diagram is composed of use cases and actors. Use cases represent the functional requirements where an actor is a role played by a user. Each use case is represented by a use case scenario(description). The use case scenario (description) is mainly composed of a Successscenario and an Extension (Alternative) scenario.

In Use Case Point Method, Firstly, the software size is calculated according to the number of actors and use cases in a use case diagram multiplied by their complexity weights. The software size is calculated through two stages. These include the Unadjusted Use Case Points (UUCP)and the Adjusted Use Case Points (UCP).The Karner's steps for doing effort estimation with Use Case Point Method (UCP) are as follows[8]:

3.1) CALCULATING UNADJUSTED USE CASE WEIGHTS (UUCW)

The UUCW is one of the factors that contribute to the estimation of software size. To calculate the UUCW, all of the use cases must be identified and classified as Simple, Average or Complex depending on the number of transactions the use case contains. Predefined weight assigned with each use case is given in Table 1.

Table 1. Use Case Classifying Complexity

Use Case Complexity level	Description	Weight
Simple	Using 1 to 3 Transactions	5
Average	Using 4 to 7 Transactions	10
Complex	Using more than 7 Transactions	15

UUCW = ∑ (Use Case in each group * WF) (1)

3.2) UNADJUSTED ACTOR WEIGHT (UAW)

Total Unadjusted Actor Weights (UAW) obtained from counting, how many actors of eachtype (level of complexity) multiplied by the weight of each type, given in Table 2.

Table 2. Actor Type, Weight and Description

Actor Complexity level	Description	Weight
Simple	Interacts through API, as Command Prompt	1
Average	Interacts through Protocol as TCP/IP, HTP	2
Complex	Interacts through GUI or Web Page	3

UAW = ∑ (Actors in each group * WF) (2)

3.3) CALCULATING UNADJUSTED USE CASE POINTS (UUCP)

Unadjusted Use Case Points (UUCP) is obtained from the sum of the Unadjusted Use Case Weights (UUCW) with Unadjusted Actor Weights (UAW).

UUCP = UUCW + UAW (3)

3.4) CALCULATING TECHNICAL COMPLEXITY FACTOR (TCF) &ENVIRONMENTAL COMPLEXITY FACTORS (ECF)

After calculating the UUCP, the Adjusted Use Case Points (UCP) is calculated. UCP isachieved by multiplying UUCP by the Technical Factors (TF) and the EnvironmentalFactors (EF). TF and EF represent the non-functional requirements of the software. TFcontributes to the complexity of the project while EF contributes to the team efficiencyand productivity.

Technical Factors taken in UCP model along with their weights are as follows:

The **Technical Factors** considered are T1 (Distributed System Required) = 2, T2 (Response Time Is Important)=1, T3(End User Efficiency)=1, T4 (Complex Internal Processing Required)=1, T5 (Reusable Code Must Be A Focus)=1, T6 (Installation Easy)=0.5, T7 (Usability)=0.5, T8 (Cross-Platform Support)=2, T9 (Easy To Change)=1, T10 (Highly Concurrent)=1, T11 (Custom Security)=1, T12 (Dependence On Third-Part Code)=1, T13 (User Training)=1.

Environmental Factors takenin UCP model along with their weights are as follows:

The **Environmental Factors**considered areF1(Familiar with Objectory)=1.5, F2 (Part time workers)=-1, F3 (Analyst capability)=0.5, F4 (Application experience)=0.5, F5 (Object oriented experience)=1, F6 (Motivation)=1, F7 (Difficult programming language)=-1, F8 (Stable requirements)=2.

Technical Complexity Factor is calculated as

$$TCF = 0.6 + (0.01 * TF) \quad (4)$$

$$\text{Where } TF = \sum_{i=1}^{13} Ti$$

Environmental Complexity Factor is calculated as

$$ECF = 1.4 + (-0.03 * EF) \quad (5)$$

$$\text{Where } EF = \sum_{i=1}^{8} Fi$$

Ti and Fi are technical and environmental factors are rated on a scale of 0 to 5 where it is interpreted as:

0	No Influence
1	Incidental
2	Moderate
3	Average
4	Significant
5	Essential

3.5) CALCULATING USE CASE POINTS (UCP)

UCP= UUCP * TFC * ECF (6)

For effort estimation, Karner proposed 20 person-hours to develop each UCP.

