# Cost Estimation Model For Reuse Based Software Products

Jasmine K.S, Dr. R. Vasantha

*Abstract*— **A fundamental task when employing software reuse is evaluating its impacts by measuring the relation of reused and developed software, the cost for obtaining reuse and the cost avoided by reusing software during development and maintenance. Different reuse related metrics exist in the literature, varying from strictly code-based metrics, aiming to measure the amount of code reused in a product, to more elaborate cost-based metrics and models, aiming to measure the costs involved in reuse programs and to evaluate the impacts of reuse in software development. The goal of this work is to evaluate the current state of the art on the reuse metrics area, with a special emphasis on reuse cost metrics and propose an estimation model for development cost by surveying hundred software companies.**

***Index Terms*—Software Reuse, Reuse Survey, Reuse metrics, Reuse cost.**

## I. INTRODUCTION

Software metrics play a very important role in software development and management [1]. Software reuse programs rely on reuse metrics, along with other software, quality and effort metrics, to evaluate their effectiveness and the relations between the different employed metrics in organizations [2] [4] [5]. The systematic collection of various metrics during the whole software development lifecycle helps determining which practices should be enforced and which ones should be avoided for productivity and quality improvements in software development process [17]. Component-based reuse is widely accepted as an important reuse strategy and component-based reuse programs heavily depend on software reuse repositories for achieving success [14] [15] [16], however the main focus on the reuse repository area is on classification and retrieval problems.

Despite the fundamental importance of reuse metrics, few reports of their systematic adoption in the software industry exist. This work aims to evaluate the current state of the art of the reuse metrics area, with special emphasis on metrics to estimate cost.

Jasmine K.S is with RVCE, Visveswaraya Technological University, Sr.Lecturer, Dept of MCA, Bangalore, India (phone: +919342969571; e-mail: jasminesadeep@yahoo.co.in).

Dr. R. Vasantha is with RVCE, visveswaraya Technological University, Prof, Dept of ISE, Bangalore, India (e-mail: vasanthaprak@yahoo.com).

### A. Software Reuse

Software reuse is the use of existing assets in some form within the software product development process. More than just code, assets are products and by-products of the software development life cycle and include software components, test suites, designs and documentation. Leverage is modifying existing assets as needed to meet specific system requirements. Because reuse implies the creation of a separately maintained version of the assets, it is preferred over leverage.

Reuse can be achieved through different modes. Compositional reuse involves constructing new software products by assembling existing reusable assets, while generative reuse involves the use of application generators to build new applications from high level descriptions [4].

Leveraging involves the modification of previously developed software for a new product. When managed correctly, leveraging can be advantageous over creating software from scratch in that it requires less time and effort. On the other hand, because it is difficult to correctly predict the impact of modifications, leveraging may result in lower quality software. It can also lead to multiple versions of a software component and consequently, an increased maintenance burden. In this situation, metrics can play an important role in decision-making.

### B. Software Metrics

Software metrics are an integral part of the state-of the-practice in software engineering. Companies are using metrics to better understand, track, control and predict software projects, processes and products.

According to Goodman, software metrics as, "The continuous application of measurement-based techniques to the software development process and its products to supply meaningful and timely management information, together with the use of those techniques to improve that process and its products" [13].

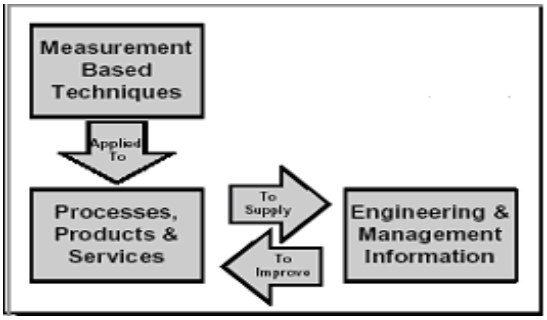


Fig1: What are Software Metrics?

Fig 1 illustrates that metrics can provide the information needed by engineers for technical decisions as well as information required by management.

