IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661 Volume 4, Issue 6 (Sep.-Oct. 2012), PP 06-12
www.iosrjournals.org

A Cost Estimation Model For Reuse Based Software Program

Shobha Rani Malik[1] , Dr. Saba Hilal[2]
[1](Lecturer, Technical Edu. Deptt. Haryana and Research Scholar at Lingayas' University Faridabad, India)
[2](Research Guide, Lingayas' University Faridabad, India)

***Abstract**: Reuse reduces total cost of product but it has its value. As per reuse principle, it is not for free, a reuse program requires a lot of set-up investment as well as additional costs for development and operational activities but benefits are neither linear nor immediate. The aim of this paper is to evaluate the current state of art and practice on cost estimation for a software reuse oriented program by extending the pioneering contribution of other researchers in this direction. A simplified software reuse cost estimation model which presents where, who, how capital investment is done for a reuse program is suggested with various cost effecting factors.*

***Keywords** - Cost effecting Factors, Cost Metrics, Software Reuse, Reuse Cost*

I. Introduction

It is accepted that to develop a reusable asset is more costly than equivalent custom tailored [1]. Reuse requires extra cost due to generalization, documentation, additional testing, library support , maintenance , architecture, design and implementation [2][3]. Software reuse have the same cost and risk features as any financial investment [4].

1.1 Prerequisites for cost

An organization have to analyze following factors related to cost before implementing a reuse program.

- Cost to implement a reuse program.
- Investment approach
- Cost to sustain the reuse program.
- Time to take to break even reuse investment.
- Expected benefits

1.2 Investment approaches

There are three investment approaches for software reuse process:

- Proactive Investment

This approach tends to require a large initial investment and is preferred in stable domain [5]. Returns on investment can only be seen when products are developed and maintained .

- Reactive Investment

This is an incremental approach to build reusable components generally used in reengineering and is preferred in unstable domain [5] .

- Extractive Investment

Combination of Proactive and Reactive approach .

1.3 Cost Factors for Reuse

- Management

This cost is spent to manage a reuse program. Reuse is as much of a technical issue as it is an organizational one [6].

- Development and implementation

This cost is spent to develop and implement a reusable program.

- Operation

This cost is spent to reuse a component.

- Investment for Support

This cost is spent to support a reuse program such as training and tool acquisition.

II. Related Work

Various authors have **e**stimated cost of a reusable component by using cost of component without reuse, calculated with the help of any approach to estimate cost of software such as algorithmic or function point analysis[7][8][9][4]. Mostly researcher [9][3][8] have estimated cost of a reusable component on the basis of size of component (salaries(costs) are according size). In the proposed scheme, a cost estimation model is suggested for a reuse oriented organization that develops reusable components where costs of reusable components are double in comparison of costs of similar components for single use and salaries(costs) depend on time(year) not on size of component. Studies[10][11][7][2][3][12] calculated relative cost writing for reuse and[2][8][10][11][12] calculated relative cost for reuse, but not estimated the effect of size, strategy, complexity, standardization, integration, modification, scale and availability on cost, [12] only shows effect of complexity and modification. Study [1] have mentioned the effect of complexity, modification, scale on cost. [13] estimated the effect of standardization and [14] estimated the effect of strategy. [15] estimated the effect of size and [16] also estimated the effect of size including integration, on cost. But none of above estimated the effect of availability and integration of components on cost. In this research, the effect of availability and integration of components on cost is estimated as well as almost all above mentioned cost effecting factors are reviewed by extending and filling the gap left by other studies. We have also reviewed some measurement tools-metrics and models to estimate costs of a software reuse program.

III. Proposed Work

3.1 Measures for cost estimate of a reuse program.

Generally following metrics and models are used to estimate cost of a reuse program.

- Reusable assets cost=∑ cost of submitted assets [17]
- Cost of development for reuse= The Relative Cost of Writing Reusable Software(RCWR) * (Cost of development for one-time use) [18] [12]

RCWR depends upon - size, strategy, complexity etc. After various studies[10][11][12][2][3][12] we found that Favro[14] suggests maximum range of RCWR=1.0 - 2.2

Poulin[2] recommended RCWR==1.5

- Operational cost of a reuse component = Relative Cost of Reuse(RCR) *(Cost of component developed from scratch) [18][12].