Effort = Size x 20 (7)

Where Effort is measured in person-hours and Size is measured in UCP.Karner proposed planning of 20 staff x hour effort per adjusted use case point. Based on the calculated UCPs, statistics from earlier projects were evaluated to visualize how much resources arerequired per UCP.After that the number of UCP were multiplied with the Mean Resources needed per UCP (MR). Karner also used the Standard Deviation of the Mean Resources (SDMR) to see how accurate the estimations are **[8]**.

4. PROPOSED ADVANCEMENT IN UCP INCORPORATING EUD FEATURES(AUCP)

End user takes some effort in programming as to satisfy their requirements. Additional Technical and environmental factors are provided to the end-user for development comfort. The additional technical and environmental cost drivers considered while providing end user development features in software are introduced below.

These EUD features can be classified into two categories as UCP model i.e.

EUD TECHNICAL FACTORS (EUD_TF)

EUD ENVIRONMENTAL FACTORS (EUD_EF)

Total seventeen EUD_TF and eight EUD_EF are considered to be included as End User Development Cost Drivers.

4.1) EUD TECHNICAL FACTORS & ENVIRONMENTAL FACTORS

Table 5. EUD Technical & Environmental Factors

Factor		Description	Weight
End User Development Technical Factors (EUD_TF)	T1	Creating throw away codes	0.5
	T2	Creating reusable codes	1
	T3	Sharing reusable code	1
	T4	Easily understandable codes	1
	T5	Security features in codes for more control by end users	1.5
	T6	Authentication features	1.5
	T7	Inbuilt feedback about the correctness	1.5
	T8	Testable codes	0.5
	T9	Tools for analyzing by debugging	1.5
	T10	Error detection tools	1.5
	T11	Availability of online help	1.5
	T12	Self – efficacy: High sense of control over the environment	1
	T13	Perceived ease of use: Apart from extrinsic motivation intrinsic motivation (enjoyment) should be present.	1.5
	T14	Perceived usefulness	1
	T15	Flexible codes	0.5
	T16	Scalability features	0.5
	T17	Ease of Maintenance	1
End User Development Environmental Factors (EUD_EF)	F1	Content Level of EUP	1.5
	F2	End User Computing Capability	1
	F3	Ease of Use & Feedback	1
	F4	Inbuilt System Assistance for EUP	1
	F5	End User Training & learning Time Constraint	1.5
	F6	Reliability of End User Code	1
	F7	End User Storage Constraint	1
	F8	Risk Factors	1.5

$$EUD\_TF = \sum_{i=1}^{17} Ti \times Wi$$

$$EUD\_EF = \sum_{i=1}^{8} Fi \times Wi$$

Where Wi is weights for the EUD_TF and EUD_EF and are rated on a scale of 0 to 5 where it is interpreted as:

0	No Influence
1	Incidental
2	Moderate
3	Average
4	Significant
5	Essential

4.2) CALCULATING ADVANCE USE CASE POINT (AUCP)

AUCP = UCP X (EUD_TF X EUD_EF)

Effort = size x 20, Where Effort is measured in person-hours and Size is measured in AUCP.

5. END USER SATISFACTION BY INCORPORATING EUD

User satisfaction can be considered as the opinion of the user about a specific computer application, which they use and the level to which that system satisfies the requirements of the end user. It also refers to the attitude or response of an end user towards information, an information system or a development tool. 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' **[17]**.

The motivating principles for EUD are that theextra effort required is comparatively less than the reward in the form of end user satisfaction. The purpose for all design and development is to achieve an optimal fit between the product and the requirements of the customer population, with minimal cost. Studies have suggested that End user satisfaction is directly related to the end users involvement in system development and in user application development.Further characteristics like end user experience, end userskills changes end users attitude towards Information System.
Maximum benefit from the end-user computing environment can be availed by providing end user training. Not spending enough resources on educating the users may have severe consequences which can result in decreased productivity and other organizational costs **[16]**.

6. CONCLUSION

Considering today's environment and coming future End user development will become the necessity though evaluating the impact of technology investments is a complex task. Current

estimates indicate that expenditure on EUC accounts for more than 50% of the information systems (IS) budget of many organizations **[18]**. On the other hand EUC enhances end user satisfaction which is the ultimate goal of Information system. It can be easily justified that the cost incurred in providing additional tools required for end user computing will be of less significance compared to the benefits that end user will get in return. Hence the study of incorporating end user development need to be studied effectively in times to come.

