CHAPTER 6
CONTRIBUTIONS OF THESIS

In this work, four object oriented metrics have been proposed to improvise the cost, effort and schedule estimation of a project. They are Method Reuse Factor (MRF) and Attribute Reuse Factor (ARF), Method Reuse Factor with Inheritance (MRFI) and Attribute Reuse Factor with Inheritance (ARFI) which are used to provide quantitative measurement for reuse in product and process.

MRF and ARFI have been validated by considering two types of data sets for Design Function points counting method and COCOMO estimation.

6.1 METHOD REUSE FACTOR

In any project estimation the project size, effort, schedule and cost are considered as important characteristics in over all planning of the project in achieving the objectives and goals. In the process of proposing new metrics, the Method Reuse Factor has been introduced as a reuse metrics in this work. This concept mainly makes use of the reusability of methods and there by reducing the size of the code.

Different projects with different KLOC has been considered in this work to show the use of proposed metrics and in the estimation of effort, time
and cost. These parameters have been calculated and applied to COCOMO model and results were obtained for cases with and without using MRF. From these it is observed that there is 20% of improvement in estimation accuracy when MRF is considered and hence a total of 85% accuracy has been established.

This analysis has been made for all the three modes of COCOMO model. In this process, 96 projects of different sizes of KLOC, have been used for comparing the cost, effort and schedule. The results were consistently providing around 85% of accuracy in all the cases.

6.2 ATTRIBUTE REUSE FACTOR WITH INHERITANCE

Attribute Reuse Factor with inheritance is a new metric that has been proposed in this research work. This Reuse Factor exactly gives how many data items are used in the derived class. In case of OODFP counting procedure, this ARFI considers the actual number of data items that are used in the derived class. The major advantage of this work is that the attribute reuse factor with inherited classes provides a 26% of improvement in the overall counting procedure. If this metric is used at the design phase then it will be much effective in reducing the design complexity.

In this work, fifty (50) different projects of different size and complexity have been taken up and they have been tested with the newly introduced ARFI. The results obtained showed that there is a consistent improvement in the estimation of errors varying between 25% to 30%.

When the metric was applied on COCOMO model, the improvement observed from organic to embedded is 20.5 to 23.5 percent in effort, 8.25 to
8.5 percent in time and a mere 26.6 to 26.8 percent in the cost over the conventional COCOMO estimation.

Though there is very less impact on lower size projects, as the size of the project increases the impact of method reuse factor for effort and cost are very much considerable.

6.3 COMPLEXITY METRIC MODEL

In this thesis work, apart from defining new metrics, a complexity metric model for Object Oriented Systems has been developed. Moreover, an empirical investigation has been carried out to find the relationship between class size, reuse level and software complexity. It is found that the overall complexity of an Object Oriented System can be expressed effectively as an weighted complexity measure that includes both class and programme complexities.

Another major contribution of this work is that it provides techniques for empirically investigating the relationship between structural properties such as Reuse level of and complexity of OO systems. The metrics data have been collected from various C++ programs and they have been analysed, in which the average depth of inheritance tree, Reuse ratio and Specialization ratio were calculated.

From these experiments, it is observed that to estimate over all complexity, of object oriented systems merely class characteristics are not enough. Hence, it is necessary consider the physical code coverage by the classes for estimating the overall complexity of OO programs that provides the Weighted Complexity Measure (WCM).
These metrics can be determined at the design level itself. Threshold values for these metrics provide a way to control complexity, error rate, and maintenance of OO programs. More empirical study helps in estimating valid threshold values for these metrics.

It is observed that class size in LOC and class size in NOA + NOM does not show proportional relationship. But LOC is considered as vital indicator of size and effort for any program in spite of many criticisms. NOM is not a better metric to estimate class size.

Empirical observations show that threshold of NOM a good measure for controlling error rate complexity and also increasing understandability. It is suggested that one should make intensive study to ensure that metrics are valid. One can bring out a complete complexity model for different OO languages.

6.4 METRICS FOR USER CENTRED DESIGN

In view of user centred design, an enhanced software reusability model and reusability attributes have been proposed.

Various software reusability metrics have been proposed to relate the user capability, enterprise benefits, interface information and user application along with the physical and special architecture. The enhancement of software reusability is so modelled as to support the required devices utilized in the human computer interactions. This is explained with the help a matrix model.
6.5 FUTURE WORKS

The application of the new metrics such as Method Reuse Factor and Attribute Reuse Factor with inheritance were applied to few projects and the results were encouraging.

However, these metrics may be extended to work with more and different types of Real time projects so that it is possible to measure the size of the project and its complexity more effectively. Further, this system may be tested with data on ARF without inheritance and MRF with inheritance with a number of projects and the validity of these metrics can be verified further.

Finally, the user centred design can be provided with new and special metrics so that the user interface design can be tested more easily.