### C. *Related Work*

Economic considerations are at the center of any discussion of software reuse. Further, different technical approaches to reuse have different investment and return on investment profiles [1][2]. Economic models and software metrics are needed that quantify the costs and benefits of reuse. Only recently the researchers started to tackle this problem [3][11][12]. But even such studies couldn't help to convince the management to understand the advantage of reuse. Most existing software engineering economic models need to be customized to each specific reuse business. Several authors have modified the cost models that are today used to estimate time and effort and for the development both of components and of applications using components [7][8][9]. Because high levels of reuse can reduce the overall cost and time to deliver applications, the extra funding and time can be directed to several alternative projects. In this paper, we discuss the implications of various approaches for software reuse in the organizations and propose an economic model for cost analysis.

## II. IMPLEMENTATION

### A. *Measuring Software Reuse Cost*

Models for software reuse economics try to help us answer the question, " when is it worthwhile to incorporate reusable components into a development and when is custom development without reuse preferable?"[19]. Generally metrics can be categorized into two namely product metrics, which determine the characteristics of components and process metrics, which measure time, cost etc.

### B. *Input Material And Data Collection*

As a first step, metrics that are weak or irrelevant to reuse are eliminated by comparing and contrasting software reuse lessons learned from literature search and utilizing the results of previous software reuse empirical studies and extracted the dominant themes apparent about reuse and identified a number of reuse practice metrics for inclusion in the survey instrument [4].

### C. *Population And Sample*

Software reuse is still an immature function. There is no census bureau representing a population of software development organizations who practice software reuse. Software intensive organizations are considered as target population for this study. This population includes companies of different sizes (in terms of No. of engineers / programmers / testers), experiences and nature. A total of 100 software organizations were responded to the survey, which includes product and service oriented companies. Table 1 shows the characteristics of the survey sample.

Table 1: Characteristics of population sample

| Nature of work | | Size of organization | | Years of experience (Years) |
|---|---|---|---|---|
| Service Oriented | Product | No. of Engineers/ Testers/ Programmers | Ratio of Developers/ Testers | |
| 80 % | 20% | 15 - 45000 | 12:5 to 5:1 | 3 to 36 |

### D. *Software Reuse Cost Estimation Models*

We can categorize the type of reuse in the context of cost estimation as follows [11]:

i) Component Reuse without Modification

ii) Component Reuse with modification

In the case of component reuse without modification, the average cost of developing using reusable components can be formulated as follows:

$$\text{Cost}_{\text{search}} + (1-p)*\text{Development}_{\text{no-reuse}} \quad (1)$$

Where $\text{Cost}_{\text{search}}$ is the cost of performing a search operation, $\text{Development}_{\text{no-reuse}}$ is the cost of developing without reuse (i.e., the cost of developing the component from scratch) and p is the probability that the component is found in the component library. It is observed that the reuse option would be preferable only if:

$$\text{Cost}_{\text{search}} + (1-p)*\text{Development}_{\text{no-reuse}} < \text{Development}_{\text{no-reuse}}$$

In the case of component reuse with modification, the average cost of developing using reusable components can be formulated as follows:

$$\text{Cost}_{\text{search}} + \text{Cost}_{\text{adapt}} + (1-p)*\text{Development}_{\text{no-reuse}} \quad (2)$$

Where $\text{Cost}_{\text{search}}$ is the cost of performing a search operation (the cost depends on the whether the search is a manual search or search using a search tool), $\text{Cost}_{\text{adapt}}$ is the cost required to adapt the component, $\text{Development}_{\text{no-reuse}}$ and p means the same as in the case of equation (1). It is observed that the reuse option would be preferable only if:

$$\text{Cost}_{\text{search}} + \text{Cost}_{\text{adapt}} + (1-p)*\text{Development}_{\text{no-reuse}}$$

< Development $_{no\text{-}reuse}$

In both the cases,

The cost saving due to reuse can be formulated using a simple equation:

$$Cost_{saved} = Cost_{no\text{-}reuse} - cost_{reuse} \quad (3)$$

In addition to the above costs, we should also consider some overhead costs associated with reuse include:

- Domain analysis [6][10]
- Increased documentation to facilitate reuse
- Maintenance and enhancement of reuse artifacts (documents and components)
- Royalties and licenses for externally acquired components
- Creation (or acquisition) and operation of a reuse repository (If the decision is to build a reusable component, then the cost of initial development and also the expected usage frequency of the component also should be considered [12]).
- Training of personnel in design for reuses and designs with reuse.

