# A Benefits Estimation Model for Software Reuse Based Program

[1]Shobha Rani Malik, [2] Dr. Saba Hilal

[1] (Lecturer,Technical Edu. Deptt. Haryana and Research Scholar at Lingayas' University Faridabad, India)
[2](Research Guide, Lingayas' University Faridabad, India )

**Abstract:**
In software engineering, reuse of artifacts is considered as silver-bullet for project development specially in component based development technic.The benefits of reuse are neither linear nor exact they are estimated with the help of metrics and models. To measure benefits of software reuse is a challenging task since they are multi dimensional mainly earlier time-to-market ,better quality, increased productivity etc. and of two dimensional characteristics- quantitative and qualitative . Reuse is highly potential in correlated domains due to similar requirements.

**Keywords:** Software Reuse Benefits estimation, Producer and User Benefits, Software Reuse Benefits increasing factors

## 1.Introduction

"Actually it is hard to see how much time or budget u have save during the project developing. it is also quite impossible to set up the goal and timetable about the reuse work in the beginning of the project" [14 ].Cost savings is the most promoted benefit for reuse, but benefits also exist in risk, schedule, and performance [ 9]. Estimation of benefits is not easy since they depend up on management, technology and organizational factors[ 38] . Generally the largest payoffs of software process assessment and improvement are in human terms - pride in work, increased job satisfaction, improved ability to attract and retain software experts and not in dollars [3][4]. Benefits include total life cycle costs saved as well as additional profits resulting from earlier completion of the product[7].

### 1.1 Software Reuse Benefits

Benefits in software engineering [48][49][50][51][26][28][8] due to reuse of artifacts are mainly as shown in figure1.

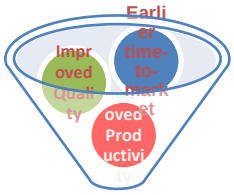



Fig 1. Reuse Benefits

Both the producer and the user enjoy benefits of reuse artifacts in a reuse program.

- User Benefits

User enjoys benefits of reusing well tested, more reliable, higher standards artifacts by reducing development and maintenance costs[1][49]with better quality , improved productivity that results additional revenue, improved sales and increased market share due to earlier market delivery of product.

- Producer Benefits

Producer takes benefits of selling product with high price, fees and royalties, [1] that results increased market share.
When producer does not explicitly charge for its components or service, reuse may be economically feasible for the user but not for the producer [7].
A producer assumes a higher cost to design and implement a reusable asset than a custom developed asset, but the consumer saves time and money by not having to develop the component [35]. If a software developer have to play both user and producer roles then for organization's benefits, creation cost should be less than consuming cost.

## 2. Related Work

Various studies [20 ][6][21][10][29][18] [19][26][24] have estimated only direct(quantitative) benefits of reuse generally measured costs avoidance that is a major factor of reuse benefits but a very few measured[28][15][13] some limited indirect(qualitative) benefits of reuse. in a In this research, a model is proposed for a reuse oriented organization which can

estimate direct as well as indirect reuse benefits by considering almost all factors that contribute in benefits of reuse by extending and filling the gap left by other studies. We reviewed some measurement tools-metrics and models to estimate benefits of software reuse and also have suggested some benefits increasing factors.

## 3. Proposed Work

### 3.1 Reuse Benefits Estimation Metrics and Models

To estimate benefits of reuse suitable measurement technique should be applied. Various metrics and models are proposed by researcher to measure reuse benefits , some of them are reviewed here. In software engineering reuse has been claimed as an important source of saving costs [2].

- Cost Saving:

$Cost_{saved} = Cost_{scratch} – cost_{with\ reuse}$ So, as the number of element increases the savings increases also[46].
$Cost_{saving} = Cost_{scratch} - Cost_{reuse} - Cost_{delivered}$ [42]
Where $Cost_{scratch}$ = cost of developing software from scratch, $Cost_{reuse}$ = slide cost associated with reuse and $C_d$ = cost of software delivered
According[13], Saving =∑((cost of artifacts * reuse rate of artifacts) – cost (avg. integration effort of artifacts* reuses of artifacts))
According [20][21] , Total savings due to reuse= [(Savings due to avoided cost -Relative cost of reuse)*No of uses]-[Cost to add the compnent to the library+Cost to maintain the component in the library]
% Savings = [0.92 - 1.5*j/i]*100 , where j = number of reusable software components that have been built, i = number of attempted instances of reuse[34].

