A Novel Software Cost Estimation Technique: Inclusion of Reusability

* P. Mangayarkarasi, ** R. Selvarani

* Research Scholar, Visvesvaraya Technological University, Belgaum India
** Professor, Dept of CSE, ACED Alliance University Bangalore, India

Abstract: *Software cost estimation is critical process for software development. It is important for efficient control and management of the whole software development process. An effective reuse of design in software engineering enhances the productivity, escalate quality, and ensure reliability. Adoption of design reusability can benefit the Small and Medium Organization by curtailing production cost as well as execution period. Hence, software developers require excellent support in the assessment of the reusability levels of the software that they are trying to develop. As the Software industry grows, software quality estimation become major concerned. To meet the software quality more powerful and accurate software quality estimation techniques always required. Our prior model DyRM introduced a technique to perform modelling of design reusability under three real-time constraints. The proposed study extends the model as DyREM by incorporating cost estimation techniques using mathematical Analysis. This model analyse the object oriented software design and calculate the cost estimation at the reusability level of the software module. This method can predict the reusability of a software system at its design phase itself and provide a feedback to measure the required effort and cost to build a system*

Key Terms

Assessment, COCOMO model, Cost Estimation, Design Reusability, Function Points, Optimization, person-Month

!. Introduction

Software cost estimation is the process to measure the required effort and cost to build the system. In the project management still it is one of the most difficult tasks. There is a lack of reliable estimation technique and remains on going challenge to software developer. Software cost estimation is more challenging to IT industry because of continuous changing resources and requirements. An ideal Software Cost Estimation Model should provide ample confidence, precision and accuracy from its predictions. Software

organizations are always looking for methods in order to reduce costs, improve quality and decrease development time. Software reuse has been focus of most organizations as it can help in decreasing cost, achieving quality product and reducing time to market a product. However, introducing reuse culture in an organization is difficult because software reuse is concerned with various aspects of software development. Technical, management and business aspects of software reuse are still under investigation [2]. In software engineering software reuse has been claimed as an important source of saving costs [7]. Activities in software reuse vary from management aspects to complex development aspects, increasing its complexity and value to the software development process. Software Reuse can be an important source of avoiding costs and to increase savings for any software development organization.

Software reusability is an attribute that refers to the expected reuse potential of a software component [1]. Software reuse not only improves productivity but also has a positive impact on the quality and maintainability of software products. Reuse can be defined as using what already exists to achieve what is desired. Reuse can be achieved with no special language, paradigm, library, operating system, or techniques. It has been practiced for many years in many different contexts. Software reuse is a technique that can have a positive impact on software engineering practice in many environments.

Reuse at earlier levels of the software life cycle provides the maximum potential for cost saving, because larger portions of the software engineering effort can be either eliminated or done more quickly, thus providing reductions in the cost of software life cycle activities. The term "software reuse" suggests that there is an archive of reusable software artifacts that can be used as part of the development of new systems. In the most common view of software reuse, the software design is considered as high level artifacts. Nowadays, most software engineering design is based on reuse of existing system or components. The level of economic benefits can be estimated by properly using metrics of reusability and return on

investment frameworks [4]. According to Ouyang [20], software reusability can be considered from two viewpoints: design-by-reuse and design for-reuse. Software-by-reuse is the use of existing application or its components to build new applications. Software-for reuse is the ability of building applications that can be used all or part of it in other applications.

2. Background

Nair and Selvarani [11] proposed a framework with the capability to compute the reusability index. The authors considered three of the Chidamber and Kemerer metrics viz. DIT (Depth of Inheritance Tree), RFC (Response for a Class) and WMC (Weighted Methods per Class). They exposed the strong relationship that exists between the design parameters and reusability factors in developing a reusability estimation model. Gill [14] highlighted the pertinent issues of software reusability for component based development on the basis of CBSE, highlighted the important issues of software reusability and high level reusability guidelines. Also the author outlined the aspects of reusability from product reliability improvement and reduction in software development costs. Fetaji et al. [15] and Singh et al. [16] carried out empirical investigation towards improving reusability as well as software consistency on the object-oriented design components. Selvarani [13] presented an empirical evaluation of the Chidamber and Kemerer metrics for assessing prediction capability using data driven techniques for mitigating defects in software. The author has carried out the investigation considering Weighted Methods per Class mainly.