RCR depends upon many factors and is proportional to complexity and modification of the reused component[1]. After various studies[2][8][10][11][12] we found that Gaffney and Durek [8] suggests minimum value of RCR=.03 and maximum value of RCR=.4 is suggested by Favaro[11] .Poulin[12] recommended RCR =.2

- $\text{Cost}_{\text{application –system}} = (\text{cost}_{\text{new- code}} * \text{size}_{\text{new- code}}) + (\text{cost}_{\text{modify- reused}} * \text{size}_{\text{modified- code}}) + \text{costs}_{\text{reuse –library}}$ [19]

In the SPC model [3],

- $\text{Cost}_{\text{software -product}}$ = cost to develop new + cost of reusing existing software.

3.2 Costs for software reuse.

Both Producer and Consumer have to pay for reuse [1][12][20][13][2][21][22][23][24] as shown in Table1.

Table 1: Producer and Consumer Costs:

Cost	Producer Cost	Consumer Costs
Development Cost	Domain analysis Modelling /design for reusability Implementation for reusability	Domain Analysis Procuring components Assessing components
Overhead Costs	Testing of parts Creation of repository (library) Maintenance of parts Management of parts Publication(marketing) of parts Training of personnel for reusability	Integration Maintenance Adaption Search -Query, Retrieval Testing Modified components Fees and royalties
Other Costs	Performing cost-benefit analysis Documenting reusable parts Making components generic Cataloguing reusable parts standardize reusable parts[13]	Performing cost-benefit analysis

3.3 costs estimation for a reuse program .

We are estimating cost of a reuse program in a hypothetical scenario of a corporation that starts its reuse initiative with domain engineering in 2007 developing reusable components as shown in Table2, that are used in applications internally as shown in Table4, and are also sold externally to corporation for a period of 4 years. The following assumptions are made with respect to this program :

- ✓ Cost of the reused components is double as compared to similar components made for single use.
- ✓ Salary of employees is fixed, not depending upon size and quantity of components .Overhead cost to make reusable components is 10 % of salary of employees and all other details are as shown in Table3.
- ✓ Cost of purchased component(y_i) that are used in applications is 25% extra of cost of internally developed component(y_i).
- ✓ Set-up cost of corporation for reuse program is $3,000 and of application- engineering cycle is $1,000

Table2: Components Details

Year	Component	Size
2007	X1	5k
2008	X2	10k
2009	X3	15k
2010	X4	20k

Table 3: Salary of Employees

Personnel	Initial Salary ($)	Increment/y (%)
component developer for reuse	1000	15
manager for reuse	800	10
librarian for reuse	700	10
Domain Analyst	800	10

Table 4: Applications Details

Year	Application	Component used internally developed	Component used externally developed	Additional-Code
2008	App(1)	X1	O_1	2K
2009	App(2)	X1,X2	O_1,O_2	4K
2010	App(3)	X1, X2,X3	O_1,O_2,O_3	6K

3.3.1Cost Structure:

Software reuse require investment at different levels from top to bottom with a cyclic cost approach[9] .We are suggesting a cost structure for above said corporation.

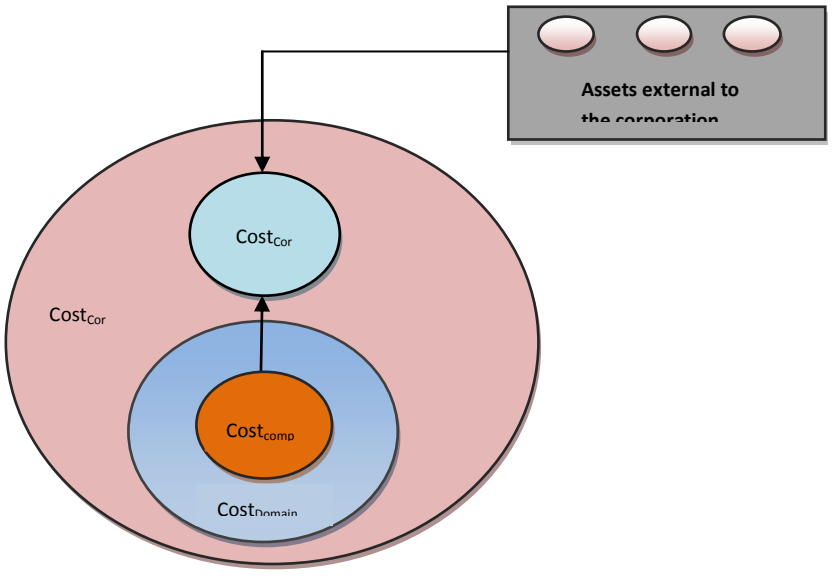


Fig 1. Cost Structure

3.3.1.1 Component Engg. Cycle Cost :

$Cost_{comp\text{-}Engg} = \sum Cost_{comp}$, Where

$Cost_{comp\text{-}Engg}$ = Component Engg. Cycle Cost

$Cost_{comp}$ = Cost for a single component

Cost for a single component:

