Chapter 7

CONCLUSION AND FUTURE WORK

This chapter consolidates the findings and attempts to bring into practice the metrics usage to the design phase of object oriented software development.

7.1 CONCLUSION

A software metric is a standard of measure of a degree to which a software system or process possesses some property. Since quantitative measurements are essential in all sciences, there is a continuous effort by computer science practitioners and theoreticians to bring similar approaches to software development. The goal is obtaining objective, reproducible and quantifiable measurements, which may have numerous valuable applications in schedule and budget planning, cost estimation, quality assurance, testing, software debugging, software performance optimization, and optimal personnel task assignments.

Object-oriented design methodology is the most powerful mechanism to develop efficient software system. Object-oriented software development approach is totally different from other traditional methods as it uses quality metrics. Quality metrics is a unit of measuring software based on quality that is used to characterize software applications and processes, effectiveness, productivity, maintainability, portability, defines the quality parameters of efficient software.
The Software Assessment Model has the lower-level design metrics which are well defined in terms of design characteristics. The quality is assessed as an aggregation of the model’s individual high-level quality attributes. The high-level attributes were assessed using a set of empirically identified and weighted object-oriented design properties, which are derived from object-oriented metrics which measures the lowest-level structural, functional and relational details of a design.

The effectiveness of this initial model in predicting Software quality attributes has been validated against real-time projects. The quality predicted by the model shows good correlation with evaluator’s assessments of project designs and predicts implementation qualities as well. For the predicted defective modules priority has assigned according to the defect severity and price will be estimated. The model’s ability to estimate the overall Software quality from design information has also been demonstrated using several functionally equivalent projects. These projects have an overall quality estimate computed by the model which has a statistically significant correlation with the assessment of overall project quality characteristics determined by the independent evaluators.

This research work attempts to bring into practice the metrics usage to design phase of object-oriented software development. The research also includes new metrics, which gives more confidence as the result of its inclusion. The Software Assessment Model that is proposed has generated quality indication to real-time project practiced in the industries. The focus of this work is on the impact of the object-oriented metrics in the design. The
design is an early stage of the software life cycle in which the benefits of software metrics can be apparent.

The result supports the concept which has consistently proved in its quality. The model has also evaluated the redesign efforts with real time adjustments quantitatively to suggest the redesigning efforts for design stage imperfection, which can aid the project management team to calculate the redesigning efforts for object-oriented projects in the design stage itself. The work also involves implementing a reverse engineering technique for object-oriented projects designed through UML for Java environment.
7.2 FUTURE WORK

Use of UML Behavioral diagrams as Input

In this work we were mainly concerned about design measures extracted from class diagrams. We are planning to investigate which are relevant metrics for dealing not only with the static structure of models, but also with its dynamic models. For such metrics, formalization of dynamic model to be used to and can also express the behaviour of the system over time. It includes support for activity diagrams, state diagrams, sequence diagrams and extensions including business process modeling.

Applying the developed tool in a large scale software project

The benefits of the developed tool would not be fully realized, until used in a large scale industrial software development project. This would help assessing the applicability of the proposed idea in the different types of project

Studying the effect of object-oriented features on ease of implementation

Other object-oriented feature implements a given re-design. Using polymorphism allows overloading and overriding, however, it increases the cognitive complexity needed to understand. Such a study would result in a sort of thresholds or best possible ranges for metrics such as other object oriented features