A Cost Estimation Model For Reuse Based Software Program

Shobha Rani Malik¹, Dr. Saba Hilal²
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 estimated 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 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= Σ 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
- Cost application-system = (cost new-code * size new-code) + (cost modify-reused * size modified-code) + costs reuse-library [19]

In the SPC model [3],
- Cost 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 Table 1.

<table>
<thead>
<tr>
<th>Table 1: Producer and Consumer Costs:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cost</strong></td>
</tr>
<tr>
<td>Development Cost</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Overhead Costs</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Other Costs</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
3.3 Costs estimation for a reuse program

We are estimating the 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 Table 2, that are used in applications internally as shown in Table 4, and are also sold externally to the 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 Table 3.
- Cost of purchased component(yi) that are used in applications is 25% extra of cost of internally developed component(yi).
- Set-up cost of corporation for reuse program is $3,000 and of application-engineering cycle is $1,000.

<table>
<thead>
<tr>
<th>Year</th>
<th>Component</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>X1</td>
<td>5k</td>
</tr>
<tr>
<td>2008</td>
<td>X2</td>
<td>10k</td>
</tr>
<tr>
<td>2009</td>
<td>X3</td>
<td>15k</td>
</tr>
<tr>
<td>2010</td>
<td>X4</td>
<td>20k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Personnel</th>
<th>Initial Salary ($)</th>
<th>Increment/y (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>component developer for reuse</td>
<td>1000</td>
<td>15</td>
</tr>
<tr>
<td>manager for reuse</td>
<td>800</td>
<td>10</td>
</tr>
<tr>
<td>librarian for reuse</td>
<td>700</td>
<td>10</td>
</tr>
<tr>
<td>Domain Analyst</td>
<td>800</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 2: Components Details

<table>
<thead>
<tr>
<th>Year</th>
<th>Application</th>
<th>Component used internally developed</th>
<th>Component used externally developed</th>
<th>Additional Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>App(1)</td>
<td>X1</td>
<td>O1</td>
<td>2K</td>
</tr>
<tr>
<td>2009</td>
<td>App(2)</td>
<td>X1,X2</td>
<td>O1,O2</td>
<td>4K</td>
</tr>
<tr>
<td>2010</td>
<td>App(3)</td>
<td>X1,X2,X3</td>
<td>O1,O2,O3</td>
<td>6K</td>
</tr>
</tbody>
</table>

3.3.1 Cost 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:
\[
\text{Cost}_{\text{comp-Engg}} = \sum \text{Cost}_{\text{comp}}
\]
Where
\[
\text{Cost}_{\text{comp}} = \text{Cost for a single component}
\]
Cost for a single component:
\[
\text{Cost}_{\text{comp}(yi)} = \text{Cost}_{\text{dev}(yi)} + \text{Cost}_{\text{Overhead}(yi)}
\]
Where
\[
\text{Cost}_{\text{dev}} = \text{Development cost of Component}
\]
\[
\text{Cost}_{\text{Overhead}} = \text{Overhead cost of Component}
\]

3.3.1.2 Domain Engg. Cycle cost:
\[
\text{Cost}_{\text{Domain-Engg}} = \text{Cost}_{\text{dev}} + \text{Cost}_{\text{Mnt}} + \text{Cost}_{\text{Training}} + \text{Cost}_{\text{Library}} + \text{Cost}_{\text{Pub}} + \text{Cost}_{\text{Std}} + \text{Cost}_{\text{others}}
\]
Where
\[
\text{Cost}_{\text{dev}} = \text{Development cost of Component}
\]
\[
\text{Cost}_{\text{Mnt}} = \text{Maintenance cost}
\]
\[
\text{Cost}_{\text{Training}} = \text{Training cost}
\]
\[
\text{Cost}_{\text{Library}} = \text{Cost of library}
\]
\[
\text{Cost}_{\text{Pub}} = \text{Publication cost}
\]
\[
\text{Cost}_{\text{Std}} = \text{Cost of standardizing a Component}
\]
\[
\text{Cost}_{\text{others}} = \text{Other costs}
\]

3.3.1.3 Application Engg. Cycle cost:
\[
\text{Cost}_{\text{App-Engg}} = \text{Cost}_{\text{Set-up}} + \sum \text{Cost}_{\text{App}}
\]
Where
\[
\text{Cost}_{\text{App-Engg}} = \text{Cost of Application-Engg. Cycle}
\]
\[
\text{Cost}_{\text{Set-up}} = \text{Cost of setting up the application}
\]
\[
\text{Cost}_{\text{App}} = \text{Cost of application}
\]
\[
\text{Cost}_{\text{Set-up}} = \text{Cost of setting up the application}
\]
\[
\text{Cost}_{\text{Training}} = \text{Training cost of personnel for reusing components in applications}
\]
\[
\text{Cost}_{\text{Tool-acq}} = \text{Cost of tool acquisition for reusing components in applications}
\]
\[
\text{Cost}_{\text{App}} = \text{Cost of application}
\]
\[
\text{Cost}_{\text{Add-Cde}} + (\sum \text{Cost}_{\text{comp-int}} + \sum \text{Cost}_{\text{comp-ext}}) + \text{Cost}_{\text{Overhead}}
\]
\[
\text{Cost}_{\text{Add-Cde}} = \text{Cost of adding code}
\]
\[
\text{Cost}_{\text{comp-int}} = \text{Cost of components in internal applications}
\]
\[
\text{Cost}_{\text{comp-ext}} = \text{Cost of components in external applications}
\]
\[
\text{Cost}_{\text{Overhead}} = \text{Overhead cost of application}
\]

3.3.1.4 Corporate Engg. Cycle cost:
\[
\text{Cost}_{\text{Cor-Engg}} = \text{Cost}_{\text{Set-up}} + \text{Cost}_{\text{Domain-Engg}} + \text{Cost}_{\text{App-Engg}} + \sum \text{Cost}_{\text{comp-int}}
\]

Cost Estimation Model For Reuse Based Software Program


A Cost Estimation Model For Reuse Based Software Program

Cost\textsubscript{Cor-Engg} = 3000 + 16863.16 + $34400.03 - 13150.36 = 41112.83

Where

Cost\textsubscript{Cor-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](image)

3.4.2 Strategies

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

![Graph 2. Strategies vs. Cost](image)

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: \( \text{Cost}_{\text{comp-X}4} / \text{Cost}_{\text{comp-X}1} = 1.40695 \), \( \text{Size}_{\text{comp-X}4} / \text{Size}_{\text{comp-X}1} = 4 \)

As \( \text{Size}_{\text{comp}} \) increases 4 times but \( \text{Cost}_{\text{comp}} \) also increases but 1.40695 times i.e not directly proportional. \( (1) \)

If \( \text{Cost}_{\text{integration}} \) is 1% of \( \text{Cost}_{\text{Overhead}} \) per component in Application –Engg Cycle, then

As Number of components increases, \( \text{Cost}_{\text{integration}} \) also increases directly proportional. \( (2) \)

As Number of components increases \( \text{Cost}_{\text{Add-Code}} \) also increases but not directly proportional. \( (3) \)
From (1) (2) (3) we can say that size effects cost of application.

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, polythilic, 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.
A Cost Estimation Model For Reuse Based Software Program

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


[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


[18] Barry Boehm, Software Reuse Economics, DARPA Workshop, C S E Center for Software Engineering University of Southern California 1697.


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


[26] Nasib S. Gill, Reusability, Issues in Component-Based Development, ngill2000@yahoo.com, ngill_2k4@yahoo.com

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