$Cost_{comp(yi)} = Cost_{dev\,(yi)} + Cost_{Overhead(yi)}$, Where

$Cost_{Overhead} = Cost_{mgt} + Cost_{Maintenance} + Cost_{Training} + Cost_{Library} + Cost_{Pub} + Cost_{Std} + Cost_{others}$

$Cost_{comp} = (Cost_{dev} + Cost_{mgt} + Cost_{Library}) + (Cost_{Maintenance} + Cost_{Training} + Cost_{Pub} + Cost_{Std} + Cost_{others})$

i,e Component cost(yi)= Personnel Salary(yi) + 10% (Personnel Salary(yi))

$Cost_{comp\text{-}Engg} = \sum Cost_{comp}$

= $ 13150.36 , Where

$Cost_{dev}$ =Development cost of Component

$Cost_{Overhead}$ = Overhead cost of Component

$Cost_{Std}$ = Costs of standardizing a Component

3.3.1.2 Domain Engg. Cycle cost :

$Cost_{Domain\text{-}Engg} = Salary_{Dom\text{-}Analyst} + Cost_{comp\text{-}Engg}$

$Cost_{Domain\text{-}Engg}$ = $ 16863.16

Where $Cost_{Domain\text{-}Engg}$ = Domain Engg. Cycle cost

$Salary_{Dom\text{-}Analyst}$ = Salary of domain analyst

3.3.1.3 Application Engg. Cycle cost:

$Cost_{App\text{-}Engg} = Cost_{Set\text{-}up} + \sum Cost_{App}$

Where

$Cost_{App\text{-}Engg}$ = Cost of Application- Engg. Cycle

$Cost_{App}$ = cost of a application

$Cost_{Set\text{-}up}$ = costs of training and tool acquisition for reusing components in aplications

$Cost_{Set\text{-}up} = (Cost_{Training} + Cost_{tool\text{-}acq})$

$Cost_{Training}$ = cost of training of personnel for reusing components in Applications

$Cost_{tool\text{-}acq}$ = cost of tool acquition for reusing components in Applications

$Cost_{App} = Cost_{Add\text{-}Code} + (\sum Cost_{comp\text{-}int} + \sum Cost_{comp\text{-}ext}) + Cost_{Overhead}$

All internally developed Components are used as *Black- Box reuse* and externally developed Components are used as *White- Box reuse . Then*

For Black- Box reuse

$Cost_{Overhead} = Cost_{integration} + Cost_{Maintenance} + (1\text{-}p)*Cost_{search} + Cost_{acq} + Cost_{others}$

For White- Box reuse

$Cost_{Overhead} = Cost_{integration} + Cost_{Maintenance} + Cost\ adap + Cost_{fr} + Cost_{others}$

Take $Cost_{Overhead}$ =20% of $Cost_{Scratch}$ for each (*Black- Box* and *White- Box reuse)* i, e RCR = .2,

$Cost_{Scratch}$ = .5*1315.036 =6575.18 (since cost of reused component is double) i, e RCWR = 2,

Salary of developer for additonal-code = 50% developer for reuse

$Cost_{App(i)} = [Cost_{comp\text{-}int\ (I)} + Cost_{comp\text{-}int\ (I\text{-}1)} + (i\text{-}1)*Cost_{comp\text{-}int\ (i\text{-}2)} + Cost_{comp\text{-}int\ (o)} + Cost_{comp\text{-}int\ (o\text{-}1)} + (o\text{-}1)*Cost_{comp\text{-}int\ (o\text{-}2)}] + [Cost_{Add\text{-}Code} + Cost_{Overhead}]$

$\sum Cost_{App} = \sum (Cost_{Add\text{-}Code} + Cost_{Overhead} + \sum Cost_{comp\text{-}int} + \sum Cost_{comp\text{-}ext})$

$\sum Cost_{App}$ = $ 33400.03

$Cost_{App\text{-}Engg}$ = 1000+ 33400.03

= $34400.03

Where

$Cost_{Overhead}$ = Overhead cost of the application

$Cost_{Scratch}$ = cost of components without reuse

Cost search = cost of performing a search operation

$Cost_{integration}$ = cost of integrating components

$Cost_{Maintenance}$ = cost of maintaining components

p = probability that the component is found in repository

$Cost_{fr}$ = cost of fee and roilites of components that are acquired from externally sources

