An Approach For Calculation Of Reusability Metrics Of Object Oriented Program

Avinash Dhole and Nehil Rao Nirmal
Assistant Professor of Department of C.S.E., R.I.T. Raipur

Abstract

Software Reuse increases the productivity and reduces the cost and improves the Quality of the software development. Reusability is one of the most important Quality Characteristic. Therefore it is necessary to measure the reusability of the module in order to realize the reuse of module effectively. By reuse the module of existing software has increased in past recent year which impact more on the software quality. Many measuring Reusability methods have been proposed for estimating the reusability of module.

1. Introduction

Reusability is the degree to which existing module or component used in Software system or new project.

Software Reuse reduces the cost, improves the quality and increases the productivity of the software development. Reusability is one of the quality characteristic. Therefore it is necessary to measure the reusability of the module in order to realize the reuse of module effectively. The practice of reuse of existing software has increased in recent years which have a great impact on the software quality.

Any Module can be reused in software development process in two ways which are:

→ Reuse the module without modification: it is the extent of ease with which the existing module can be used in the new development.

→ Reuse the module with modification: it is the extent to which some part of the module with modification can be reused in the development.

If we reuse the oldest module which is already develop in new system it increase the productivity because effort of the programme is reduce and it correctness is also increase because the exist system already tested.

2. Why reuse is important

Many Organization and department, by increasing the level of the software reuse, save the time and development cost taken to develop the software. U.S. Department saved 300 million $ by increasing the 1% software reuse.[1] Reusability measurement is providing the way to build and identify the reusable modules from existing program. Existing programs contain the knowledge and experience of the developers who are expert in particular application domain. So if we extract information from existing program which meet the needs of the software organization then it is beneficial for the organization.

These are the some example where reuse helps

- In Missile Systems Division (MSD) using the software reuse concept it increased the 50% productivity.[3]
- American Navy uses the reusable modules which reduce 26 % of labor required to develop and maintain the Restructured Naval Tactical Data Systems (RNTDS).[4]
- Magnavox saw if we are using the reusable modules to develop the Force Fusion System Prototype (FFSP), it reduces the 20% of the development time of estimated time for developing the new system.[3]

3. What can be reused?

In Software development life cycle Reusability concept is not limited to only coding phase. We used it from requirement phase to last stage of the software development. There are various phase in software development where reusability is used: Code,

Requirements, Architectural/design documentation, Test Plan, Specification and Design.[7][8]

4. Definition of reuses types

Bieman et al's [9] defined the several types of reuse and defined the three types of reuse in the three perspectives. These are explained as follows:

A) Public Reuse: Fenton define the public reuses as "the proportion of a product which was constructed externally "[5].

Public reuse = length (E) / length (p);

E is the code developed externally. P is the new system including E.

B) Private Reuse: Fenton defines private reuse (or perhaps more appropriately internal reuse) as the "extent to which modules within a product are reused within the same product" [5]. Fenton uses the call graph which represents the flow connection of the module .In these graphs node represents the module and they are connected through edges. If one node calls another node then edge displays the connection between them. Fenton provides the formula for calculating the private reuse in call graph as follows: [2]

R (G) = e – n + 1;

e is total no of edges in graph. And n is the number of the nodes in graph.

C) Leveraged Reuse: In Leveraged reuse means a modification of reuse is allowed. [6][8]

D) Verbatim Reuse: In Verbatim reuse means a modification of reuse is not allowed.[6][8]

5. Methods for calculating reusability

Reusability is depending on the portability, adaptability, understandability, maintainability and reliability. Here we are dealing with the java program that way portability is not issue for us. And we are dealing with static code therefore we are not considering the reliability as an affect which affects the reusability, because reliability is measured in terms of mean time and fault which is measured during the execution of the program.

Understandability depends on the complexity, size and documentation level of the programs. [1] Complexity is of two types' structure complexity and inheritance complexity.