To maximize reuse profits, by analyzing process, organizational and technical aspects with reduced asset development cost and management cost, We have to consider the following points

i) When and Where Capital investment is to be made

Two approaches were observed namely, proactive and reactive. 80% of identified population supported proactive approach and 20% supported reactive approach. If the domain is stable, where the product features can be predicted, organizations can go for upfront investment to develop reusable assets (proactive approach). If the domain is unstable, reusable assets can be developed as when required (reactive approach). This approach may result in reengineering and retrofitting existing products with reusable assets, if there is no common architectural basis.

ii) Whether to go for a dedicated team for development/distribute/maintain assets or not and associated costs involved.

Again the Cost no-reuse and cost reuse depends on the % size of parts (components) reused and % of parts (components) not reused.

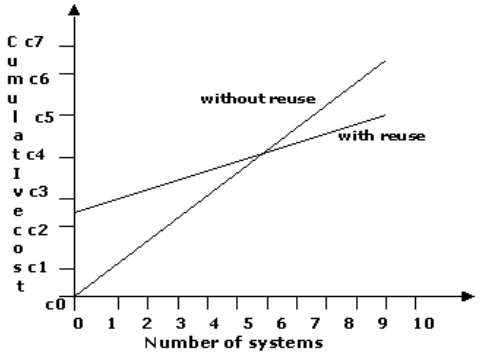


Fig 2: Cumulative costs of software systems without reuse vs. with reuse

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.

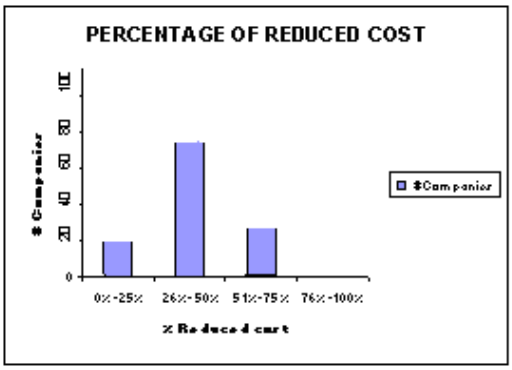


Fig 3: Percentage of reduced cost of software products due to reuse

Fig 3 illustrates the percentage of reduced cost in the surveyed software companies. There are 20 companies, their % of cost reduction due to reuse is between 0% and 25% and there are 70 companies, their % of cost reduction due to reuse is between 26% and 50% and there are 30 companies, their % of cost reduction due to reuse is between 51% and 75%. Also there are no companies they are completely saved their development cost due to reuse.

## III. Conclusion

Software reuse is only relevant when it has positive economical impacts in organizations and reuse metrics and models are fundamental in assessing these impacts. The study, which proposes a cost estimation model in the reuse context, suggests the cost associated with different stages of software development. The study shows that there is an up-front investment and proportional costs (cost per reuse). Also it is found that reuse density (the ratio of number parts reused to the size of the system) play an important role in estimating cost savings due to reuse. An extensive work relating more elaborate software structure reuse metrics and economical factors, with a wide variety of representative set of industrial organizations (i.e., software organizations only involved in development, only involved in maintenance and involved both in development and maintenance) is still necessary to standardize the reuse cost metrics field.

## Acknowledgment

This study has been possible thanks to the support and collaboration of several industry people they are directly or indirectly contributed towards the successful completion of this research.

## References

[1] 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.

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

[3] W. Frakes and C. Terry, "Software Reuse: Metrics and Models", ACM Computing Surveys, Vol. 28, No. 2, June 1996.

