AN OPTIMIZATION OF EFFORT AND COST ESTIMATION BY CODE REUSABILITY CONCEPT

Geetika Batra¹, Kuntal Barua²
¹(M. Tech Scholar, Computer Science Dept, L.N.C. T Indore, India)
²(Asst. Prof, Computer Science Dept L.N.C. T, Indore, India)

ABSTRACT

In software industry, a large amount of effort and cost is required to develop a solution for real world problems; additionally the rework, changes and maintenance are some factors which increase the project cost and efforts. Estimating cost and effort is a challenging and essential task. To overcome this, code reusability concept is introduced, an important aspect of object oriented programming by which piece of source code can be used in other software projects or in other time to add new functionalities with minor or no alteration. In this paper, the proposed model is based on the concept of estimation by analogy, which describes the comparison of proposed project to previously completed similar projects that resides in code repository. Innovative idea of proposed model, search and retrieval techniques, leads to achieve the result of code reusability to decrease the cost and effort estimation.

Keywords: Cost estimation, Effort estimation, LOC, Project Management, Software engineering.

1. INTRODUCTION

Estimation is a frequent occurring and most challenging task for the software industries [5]. As, software possess the characteristics of increasing its cost and effort with time, it’s a tricky task for calculation of cost and effort. Introducing the layered technology known as Software Engineering is used for development of software which includes quality focus, process, methods and tools. Development of software is a major activity which is achieved by the software development life cycle (SDLC) [13]. A structured step by step for developing the information system. It includes five phases that results to a desired product. After completion of the project, there are some escape factors of the software industries like rework, project extension maintenance. These factors lead to increase the time, complexity, cost and effort. To conquer through those factors reusability of codes and component based model techniques can be used.
Software engineering is a complex domain and requirement of cost and effort estimation is important for the software development. There are various methods through which the estimation problem can be solved. Development of Cost models like COCOMO, PUTNAM, SEER-SEM, CHECKPOINT [2][9] etc and many other methodologies were used, as it causes failure at some point which clearly concludes that each methods have their own advantages and disadvantages. Each and every model is not suitable every time. The concept of Object Oriented Programming i.e. code reusability overcomes the failure of those methods.

The searching and retrieval of the old code from the code repository can be accomplished by search engines, or some of the following techniques can be used like Cosine Similarity, Euclidean distance, Levenshtein distance. Reusability is a great concept but not all components and methods are available to reuse in the different conditions practically in real world applications, due to change in requirement or environment [3]. Additionally the component based development provides the reusability for reducing cost and effort during software development phases. But in practical there very few components that are ready to directly use in other new projects due to the system specifications and environmental variable. Thus, there is a requirement to develop a new concept that search reusable code directly from old code repository, using developer’s guidelines and provide the definition for method level codes that are directly implement on the new systems. The search in a code repository by cosine similarity [11] where there is a measure of similar two strings is done, which helps to reach our destination of searching matched codes from old repository.

This Paper includes the basic concepts of software effort estimation and project management on the basis of cost estimation, Section 2, Literature Review gives the description of the existing work and the problem description of that work. Section 3, consists of Problem Identification and its solution, Section 4 includes the Proposed System Architecture ,Section 5 includes Implementation process and lastly Conclusion & future work.

2. LITERATURE REVIEW

The different methods for estimation using different estimates model like parametric and non parametric, concludes that estimation of effort is not exact science, methods need improvements Many factors have impact on the software development process. These factors are not only technical, human, but also political and their impact can never be fully predicted. These are only not responsible as even insufficiently accurate estimates are far better than none [1]. The most common and widely used effort estimation techniques and metrics used in both function oriented and object oriented development environments concludes that complexity increases with time and which leads the problem of cost and effort. It also concludes that not only the metrics which can responsible for accurate prediction, but also it is how and when they are being used. [2]. Reuse of Web application design is already being employed in practice by the larger development organizations. The Conceptual approach to maximize design reuse in web application draws the conclusion to reuse the development part also [3].An approach called Executable Examples Archive (Exemplar) for finding highly relevant software projects from large archives of applications i.e. code reusability concept. Conclusion drawn that call of Application Programming Interface can improved the performance of search engine. As the above approach is based on the search engine [4].The overview of software effort estimation using different techniques gives many suggestions and made comparison between empirical and analogy based software effort estimation for better results. As for future work it is also suggested that data mining techniques can be applied with analogy techniques to get the accurate data set from large dataset and it will give a better results. [5]. the survey shows each methods and techniques have their own pros and cons by the time. A new approach for optimization based on fuzzy logic, analogy based reasoning and linguistic quantifiers is proposed to improve the performance of the effort in software project when they are described in either numerical or
categorical data. As fuzzy analogy can be better for the cost estimation but it may be complex for the code reusability [6]. The search and retrieval technique can be effectively implemented by Levenshtein distance which is a simple metric which can be an effective string approximation tool [12]. As other technique Cosine Similarity, is a similarity between two vectors of an inner product [11].

The study gives a clear understanding that various SDLC models when employed for developing different software then they may generate successful results owing to the fact that circumstances, resources, requirements, etc do vary for developer side as well as for client side[13]. The use of SDLC for the development is must, as the avoidance of it causes rework, never ending projects, unclear requirements, bugs etc. In one of the research paper, describes the cost factors of reusable components that divide the identifications and acquisition costs, modification costs, new development costs, integration and testing costs, infrastructure costs. Analysis also includes that not only one model, but more can be used for the code reusable estimation [15] Some of the research questions are also answered by different surveys and by this survey it shows probably used methods, where the project managers are unsuccessful to work etc the research questions are [14]:- a) To what limit, software development projects diverge from the original plan, in terms of cost, functionality, effort and schedule? b) Which methods are used to estimate software effort, and do these systematically differ in accuracy? c) How important is accurate effort estimation perceived to be, and to what extent is the level of accuracy considered a problem in the software industry? d) What are the main causes for software projects to deviate from their original plan?