Cost adapt = cost of adapting components

$Cost_{others}$ = any other cost for reusing components

3.3.1.4 Corporate Engg. Cycle cost :

$Cost_{Cor\text{-}Engg} = Cost_{Set\text{-}up} + Cost_{Domain\text{-}Engg} + Cost_{App\text{-}Engg} - \sum Cost_{comp\text{-}int}$

$Cost_{Cor\text{-}Engg}$ = 3000+16863.16+$34400.03- 13150.36
=41112.83

Where

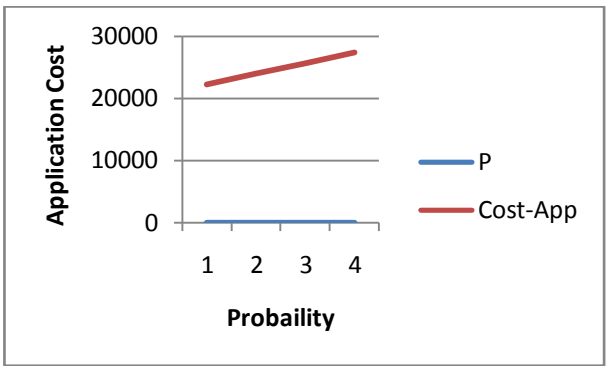
$Cost_{Cor\text{-}Engg}$ = Corporate Engg. Cycle cost

3. 4 Cost effecting factors in software reuse scenario .

Cost effecting factors in context of above said corporation:

3.4.1 Availability

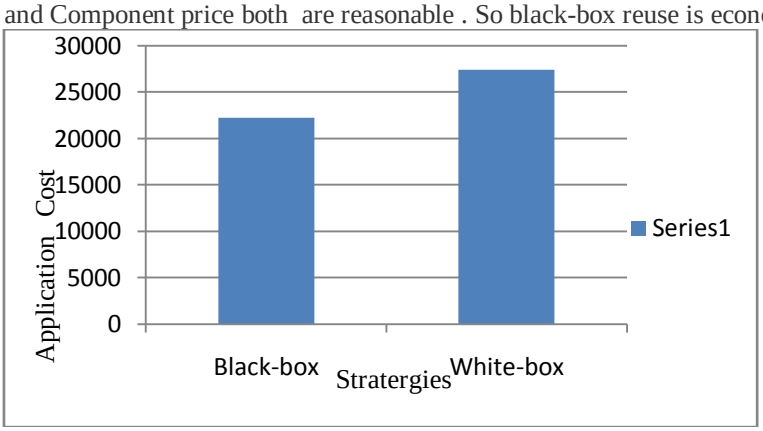
We are developing an application of the year 2010 (cost of all components is taken of year 2009) considering that if a component is found in organization 's own library it is used as black box component otherwise as white box (externally used components). RCR depends upon value of p (probability that a component is found in repository)- as value of p decreases, number of externally used components increases .So cost of the application increases since externally used components more costly as well as RCR increases due to increment in externally used components (white box components) that require modification . RCR is proportional to modification of the reused component and may be up to .9 [1]



Graph1. Availability vs. Cost

3.4.2 Strategies

Components from market with lower customization cost, may off set the higher acquisition costs of black-box reuse (CBD form) [14].But in our model organization itself develops black-box components according its needs so Search cost and Component price both are reasonable . So black-box reuse is economical.



Graph 2. Strategies vs. Cost

3.4.3. Size

Cost is proportional to Size, is a very useful prediction about cost when variation is desired in size of reused product.[15] .So size of reused product should be effective since that incorporates reusable component engineering and integration efforts [16] and ultimately impacts project(application). Integration of components increases size of project so effect on cost should be predicted since it play a lot in overhead costs.