[4] E. Henry and B. Faller, "Large-Scale Industrial Reuse to Reduce Cost end Cycle Time", IEEE Software, Vol. 12, No. 5, September 1995.

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

[6] D.M. Balda and D.A Gustafson, " Cost-estimation models for the reuse and prototype software development", ACM SIGSOFT, pp. 42-50,July 1990.

[7] J. Poulin, "Measuring Software Reusability", Proceedings of the 3rd International Conference on Software Reuse: Advances in Software Reusability, IEEE, 1994.

[8] Boehm B.W and Papaccio P.N. "Understanding and controlling software costs", IEEE transactions on software engineering, 14(10), 1462-77

[9] Malan R. and Wentzel K. "Economics of software reuse revisited", Proc. 3rd Irvine Software Symposium, University of California, Irvine, 30 April, pp.109-21.

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

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

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

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

[14] 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.

[15] R. D. Banker, R. J. Kauffman, D. Zweig, "Repository Evaluation of Software Reuse", IEEE Transactions on Software Engineering, Vol. 19, No. 4, April 1993, pp. 379-389.

[16] 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.

[17] 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.

[18] J. A. Favaro, Comparison of Approaches to Reuse Investment Analysis. In Proceedings of the Fourth International Conference on Software Reuse, IEEE Computer Society Press, Los Alamitos, CA, 1996, pp.136-145.

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

[20] 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.

**Jasmine K.S** born in the Ernakulam District Of Kerala state on October 14th in the year 1971. She received BSc degree in Mathematics from Mahatma Gandhi University, Kerala in 1991, MSc degree in computer science from Kerala University, Kerala in 1994 and M.Phil degree in computer science from Bharathidasan University, Tamilnadu in 2005. She is currently doing her PhD in computer science in Mother Teresa University, Kodaikanal, Tamilnadu.

She is a Senior lecturer in the Department of MCA, R.V.College of Engineering, Bangalore. Since from 1995, she is working as a lecturer in the field of computer science. During 98-99,She held a visiting faculty position at Visveswarapura College of science, Bangalore. She has authored 12 research papers in the national and international level conferences. Her research interests include Software reuse, Software performance, Software testing, data mining and experimental software engineering.

Ms. Jasmine is the member of Indian society for technical education (ISTE), Computer society of India (CSI) and International Society for computer applications (ISCA).

**Dr.R.Vasantha** received BSc degree, majored in Physiscs, Chemistry and Mathematics from University of Mysore, India in 1976, MSc degree in Mathematics from Manasa Gangotri, University of Mysore, India in 1978 and PhD from Indian Institute of Science, Bangalore, India in 1985.

She is a professor in the Department of Information science and Engineering, R.V.College of Engineering, Bangalore. She got more than 20 years of research experience. During Oct.1991-Oct.1994, she worked as a Scientist in National Aeronautical Laboratory, Bangalore, India, on Turbulence modeling of aerofoil.Sept1988-Sept.1991: Worked as Research Associate in the Dept. of Mechanical Engineering, University of New South Wales & University of Sydney, Sydney, Australia on turbulence modeling. Oct.1987-Sept.1988: Worked as a Senior Research Associate in the School of Mathematics, University of East Anglia, Norwich, England, on the initiation of detonation waves, Oct.1986-Oct.1987: Worked as a Senior Visiting Fellow in the School of Mathematics, University of East Anglia, Norwich, England. Aug.1985-Aug.1986: Worked as Research Associate in Dept. of Aerospace Engineering, IISc, Bangalore, India. She also has many years of teaching experience. During 1978-1980, she worked as a Lecturer, 1994-2002: Worked as Associate Professor of Mathematics, 2002-2006: Worked as Professor & HOD of Mathematics in an Engineering College in India. Her research interests are in the field of computational fluid mechanics. She is the author and coauthor of several publications appearing in international journals, books, and conference proceedings in the fields of Applied Mathematics and Computational fluid mechanics.

Dr. Vasantha is a Gold medal contender for PhD dissertation. And also she Won Scholarship from B.Sc. till the end of Ph.D.