Benefits = activity cost without reuse- activity cost with reuse [5 ].
Benefits = ∑(Development without reusable products - Cost of adptation)- Total cost of all resources[6 ].
Benefits = graphs of the reuse cost ratio (cost with reuse/cost without reuse) versus the breakeven point. [29].
In the GTE-Contel model[11]
Net benefits =∑ (net cost no reuse - net cost with reuse reuse) –total reuse investment
Benefits (system S) = [Cost of developing S without reuse - Cost of developing S with reuse]/ Cost of S with out reuse. [18]
These benefits are sensitive to implementations, reuse strategy and reused components cost.
As reuse leverage increases, benefits also increases [ 7].
Reuse Cost Avoidance (RCA)= Development Cost Avoidance (DCA) +Service Cost Avoidance(SCA) [19].
Benefit investment = (cost of project without reuse - actual cost of project with reuse)/ cost of producing reusable components[26]

- Quality :

It is very hard to quantify quality benefit due to its intangible nature. It is generally measured either in savings in maintenance or in finding and removing errors.
According[17] , Defect removal efficiency $=DRE_i = E_i / (E_i + E_{i+1})$, where $E_i$ = the number of errors found in the $i^{th}$ component and $E_{i+1}$ =the number of errors found after integrating $(i+1)^{th}$ component with the $i^{h}$ component.
Quality is savings in maintenance[28].
Quality is finding and removing errors[15].
High level of reuse correlates with a low defect [52].
Reuse rate, development time and decreases in number of errors are highly correlated [53].
Adequacy (library contents) = number of artifacts reused + number of not available but required artifacts + number of artifacts in library and Goodness (artifacts repository)= Defect density of reused artifacts + Avg. reusability score [13] .
Quality of an instrument (Q) = B/R Where Q <B(producer activities) and R(consumer activities).[31]

- Productivity:

According [18] , Productivity = size of the system /cost spent to develop system i,e
Productivity = a(1+ Reuse benefit )b , a, b are Coefficients estimated with standard least squares regression
Productivity= (Reuse KNCSS +Modified KNCSS)/ Product Total KNCSS]*100 [10].
Productivity =1/cost [16].

### 3.2 Benefits estimation for a reuse program

We are estimating benefits 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 Table1, that are used in applications

internallyasshown in Table2, and are also sold externally to corporation for a period of 4 years.

Table1: Components Details

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

The following assumptions are taken 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(yi ) that are used in applications is 25% extra of cost of internally developed component(yi).
- ✓ Sell price of internally developed components and applications is 25% extra of their respective costs.
- ✓ Set-up cost of corporation for reuse program is $3,000 and of application- engineering cycle is $1,000 .

**Table 2: 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 |

**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 |

**3.2.1 Benefits Structure:**

In above said organization , a component developer roles as creator (producer) in Domain Engg. cycle and component user as consumer in Application- Engg. cycle ,so all benefits are of Corporate. It is not possible to quantify all benefits of reuse specially qualitative benefits but it is tried to measure them. Benefits of reuse program for above said corporation are structured as shown in Fig.2. Time value of money is not considered here.

