ISSN XXXX XXXX © 2018 IJESC

Research Article Volume 8 Issue No.4

Decision Making On Resource Allocation for Strategic Planning

Sowmiya.M[1], Shamilii.D[2], Maithili.K[3]
Student[1, 2], Assistant Professor[3]
Department of CSE
Kingston Engineering College, Katpadi, Vellore, India

Abstract:
The use case point (UCP) method is one of the most commonly used size estimation methods in software development. Applicability of UCP size for the project effort estimation is thoroughly investigated; however, little attention is devoted to the effort estimation of particular task types. The authors have created and cross-compared prediction models for estimating task-type efforts by means of UCP size using an Online analytical processing model and R packages on a set of 32 real-world projects, with the goal of facilitating analysis of the correlation between project sizes and effort required to complete task types. Requirements, scoping, functional specification, and functional testing task types have up to two times better estimation accuracies than project effort. Implementation has slightly better accuracy than the project effort, while the other task types are not correlated to the UCP size. Using estimates of the most correlated task types and other techniques, such as expert judgment for others, we improved the overall project effort prediction accuracy and decreased the error from 26 to 16%.

1. INTRODUCTION

One of the most important activities in software development is effort estimation. A reliable effort estimate is crucial for planning and management of software projects. Effort estimation methods could be based on subjective expert judgment or formal estimation models. Formal models use functional dependency to estimate effort (Effort*) using some value that quantifies project size, as in (1).Effort* = f (Size)
(1)The functional dependency might be either an exact function, or an algorithm that can be used to estimate the effort based on the size.

2. UCP METHOD

In the OOAD methodology, software requirements are captured and documented with use case models and scenarios. UCP is a size metric for quantifying system size using its use case model. To calculate the size of the system, a numerical weight is assigned to each use case depending on a number of transactions in the use case scenario (i.e. 5 for less than four transactions, 10 for four to seven, and 15 for more than seven). Weights are also assigned to the users of the system based on their type (i.e. 1 for simple users who are accessing system via well-defined API, 2 for users who are using some standard protocols and 3 for human users accessing the system via a GUI) [2]. The sum of all weights of use cases and users represents an unadjusted use case point (UUCP) size, which is adjusted with a factor that represents an influence of non-functional requirements assessed using 13 technical complexity factors (TCF) and 8 environmental complexity factors (ECF). The rule for calculating the final adjusted UCP size is given in (2).

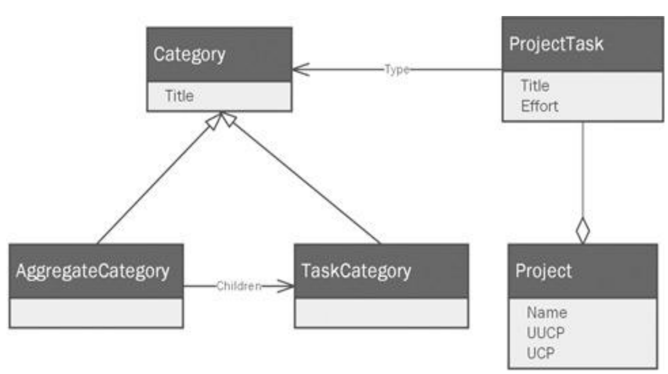
$$\mathrm{UCP} = \mathrm{UUCP} * \left(0.6 + \sum_{i=1}^{13} \mathrm{TCF}_i/100\right) * \left(1.4 - 0.03 * \sum_{i=1}^{8} \mathrm{ECF}_i\right)$$

3. OBJECTIVES

Our hypothesis is that the correlation between the UCP/UUCP size and the effort required to complete individual project task types is not uniform. The goal of our research is to evaluate how effectively these sizes can be used to estimate the effort required to complete particular project task types, identify, and differentiate task types that could be more accurately estimated using UCP size from the ones that degrade estimation accuracy.

