A Minimization of Software Cost and Effort Estimation using Code Reusability Concept by Analogy Estimation Technique

Geetika Batra¹, Kuntal Barua² and Dr. M. K Rawat³
CSE, RGPV, LNCT, Indore M.P, India¹, CSE, RGPV, LNCT, Indore M.P, India²
CSE, RGPV, LNCT, Indore M.P, India³
batra.geetika@yahoo.com¹, kunta.lbarua@gmail.com², drmkrawat@gmail.com³

Abstract
In Software industry, reduction and accurate results for cost and effort estimation is in challenging period. The different techniques with its own pitfalls always lead to novel approach for assessment. In this paper, we presents an implementation and investigation report about an optimization of software cost and effort estimation. The proposed concept for such essential and critical task of code reusability based on analogy estimation technique with search and retrieval by using cosine similarity technique with different parameters to be measured from historic projects which are stored in code repository. This increases the precision and assist in deduction of estimation.

Keywords: Analogy Estimation Technique; Cosine Similarity Cost Estimation;Software Effort Estimation;LOC.

1. Introduction

Estimation is an important and crucial task that has its own significance in software development. It is determined by significant old projects and past experience [1]. Human effort is also one of the aspects as many cost model emphasis on this facet and give the assessment in terms of person months [2]. This value provides base for others values appropriate for software projects like cost and time taken for developing software products [3]. Accuracy plays an important role in estimation, as it is critical to both developers and consumers. Both overestimation and underestimation misleads jobs performed by a programmer [6]. Cost acuity is a stiff job to execute in field of software, estimation comprises of product size, effort needed, projects delineate and entire cost of project to be delivered [5]. There are different techniques and models used for number of reasons during software cost estimation and those are budgeting, planning of projects and improved investment analysis [4]. In this paper we include the investigation of different approaches those are suitable for navigating and searching for code blocks during implementation phase, in order to work out the costing and efforts of implementation.

2. Background

The investigations of different techniques and models for software cost estimations are classified in various ways like model- based, learning oriented, expertise based, regression based, composite and dynamics based. Through study of these classifications that consists different models and methods the conclusion is drawn, neural-net and dynamic based is less mature to use for estimation. No model is preferred over the other [4]. The models and techniques which are as lime light for estimation are Putnam & Mayer-SLIM Boehm-COCOMO, Capers Jones-Checkpoint Jensen-SEER-SEM, [7] [8].The learn of different techniques are also classified as Wideband Delphi Technique, Estimation by analogy, Top-Up approach, Bottom-Up approach, Price-To-Win, PROBE, COCOMO-II, The Planning Game [2]. Software metrics also plays an important role in assessment of cost and effort. Various software sizing techniques are mentioned in research as Line of Codes (LOC), Software Science(Halstead’s Equation),Function Point, Feature Point, Object Point [6] [8].The improvement of accuracy in estimation is centre of attraction; comparative analysis study of models and rapid development of new models and techniques proves to be supportive and innovative.

The main categories of model are algorithmic and non-algorithmic having its own potency and limitation. Selection of models and estimation technique is one aspect of accuracy. The estimates depend on effort, cost, project duration etc. Cost estimation always remains composite problem and researches should indulge to move toward with new technique for this task [9]. Some of the approaches and models which are designed are- Gray Relational Analysis with Genetic Algorithm, Ordinary Least Square Regression (OLS), Robust Regression (RoR), Ridge Regression (RiR), Multivariate Adaptive Regression Splines (MARS), Classification and Regression Tree (CART), Case Based Reasoning (CBR), Analogy weights with genetic algorithm [1]. Other advancement like use of artificial neural network can prove to be a leading idea for accurate estimation [6]. Fuzzy logic approach with a scheme of two-sided Gaussian membership function in this
technique is used for software development time estimation and shows good result [10]. The analogy based reasoning methods also establish the good reputation in new techniques of estimation used in various fields encompasses fuzzy logic [11] [12]. 2CEE (21st century Effort Estimation) is one of the models developed by JPL for NASA, based on data mining system and machine learning techniques [5]. The review concludes that 60-80% projects endure overruns of cost, effort and project schedule. The questions are answered for the sake of accuracy and lack of accurate results in evaluation, choice and combination of methods to be used. Requirement of analysis and planning is another demanding factor for accurate estimation [13]. Use of all these methods and techniques are useful for the task of estimation. Apart from these wherever there is an impact of cost the code reusability technique is useful for organization. The terms up-front investment, proportional cost and reuse density helps in cost saving [14]. The reuse perception conducts search technique and the survey results some of the methods like PageRank algorithm [20], Levenshtien distance [22] Euclidean distance [23], Cosine similarity [21]. Each method has its own advantages and disadvantages. And one of the methods can be used for searching and retrieving technique. This section leads to identify some problems faced by research in the field of software cost estimation.