In our model : $Cost_{comp\text{-}X4} : Cost_{comp\text{-}X1}$ = 1.40695 , $Size_{comp\text{-}X4} : Size_{comp\text{-}X1}$ = 4

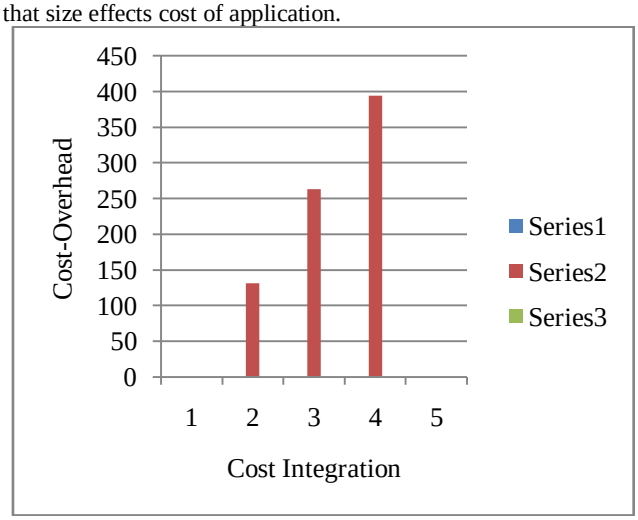
As $Size_{comp}$ increases 4 times but $Cost_{comp}$ also increases but 1.40695 times i,e not directly proportional . (1)

If $Cost_{integration}$ is 1% of Cost $_{Overhead}$ per component in Application –Engg Cycle, then

As Number of components increases , $Cost_{integration}$ also increases directly proportional . (2)

As Number of components increases $Cost_{Add\text{-}Code}$ also increases but not directly proportional. (3)

From (1) (2) .(3) We can say that size effects cost of application.



Graph 3. Size Vs. Cost

3.4.4 Scale

Scale of Reuse ranges (1.15-1.75) for Across Projects to Across Multiple product-line [1]. In our model , Components are used across projects and program. So cost increases as scale of reuse increases.

3.4.5 Standardization

For better understanding and optimal integration a component should be interface and functionality standardized for which it have to pay otherwise pay for adaptation and integration[26]. As standardization level of reusable components increases , cost also increases[13].

3.4.6 Failure

Reuse failure mean that either a component is not reused or the cost of reusing exceeds the cost of developing the component from scratch. Failures in the sense of programmer's time, motivation, accountability differently effects cost and incentives so it should be minimum.[27].

3.4.7 Quality

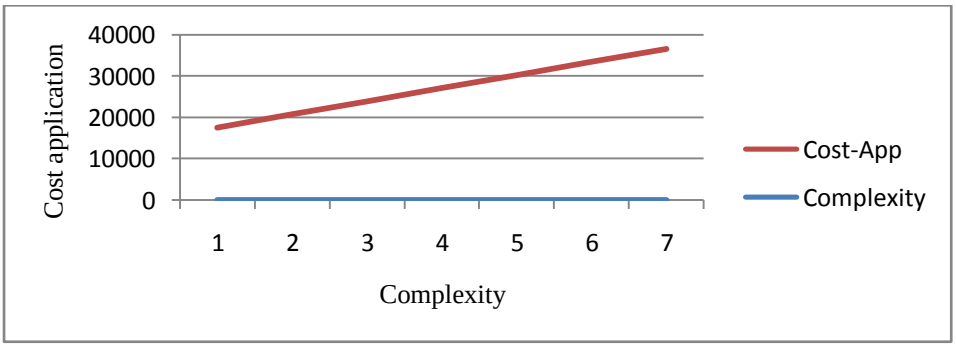
Quality characteristics demand a lot, perhaps the most costly feature of a reusable software.

3.4.8 Complexity

Complexity of component depends upon its structure that may be monolithic , polylithic , graph, menu or mask[18].Complexity of reused component reflects both costs- development and operation. RCWR and RCR are proportional to complexity of the component[1].We have taken Complexity as RCWR in component-Engg. Cycle that reflects Complexity of development and RCR as operation Complexity in Application - Engg. Cycle. RCWR+ RCR=total Complexity of a reused program.

As Complexity of reused components increases ,cost of component-Engg. Cycle and ultimately Application - Engg. Cycle also increases almost proportional to Complexity of a reused program

.



Graph 4. Complexity vs. Cost

IV. Conclusion

Producer as well as user have to pay for a reuse program. Some measures(metrics and models) to estimate the cost of a reuse program are reviewed. In this research, a simplified and enhanced cost estimation model is suggested for all cycles of software reuse organization .The Proposed study also results that a application (project) cost depends upon domain analysis and size , reuse rate ,quantity, availability in repository, strategy, failure ,standardization of components. Corporate(organization) investment depends upon infrastructure and number of applications made.

V. Future Work

For cost-benefit analysis of proposed scheme ,we will suggest a benefits-estimate model and further estimation of economic worth of proposed scheme for organization .