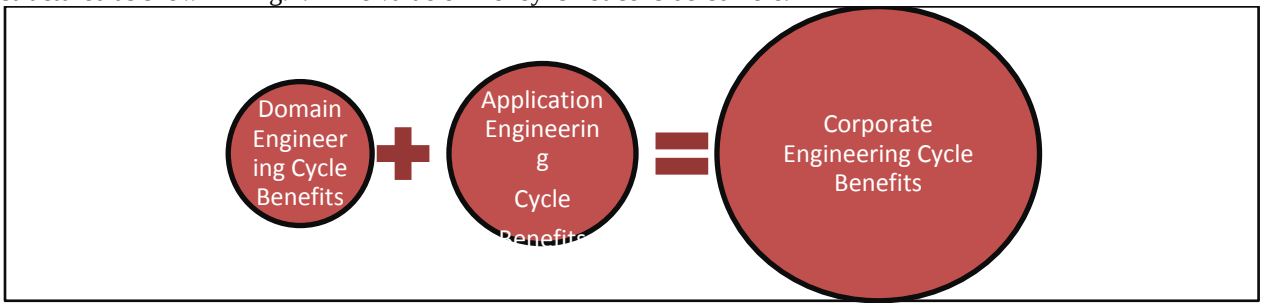


**Fig 2. Benefits Structure**

Corporate Engineering Cycle Benefits= Domain Engineering Cycle Benefits from selling components internal and external to the corporation + Application Engineering Cycle Benefits from selling applications external to the corporation.

**3.2.1.1 Domain Engg. Cycle Benefits**

$Benefits_{Domain\text{-}Engg} = Benefits_{comp\text{-}Engg} - Salary_{Domain\text{-}Analyst}$

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

Where

$Benefits_{Domain\text{-}Engg}$ = Benefits of Domain Engg. Cycle

$Benefits_{comp\text{-}Engg}$ = Benefits of Component Engg. Cycle

$Benefits_{comp}$ = Benefits from selling a component internal and external to the corporation

$\sum Benefits_{comp} = \sum Direct\ Benefits_{comp} + \sum Indirect\ Benefits_{comp}$

It is assumed that all components (X1, X2, X3) made before 2010 are sold internal and external to the corporation and fees and royalties are 1% of sell price for externally sold components. Component X4 made in 2010 is not included in cost-benefit analysis since it will be used internally in next plan(App4,App5,App6) and sold externally in next coming year(2011) according proposed scheme.

- $\sum Direct\ Benefits_{comp}$ = [ Saving due to sell - Additional cost due to make reusable]

[(Total sell price -Total cost to make reusable components) + fees and royalties ] - [Total cost to make reusable components - Total cost to make for no reuse]

=[{(Sell price of components internal to the corporation+ Sell price of components external to the corporation)- Total cost to make reusable components } +fees and royalties]- [Total cost to make reusable components -Total cost to make for no reuse ]

[{(9281.25+11601.56)- 9281.25}+116.0156] - [ 9281.25 -4640.625]

=11717.58-4640.625 = $7076.953

- $\sum Indirect\ Benefits_{comp}$ = increased market share +Improved quality Increased market share- due to revenue from selling components external to the corporation + Improved quality and productivity in terms of increased experience of staff (more trained staff) in same environment.

$Benefits_{Domain\text{-}Engg} = (7076.953\text{-}2648) + \sum Indirect\ Benefits_{comp}$

$Benefits_{Domain\text{-}Engg} = \$4428.953 + \sum Indirect\ Benefits_{comp}$

**3.2.1.2 Application -Engg. Cycle Benefits**

$Benefits_{App\text{-}Engg} = \sum Benefits_{App}$

Where

$Benefits_{App\text{-}Engg}$ = Benefits of Application Engg. Cycle

$Benefits_{App}$ = Benefits of a application

$\sum Benefits_{App} = \sum Direct\ Benefits_{App} + \sum Indirect\ Benefits_{App}$

$\sum$Direct $Benefits_{App}$ = [ Saving due to sell- Saving due to reuse]

= [Total sell price-Total cost with reuse ]

= [Total sell price of applications external to the corporation – (Total cost of applications with reuse +Set-up cost -Total cost of repeated components) ]

= [(Total sell price of applications external to the corporation – Total cost of applications with reuse) - Set-up cost + repeated cost of components ]

= (.25* Total cost of applications with reuse) -1000 + cost (2X1+X2)

= (.25* 45285.19 +2*2750+ 3080)-1000

= 11321.3+ 5500 + 3080-1000

=19901.3-1000 = $18901.3

- $\sum$Indirect Benefits$_{App}$ = additional revenue due to earlier market delivery of product + increased sales (increased productivity due to reusing components +Improved quality ) +increased market share

**3.2.1.3 Corporate Engg. Cycle Benefits**

$Benefits_{Cor\text{-}Engg}$ = $Benefits_{Domain\text{-}Engg}$ + $Benefits_{App\text{-}Engg}$ –Set-up cost , Where