3. Problem Identification

The analysis from various research and survey clarify that accurate estimation for software effort and cost is not the result of single approach but the combination of different techniques and models. The correct use of hybrid scheme leads to provide better result else can results inaccuracy in evaluation [5] [15]. Each cost estimation models and various methods used in assessment, had its own pros and cons. Survey and review cannot declare that this precise model is best in every circumstances. No model is preferable over other as in each phase different models are used therefore single model is not favorable [7]. Therefore the suggestions of using combination are required. Software estimation will always remain complex problem and new techniques will be developed day by day for better outcomes. Many factors are responsible for accurate estimation one of them plays significant role in this field i.e. software metrics used for calculations. The survey mentions different metrics concludes that feature based, object oriented, function point is enhanced than Lines of codes as review cited though function point is fine but it responsible for the overruns of time and budget. The extensively approach used by the software companies is LOC which is traditional way but easier for assessments [13] [16]. The various models are introduced and numerous techniques are described by researchers but in case of accurate results those models are not much consistent for calculations. Nevertheless most admired approach i.e. expert judgment or estimation by analogy is frequently used [11] [13] [15].

Concept of reuse of web application sounds to be good results for cost and time saving, but the other aspects can also be reused purposely [17]. The reusability of code for effort and cost estimation using historic projects code also proves for the same purpose, but the code should match and with the help of search engine the relevant code can be searched[18] [19].

4. Proposed Solution

The above problem identification leads to result that hybrid techniques and LOC as a software metrics can play a major function in estimation. The proposed solution use estimation by analogy approach with reusability concept and use of searching and retrieving techniques. The Proposed methodology describes that it contains code repository for the reusability. To support this approach analogy estimation technique is used for comparison of newer requirements from previous codes. Finally the best technique for the retrieval is chosen. The following steps can be used for calculation of cost and effort, so as to provide better results and this proposed solution becomes one way to overcome a problem that is identified in above section.

For the reuse of code it is compulsion to collect historic projects and design a code repository [19].

- Let the code repository accommodate \( NP \) number of projects; consider each project contains \( L_i \) code lines. So, overall repository contains-

\[
\text{Total Code Lines} = \sum_{i=1}^{NP} L_i
\]  

- Moreover the historic projects haves some preliminary cost which indicates old cost, at the time of deployment which is denoted by \( OC \) where, \( C = \{C_1, C_2, C_3, \ldots, C_n\} \) Thus, repository code having the total cost-

\[
\text{TCRC} = \sum_{i=1}^{NP} OC_i
\]

\[
\text{TCRC} = \text{Total Cost of repository code}
\]
Due to effect of time each project is affected by a cost (as software cost always increases with time) denoted by $T_c$ where, $T_c$={t1, t2, t3,……tn}. Thus, from equation (2)

$$\text{COST}_T = \sum_{i=0}^{N} \text{OC}_i + T_c$$

(3)

$\text{COST}_T =$ Total cost of project with time effect

- Evaluate the cost of individual code lines from equations (1), (2) and (3)

$$\text{LOC} = \frac{\sum_{i=1}^{NP} L_i}{\text{COST}_T}$$

(4)

$\text{LOC}=\text{Lines of Codes}$

- Suppose, if we have RES_u code to reuse then

$$\text{ECR} = \text{RES}_u \times \text{COSTLOC}$$

(5)

$\text{ECR}=$Estimated Cost in Reuse

- The new project which is to be developed by a software company have some estimated cost $E_c$ of that project then from equation(5) the reduced cost of project can be calculated-

$$\text{Reduced Cost} = E_c - \text{ECR}$$

(6)

The above steps are used for the calculation of estimation all the equations plays significant role; the remaining part i.e. of retrieval from the code repository is done by cosine similarity technique. For this the following algorithm is proposed:

- Declare and define a function with argument and variables to store various results in context of match of code from the repository
- Now, compare all the parameters that is to be matched for the reusability from database using if condition like: Abstract, static, access modifier method return type etc
- If the condition satisfies then increment the value of variable which is initialize by 0 i.e i=0 to i++ in every condition.
- Compare the required number of arguments and size with declared number of argument.
- If matched then again increment the value i2.
- Calculate the distance for relevant search of arguments.
  $$d=\frac{i2}{\text{required number of arguments}}$$
- Add the above result with variable i
  $$i=i+d$$
- Finally 9 parameters are emphasized while matching the new codes from the historic code and therefore d should be declare as
- After declaring d=9, i is calculated and returned with some results.
  $$i=i/d$$
- return i

As result of i is very important it shows the number of parameters matched from that of total 9 and it effects the search n retrieval techniques in terms of accuracy, memory used search time etc. The following proposed solution is against problem help for the decline of cost and effort in development of software.