7. FUTURE WORK

There are some limitations in this study, which need tobe examined in further research.The importance of end user computing will be of high importance to IT companies in the domain of cost benefit analysis. It will help the IT companies determine and minimize their project cost and optimize the available resources. The study of incorporating end user development will need to be studied effectively in times to come. In future the authenticity of the Enhanced UCP formula will be evaluated based on the actual data of the IT companies. Interpretation and generalization of our study's findings should be done cautiously.

REFERENCES

[1] These figures are judgment-based extensions of the Bureau of Labor Statistics moderate-growth labor distribution scenario for the year 2005 [CSTB 1993; Silvestri and Lukaseiwicz 1991]

[2] Burnett, Margaret M. and Scaffidi, Christopher (2013):End-UserDevelopment.In:Soegaard, Mads and Dam, Rikke Friis (eds.). "The EncyclopediaofHuman-Computer Interaction, 2nd Ed."

[3] Boehm, B., et al. Cost Models for Future Software Life Cycle Processes: COCOMO 2.0. Annals of Software Engineering Special Volume on Software Process and Product Measurement, J.C. Baltzer AG Science Publishers, Amsterdam, The Netherlands, 1995.

[4] Scaffidi, C., Shaw, M., and Myers, B. The "55M End User Programmers" Estimate Revisited. Technical Report CMUISRI- 05-100, Carnegie Mellon University, Pittsburgh, PA, 2005.

[5] Paternò, F. (2001). Model-based Design and Evaluation of Interactive Applications, Springer Verlag.

[6] Liebermann, H. (2001). Your Wish is My Command: Programming by Example. San Francisco, Morgan Kaufmann.

[7] B. Boehm, Software Engineering Economics, Prentice Hall, Englewood Cliffs, N.J., 1981. Boehm 1981, pp. 58-59

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

[9] Clemmons, Roy K., 2006, Project Estimation With Use Case Point. Diversified Technical Services, Inc.

[10] "Object based designing of pattern using U2ML" in proceeding of International conference of advances in computer vision and information technology (ACVIT2007)

[11] Hartmann, B.; Wu, L.; Collins, K.; Klemmer, S.R. Programming by a Sample: Rapidly Creating Web Applications with d.mix. In Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, Newport, RI, USA, 7–10 October 2007; pp. 241–250.

[12] Kovatsch, M.; Weiss, M.; Guinard, D. Embedding Internet Technology for Home Automation. In Proceedings of IEEE Conference on Emerging Technologies and Factory Automation (ETFA), Bilbao, Spain, 13–16 September 2010; pp. 1–8.

[13] Ousterhout, J.K. Scripting: Higher level programming for the 21st century. Computer 1998, 31,23–30.

[14] Miller, R.C. End User Programming for Web Users. In Proceedings of the End User Development Workshop at CHI Conference, Ft. Lauderdale, FL, USA, 5–10 April 2003.

[15] Won, M.; Stiemerling, O.; Wulf, V. Component-based approaches to Tailorable systems. In End User Development; Lieberman, H., Paternò, F., Wulf, V., Eds.;Springer: Dordrecht, The Netherlands, 2006; Volume 9, pp. 115–141.
[16] Maryam Alavi. Some thoughts on quality issues of end-user developed systems. SIGCPR '85: Proceedings of the twenty-first annual conference on Computer personnel research, pages 200 – 207,1985.
[17] Doll, W. J., & Torkzadeh, G. (1988). The measurement of end-user computing satisfaction. MIS Quarterly, 12(2), 259-274.
[18] Nord, G. D., & Nord, J. H. (1994). Perceptions & attitudes of end-users on technology issues. Journal of Systems Management(November), 12-15.

AUTHOR BIOGRAPHY

Author Archana Srivastava has done M.Sc(Maths), MCA, M.Tech(IT) and is working as Asst. Professot at Amity Institute of information Technology, Amity University, Lucknow, India. She is persuing phD in software engineering from Amity University. She has more than 15 years of teaching experience.

Co-Author Dr. S.K.Singh is Professor and Programme Director in Amity Institute of Information Technology, Amity University, Lucknow, India. He has done M.Sc(Maths), MCA, M.Tech(IT) and PhD in Applied Computer Science. He has more than 20 years of teaching experience. Till date he has published over 20 research papers in National and International journals.

Co-Author Syed Qamar Abbas completed his Master of Science (MS) from BITS Pilani. His PhD was on computer-oriented study on Queueing models. He has more than 20 years of teaching and research experience in the field of Computer Science and Information Technology. Currently, he is Director of Ambalika Institute of Management and Technology, Lucknow. He is actively involved in academic and research work. Till date he has published over 70 research papers in National and International journals.