RESEARCH ARTICLE OPEN ACCESS

# Cost Estimation Using Parameterized For Use Case Point Model (P-UCP)

Pragya Jha, Dr. Rajani Kanta Malu
(School of Computer science and Engineering KIIT University, Bhubaneswar, Odisha, India)
(Professor, School of Computer Engineering Kalinga Institute of Industrial Technology, KIIT University Bhubaneswar, India)

## ABSTRACT

Estimating the cost of development is one of the most important and scaring tasks for a software project engineer. A lot of cost estimation models were reported become obsolete because of rapid changes on technology. Earlier methods were only applicable in procedural software estimation. There is a paradigm shift from procedural to object oriented software development. Gustav Karner [3] model for use case point estimation method was further enhanced in this paper for better evaluation. Additionally use case narratives [4] and sixteen most influential environment factors [2] were used for cost estimation.

***Keywords-*** use case point method (UCP), Function Point, Software cost estimation.

## I. INTRODUCTION

Software, being an intangible product, is hard to develop because there are so many unknowns in the development process. Even when using a well-defined methodology, the development cost of a well-defined application is not easy to predict. Some key factors that contribute to this difficulty include the precise set of functionalities to be implemented, the risks associated with the development process, and the knowledge and experience of the development team. Among these, the set of functionalities to be implemented is the most influential factor. Function Point (FP) metrics were designed to consider functional requirements instead of lines of code. Albrecht came up with the Function Point (FP) metrics in 1979 [1]. FP uses five parameters: number of inputs, number of outputs, number of inquiries, number of internal logical files and number of external logical files. Consequently, FP is based on the number of interactions and the size of data to be used in the end product. While FP eliminated the need for lines of code or delivered source instructions, and was widely used because of its independence on Development platform and environment, it does not seem to be applicable to software product developed using the object-oriented methodology. In particular, the notion of internal and external logical files is somewhat harder to identify in the object-oriented paradigm. Gustav Karner [3] came up with the notion of Use Case Point (UCP) which is somewhat similar to the notion of Function Point but based on use cases. One of these methods or techniques is a Use case modelling [6] that is a popular and widely used technique for capturing and describing the functional requirements of a software system. Periyasamy [4] extend the UCP model with a focus on internal details of each use case and Alwidian [2] that enhance sixteen environment factor were used for the software estimation. This paper describes the twenty four environment factors which are strongly related to our environment with use case narrative to show its effects on cost estimation process.

## II. AIMS AND OBJECTIVES

The main objectives of this research is to make UCP method applicable in environment by adding the main of these factors that can be summarized as follows [2]:

- Satisfied Resources (**SATR**).
- Financial Motives (**FMOT**).
- Generalized Job Description (**GJOB**).
- Client Type (**CLIT**).
- Religious Events (**REVT**).
- Unified Evaluation System (**USYS**).
- Unscheduled Events (**EVNT**)
- Continuous Training (**TRAN**).
- Imposed Partner (**IMPO**).
- Job Turnover (**TURN**).
- Decision Making (**DMAK**).
- Increasing Task (**ITAX**).
- Unified IT Strategies (**ITST**).
- Job Respect (**RSPT**).
- Terminology Concept (**TCON**).
- Income Level (**INCM**).

The extended UCP methodology uses every aspect of a use case model such as actors, use cases, associations between actors and use cases, relationships between actors, relationships between use cases and finally detailed narrative of each use case. The last one is important because it describes the missing details of a use case diagram, while others can be directly extracted from a use case diagram itself.

This makes the significant difference between UCP method given by Karner [3] and e-UCP [4].

**Table 1: Sample Use Case narrative**

| | |
|---|---|
| Use case name | A descriptive name of the use case. |
| Purpose | A brief description of the tasks to be implemented by this use case |
| Input parameters | Input parameters to the use case |
| Output parameters | Output parameters returned from this use case. |
| Primary actor | The list of primary actors who invoke this use case. |
| Secondary actor ` | The list of secondary actors that are used by this use case. |
| Precondition | Condition that must be true before invoking the methods that implement this use case. |
| Post-condition | Condition that must be true after the methods that implement this use case complete. |
| Successful scenario | A sequence of instructions that explain the successful scenario of invoking this use case. |
| Exceptions | A set of conditions that may make the use case fail when invoked |
| Includes list | Other use cases that are included in this use case. |
| Included by list | Other use cases that include this use case. |
| Extends list | Other use cases that are extended by this use case. |
| Extended by list | Other use cases that extend this use case. |
| Additional Remarks | Information entered here will be passed to other UML models. |

From the list of entries in Table 1, only the following entries will be used by the e-UCP model: Input parameters, Output parameters, Precondition, Post-condition, Successful scenario, and Exceptions. Other entries in the table are used for understanding the application domain and for deriving other UML diagrams [2].