5. Implementation

The proposed solution provides a way to assess the cost which supports in reduction by reusability concept. As estimation by analogy is suitable for this proposed solution as it helps in comparison of proposed project with previous projects those are similar [2] [12].According this concept and solution of the problem find in previous research the design can be projected. The proposed architecture helps in implementation of this solution as it consists of subsystem for completion of system. Various parts of this system are [19]

- System code depository: It is a gathering of different formerly developed projects with some supplementary information such as cost, numbers of functions, methods and classes which are used for development. In this, we consider java classes for estimation.
- User novel project necessity: The requirement for new project according to the condition is emphasized in this subsystem.
- Selection of similar projects from deposit: Ability to select previous codes via search techniques.
- Practical Requirement: User inputs their precise requirement like return types, access modifiers number of arguments classes etc for exploration.
- Function Definition: Search of the nearest code and functions according to the requirement and list those who actually met from repository.
- Effort and cost estimation: Cost and effort are measured as consequence of codes that is reused.
The architecture can be comprehended with the help of system flow which describes the different stages that is performed in implementation. The concept of analogy with use of LOC metrics and search technique, cosine similarity [24] proves to be very supportive for execution. The steps for work flow of proposed solution which is used in proposed architecture are [9][19]:

- Analysis of input code
- Search for all classes in source code
- Search for all member functions from source code
- Generate Matrix by developer of selected codes
- Find the desired class or member function.
- Create List of Matched Function and utilize for cost and effort reduction

These steps leads the required result and when this flow is performed for java class and methods in java programming language the following screen shots will developed which mentions that code repository is firstly developed and the new projects can get help from this particular database. The methods which are similar to the historic projects can be reused and results in the reduction of cost and effort which is the goal of research. Calculation of cost depends upon LOC; the technique which leads the completion of estimation is analogy concept, most popular among all the techniques. Search and retrieval of previous code for the new projects becomes lime light and the cosine similarity is used for this purpose.

6. Result Analysis

The implementation describes that concept of analogy estimation with cosine similarity as search technique of previous code similar to the new required code. As, while searching the code from the repository that consists of methods, functions, class etc there are some factors on which search engine works effectively. Those factors are memory used; search time, relevant accuracy and error rate.

These factors can be represented in graphical representations.

![Memory used](image1)

**Fig 1. Memory Used For Search**

This graph shows that how much memory is used while searching the required code in KB.

![Accuracy %](image2)

**Fig 2. Accuracy in search**

The graph shows accuracy how much accurate is the data which is retrieve from the repository which describes in percentage.

Error rate can also be found with the help of accuracy values. And the graph will form vice-versa. The formula for finding error rate:

\[
\text{Error rate} = 100 - \text{Accuracy}\%
\]
This graph shows the search time, according the requirement and code found in the database depends on time in milliseconds.

All the results of implementation show that the new approach of reusability with search technique proves to be successful in the estimation of cost and effort in reduced format. The analogy concept of estimation still being a success in software companies plays a significant role in this research. LOC metrics that prevents overrun in cost budget and cosine similarity which makes the research easier and leads to retrieve the require code.

**Analysis**

The following above graph describes that accuracy, search and memory used for desired code with help of cosine similarity. For analysis aspect the search metrics can be compared with another method and here cosine is compared with Euclidean distance as a match and retrieval technique. As this method depends on point representation the researchers conclude that distance can be a bad technique as compare to angle. The reason behind this concept is the shorter the document the distance works in accurate manner while for larger document it provides less accuracy [25].

The comparison can be made with the help of graphs the accuracy of both methods in case of relevant codes and search time can be compared in ms, that how much time is taken for searching the codes from code repository.

As cosine similarity is found better than Euclidean distance in case of the search engine parameters. From the following graphs it is shown.

Similarly search time can also be compared between Cosine similarities and Euclidean.

**7. Conclusions**

Due to study we observe that there are various kinds of cost and effort estimation models exists but in order to reduce the cost and effort during implementation phase there are fewer supporting model available. The proposed work is an effort in the direction of reducing cost and effort for java driven codes. The implemented model is suitable to search a piece of code in a large code repository with effect to performance. The experiment results reflect
that effectiveness of search by uses less memory and time resources with high accurate results.

In future work a new extension of proposed model is presented for enhancing the concept of agile software development technology which is more promising technique of project management.

Acknowledgments

This study has been possible thanks to the support and collaboration of Computer Science & Engineering Department, directly or indirectly contributed towards the successful completion of this research.

References


[20] Eduard Kuric and Mária Bielíková “Search in source code based on identifying popular fragments”, Institute of Informatics and Software Engineering, Slovak University of Technology in Bratislava Ilkovičova 3, 842 16 Bratislava, Slovakia, kunec, bielik@fit.stuba.sk, “unpublished”.


[22] Rishin Haldar and Debajyoti Mukhopadhyay, “Levensstein Distance Technique in Dictionary Lookup Methods: An Improved Approach”, Web Intelligence & Distributed Computing Research Lab Green Tower, C-9/1, Golf Green, Calcutta – 700095, India Email: (rishin,debajyoti.mukhopadhyay)@gmail.com, “unpublished”.