3. PROBLEM IDENTIFICATION

Software engineering is a composite province, where each and every product with the same name can have different functional point of view, thus a huge amount of cost and efforts are required to deal with them. As a result of study of research articles and papers we found the various methods proposed and implemented to estimate the cost and efforts but too minor work is found under the reduction of the cost and effort of the software development.

The concept of reusability becomes the solution, but with the big problem that all methods and components cannot be used as by the time requirements and technology changes customers needs new codes and design. Moreover the component based development also provides the reusability for reducing cost and effort during software development phases, but practically there very few components are used in other projects due to the system specifications and environmental variable.

3.1 SOLUTION DOMAIN

The solution specified for the problems recognized in the real world and beyond the existing solution for reducing cost and efforts in software development is.
1. Search of similar functionality and similar code size project from the old projects, which results in similar cost and effort so as to reduce the both.
2. Search of Basic components like User defined components, Member functions and methods for the new projects which help in reducing the effort and cost.
3. If Pre-Existing Code consist of x code lines and f functions, and after code analysis, if u functions and methods are used then:

   \[
   \text{Percentage of Effort and cost reduced} = \left( \frac{u}{f} \right) \times 100
   \]

4. Code search Engine will help to search the relevant class and methods for the new projects from the code repository, so the cost and effort can be reduced.
After Implementation performance of search engine is measured over different parameters: Accuracy, Error Rate, Memory Consumed, Search Time

3.1.1 SYSTEM FLOW

The above described problem can be solved by the proposal of a new semi-automatic feedback basis code analyzer that helps to review how much code lines are reused and where the code is written in old code. This system flow mentions the steps followed by the given solution domain for the problem identification. Match the methods and classes from the old source code for the new project is a challenging task. The significance of this diagram is that the work should be done in a sequence so as to produce the desired output for that it should be completed by the following steps:

![System Flow Diagram]

4. PROPOSED SYSTEM ARCHITECTURE

The architecture proposed is a high level view for identification of the recognizable functional utilization of the system which consumes more than one processing steps to observe the user and previously written code behavior. The various subsystem involved in the proposed system is given below.

- System code repository: It is a collection of different previously developed projects and source codes with some additional information such as cost, time and numbers of functions which are used for development. In this project we consider java classes for evaluation.
- User new project requirement: A new project that is required to develop, the developer team can add different project modules and their related client information.
- Similar project from code repository: User must able to select previously written code form the source code database and analyze them for finding the reusable codes.
• Functional Requirement: User inputs their specific requirement like access modifiers; return types, number of arguments etc for search.
• Function Definition: According to the user input system search for most nearest code definitions and list those function that are met actually required by the developer.
• Cost and effort estimation: The functions found in the previous code repository the cost and effort are measured.

![Diagram](image)

**Fig 2** shows the proposed system architecture

5. IMPLEMENTATION

The proposed solution is based on the concept of cost estimation by analogy, using LOC as a size metrics. According to it, the previous projects resides in database known as Code repository, through search and retrieval technique required code can be traced from that repository. This idea leads to cost and effort estimation of developing project. The following algorithm in the form of formulas can be used for the reduction of cost using the concept of reusability. The steps are as follows.

1) Let our code repository contains \( N \) number of projects and consider each project contains \( X_i \) code lines. So, overall repository contains-

\[
TCL = \sum_{i=1}^{N} X_i \tag{1}
\]

\( TCL \) = Total Code Lines

2) On other hand, the projects have some initial cost at the time of deployment which is denoted by \( C \) where, \( C = \{C_1, C_2, C_3, \ldots C_n\} \)

Thus, repository code having the total cost-

\[
TCRP = \sum_{i=1}^{N} C_i \tag{2}
\]

\( TCRP \) = Total Cost of Repository Project
3) Due to time effect each project is affected by a cost (as software cost always increases with time) denoted by $T_f$ where, $T_f = \{t_1, t_2, t_3, \ldots, t_n\}$ 

Thus, from equation (ii) 

$$C_T = \sum_{i=0}^{N} C_i + T_f$$  

(3) 

$C_T$ = Total cost of project with time effect 

4) Evaluate the cost of individual code lines from equations (i), (ii) and (iii) 

$$\text{Lines Of Codes} = \frac{\text{Total Code Lines}}{C_T}$$  

(4) 

5) Suppose, if we have $R_u$ code to reuse them then 

$$\text{Estimated Cost in Reuse} = R_u \times \text{LOC}$$  

(5) 

By the above calculations, cost can be easily calculated for the proposed solution and implementation part can run smoothly 

6. CONCLUSION & FUTURE WORK 

Software engineering is a kind of team work for finding real world optimum solution for any existing problem. As previous source code analysis helps to improve or reduce the cost. And also make efforts to make search accurate methods and function on previously developed projects by using software reusability. The analogy based cost estimation with a relevant search code engine similarity proves to be a most advantageous and efficient method for search codes and functions, by which user are able to find most suitable codes to reduce the cost and effort estimation. As the Cost and Effort depends on many factors, apart from the class and methods reusability. Others can be used in future. Different techniques can be used for the search Engine, the future work in above implementation is to use of cosine similarity for searching technique. The search engine can be based on some more performance parameters. 

REFERENCES 


