Mathematical Modeling for Software Reusability

Dinesh Kumar Saini¹
Faculty of Computing and Information Technology,
Sohar University, Sohar, Sultanate of Oman,
E-mail: dinesh@soharuni.edu.om

Sanad Al-Maskari²
Faculty of Computing and Information Technology,
Sohar University, Sohar, Sultanate of Oman,
E-mail: sanad@soharuni.edu.om

Lingaraj A Hadimani³
Dept. of Electronic and Computer Engineering,
Caledonian College of Engineering, Muscat, Sultanate of Oman,
E-mail: lingaraj@caledonian.edu.om

Abstract - This paper describes the major issues in the software reusability and metrics established to quantify the concept. The concept of reuse in the software development environment involves decisions related to reuse or not in the development environment. Reusability brings financial benefits as well as it reduces development time, development cost, and brings direct and indirect benefits which are hard to quantify. A piece of code unit or a component can be used, and based on these two parameters cost and benefits analysis of reusability is quantified with the help of already established mathematical formulas and software metrics for reusability will be established. The research framework and conclusions should provide a useful reference for persons interested in ways to determine the reusability and its quantified mathematical validated approaches to measure the benefits of the reusability.

Keywords - Mathematical model, Reusability, metrics, software development environment

I. INTRODUCTION

For software development managers to effectively manage and control software development projects they need to incorporate a measurement process into their decision making and reporting process [3]. Software measurement process is a systematic method of measuring assessing and adjusting the software development process using objective data [1]. Software reusability is one of the hot issues in the industry. Reusability is a major issue in the software engineering [9, 10]. In the last decade academicians have proposed many ideas and models but only a few had a direct impact on the software industry. A mathematical modeling effort is implemented to quantify the concept of reusability. Reusability plays very crucial role in the large-scale software development environments. Effort is made to establishing a realistic return on investment on a reuse program [4, 6]. It is essential to inserting reuse into a corporate software development process, and that clearly stating the potential benefits of reuse in financial terms has proven to be a powerful motivator.

A reuse metrics and return on investment (ROI) model that distinguish the savings and benefits from those already gained through accepted software engineering techniques are defined [2]. Three reuse metrics are derived from readily available and observable software data elements. The metrics are used in the return on investment model to establish sound business justification for reuse. The paper examines different modeling approaches and methods to measure software reusability. Knowing what makes software reusable can help us learn how to build new reusable components and help us to identify potential useful modules in existing programs [5, 8].

The paper begins by establishing various modeling efforts to identify potential useful modules in existing programs [5, 8]. The paper examines various benefits of reusability and proved it with the help of mathematical modeling. Formulas were derived for calculating the potential benefits in terms of measurable quantities like cost, time and efforts [7].

II. EXPERIENCE OF SOFTWARE REUSE:

The objects for reuse can be any kind of information handled during software development not only concrete information such as code, design, requirements, and test cases, but also abstract information such as development processes and domain knowledge [10, 12].

A. Various Reusability Metrics

The reusability metrics are established by the following two methods and these methods are Empirical methods and Qualitative methods [13]. Empirical method is consists of the following modeling techniques which are based on the criteria of

- Module oriented
- Size Based
- Reliability based
- Complexity based
- Component based.

Qualitative methods consists of the following modeling techniques and these are as follow

- Module oriented style guideline
- Component oriented certification guidelines
- Quality guidelines.

Domain modeling is one of the important concepts which should be implemented while deciding the reusability, so modeling framework is outlined, and then meta-schema is introduced. This is followed by description of the domain abstractions.

III. VARIOUS MODELING TECHNIQUES FOR REUSE MEASUREMENTS

Cost and productivity

\[ C = (B - 1) R + 1 \] and productivity \( P = 1/C \)
C = Cost of software development  
R = Proportion of reused code in the product  
B = Cost relative to that of all new code of incorporating reused code into the product.  

Cost of development model  
\[ C = (B + \frac{E}{N-1}) R + 1 \]  
E = Cost of developing a reusable component relative to the cost of producing a component that is not to be reused.  
N = The number of uses over which code cost will be amortized.  

Quality of investment  
\[ Q = \frac{B}{P} \]  
If Q < 1 then the reuse effort resulted in a net loss.  
If Q > 1 then the investment provided good returns.  

IV. SOFTWARE REUSABILITY DEVELOPMENT BENEFITS  

DB = \[ \sum_{i=1}^{n} \left( \text{average Normal Code unit cost} - \text{reused unit code cost} \right) \]  
There is non-linear Relationship between system size and system costs.  

