CHAPTER 6

DISCUSSION AND CONCLUSION

The reliability of software is more critical issues in real time systems. Software reliability is defined as the probability that software will perform its intended function during a specified period of time under stated conditions Smidts (1998).

Software reliability models used for estimating reliability of the system. However, the widely used point estimators are subject to random variations in the data, resulting in uncertainties in these estimated parameters. Thus there are many unanswered questions concerning the underlying assumptions, accuracy, competence, applicability and validity of software reliability models.

In literature review, internal quality is the main criteria to achieve reliability .If that is poor, then it will increase the maintenance costs. For improving internal quality, the reliability models Dohi et al. (2004) are used. Early prediction models uses characteristics of the software development process from requirement to design and test, and extrapolate this information to predict the behaviour of software during operation Sing (2001), Cortellessa (2002) and Wood (1996). Software reliability growth models (SRGM) analysis the failure occurs during testing of software and operations on real world. In earlier prediction model, trends are observed in the failure data to derive reliability prediction.
6.1 OBJECTIVE OF THE RESEARCH

The objective of the proposed work is focused to develop models for fault forecasting, prevention and removal. It involves formulation of the fault/failure relationship, an understanding of the operational environment, the establishment of software reliability models, developing procedures and mechanisms for software reliability measurement, and analyzing and evaluating the measurement results. The following works are carried out to achieve the aforementioned objective

- Reliability Factor Approach for uncertainty analysis in software development
- Enhanced Software Reliability model based on Logistic Effort Test Function.
- Improving Internal Quality of the Software Using Intelligence Code Evaluator.
- Procedure for analyzing Reliability Data.
- Co-operative approach and Enhancement techniques to improve Debugging activities.
- An Effective Approach To Support Multi-Objective Optimization In Software Reliability Allocation For Improving Quality
6.2 DISCUSSION AND CONCLUSION

Reliability Factor Approach work gives the solution for uncertainty problems in reliability modeling on system level. This safe growth model solves the challenges for the dearth of data by using quality factors. The model gives expert knowledge, historical data, and developmental environments. This expert knowledge involved in analyzing the uncertainty and for compensating insufficient failure data. After analyzing the problem, this work further extends to more complicated systems that contain numerous components, each with its own respective distributions and uncertain parameters.

The second approach presents software reliability models based on logistic effort test function applying logistic testing effort function to the Yamada ‘S’ Shaped model the reliability of the software can be improved by predicting more no of errors when compared to the normal Yamada ‘S’ shaped model. Logistic testing effort function based Yamada ‘S’ shaped model predict more no of failures than any other model as evident from the table. Hence Yamada ‘S’ shaped model with logistic testing effort function can be applied to time & failure sensitive project and generic software.

A new approach to evaluate source code quality and source code profiling. In order to do so, it defines source code metrics and analyzes its internal behaviour with data mining and machine learning procedures. In the process, a dataset created by a data mining algorithm is the reference classified data source to set up a neural network. The trained multilayer perception has shown excellent precision to classify this sort of data. An
expert system analyzes the classification results and identifies every inaccurate source code usage. In order to test the prototype, a web user interface is being developed. Therefore, this feature will make community.

An effective software reliability allocation technique is implemented with consideration of reliability, cost, and schedule. This is the first system to consider schedule and support the simultaneous optimization of multiple objectives (Reliability, Cost, and Schedule). This helps us to establish schedule planning, which is very important in today’s business. This aspect of the proposed system can increase the effectiveness of resource uses such as cost and schedule. Furthermore, a scheme for adaptation of NSGA-II is suggested in software reliability allocation area. The foundation of multi-objective optimization in software reliability allocation is prepared. Further researches are possible based on the proposed system, which considers activities of software reliability engineering in the early phases and schemes for the improved selection and the optimal solution.

The previous system had the disadvantage of having test with high average sample number which required high resolving power. This is overcome by the proposed system which has the advantage of economy in time requirement and cost. The proposed planning methodology is capable of yielding a highly efficacious test which increases reliability and feasibility of decisions.

The proposed framework helps to understand about the on-going project and also used to infer the current and future situations in the project or reconstruct the project. The applications of the proposed procedures are explained through two real data sets. This studies show that the proposed
simulation procedures can analyze the influence on the performance, and the cost related to software debugging when the number of allocated debuggers changes. This useful, important information can guide project managers in the estimation and adjustment of the staffing needs for debugging systems.