4. DATA MODEL

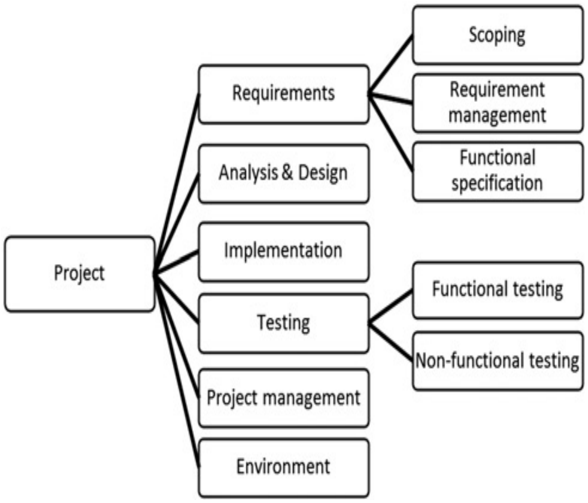
In the underlying research, we have used a set of 32 commercial projects in various domains (e.g. publishing, insurance, banking, recruitment, and educational software). All of them are standard three-tire applications based on Microsoft technologies (ASP.NET,C#,SQL Server) and have all required documentation necessary for determining the UCP size. All of them have a similar architecture and most of the effort is spent on implementation of application/database layers. Project efforts are in the range from 2 to 14 person-months. The data set is an extension of 27 projects used in our previous research [10], where additional information about the data set can be found. For analysis purposes, we have defined a model that will describe the structure to analyze, as shown in Fig. 1.



5. ANALYSIS

To evaluate applicability of the UCP method for task-type effort estimation, we have built an effort estimation model using linear

Regression based on a project data set that contains information about UCP/UUCP sizes and project efforts. Since both UUCP and UCP are valid and available measures obtained by the UCP method, we introduced both of them in our



Analysis in order to establish which of these two measures is better correlated to the efforts required for particular task types. The adjustment factors from the original UCP method have been redefined or made less significant in current research [11, 30] in order to improve the accuracy of project effort estimation. Thus, we intended to determine which of these two measures is more suitable for task-type effort estimation. We have sliced both effort estimates and actual values in our OLAP cube by task types defined in our taxonomy, built a separate linear regression model for each of them, and evaluated whether the accuracy is increased or decreased comparing to the baseline estimation model. This approach reveals for which task types we can successfully use UCP size for effort estimation, and for which task types there is degradation in effort estimation accuracy.

5.1 EVALUATION CRITERION

We have used the magnitude of relative error (MRE), as shown in (3), as an evaluation criterion to determine the accuracy of effort estimates.

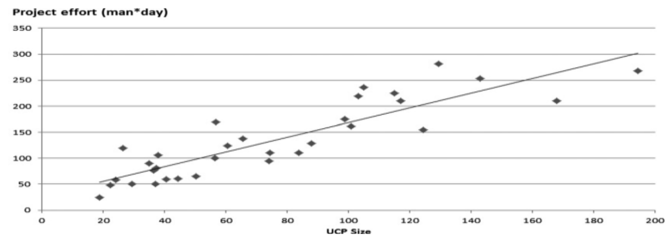
$$\mathrm{MRE} = \frac{\left|\mathrm{Effort}^{*} - \mathrm{Effort}\right|}{\mathrm{Effort}}$$

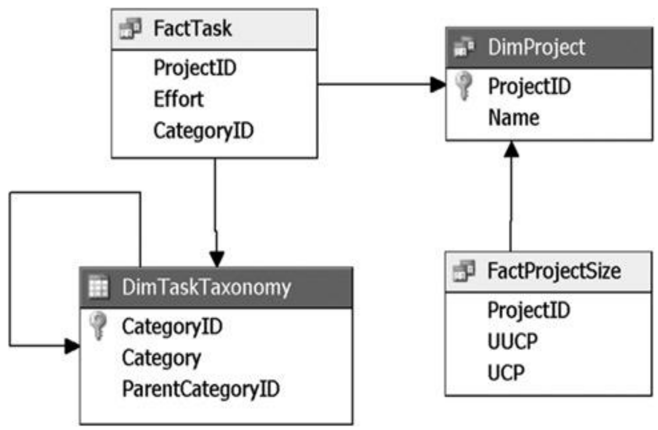
MRE compares the absolute difference between the estimated effort (Effort*) and the actual effort (Effort) to the actual value. We have also used the magnitude of error relative to the estimated value (MER), which compares the absolute difference to the estimated value instead of the actual value, and a Z-score which represent a ratio between estimated and actual value [31].To quantify an overall accuracy of the prediction model applied to our project data set, we have used the mean magnitude of relative error (MMRE) and mean magnitude of error relative (MMER),which represent average values of MRE and MER determined on the individual projects in the data set, Pred (25) that indicates how many of the predictions are within 25% of the real value, and the coefficient of determination (R2). Although these parameters are highly criticized [31–33], they are the most widely used evaluation criteria for assessing the accuracy of software prediction models

