CHAPTER 8

CONCLUSION AND FUTURE SCOPE OF WORK

8.1 OVERVIEW

The research aimed at providing a mechanism for selecting a reliable Open Source Software by means of Metric Suites. Initially, this research validated three different metric suites and identified their predictability of corresponding fault proneness in the various components of OSS. The metric suites such as Chidamber & Kemerer, Robert C Martin and McCabe’s were utilized for the research. The intention behind the choice of metric suites is that these have been validated as better predictors in the object oriented designs. CK metrics is widely accepted as good quality predictor and the other two metrics suites required empirical validation in the OSS design and hence these were undertaken. Different versions of Rhino, an OSS, were taken for validating the metric suites. The utility of Open Source Software was examined to ascertain its reliability. The scope of research limits itself to the OSS developed using Java. Algorithms, to identify the error rate of the classes, methods & packages, were formulated to conform to the aim of the research. The results of the research emerge to be good to validate and to provide guidelines for the users in the process of identifying reliable OSS and also in the development of OSS.

8.2 FINDINGS AND CONTRIBUTIONS

The research was carried out to validate the metric suites in the open source environment and to provide guidelines for the appropriate selection of metric suite for computing complexities of OSS and eventually put forward an automated mechanism for gathering metric values. When the proposed mechanism is implemented, the problems faced, in the development and the usage of the Open Source Software, are addressed.

8.2.1 Empirical Validation of Metrics Suites

A typical empirical validation of object-oriented metrics proceeds by investigating the relationship between each metric and the outcome of interest. The validated metrics were Chidamber and Kemerer metrics, Robert C. Martin Suite and McCabe’s Metric Suite. Rhino software was analyzed to predict the software quality with different versions in terms of fault proneness with these metric suites. The OSS was analyzed through metrics suites to extract the corresponding metrics. These metric values were correlated with the bug report obtained from reliable sources. From the outcome of the analysis, it is concluded that the CK metric suites are effective in detecting the complexity of the methods, the McCabe metrics suite are a good predictor of class complexity and the Robert C Martin metrics suite is better in predicting the faults in terms of the packages. The empirical analysis aids software professionals to find out the metric suites that can predict faults while developing the quality software products using the OO approach in OSS design.

8.2.2 Fault Tolerance in OSS

Detection of tolerance level for the given OSS is achieved through the methodology proposed in this research. This process is automated by means of an algorithm in Chapter V. This process will be useful for the OSS user community in identifying the reliable software among various alternatives. In this research, the possible number of fault that could occur in an OSS is identified through the proposed algorithm. As the process is automated, it eases the work load of the users in the OSS community. In this work, the algorithm is applied to Java-Based Open Source Software. Thus, by implementing the efficient algorithm to carry out the process of tolerating the fault in the Open Source Software up to the certain limit is carried out to meet the scope. Thus this research work will certainly help the users and make the rate of accessing the Open Source Software increase.

8.2.3 Mapping Class Path File

An automated search engine to deduce the errors occurring in the JAR file mapping and the mapping up of correct path automatically by detecting the correct path of the JAR file is designed. This work has proposed an efficient methodology to be used in mapping of class path file. This is useful in the development of the program in the use of JAR files. If the class paths in the JAR files are mapped precisely then the developed open source software will be successful. The error occurrence due to the JAR file mapping is reduced with the use of the automated technique.

8.2.4 Reliability Framework for OSS

Reliability is a key factor in the reuse of OSS by the user. A framework comprising of empirical validation of the metrics suites and threshold values, fault proneness and tolerance value are used. The proposed framework constitutes of two phases, where the first phase empirically validates the metric suites and the second phase analyses the software with the fault proneness and tolerance value. Thus, a suggestion is provided based on the reliability of the OSS.

8.3 APPLICABILITY OF THE RESEARCH

The research is of great pragmatic value to the OSS community since it can be used by the users of the OSS community in identifying the reliable OSS from among the various alternatives. For the OSS developers, this research would be of immense help in providing quality open source software by applying various reliability mechanisms proposed in this research. The same mechanism shall be provided as an open source which could be at the disposal for the OSS.

8.4 LIMITATIONS

In this research work the metric suites adopted for empirical validation are the selected three metric suites only, namely: Chidamber and Kemerer Metric Suite, Robert C Martin’s Metric Suite and McCabe’s Metric Suite. Many other metrics are not taken into consideration. The Rhino open source software alone is used for testing and analysis; the experimental verification could be varied in order that the methods proposed are validated for competence. The aspects taken into account to probe into the OSS were fault proneness, error rate. The prospects could involve overall aspects of the OSS such as maintainability, extendability and the like.

8.5 FURTHER RESEARCH

The scope has been limited to only Java based OSS, this could be extended to other OO systems. This research work could be carried out for varied Open Source Software environment by extending the features and aspects of this work. Inclusion of other aspects to rectify the possible errors automatically, which will reduce the user’s burden, may be implemented.

Metric suites taken by and large may enhance the research work on validations of the open source software environment. The metrics suite can be validated for other quality indicators, namely, reliability and maintainability.