Volume 4, Issue 9, September 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Problem Formulation Paper on a Tool to Evaluate the Performance of UCP

Dr. Bikrampal Kaur[1], Ms. Dipeeka Rani Bathla[2]

[1]Professor, Department of CSE, Chandigarh Engineering College, Mohali, Punjab, India

[2]Research Scholar, Department of CSE, Chandigarh Engineering College, Mohali, Punjab, India

Abstract-Several tools to support calculating UCP have been developed. As the role of software in the society becomes larger and more important, it becomes necessary to develop high-quality software cost effectively within a short period. In order to achieve this goal, the entire software development processes need to be managed based on an effective project plan. Use case point (UCP) method has been proposed to estimate software development effort in early phase of software project and used in a lot of software organizations. Intutively, UCP is measured by counting the number of actors and transactions included in use case models. To estimate the effort in earlier phase UCP method has been proposed. UCP is measured from a use case model that defines the functional scope of the software system to be developed. It is influenced by the function point methods and is based on analogous use case point. There are several experience reports that show the usefulness of UCP for estimation of software size.

Keywords- UCP, Technical Factor(TCF), Environmental Factor(EF), Unadjusted UCP(UUCP).

I. INTRODUCTION

It is essential to estimate various undesirable phenomena which happened during the project and take measures to prevent them in advance to construct distinct plan. The subjects of estimation in the area of software development are size, effort invested, development, time, technology used and quality. Today, software are based on object oriented paradigm and OOP languages. Developers use Unified Modeling Language (UML) notations and diagrams for estimation of each aspect of software development. So the main factor is to use such a technique that supports the OOPs for estimation purposes. Main approach of this thesis is to implement Use Case Point estimation (UCP) Technique to overcome the drawbacks of FP which was considered as a procedural oriented programming. This paper describes the problem of introducing use case point method to software projects. To effective introduction of UCP method, at first we develop a use case measurement tool.

II. USE CASE POINT METHOD

This section briefly explains the measurement steps in the use case point method.

Use Case Model

UCP is calculated from use case model. The use case mainly consists of two documents system and subsystem documents and use case documents that include the following items: system name, risk factors, system-level use case diagram, architecture diagram, subsystem descriptions, use case name, brief description, context diagram, preconditions, flow of events, post conditions, subordinate use case diagrams, subordinate use cases, activity diagrams, sequence diagrams, user interface, business rules, special requirements and other artifacts.

Here we explain the main item used to calculate UCP. They are system-level use case diagram and flow of events. System- level use case diageams showing all the use cases and actors in the system. Flow of events includes a section for the basic path and each alternative path in each use case.

Counting UCP

Intutively, UCP is measured by counting the no of actors and transactions included in the flow of events with some weight. A transaction is an event that occurs between an actor and the target system, the event being performed entirely or not at all. Effort estimation based on UCP method is conducted through the following steps 1-6:

Step 1: Counting Actors Weight

The actors in the use case model are categorized as simple, average and complex. Its another system interacting through a protocol such as TCP/IP. Complex actor maybe a person interacting through a GUI or webpage.

Step 2: Counting Use Cases Weight

Each use case is defined as simple, average or complex depending on no. of transactions in use case description including secondary scenarios. A transaction is set of activities which is performed entirely or not at all.

Step 3: Calculating UUCP
Its calculated by adding the total weight of actors to the total of use cases.
Step 4: Weighting TF and EF
The total effort to develop a system is influenced by factors beyond the collection of use cases that describe the functionality of the intended system. A distributed system will take more effort to develop than a non distributed system. The impact on use case points of the technical complexity of a project is captured by assessing the project on each of thirteen factors. Many of these factors represent the impact of a project's nonfunctional requirements on the effort to complete the project. The project is assessed and rated from 0 (irrelevant) to 5 (very important) for each of these factors. In Karner's formula, the weighted assessments for these twelve individual factors are next into what is called the **TFactor.** Environmental factors also affect the size of a project. The motivation level of the team, their experience with the application, and other factors affect the calculation of use case points. The weighted assessments for these eight individual factors are summed into what is called the **EFactor.**
Step 5: Calculating UCP
These are calculated as follows : UUCP*TCF*EF
Step 6: Estimating effort
By multiplying specific value man hours by UCP, estimated effort can be obtained. A factor of 20 man-hours per UCP is suggested.

