A COMPARATIVE ANALYSIS OF OBJECT POINT METHOD WITH USE CASE METHOD IN SOFTWARE EFFORT ESTIMATION

Mr.M.Senthil Kumar
Assistant Professor(S.G)
Department of CSE,
Valliammai Engineering College,
Chennai, India
msen1982@gmail.com

Dr.B.Chidambara Rajan
Principal/Professor,
Valliammai Engineering College,
Chennai, India
profbcr@yahoo.com

ABSTRACT

The important work in software development process is to develop projects within estimated time and cost. As accuracy and effort plays a vital role for software development system. Several techniques are available for effort estimation. Now it is a big question, which method is worth for the estimation. Approach: The Question then became how to enable the estimation using several techniques. In this paper we perform the performance study of object point method and use case method to make an intelligent application for the purpose of effort estimation. On the basis of the following parameters in terms of SLOC, complexity, productive rate the performance study of these methods is accomplished. Conclusion: This study discussed all the points above, identity open problems and future research directions in the field of software effort estimation.

Keywords: Software Effort Estimation; Object point method; Use case Method; Complexity;

I. INTRODUCTION

The most complex work in software development is effort estimation. Because there is a rapid changes in the software development environments, techniques used and people requirements [1]. The challenging issue for estimators to estimate the software effort in the life cycle phase. Sometime there is a huge deviation between the actual cost and an estimated cost and hence accurate estimate is highly desired. Recently many techniques have been used to drop off the problem of estimation which used to be overestimated or underestimated. Both conditions are highly risky for software development. If the estimate is very low, then there will be more pressure to complete the project within short period [2]. On another side, if the estimate is very high, then more cost and resources are required, that exceeds our budget.

The goal of this paper is to explain the importance of effort estimation and compare two methods for estimation.

2. MATERIALS & METHODS

This section 2.1 deals with important factors which dominates the effort estimation.

Size. In a software development context, it is the complete set of business functionalities that the end user gets when the product is deployed and in use.

Effort. The person months required to produce the software application of a given size is an effort.

Quality attributes. What we expected from the product or work, it may be differ based on the team skills.

Cost. The Estimated budget is required to complete the project within the scheduled time.

3. USE CASE METHOD

This method mainly focuses on the use case diagram of the UML model [3]. System-level use case diagram includes one or more use case models showing all the use cases and actors in the system."Fig.1" shows an example of system-level use case diagram for online ticket reservation systems. Initially, Use case point is measured by assuming the number of actors and transactions included in the flow of events with some weights.

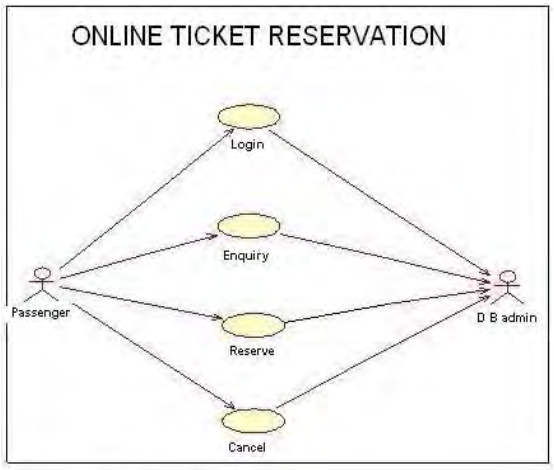

Fig.1 Use Case Model

3.1 Counting Actors:

It counts the actors who are all participated in the event. The actors in the use case are categorized as average, simple or complex

3.2 Counting Use case weights:

The weights are prioritized as simple, average and complex based on the number of transactions.

3.3 Use case Transactions & Operations:

The Use case transactions are the events take place between the user and administrator. The operations are the steps in the event.

3.4 Calculating Unadjusted use case points:

It is calculated by adding the total weights for actors to the total for use case weights and number of transactions and operations.

3.5 Calculating Use case points:

The use case points are calculated by multiply the unadjusted use case points with the technical complexity factors and environmental factor.

UCP=UUCP*TCF*EF (1)

4. OBJECT POINT METHOD:

It measures the size of an application by counting the no of screens, reports & interfaces (Known as object) required completing the coding. The Objects themselves can be classified into different level of complexity such as simple, average &complex. The object point method requires the design phase of the software development life cycle. Then only we can assume the screens & reports of the project [4].