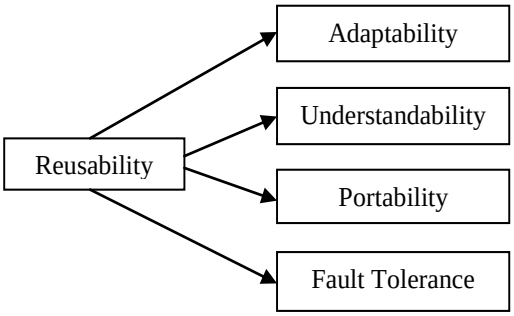


Figure 1

Adaptability: Today suddenly business environment or business need is changed, so handling this situation adaptability is one of the important factor or weapon. Business market situation is change frequently so our software system should be adaptable to satisfy this requirement. In object oriented concept using the ability to build adaptable software. It doesn't mean whatever software we develop from OOP is always adaptable. For make adaptable module we concentration on coupling and cohesion of the module. If the coupling of module is low and cohesion is high that means module is easily adapt in new environment from old environment.

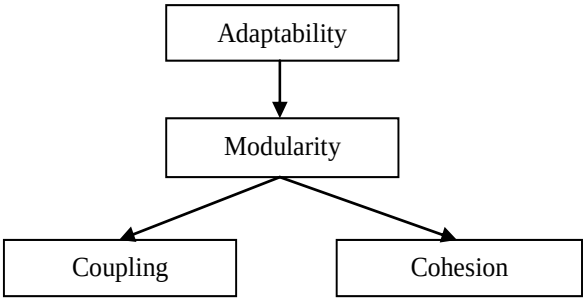


Figure 2

Coupling can occur if

- A method of one class is invoked from another class
- An attribute of one class is modified / used by a method of another class
- An attribute is defined in terms of something defined in another class

We are using the Coupling Metrics for determine Coupling between the classes. This metrics is also

determining the indirect Coupling between the classes. Suppose system having the Set of class C = {C1, C2, C3….Cn} and Mj = {M1, M2, M3 ….Mk} is the set of the methods which is having in class Cj.And Ri,j is defined as no of the method and instance variable in class Cj which is called by class Ci. [9]

We define Direct Coupling between class I to class j CoupD (i,j), as ratio of number of methods of class j called by class I to total no of the methods in the class i.

Measure the coupling of the class is defined as

$$\text{Class Coup} = \sum \text{Coup (i,j)} / (m*m-m);$$

We use the Cohesion Metrics for determining Cohesiveness of the class. This metrics also determines the indirect cohesion between the methods.
Suppose class have a set of methods M = {M1, M2, M3… Mn} and Vj = {V1, V2, V3… Vm} is the set of instance variable used by method Mj. [9]

We calculate the direct similarity of two methods Mi and Mj by using this formula.

$$SimD(i,j) = |V_i \cap V_j| / |V_i \cup V_j|$$

The cohesion of the class is defined as

$$ClassCoh = \sum \text{Sim (i, j)} / (m*m-m)$$

Understandability: is the degree in which the meaning of the system or module should be clear to the user or developer. Understandability depend on the following factor, these are documentation level, complexity and size. If module are well documented then understandability of the module is high i.e. module having more comment line so new developer understand module code easily, because what does function do describe in the beginning of the function. If the Complexity of the module is high then module is difficult to understand.

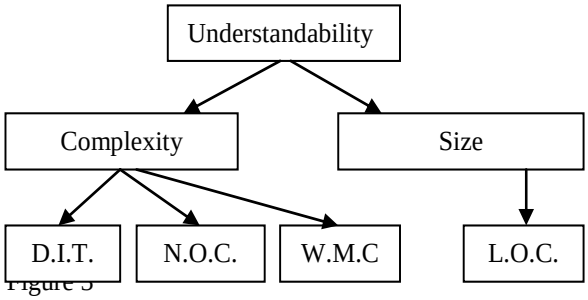


Figure 3

Weighted Method per Class (WMC): this metrics is used for calculating the structure complexity of the programs. WMC is sum of complexity of the all methods which is implemented in class. And method complexity is measured by using Cyclomatic Complexity.
Suppose class is having the methods (m1, m2, and m3…mn) and complexity of the methods are (c1, c2, and c3…cn) then

$$WMC = c1+c2+c3+\ldots. +cn;$$

Cyclomatic Complexity has foundation of the graph theory and is computed in one of the 3 ways. [8]

Number of regions in flow graph.