## III. USE CASE POINT METHOD

Use case point method(hereinafter referred to as UCP) is proposed by Gustav Karner [3] of Objectory Company (renamed by rational company) in 1993. This UCP contains steps:

### 3.1 Determine and compute the Unadjusted Use Case Points (UUCP)

UUCP is composed of Unadjusted Use Case Weight (UUCW) and Unadjusted Actor Weight (UAW)
UUCP = UUCW+UAW

This step includes two inner steps as follow:- Identify, classify and weight actors Each identified actor is given a weighting from 1-3 as shown in table1, that corresponds to simple, average, and complex. Human actors are always classified as complex and receive a weighting of 3. Systems to which the new system will interface (legacy systems) are either simple or average depending on the mechanism by which they are addressed.

**Table 2: Actors with weighting factors**

| Actor Type | Complexity weighting factor |
|---|---|
| Simple | 1 |
| Average | 2 |
| Complex | 3 |

$$\text{UAW}=\sum a_i X w_i$$

- Identify, classify and weight use cases

After calculating the Total actor weight, it is necessary to create a use case model and to then classify each use case as being simple, average or complex. The method specifies ignoring use cases that are associated through extend or include stereotypes. This classification can be done depends on the basis of the number of transactions in a use case.

**Table 3: Use case with weighting factors**

| Use case type | No of transaction | Complexity weighting factors |
|---|---|---|
| Simple | <=3 | 5 |
| Average | 4 to 7 | 10 |
| Complex | >=7 | 15 |

$$\text{UUCW}=\sum_{i=1}^{m} u_i X w_i$$

### 3.2 Determine and compute the Technical Complexity Factors (TCF)

Thirteen standard technical factors exist to estimate the impact on productivity that various technical issues have on a project. Give each one of these a rating on the scale of [0 – 5]. A '0' rating means the factor is irrelevant to the project, a '5' rating means it is essential. Then for each factor it is necessary to multiply its score by its weight and sum the results. The final result is then substituted into a Technical Complexity Factors .

**Table 3: weighting factors for TCF**

| Environment Factors | Weights |
|---|---|
| Distributed systems | 2 |
| Application performance, objectives in either response or throughput | 1 |
| End user efficiency | 1 |
| Complex internal processing | 1 |
| Reusability of source code | 1 |
| Installation ease | 0.5 |
| Ease of usability | 0.5 |
| Portability | 2 |
| Concurrency | 1 |
| Special security | 1 |
| Direct access for third parties | 1 |
| Special customer training provided | 1 |
| Ease to change | 1 |

Technical Complexity Factor (TCF)

$$\text{TCF}=0.6+0.01 \sum_{i=1}^{13} fiXwi$$

### 3.3 Determine and compute the Environmental Factors (ECF)

There is eight environmental factors. Each factor is rated on a scale from [0-5]. Again, 0 means the factor has a strong negative impact for the project. 3 is average. 5 mean it has a strong positive impact for the project. Each factor is calculated by multiplying its score by its weight and producing a sum of the results. The final result becomes the Environmental Complexity Factors.

**Table 4: Weighting factors for ECF**

| Environment Factors | Weights |
|---|---|
| Familiar with RUP | 1.5 |
| Application experience | 0.5 |
| Object oriented experience | 1 |
| Analyst Capability | 0.5 |
| Motivation | 1 |
| Stable Requirements | 2 |
| Part time workers | -1 |
| Difficult programming language | 2 |

Environmental Complexity Factor (ECF) =

$$\text{ECF}=1.4 + (-0.03) \sum_{i=1}^{8} fiXwi$$

## IV. ADDITIONAL INFORMATION

**Weights for Use case Narrative Parameters**

A use case diagram must be supported by use case narratives [4]. Table 1 shows the structure of a use case narrative used in this methodology. Though there is no standard for the structure of a use case narrative, the authors found that the use case structure illustrated in Table 1 contains all information that many practitioners use. With this assumption, Table 5 describes the weights associated with the different parameters of a use case narrative.

