ABSTRACT

Software Engineering is the application of systematic, disciplined and quantifiable approach to the development, operation, and maintenance of Software. It is a direct subfield of Computer Science and has some relations with Management Science. Software development life cycle involves requirement analysis, design, coding, testing and implementation. Software Quality is the most important one, since the success of a Software Engineer relies on the development of failure free software. One of the main quality factors is reliability. Reliability can improve through Software Reliability Models, analyse of reliability data, proper utilization of debuggers, internal quality factors and evaluating the measurement results.

Challenges in software reliability are the lack of available failure data from a single test, which often makes modelling difficult. Further the data poses a bigger challenge in the uncertainty analysis of the software reliability modelling. Bayesian statistical method and Logistic test effort model are proposed to quantify the uncertainties in the software reliability model of system with correlated parameters. It will help to improve Software Quality.
Software quality is assessed by a number of variables. These variables can be divided into external and internal quality criteria. External quality is what a user experiences when running the software in its operational mode. Internal quality refers to aspects that are code-dependent, and that are not visible to the end-user. Internal quality is meaningful to the developer only. The minimum size of source code will lead to reduction in debugging time and cost. This Research proposes a software quality support tool, a Java source code evaluator and a code profiler based on computational intelligence techniques to reduce schedule slippage of development activity. It gives a new approach to evaluate and identify inaccurate source code usage and transitively, the software product itself.

Reliability is the indirect measure to improve the quality of the software. Testing is the process of running a system with the intention of finding errors. Failure Data used for the testing should be reliable. This proposed system focus on improving test by using low average sample number in short term which is having the advantage of economy in time requirement and cost. It produces optimum truncated test called binomial Sequential Probability Ratio Test. The goal of the work is to develop a methodology for planning short truncated test for improving reliability of the software.
Many approaches have been given like rate based approaches for the reliability of the software and to analyze the reasons for the failure of the software. Criteria for the reliability of the software, number of debuggers or developers available are not taken into account. Newly detected faults have to wait for some time since all the debuggers will be busy in detecting the faults which they found previously. Time taken to remove the fault is taken into consideration and the main fact relies in it is that less number of faults been removed when compared to the number of faults detected.

Currently, there are several software reliability allocation techniques, but schedule planning and effective resource use are unavailable because they do not support multi-objective optimization. It is an effective software reliability allocation technique that supports multi objective optimization. This approach assists schedule planning as well as effective resources allocation. This research can aid to increase the effectiveness of resource uses and to establish schedule planning. It will lead to improve Software Reliability.

**Keywords:** Reliability, Fault, Failure Rate, Mean Time Between Failures, Genetic Algorithm, Reliability Growth Model, System Testing, Queing System, Project Management.