Capers Jones [17] on 2D structure of projects which define activities carried on various projects, the paper also calculated through industrial survey, theoretical analysis and total number of the activities. After this the author describes the areas on which the estimations are carried out and mentions the groups which are involved in the software development. Nair and Selvarani [12] carried out a complete analysis of the relationships that exist between internal quality attributes in terms of the complete suite of Chidamber and Kemerer metrics and the reusability index of software systems. The authors presented a new regression technique for the purpose of mapping the association between reusability and design metrics. Jasmine K.S and Vasantha [22] states that reuse density play an important role in estimating cost saving due to reuse. They also suggested that software reuse is only relevant when it has positive economic impacts on organisation and reuse metrics. Ali Arshad et al [23] Carried out a survey in the companies and proposed that

accuracy of cost estimation mostly relay on model based estimation method and expert judgement. They also concluded that cost estimation focuses to be on industries, project and product. Shobha Rani Malik and Dr. Saba Hilal [24] approached enhanced and simplified cost estimation to be followed for all stages of software development and also specified that accuracy of cost estimation depends on all the attributes of Application.

3. Problem Definition

Smaller and Medium Enterprises (SME) doing business in software project development are backed up by smaller numbers of skilled human resources. For better revenue-making strategies, such SMEs very often adopt the strategical means of optimizing their limited human resources to handle high numbers of software projects. Usually, such an organization can scale up profit if it undertakes business contracts of similar types of products to the maximum number of clients. Hence, the organization focuses on adopting design-reusability for catering to the client's requirements with a limited availability of human resources. The more design reusability, the greater the profit of SMEs. [18] Hence better optimization of human resources can be done if one developer can reuse the code or work on a new design with higher degree of reusability to a maximum extent.

In order to understand the problem scenario, let us consider the criticality of the considered constraints for the proposed system with respect to three different constraints such as human resource, cost, and work schedule.

Human Resource: Smaller organization has smaller number of human resources who are always endowed with unproportionate work load. Although, there is always a breakpoint for the capability of the human resources, but it is not possible to identify the variable capacity of multiple human resources. For an example, for a give same job, one employee can have more efficiency as others. Hence, modelling human resource as a constraint is one of the challenging tasks if considered along with design reusability concept [18].

Work Schedule: There is always constant work timing for all the human resources. However, the challenge is how much of the same work schedule can be used by an employee to design a new component with maximum design reusability? There may be multiple unseen factors that may have positive or negative influence on the employee's productivity (in perspective of designing reusable components) in static work schedule [18].

Cost: The term cost will mean amount of resources required by an employee to develop a new software project with maximum reusable components. For better margin of profit, cost of new development must be as low as possible. Although, cost cannot be lowered down to zero, but it is challenging to perform predictive modelling to foresee some approximated value of cost of new development of upcoming software project [18].

4. Proposed System DyREM – Dynamic Reusability Estimation Model

The proposed formal model called Dynamic Reusability Estimation Model (DyRM) is to calculate primary cost, effort and work schedule to develop a Reusability software system. Our model estimates application by considering parameters like reusability, time duration, human effort and cost. This model is flexible because estimated cost varies with change of requirement. The empirical experiment is conducted in a controlled research environment using the SPSS tool to analyze the reusability. Figure 1 shows the architecture of the Dynamic Reusability Estimation Model. The threshold factor plays a critical role in this analysis, which is limited to 70 percent, that means if the existing development requires less than 70 percent of reuse the new design is highly cost effective. If it is more than 70 percent, it can be considered equivalent to new development. The efficiency of the proposed model is analysed by considering the skills of analysation, qualification

and adoption of the reusable design components based on the threshold value.

The domain model helps in analysing user requirements with respect reusability and also analysis whether the reusable component can be used as itself or with modifications. Software architecture provides input for the design of the application and also provides general patterns and the context for interpreting the detailed level. Structural model gives `the view of the system with respect to entities, attributes and their relationships. It mainly focuses on the architectural style. The analysis and architectural design focus on reusable component development.