$Benefits_{Cor\text{-}Engg}$ = Corporate Engg. Cycle benefits

i, e $Benefits_{Cor\text{-}Engg}$ = [(Direct Benefits) $_{Domain\text{-}Engg}$ + $(Direct\ Benefits)_{App\text{-}Engg}$] + [$(Indirect\ Benefits)_{Domain\text{-}Engg}$ + $(Indirect\ Benefits)_{App\text{-}Engg}$ ] -3000
=[4428.953 +18901.3]-3000 + [$(Indirect\ Benefits)_{Domain\text{-}Engg}$ + $(Indirect\ Benefits)_{App\text{-}Engg}$ ]

= $20330.25 + [$(Indirect\ Benefits)_{Domain\text{-}Engg}$ + $(Indirect\ Benefits)_{App\text{-}Engg}$ ] = $20330.25 + Increased market share- due to additional revenue due to earlier market delivery of product and selling components external to the corporation+ Increased sales (Improved quality +increased productivity due to reusing components) +Improved quality and productivity in terms of increased experience of staff (more trained staff) in same environment .

**3.4. Benefits increasing factors in software reuse scenario**

Following efforts can help to increase benefits in context of above said corporation:

- Rich Component repository

Organization ‘s own repository should be rich of components since in house built components are lesser cost than components purchased from market. Library should be updated according needs and new components should be regularly introduced [39].

- Maximize black-box reuse

Black-Box Reuse(CBD form ) is costly due to inefficiency in search techniques and components price [40] but in house built black-box components are designed according internal reuse needs and forward looking reuse as well as this reduce the understanding time since a reusable asset require more understanding time than any other part of reuse process [22].So search cost and component price are very reasonable in house built black-box component.

- Minimize Complexity

In our model , as Complexity of component increases ,cost increases also .So try to Minimize it.

- Effective Size

Cost increases according size , so it should be effective and can be calculated as

Effective size = (existing size × (0.4 × redesign% + 0.25 × reimplementation% + 0.35 × retest %)) [32].

- Avoid Reuse Failure

Failures of reuse increases programmer's time, decreases motivation, accountability and incentives so it should be avoided [33].

- Start with small scale and grow-up

At initial level organizations should start from low level of reuse with small size of components since they are easy to adapt[38 ] and make and then grow-up with increasing size of components.

- Include reuse as integral part

Reuse should be as an integral part of software life cycle process[28] by providing central support for reuse activities .

- Improve Quality

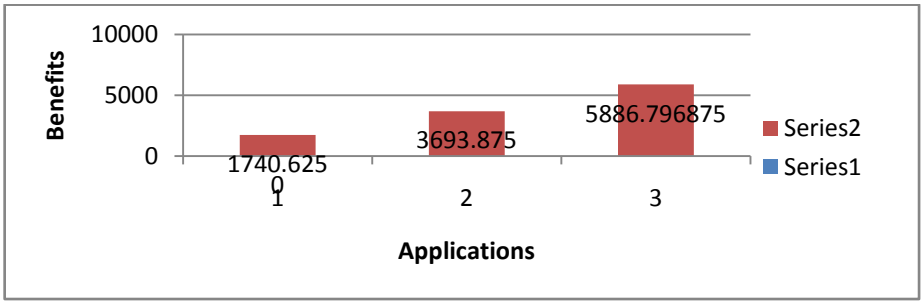
Quality benefits can be achieved by error reduction and standization . Reuse rate of component should be increased since defect are fixed each time that results –improved quality[10]. Try to remove defects in earlier phases as soon as possible since cost of prevention and debugging can be amortized number of uses [27] and the longer it takes to identify an error, the costlier it is to fix[35][36 ] .Initiatives should be provided to establish methods to avoid post release errors[37].

- Minimize product delivery time

Try to avoid missing a market window since in reused oriented environment ,sometimes earlier time-to-market can be more profitable and competitive advantage than direct cost reduction[41][44][45][47].

- Use common technical standards [12].
- Increase Scale of Reuse

In our model , Components are used across projects. For example , as component X1 is used in App1,App2 and App3 then benefits increase also according scale as shown in Graph1.




**Graph1. Scale Vs. Benefits**

- Share knowledge and experiences

Organization can introduce reuse groups and forums where people can discuss problems so that experiences should be shared by other organizations [23] .