5.2 Baseline project effort estimation model

As a first step in the research, we had to check if a linear dependency, originally proposed by Karner [2], is applicable on our data set. Since UCP sizes are highly correlated to effort values (with R2 value 0.78), we have created linear regression model shown in Fig. 3. Since we could not reject null hypothesizes described in the previous section, we can confirm that linear regression model is applicable on our data set. Baseline linear model has MMRE value 26%, Pred (25) value 56%, and a p-value less than 0.001.

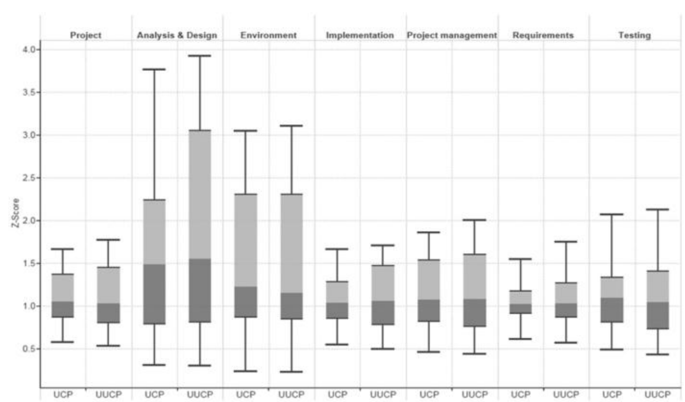
5.3 Analysis model





5.4 Results

We have created a set of MDX queries for our estimation model. This MDX query determines the R2 coefficient of linear regression between the UUCP size and task-type efforts in the fact table across all projects, as well as the mean relative errors of the estimation model.



5.5 Hybrid approach for project effort estimation

In the underlying analysis we have identified three task types that can be estimated with MMRE between 16 and 18%. Since applying UCP size to estimate the rest of the task types yields higher errors, we originated from a single company and they are relatively specific to the Microsoft technologies, which might introduce some kind of bias into the results.

```
WITH
MEMBER [Project Size] AS UCP
MEMBER Estimate AS
       LinRegPoint(
              [Project Size],
              Except(Projects, [Dim Project].[Project].CurrentMember),
              Effort, [Project Size])
MEMBER MRE AS ABS(Estimate-Effort)/Effort
SELECT
       [Task Types]*[Dim Project].[Project].CurrentMember ON ROWS, MRE ON COLUMNS
FROM UCPCube
```

5.6 VALIDATION

Our OLAP model enabled us to create a single MDX query that provides us data needed for LOOCV validation. MRE distributions for Requirements, Implementation, and Testing match the MRE distribution for the project.

6. THREATS TO VALIDITY

A threat to validity of results presented in this paper is a moderate number of projects in our data set, although it is comparable with other researchers presented in this paper.

7. REFERENCES

[1]. Boehm, B., Horowitz, E., Madachy, R., et al.: 'Software cost estimation with COCOMO II' (Prentice–Hall, 2000), ISBN-10: 0137025769

[2]. Karner, G.: 'Resource estimation for objector projects', Object. Syst., 1993

[3]. Kassab, M., Neill, C., Laplante, P.: 'State of practice in requirements engineering: contemporary data', Innov. Syst. Softw. Eng., 2014, 10, (4), pp. 235–241