The analysis model is used to analyse and determine any elements of the model that point to existing reusable components. It mainly focuses in extracting information from the requirements model in a form that lead to specification matching. If specification matching yields components fit in the needs of the new application the design model can extract the matched components from a reusable repository and use them in the design of new systems. If design components cannot be found the software designer create them from the scratch. When the designers create a new component it should be considered as design for reuse. The cost estimation of application with the reusable components are described in the next section.
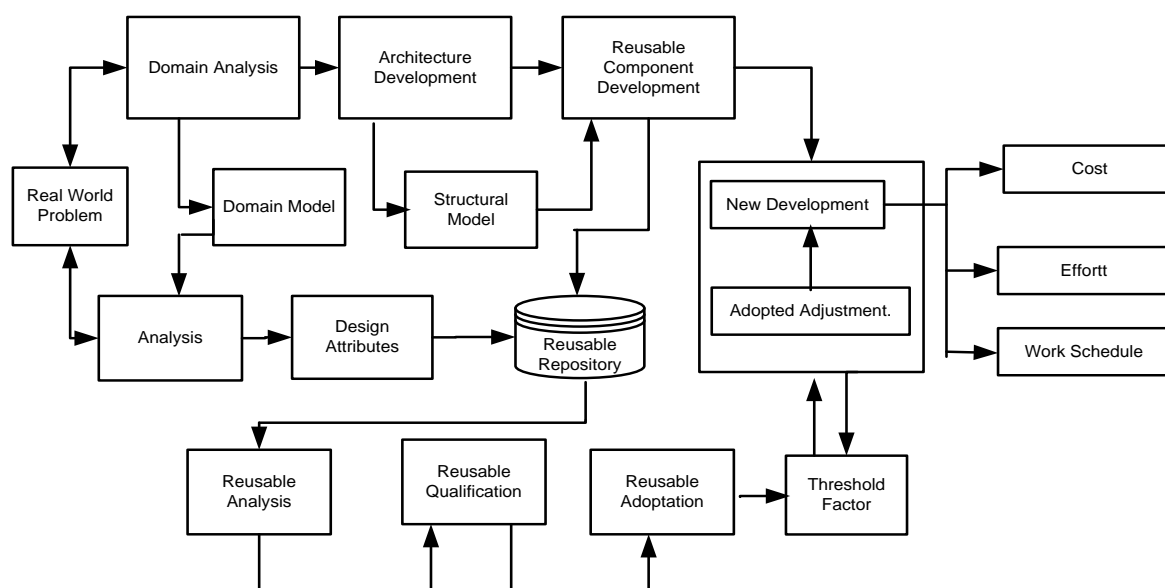

Fig 1: Dynamic Reusability Estimation Model

5. Software Reuse Cost Factors

Economic considerations are the centre for any discussion of software reuse. Further, different technical approaches for reuse have different investment and return on investment profiles [5][6]. Economic models and software metrics are needed to quantify the costs and benefits of reuse.

In this work, Function Point Analysis (FPA) and Constructive Cost Model (COCOMO - Barry W. Boehm) have been used to estimate software project cost with the case study of small organic mode which covers relatively small and simple five software projects conducted by small teams with good application work experience for a set of less than rigid requirements as a Supply Chain Management   , Customer-Relationship Management and Enterprise Relationship Management projects. The type of cost with reuse in the context of estimation as follows [19]:

a. Reusable Assessment, Identification and Selection Effort Cost -> RAISE

The effect of Assessment for finding reusable components that matches with customer requirements. Identification of finding the component that fit into the architectural style specified for the system and Selection of reusable components with white box, gray box and black box wrapping.

b. Reuse without Modification Cost ->RWOM

The effects of the reuse factors of the requirements transported without any modifications, consider as a block box transfer without understanding the structure and making changes to it.

$$Cost_{RWOM} = \sum_{i=1}^{n} (RSLOC * (1 - \% reuse / 100)) / RPROD$$
------------------------------- (1)

RSLOC – Reusable Source Line of Code

RPROD – Reusable Productivity

c. Reuse with Modification Cost-> RWM

The effects of the reuse factors of the requirements transported with adapted minor modifications, consider as a white box transfer with some development effort.

$$Cost_{RWM} = \sum_{i=1}^{n} (RASLOC * (1 - \% reuse\_atopted / 100)) / RAPROD$$
------------------ (2)

RASLOC – Reusable Adopted Source Line of Code

RAPROD – Reusable Adopted Productivity

d. Auto translation Cost -> AT

The effects of automatically generated code form the system model. This is the form of reuse where standard templates are embedded in the generator.

$$Cost_{Auto} = \sum_{i=1}^{n} ((ATSLOC * AT / 100) / ATROD)$$
----------------------- (3)

ATSLOC – Automatic Translation Source Line of Code

AT – Percentage of Automatically generated Code

ATPROD - Automatic Translation Productivity

e. New Development Cost->ND

The entire components cannot be reused as it is, based on the threshold value 30% of the code needs to be written. The new development should consider their work as reusable product in future while writing the new work form the scratch.

$$Cost_{ND} = \sum_{i=1}^{n} (a[size(NSLOC(1 + BRAK / 100)]^{b} * EM$$
------------------------------ (4)

NSLOC – New Source Line of Code

BRAK – Percentage Code Discarded due to Requirement Volatility

EM – Effort Multiplier

f. Overhead Cost->OC

Overhead Costs associated with reuse includes the Domain Analysis, data gathering, Reporting, Maintenance of documents and components, conservation of Repository and Training facilities.