REFERENCES

[1] M.R.V. Chaudron, Software Engineering Economics with special attention to Component-& Reuse-based Software Engineering,www.win.tue.nl/~mchaudro,Dept. of Mathematics and Computing Science Eindhoven University of Technology, Based on IBM Systems Jnl, 1993, Vol 32, No, 4

[2] Tracz, Will, Confessions of a Used Program Salesman Institutionalizing Software Reuse, New York: Addison-Wesley Publishing Company, 1995.

[3] Margano, J. and T. Rhoads,Software Reuse Economics, Cost-Benefit Analysis on a Large Scale Ada Project, In Proceedings, International Conference on Software Engineering, Melbourne, Australia, 11-15 May 1992: 338-348.

[4] Barnes, B. and T. Bollinger, Making Reuse Cost Effective, IEEE Software, Jan. 1991, 8(l): 13-24.

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

[6] Jared Fortune and Ricardo Valerdi Los Angeles, Considerations for Successful Reuse in Systems Engineering, University of Southern California , CA, Massachusetts Institute of Technology, Cambridge, MA.

[7] Gaffney, J. and R. Cruickshank., "Productivity Consortium, Reston", SPC-TR-88-015 Version 1.0, Software, VA, April 1988.

[8] [Gaffney, J. and T. Durek. Software Reuse —Key to Enhanced Productivity: Some Quantitative Models, Information and Software Technology, June 1989, 31(5):258-267.

[9] Senta Fowler Chmiel , An Integrated Cost Model for Software Reuse, for the degree of Doctor of Philosophy in Computer Science, Dissertation submitted in partial fulfillment of the requirements to the College of Engineering and Mineral Resources at West Virginia University, Morgantown, West Virginia,2000

[10] Lim, W. Effects of Reuse on Quality, Productivity and Economics. IEEE Software, September 1994, 11(5): 23-30.

[11] 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.

[12] Jeffrey S. Poulin, Measuring software reuse, principles, practices, and economic models, Addison-Wesley Longman Publishing Co., Inc., Boston, MA,USA, 1996.

[13] Giancarlo Succi and Francesco Baruchelli , The cost of Standardizing Components for Software Reuse, Supporting Article ,University OF Calgary, Calgary, Alberta, Canada, University DI Trento, Trento, Italy.

[14] By T. Ravichandran and Marcus A. Rothenberge ,Software Reuse Strategies and Component Markets", Communications of the ACM August 2003/Vol.46 No. 8 Pp. 109-114

[15] O. K. Harsh and A. S. M. Sajeev, Component-Based Explicit Software Reuse, Engineering Letters, 13:1, EL_13_1_4 (Advance online publication: 4 May 2006).

[16] Randall W. Jensen ,Estimating the Cost of Software Reuse, Software Engineering, Inc.

[17] Basili , In the context of the Goal Question Metric , Lin98.

[18] Barry Boehm, Software Reuse Economics, DARPA Workshop, C S E Center for Software Engineering University of Southern California 1/6/97.

[19] Raymond, G. and D. Hollis, Software Reuse Economics Model ,In Proceedings of Eighth Annual Washington Ada Symposium Summer SIG Ada Meeting, McLean, VA, June 1991:141-155

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

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

[22] Marcus A. Rothenberger; Derek Nazareth, A COST-BENEFIT-MODEL FOR SYSTEMATIC SOFTWARE REUSE, School of Business Administration University of Wisconsin – MilwaukeePO Box 742,Milwaukee, WI 53211, USA,E-mail: {rothenb; derek}@uwm.edu

[23] Dr. Jeffrey S. Poulin, An Agenda for Software Reuse Economics, International Conference on Software Reuse 15 April 2002, Lockheed Martin Systems Integration,MD 0220 Owego, NY 13827,Tel: (607) 751-6899,Jeffrey.Poulin@lmco.com

[24] Malan R. and Wentzel K. "Economics of software reuse revisited, Proc. 3rd Irvine Software Symposium, University of California, Irvine, 30 April,1993 pp.109-21.

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

[26] Nasib S. Gill, Reusability, Issues in Component-Based Development, nsgill2000@yahoo.com, nsgill_2k4@yahoo.com

[27] Robert G. Fichman and Chris F. Kemerer , Incentive Compatibility and Systematic Software Reuse, Appeared in: Journal of Systems and Software, New York; Apr 27, 2001; Vol. 57, Iss. 1; pg. 45