- Make components according market needs

Design and functionality of components market should be according needs.

- Up gradation and incentives for staff

Organization should establish refresher courses, training for staff up gradation to compete market trends and recognition, royalty , lottery award and preferential funding policy for reuse achievements[25] . Set targets and use competitions to drive reuse [30] to increase and promote the reuse level.

- Improve Continuously

Perform walkthroughs, reviews , feedback from users and monitor reuse efforts throughout development life cycles.

## 4. Conclusion

Both producer as well as user enjoy benefits in a software reuse scenario specially user. Benefits may be operational or strategic ,direct or hidden, quantitative or qualitative. Some measures(metrics and models) to estimate the benefits of a reuse program are reviewed. In this research, a simplified and enhanced benefit estimation model is suggested for all cycles of software reuse development .The Proposed study also suggests that rich repository, black-box strategy, failure avoidance development according market requirements, better quality , standardization of components as well as continuously improvement with incremental business strategy by keeping staff up graded, happy and sharing knowledge ,experiences with others ,the corporate(organization) can increase its benefits.

## 5. Future Work

Economic worth of proposed scheme for above said organization will be estimated.

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

[3]. Krasner, H. ,The Payoff for Software Process Improvement: What it is and How to Get it. In K. Emam & N. Madhavji(Eds.), Elements of Software Process Assessment and Improvement (pp. 151-176) , Washington, DC: IEEE Computer Society Press,1999.

[4]. Ronald G. Wolak, DISS 725 – System Development: Research Paper 4,Software Process Assessment and Improvement Models,wolakron@nova.edu.

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

[6] . Bollinger, T., and S. Pfleeger. Economics of Reuse: Issues and Alternatives. Information and Software Technology, December 1990, 32(10): 643-652.

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

[8]. Hui Zhou and Monan Yao, Software Reuse And Off shoring : A Study Of Benefits, Difficulties And Feasibility, Bachelor of Applied Information Technology Thesis, Report No. 2010:035,ISSN: 1651-4769

[9]. Jared Fortune1, Ricardo Valerdi2, Barry W. Boehm3 and F. Stan Settles4 . Estimating Systems Engineering Reuse, 7th Annual Conference on Systems Engineering Research 2009 (CSER 2009)

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

[11]. 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. 141-155.

[12]. McClure, C., "Software Reuse: A Standards-Based Guide", IEEE Computer Society, 2001, ISBN 076950874X.

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

[14]. Henrik, a software development professor in Sweden.

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

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

[17]. K. S. Jasmine, and R. Vasantha,' DRE - A Quality Metric for Component based Software Products, World Academy of Science, Engineering and Technology 34 2007.

[18]. Devanbu, P., S.Karstu, W.Melo, and W.Thomas. Analytical and Empirical Evaluation of Software Reuse Metrics. Proceedings, International Conference on Software Engineering, Berlin, 1996.

[19 ]. Poulin, J. The Economics of Software Product Lines. International Journal of Applied Software Technology , March 1997.

[20]. Poulin, J., and J. Caruso. A Reuse Metrics and Return on Investment Model. In Advances in Software Reuse; Proceedings of the Second international Workshop on SoftwareReusability, Lucca, Italy, 24-26 March 1993: 152-166.

[21]. Lim, W. Reuse Economics: A Comparison of Seventeen Models and Directions for Future Research, Proceedings, International Conference on Software Reuse, Orlando, FL, 23-26, April 1996: 41-50.

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

[23]. Ryan Gerard, Robert R. Downs, James J. Marshall, and Robert E. Wolfe.The software reuse working group: A case study in fostering reuse. In IRI, pages 24{29. IEEE Systems, Man, and Cybernetics Society, 2007.