Hence the overall cost estimation for the application is derived by combining the calculated equations form 1 to 4 as

$$Cost = Cost_{ND} + \prod RAISE[Cost_{RWOM} + Cost_{RWM} + Cost_{Auto}] + OC$$

6. Result Discussion

This section discusses about the outcomes accomplished from the proposed study. The result discussion is carried out in two stages. The first stage of the result discussion is carried out with respect to total amount of reusability factor found from the existing constraints, whereas the second stage of the result discussion is carried out with respect to cost estimation including the cost of new development in this model. For this purpose, we consider simple five software projects with higher number of modules developed in Java. The details of the projects and industry are not mentioned explicitly as per the Non-Disclosure Agreement. The overall numerical outcome is just an approximation of 5 such types of projects (2 projects in supply chain, 2 projects in customer-relationship management, and 1 project in enterprise relationship management). All the five projects have been developed with the two possibilities of with reusability and without reusability.
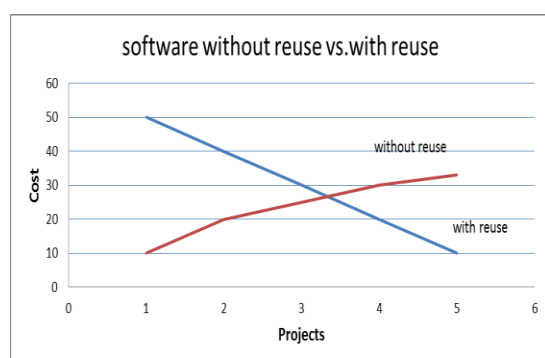

Fig 2: Cumulative costs of software systems without reuse vs. with reuse

Fig 2 illustrates that for a reuse oriented software development; there will be an initial cost increase. Then gradually cost will decrease due to reusing the same component again and again across similar products.

Conclusion

Software reuse is relevant when it has positive economic impacts in organizations and its reuse metrics. The proposed system introduces a formal model DyREM that evaluates the challenging scenario of SMEs related to design-reusability. It will evaluate the change in reusability requirements in terms of effort, work schedule and new cost. In this model, the design attributes build a base to establish a relationship between the above said parameters and estimation cost with respect to design reusability. In the first phase of DyRM the reusability components are analysed, identified and adopted. In the second phase cost of the projects are estimated by considering two aspects of components with reusability and without reusability. The model shows that reuse density (the ratio of number artefacts reused to the size of the system) plays an important role in estimating cost.

References

[1] Selvarani R, Nair TRG (2009) Software Reusability Estimation Model Using Metrics Governing Design Architecture, International Book: Knowledge Engineering for Software Development Cycles: Support Technologies and Applications. Engineering Science Reference, IGI Publishing, USA, DOI: 10.4018/978-1-60960-509-4.ch011

[2] Kevin D. Wentzel. Software reuse - facts and myths. In Proceedings of the 16th international conference on Software engineering, ICSE '94, pages 267{268, Los lamitos, CA, USA, 1994. IEEE Computer Society Press.

[3] Paul Goodman, Practical Implementation of Software Metrics, McGraw Hill, London, 1993.

[4] B. Clark.2002. Eight Secrets of Software Measurement, IEEE Software, vol.19. Iss.5., pp. 12-14

[5] J. Poulin and J. Caruso, "A Reuse Metrics and Return on Investment Model", Proceedings of the 2nd Workshop on Software Reuse: Advances in Software Reusability, IEEE 1993.

[6] W. Frakes and C. Terry, "Reuse Level Metrics", Proceedings of the 3rd International Conference on Software Reuse: Advances in Software Reusability, IEEE, 1994.

[7] Barry W. Boehm. Software Engineering Economics. Prentice Hall, Engle-wood Cli_s, NJ, 1981.

[8] Roger S. Pressman, Software Engineering-A practitioner's Approach, Fifth edition, McGraw-Hill, 2001.

[9] T.R.GNair and R. Selvarani. 2010. Estimation of Software Reusability: An Engineering Approach, Association for computing Machinery (ACM) – SIGSOFT, USA, Vol.35, Iss.1

[10] Ivan Jacobson, Martin Griss and Patrik Jonsson, Software Reuse- Architecture, Process and Organization for Business Success, ACM Press, 2000.