**Table 5. Weights for Use case Narrative Parameters**

| Use case Narratives Parameters | Weights |
|---|---|
| Input parameter | 0.1 |
| Output parameter | 0.1 |
| A predicate in Precondition | 0.1 |
| A predicate in Post condition | 0.1 |
| An action in successful scenario | 0.2 |
| An exception | 0.1 |

$$\text{UUCN}= \sum_{i=1}^{m} uc_n Xw_i$$

**UCP method by using new environmental factors:**

Adding a set of sixteen influencing and suitable factors which strongly related to our environment. These factors as shown in table 6. in UCP method we have eight environmental factors and by using the new factors, number of factors will be 24 factors, so that the equation of environmental complexity factor will be :-

Environmental Complexity Factor (ECF)

$$=1.4 + (-0\text{-}03) \sum_{i=1}^{24} F_i XW_i$$

**Table 6: New Environment with weighting factor**

| Suggest Factors | Weight |
|---|---|
| Financial Motives | 1.2 |
| Generalized job description | 0.9 |
| Satisfied Resources | 0.9 |
| Client type | 1.2 |
| Religious Events | 1.2 |
| Continuous Training | 1 |
| Unified Evaluation system | 0.8 |
| Job Turnover | 0.8 |
| Imposed Partner | 1.2 |
| Decision Making | 0.8 |

| | |
|---|---|
| Increasing Task | 1.2 |
| Unified IT Strategies | 0.9 |
| Job Respect | 0.7 |
| Terminology concept | 0.8 |
| Income Level | 1.2 |

## V. OUR NEW METHOD TO CALCULATE UUCP

UUCP=UAW+UUCW+UUCN
p-UCP (with additional factors TCF and UUCN)
p-UCP= UUCP*TCF*ECF

## VI. CONCLUSION

In this paper author exploring the use case model in details that includes use case narrative and sixteen new environment factor. As, the result, the UCP calculated by new parameterized. The approach was able to classify the software system with respect to the level of code quality and the better way to calculate the use case points.

## VII. FUTURE WORK

The paper is a framework for cost estimation using the use case points, which can be finally translated into lines of code [5]. The author of this paper continue to work in UML class diagram which will provide some sort of evolution of initial cost estimation and also UCM relationship such as <<include>> and <<extend>> should be investigated in the cost estimation process. The authors have also planned to revise these factors in order to improve the calculations. Development of concurrent and real time systems is a challenging job now a day that needs to be estimated.

## VIII. ACKNOWLEDGEMENTS

This work was supported by KIIT University, Bhubaneswar, India. Our thanks to respected G. N. JHA for his valuable suggestions during various discussion sessions .

## REFERENCES

[1] Albrecht, A.J., 1979. Measuring Application Development productivity. IBM Applications Development Symp, GUIDE Int. and SHARE Inc., IBM Corp., Monterey, CA, Oct. 14-17, 1979.

[2] Jaber Alwidian, Wael Hadi ITC department CIS department Arab Open University Philadelphia University Riyadh-Saudi Arabia Amman- Jordan Enhancing the Results of UCP in Cost Estimation Using New External Environmental Factors ,IEEE 2012, 2012 International Conference on Information Technology and e-Services

[3] Karner's works on use case points was written in his diploma thesis titled "Metrics Objectory", Page no-126. Object Oriented Software Engineering. PHI Learning Private Limited. Yogesh Singh, Ruchika Malhotra Kasi

[4] Periyasamy and Aditi Ghode Department of Computer Science University of Wisconsin-La Crosse La Crosse, WI 54601 Cost Estimation using extended Use Case Point (e-UCP) Model ,IEEE 2009.

[5] Smith, J. The Estimation of Effort Based on UseCases . Rational Software, White paper. 1999.

[6] UML 2.0 Reference Manual, Object Management Group, www.omg.org, 2003.