[24]. Malan, R. and K. Wentzel. Economics of Reuse Revisited. Technical Report HPL-93-31, Hewlett Packard Laboratories, April 1993.
[25]. Reuse incentives, Planning Initiation Implementation Maturity, http://www.rise.com.br 43/46
[26]. Poulin, J. S., Caruso, J. M., and Hancock, D. R. The business case for software reuse. IBM Systems Journal, 32(4), 567-594.
[27]. M.D. Lubars,Affording Higher Reliability Through Software Reusability,Software Eng. Notes, Oct. 1986, p.39.
[28]. nta 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
[29]. Schimsky, D. Software Reuse --Some Realities. Vitro Tech Journal, Summer 1992, 10(1):47-57.
[30]. Dr. Jeffrey S. Poulin Measuring Software Reuse Federal Chief Architects Forum11 May 2004, Jeffrey.Poulin@lmco.com,http://home.stny.rr.com/jeffreypoulin

[ 31]. Sarbjeet Singh, Manjit Thapa, Sukhvinder singh and Gurpreet Singh, Software Engineering - Survey of Reusability Based on Software Component," International Journal of Computer Applications (0975 – 8887) Volume 8– No.12, October 2010.

[32]. G.N.K.Suresh Babu and Dr.S.K.Srivatsa, Analysis and Measures of Software Reusability, International Journal of Reviews in Computing ,© 2009 IJRIC.

[33]. Robert G. Fichman ,Incentive Compatibility and Systematic Software Reuse, Appeared in: Journal of Systems and Software, New York; Apr 27, 2001; Vol. 57, Iss. 1; pg. 4

[34]. L. Amar and J. Cofey. Measuring the benefits of software reuse – examining three different approaches to software reuse. Dr Dobbs Journal, 30:73,76, 2005.
[35]. De Marco, T., Structured Analysis and System Specification, Prentice Hall, Englewood Cliffs, NJ, p. 26 (1979).
[36]. Grady, R. B., "Practical Results from Measuring Software Quality," Commun.ACM 36, 62–67 (1993).
[37]. William R. Bitman ,Balancing Software Composition and Inheritance to Improve Reusability, Cost, and Error Rate,Johns Hopkins APL Technical Digest, Volume 18, Number 4 (1997).
[38]. Mohsin Irshad ,Master Thesis Software Engineering Measuring Cost Avoidance Through Software Reuse, A model to measure costs avoided through software,reuse and guidelines to increase profits through,software reuse, School of Computing, Blekinge Institute of Technology,SE-371 79 ,Karlskrona, Sweden, 2010
[39]. Michael F. Dunn and John C. Knight. Software reuse in an industrial setting: a case study. In Proceedings of the 13th international conference on Software engineering, ICSE '91, pages 329{338, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.
[40]. 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
[41]. William T. Ward. Calculating the real cost of software defects. Hewlett-Packard Journal, pages 55-58, October 1991.
[42]. A Comparison of Approaches to Reuse Investment Analysis John Favaro
[44]. Preston G. Smith and Donald G. Reinertsen. Developing Products in Half the Time. Van Nostrand Reinhold, New York, 1991.
[45]. Martin L. Griss,Software Reuse:From Library to Factory,Software Technology Laboratory,HPL-93-67,July, 1993
[46]. 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
[47]. George Stalk Jr. Time - the next source of competitive advantage. Harvard Business Review, 66(4):41-51, Jul-Aug 1988.
[48]. Parastoo Mohagheghi , R. C. (2007). "Quality, productivity and economic benefits of software reuse: a review of industrial studies." Empirical Software Engineering Volume 12(Issue 5): pp: 471-516
[49]. Sommerville, I., "Software Reuse", Software Engineering, 7th Edition, Chapter 18, 2004 http://www.comp.lancs.ac.uk/computing/resources/IanS/SE7/Presentations/PDF/ch18.pdf
[50]. Mili, H., Mili, A., Yacoub, S., and Addy, E. (2002) Reuse-Based Software Engineering.
[51.] Anthony M. Wilson, Joseph G. San Miguel and Michael W. Boudreau, The Impact of Software Reuse on The Cost of Navy Sonar and Fire Control Systems, Naval Postgraduate School Monterey, California Thesis, Approved for public release; distribution is unlimited , June 2009
[52]. Agresti w. and Evanco W. , Projecting Software Defects in analyzing Ada designs,IEEE Trans.Softw. Engg.18,11,988-997,1992
[53]. Browne J., LeeT. and Werth J. , Experimental Evaluation of Reusability-Oriented Parallel Programming Environment, 16,2,111-120, 1990