**Line of Code**

The express “lines of code” (LOC) may be a metric by and large utilized to assess a software program or codebase concurring to its estimate. It could be a common identifier taken by including up the number of lines of code utilized to type in a program.

**Total Comments**

In computer programming, a comment may be a programmer-readable clarification or explanation within the source code of a computer program. They are included with the reason of making the source code less demanding for people to get it, and are by and large overlooked by compilers and translators.

**Weighted Method Count (WMC)**

The Weighted Method Count or Weighted Method per Class metric was originally defined in A Metrics Suite for Object Oriented Design. The WMC metric is characterized as the sum of complexities of all methods declared in a class. This metric is a great indicator how much exertion will be vital to preserve and create a particular class.

**Lack of Cohesion between Methods (LCOM)**

Lack of cohesion in methods. Cohesion refers to the degree of the intra-relationship between the elements in a software module such as packages and classes. It is ideal that each element has a strong relationship in the module by achieving a particular functionality. The LCOM metric indicates a set of methods in a class is not strongly connected to other methods

*𝐿𝐶𝑂𝑀= (𝑚∗𝑎/𝑣)/m-1*

Where,

I. m: the number of methods in the class

II. a: the number of methods in a class that access an instance variable.

III. V: the number of instance variable

**Attribute Hiding Factor (AHF)**

This is the degree of the invisibilities of attributes in classes. The invisibility of an attribute refers to the rate of the overall classes from which the attribute is invisible. It can be calculated by summing the invisibility of each property with regard to the other classes within the project. Within the previous calculation of invisibility, private=1, public=0, protected = Size of the legacy tree / Number of classes. AHF is a fraction.

*Attribute hiding factor = (Sum of the invisibilities of all attributes defined in all classes) / (Total number of attributes defined in the project)*

**Method Hiding Factor (MHF)**

MHF is the measure of the invisibilities of methods in classes. The invisibility of a method refers to the percentage of the total classes from which the method is invisible. It can be calculated by summing the invisibility of each method in respect to the other classes in the project. In the previous calculation, private=1, public=0, protected = Size of the Inheritance tree / Number of classes. MHF is a fraction.

*Method hiding factor = (Sum of the invisibilities of all methods defined in all classes) / (Total number of methods defined in the project)*

**Attribute Inherited Factor (AHF)**

This is the fraction of class attributes that are inherited. The expression for calculating it requires summing the inherited attributes for all classes from its super-classes in a project.

*Attribute inheritance factor = (Sum of inherited attributes) / (Total number of available attributes)*

**Method Inherited Factor (MHF)**

This is obtained by dividing the total number of inherited methods by the total number of methods. The total number of inherited methods is obtained by summing the number of operations that a class has inherited from its super-classes.

*Method inheritance factor = (Sum of inherited methods) / (Total number of inherited methods)*