Cyclomatic Complexity defined in flow graph as follow

$$C (G) = E – N +2;$$

Where E is the no. of the edge in the graph and N is the no. of the nodes in graph.

Cyclomatic Complexity defined in flow graph as follow

$$C (G) = P+1;$$

Where ‘P’ is number of predicate nodes in the graph. Statement where we are taking some decision are called predicate node.

Depth of Inheritance Tree (DIT): This metric is used for measuring the inheritance complexity for the programs, when programmer uses the inheritance in his program then this Metric can be used.

DIT is the Maximum depth from the root node of tree to particular node. Here class is represented as a node. Deeper node in the tree has more no of the methods because they inherit the more classes in the tree and it makes the class more complex.

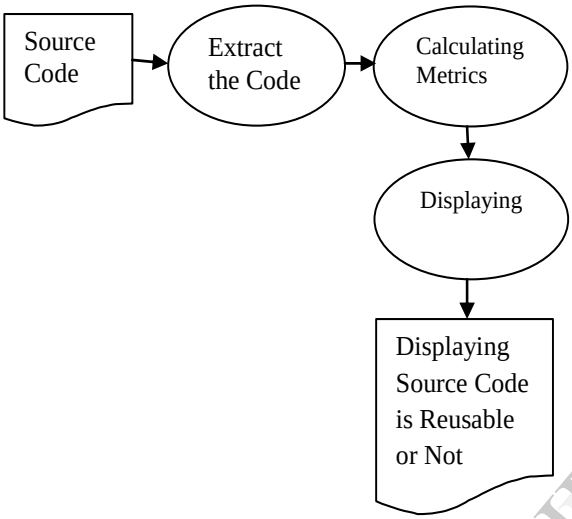
Number of Children (NOC): NOC is defined as Number of the Sub- Classes of the Particular class in hierarchy of the class. If class has more children then it requires more testing because subclass may misuse the super class.

Lines of Code (LOC): This metrics used for measuring the size of the program by counting the number of line in program. Lines of Code (LOC) counts all lines like as source line and the number of statements, the number of blank lines, and the number of comment lines.

Portability: is depended on the independency of system. Java file or .class files are system independent. We can find the relevant reusability of program through the system independency.

6. Approach for identification of reusable module

For identification of reusable module consist following step and these are: Analyzing the source code and calculating the all metrics and displaying the result. These steps are shown in figure 4.



7. Conclusion:

The Purpose of this method is finding the approach and way to calculate reusability of object oriented programs. Reusability is one of the quality attribute and it is of prime importance in object oriented software development as reusability leads to increase in developer productivity, reduce development cost as well as reduce time to market.

8. References:

[1] Anthes, Gary I I., "*Software Reuse Plans BringPaybacks*," Computeworld, Vol. 27, KO. 49, pp.73-76.

[2] [BB81] J.W. Bailey and V.R. Basili. "*A meta-model for software development resource expenditures"*. Proc. Fifth Int. Conf. Software Engineering.Pages107-116. 1981.

[3] [CDS86] S.D. Conte, H.E. Dunsmore, and V.Y. Shen, "*Software Engineering Metrics and Models"*. *Benjamin*"Cummings, Menlo Park, California 1986.

[4] [Boe81] B. W. Boehm. "*Software Engineering Economics"* .Prenntice Hall, Englewood Cliffs, NJ, 1981.

[5] [Fen91] Norman Fenton. "*Software Metrics A Rigorous Approach"* .Chapman & Hall, London, 1991.

[6] [Sel89] Richard W. Selby. "*Quantitative studies of software reuse"*. In Ted J. Biggersta and Alan J. Perlis, editors,

[7] Software Reusability Vol II Applications and Experiences, Addison Wesley, 1989.

[8] http://www.indiawebdevelopers.com/articles/reusability.asp

[9] [CK93] Shyam R. Chidamber, Chris F. Kemerer, "A *metrics suit for object oriented design"*,1993