Reused Cost = Normal Cost \times (1-RCR)  
(RCR) = Relative cost of Reuse.  

Reuse with modification  

Reused Cost modified = DB (Reused Cost modified)  
DB = Normal Cost – (Reused Cost - modification unmodified + modification cost).  

V. REUSED CODE UNITS AND COMPONENTS  

If reuse is not component based, the part, which has been reused rather than developed, can be considered one component.  

DB = \[ \sum_{i=1}^{\#\text{systems}} \sum_{j=1}^{\#\text{components}} \sum_{k=1}^{\#\text{reuses}} \sum_{l=1}^{\#\text{code units}} (C_1 - C_2) \]  

\[ C_1 = \text{Normal Cost}_{\text{.business}}, C_2 = \text{Reused Cost} - \text{cost without reuse}\_{\text{business}}, C_3 = \text{cost with reuse}_{\text{business}} \]  

Instead of summing we can take averaging for component reuses.  
\[ \sum_{i=1}^{n} x_i = \frac{1}{n} \left( \sum_{i=1}^{n} x_i \right) \times n = \text{average } x \times n. \]  

DB = Normal – avg. reused  

Average normal costs = \[ \frac{1}{\#\text{components}} \left( \sum_{i=1}^{\#\text{components}} \text{avg. normal cost}_i \right) \]  

Reused cost = \[ F^b + U^b + I^b + N^b + P^b + O^b + F^w + U^w + M^w + I^w + N^w + P^w + O^w. \]  
Where b = black box reuse (without modification).  
w = white box reuse (with modification).  
F = cost to find reusable software (location cost).  
U = cost to understand the reusable software.  
I = cost to integrate reusable software.  
M = cost to modify the reusable software (white box only).  
N = cost to develop new software if the reuse attempt fails.  
P = an incentive payment to the reusable software producer.  
O = other reused costs not mentioned above.  

Development cost  
\[ DC = \sum_{i=1}^{\#\text{components}} \sum_{j=1}^{\#\text{code units}} (\text{Normal cost}_{i,j} - \text{Reused cost}_{i,j}) \]  

VI. MAINTENANCE COST AND BENEFITS FROM REUSABILITY  

The cost of maintenance with reuse is the same as that without reuse.  
High quality reusable software results in consumer benefits. Quality can increase directly through extra testing by the producer and also indirectly through feedback (bug report) from the consumers of the software.  

MB = \[ \sum_{i=1}^{\#\text{systems}} \sum_{j=1}^{\#\text{components}} \sum_{k=1}^{\#\text{reuses}} \sum_{l=1}^{\#\text{code units}} (C_1 - C_2) \]  
\[ C_3 = \text{cost without reuse}_{\text{business}}, C_4 = \text{cost with reuse}_{\text{business}}. \]
MC = \sum_{i=1}^{n} \sum_{j=1}^{m} \left[ \text{Producer's maintenance cost}_{ij} \right]

Software Reusability metrics, models and analysis carried out suggests quite strongly that Reuse Software in the circumstances where there are economic and financial benefits to be gained and this we can save the clients money and can have better customer Relationship and can compete in the market with competitors.

Software industry treats reuse in a financially desirable way. In the software industry, an investment should pay back.

In the software industry the accuracy of the results of reusability are directly related to the quality of the data that is fed into the model.

Accuracy is nothing but closeness to reality. Results may not give always accurate results so for this sensitivity analysis is done.

VII. NVP ANALYSIS FOR SOFTWARE REUSABILITY

\[ NVP = \sum_{j=1}^{n} \frac{CFY}{(1+d)^{y}} \]

\[ PI = \frac{\text{Total Benefit}}{\text{Total Cost}} \]

Benefits of software Reusability and mathematical modeling

Mathematical modeling is proved to be very useful for validation and verification of the software reusability metrics.

The other benefits of the software metrics are:-

Development Benefits

Quantification of the benefits and costing validation

Use of economic models for validation

Economic models of reuse can help in taking decision concerning reuse investment

Economic models tell about financial property of reuse, cost saving and profitability ratio.

Reuse metric emphasize on quantity of reuse in a system and value addition through reusability.

VIII. CONCLUSION

The concept of reuse in the software development environment involves decisions related to reuse or not in the development environment. Reusability brings financial benefits as well as it reduces development time, development cost, and brings direct and indirect benefits. We have quantified those parameters which quantify software development cost A piece of code unit or a component can be used, and based on these two parameters cost and benefits analysis of reusability is quantified with the help of mathematical formulas and software metrics for reusability.

REFERENCES