III. LITERATURE SURVEY

Work done in Use Case Point Method and Effort Estimation based on use case model are given below:-

"Software Size Measurement and Productivity Rating in a Large-Scale Software Development Department", (M. Arnold, P. Pedross, 1998): The authors proposed that the use of the use case point method is accepted to estimate the size . In this paper Arnold and other compare the object oriented methods like OOSE, OMT, OOAD, UML, ROOM, OBA, and Syntropy. Although it has been accepted by the researches that a number of syntactic and semantic definitions existed for these methods, they also conclude that no method provides the technique for size measurement based on use cases and scenario. Researchers also described that since the language concepts for documentation are not well understood, it would be important to define the language concepts more precisely in advance.

"The Estimation of Effort Based on Use Cases", (John Smith 1999): The author proposed a framework to estimate LOC from use case diagram. The framework takes account of the idea of use case level, size and complexity, for different categories of system and does not resort to fine-grain functional decomposition.

"**Effort Estimation Tool Based on Use Case Points Method", (Shinji kusumoto, Fumikazu matukawa, Katsuro inoue, Shigeo hanabusa, Yuusuke maegawa):** To effective introduction of UCP method, the author has developed an automatic use case measurement tool, called U-EST. This work describes the idea to automatically classify the complexity of actors and use cases from use case model.

"**Cost Estimation using Extended Use Case Point (e-UCP) Model", (Kasi Periyasamy and Aditi Ghode, 2009):** This research focus on the internal details of use cases. For this a focus on the use case narratives, uses the relationships between the entities in a use case diagram, and hence closely estimates the size of the software product to be developed. The authors extended the original UCP model with additional information obtained from use case narratives also changes some parameters value.

"A Linear Use Case Based Software Cost Estimation Model", (Hasan.O. Farahneh, Ayman A. Issa, 2011): This research reports on the development of new linear use case based software cost estimation model applicable in the very early stages of software development being based on simple metric. Evaluation showed that accuracy of estimates varies between 43% and 55% of actual effort of historical test projects. These results outperformed those of well known models when applied in the same context. Further work is being carried out to improve the performance of the proposed model when considering the effect of non-functional requirements.

IV. PROBLEM FORMULATION

This paper proposes to create a tool which checks the performance of Use Case Point Method which is a object oriented method based on the prevailing technologies in present day market to have better and efficient results with less effort, time and cost in comparison to earlier used methods such as Function Point Method, COCOMO I, COCOMO II, Line of code which suffered several errors and shortcomings.

REFERENCES

[1] Gautam Banerjee, *"*Use Case Points- An Estimation Approach*"* Aug. 2001.
[2] Mel Damodaran, **"**Estimations using Use Case Points"
[3] Encyclopedia of Software engineering, volume 2, second edition
[4] Nancy Merlo – Schett, **"**Seminar on Software Cost Estimation*",* Computer Science, 2002
[5] Rajiv aggarwal book. K k aggarwal.

[6] Bharat Bhushan Aggarwal

[7] Hareton Leung Zhang Fan, "Software cost estimation"**,** Department of Computing, The Hong Kong Polytechnic University, pg no.2-8.

[8] B.W. Boehm et al "The COCOMO 2.0 Software Cost Estimation Model", American Programmer, pp.2-17, July 1996.

[9] Barry Boehm, Bradford Clark, Ellis Horowitz, Chris Westland, "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0*" Science Publishers, 1995.

[10] Costar tool table.

[11] COCOMO MODEL