[11] Jorge Cláudio Cordeiro Pires Mascena, Eduardo Santana de Almeida, Sílvio Romero de Lemos Meira, " A Comparative Study on Software Reuse Metrics and Economic Models from a Traceability Perspective", IEEE, 2005.

[12] V.R. Basili, L. C. Briand, W. M. Thomas, "Domain Analysis for the Reuse of Software Development Experiences", Proceedings of the 19th Annual Software Engineering Workshop, NASA/Goddard Space Flight Center, November 1994.

[13] Lim, W. C. (1994). Effects of reuse on quality, productivity, and economics. IEEE Software, 11(5), 23-30.

[14] Gill, N. S. Reusability Issues in Component-Based Development, ACM SIGSOFT Software Engineering Notes, 28(4): 4 – 4 (2003).

[15] B. Fetaji, N.Reci, & M.Fetaji.(Retrived 2015). Analysing and Devising a Model for Trust-worthy Software. Recent Advances in Electrical and Computer Engineering

[16] P. K. Singh, O.P. Sangwan, A. P. Singh and A. Pratap. (2014). A Quantitative Evaluation of Reusability for Aspect Oriented Softwareusing Multi-criteria Decision Making

Approach. World Applied Sciences Journal, Vol. 30 (12), pp.1966-1976

[17] Capers Jones, "Software Cost Estimation in 2002", The Journal of Defense Software Engineering, June 2002.

[18] R. Selvarani and P. Mangayarkarasi, "A Dynamic Optimization Technique for Redesigning OO Software for Reusability", SIGSOFT Softw. Eng. Notes, Vol. 40, 2, pp.1-6, 2015

[19] B.H.Barnes and T.B. Bollinger, " Making reuse cost-effective", IEEE software, Vol. 8,No.1, pp.13-24, Jan 1991.

[20] Y. Ouyang and D. L. Carver, "Enhancing design reusability by clustering specifications," presented at
the Proceedings of the 1996 ACM symposium on Applied Computing, Philadelphia, Pennsylvania, United States, 1996.

[21] Jorge Cláudio Cordeiro Pires Mascena, Eduardo Santana de Almeida, Sílvio Romero de Lemos Meira, " A Comparative Study on Software Reuse Metrics and Economic Models from a Traceability Perspective", IEEE, 2005.

[22] **Jasmine K.S and Dr.R.Vasantha** "Cost Estimation Model For Reuse Based Software Products", Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I IMECS 2008, 19-21 March, 2008, Hong Kong

[23] Ali Arshad, et al, "A Survey on Software Cost Estimation in the Pakistani Software Industry", IJCER, Volume 3, Issue 1, February 2014

[24] Shobha Rani Malik and Dr. Saba Hilal "A Cost Estimation Model For Reuse Based Software Program", IOSRJCE, ISSN: 2278-0661 Volume 4, Issue 6 (Sep.-Oct. 2012), PP 06-12, www.iosrjournals.org

[25] S. Henninger, "An Evolutionary Approach to Constructing Effective Software Reuse Repositories", ACM Transactions on Software Engineering and Methodology, Vol. 6, No. 2, April 1997, pp. 111-140.

[26] Hafedh Mili, Fatma Mili, and Ali Mili, "Reusing Software: issues and research Directions", Proc. IEEE trans. software Engineering, Vol 21, No.6, June 1995.

[27] W. Lim, "Effects of Reuse on Quality, Productivity, and Economics", IEEE Software, Vol. 11, No. 5,September 1994.

[28] Shaw, M. (1995), "Cost and Effort Estimation", CPSC451 Lecture Notes. The University of Calgary.

[29] Leach, Ronald J. "Software Reuse: Methods, Models, Costs", second edition, 2012,p.661 cm, ISBN 13:9781939142351

[30] Roger S. Pressman, Software Engineering-A practitioner's Approach, Fifth edition, McGraw-Hill, 2001.

[31] J. Guo, Luqui, "A Survey of Software Reuse Repositories", 7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems, April 2000, pp. 92-100.

[32] Walker Royce, "Software Project management-unified Frame work" Pearson.First edition 2006,ISBN : 978-81-7758-378-6
[33] Sommerville, "software Engineering", Seventh Edition, pearson Publication 2004
[34] CH. F. KEMERER, An empirical validation of software cost estimation models. Communications of the ACM, 30(5) (May 1987), 416–429.