For estimating the size of an application using object point analysis required many information such as scope of the application, data being maintained by the application input & output of the application like data, reports and queries.

"Fig.2" clearly explains the object points looks at the number of objects (screens, reports and interface) likely to be generated in the software. The effort estimated for objects is similar to that of SLOC and its focused on the code and test phase of the project execution life cycle.

Effort=app size*Complexity*Productivity rate (2)

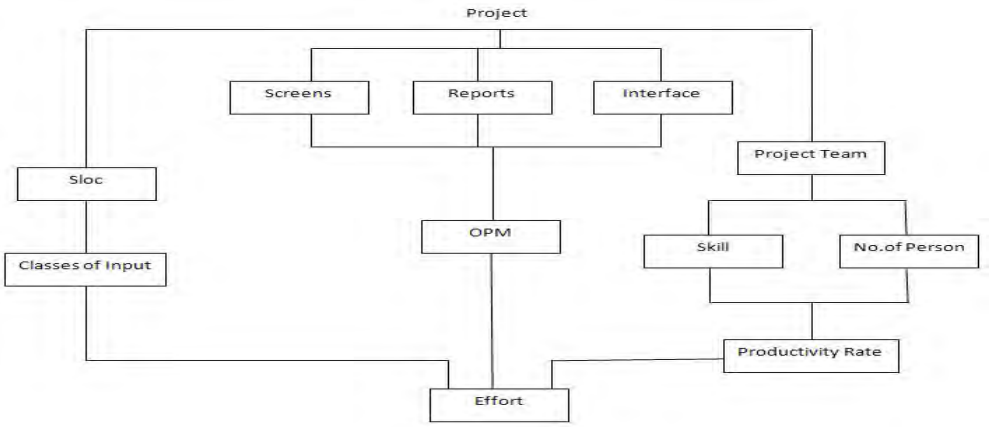

Fig.2.Object point method

The Effort is calculated by multiplying application size of the project with the complexity of the project and its productivity rate.

5. DISCUSSION

5.1 Preprocessing:

First we need to create the UML diagrams for the use case methods, but in case of object point methods, we need not use the UML diagrams [5]. So processing step is not required in object point analysis.

5.2 Estimation:

In the Use case methods, the actors and transaction weightages are calculated. Based on the weights, the unadjusted use case point is calculated. By using UUCP, we can calculate the effort of the project [2].

In the object point method, the size of the application, reports, forms are calculated. Based on the value, the effort of the project is calculated. The calculation point in the object point method is easier than the use case method [3].

5.3 Results:

In Use Case method, calculating unadjusted use point in shorter time with high judgments. So the estimators must be high in experience and skills related to the project. In the object point method, estimation is done by more calculating values of size and other parameters. This method is depended on the design phase of the software development life cycle.

6.CONCLUSION:

Based on the comparative analysis of both methods, the use case methods need more judgments and experience for classifying the types of actors and use cases. It needs expert judgments for calculating the unadjusted use case points. The object point method needs more analytical and stoical methods for calculating effort. This method allows the integration of numerical data and expert knowledge. It can be a powerful tool to tackle important problems in software engineering such as cost and quality prediction.

This work attempts to purpose the applicability of the estimation approach and suggested that the objet point method may be conceived as a fairly good approach in effort estimation in software development. Further work will utilize this comparative analysis into effort estimation and further improvement will be implemented.

REFERENCES

[1] Senthil Kumar. M and Chidambararajan.B,2012 "An Automated Neuro Model for Software Effort Estimation and Rmmi Using Competitive Learning. International Journal of computer technology and applications" vol-3: 2060-2065.

[2] Vijay .J.F and Manoharan .C.2009 "Initial Hybrid method for analyzing software estimation, benchmarking and risk assessment using design software" Journal of computer science 5:7 17-724.

[3] Briand.L.C and wvest.2002, "Empirical studies of quality models in object oriented systems.Advance in computing" 56:97-166: DOI: 10.1016/30065-2458.

[4] Pedrcyz.w and peters.J.F.1998,"Computational intelligence in software engineering" 1st edn, world scientific publishing company, USA PP: 485.

[5] Keung.J, Jeffery.R and Kitchenham.B.2004 "The Challenge of introducing new software cost estimation technology into a small software organization", Proceedings of the Australian software engineering conference, IEEE explore press, USA.pp:52-59.