Conclusion and Future Scope

It is evident from the literature review that numerous models have been developed to measure, estimate and predict the reliability of computer software. Software reliability has received much attention because reliability has always had obvious effects on highly visible aspects of software development. A large number of software reliability growth models (SRGMs) which relate the number of failures and execution time have been discussed in the software reliability engineering literature. But no SRGM can be claimed to be the best as the physical interpretation of testing and debugging changes due to numerous factors e.g. design of test cases, defect density, skills and efficiency of testing team, availability of testing resources etc.

Every phase in Software Development Life Cycle (SDLC) is important in the generic framework yet testing phase is an extremely important component of SDLC where around half the developmental resources are consumed. Verification and validation of the software takes place in this phase only. It is endeavoured to remove the faults lying dormant in the software. The software testing involves running the software and checking for unexpected behaviour of the software output. The successful test can be considered to be the one, which reveals the presence of latent faults. During testing, resources such as manpower and time are consumed but with increasing importance of cost and time during software development, efficient management of testing phase becomes a high priority issue for an organization.

Thus, testing process in a sense, determines the nature of the failure data. The fault detection rate strongly depends on some parameters like skill of test team, program size, software testability, defect density and resource allocation. It has also been observed that the fault detection rate for all the faults lying in the software differs on the basis of their severity as well. In most of the NHPP based Software Reliability Growth Models, the fault detection rate is assumed to be constant. During a software testing process, there is a possibility that the underlying fault detection rate is changed at some time point called Change Point. This would result in a software failure intensity function either increasing or decreasing monotonically. The work in this thesis inculcates the concept of change point along with the assumptions based on NHPP based modeling approach.
Acknowledging the aforesaid facts various approaches have been used in this thesis to develop Reliability growth models;

(i) **Based on imperfect debugging**: Whenever the debugging process takes place there are very much chances that the bugs are not debugged perfectly or some new bugs comes into the picture. We term this phenomenon to be imperfect debugging. This is a practicality that has been used while developing some SRGMs in this thesis.

(ii) **Based on the severity of errors in the software**: Faults are classified into different categories as simple, hard and/or complex faults. This categorization has also been extended to n-types of faults. Some of the existing research incorporates this phenomenon considering that the fault removal rate is different for different types of faults and remains constant during the overall period of testing. The concept of change point has been implicitly used in modelling framework.

(iii) **Based on Testing Effort**: Various SRGMs have been proposed in the Software Reliability Engineering literature under different sets of assumptions. But most of them do not consider the consumption pattern of resources such as computer time and manpower during testing. More realistic SRGM can result if the reliability growth process is studied with respect to the consumption pattern of the testing effort, measured by CPU hours, number of executed test cases, man-hours, etc. The theory developed in the present study has addressed this consumption done during testing phase by making use of the frequently used effort functions namely; Exponential, Rayleigh, Weibill, Logistic.

(iv) **Based on Unification Schemes**: The plethora of SRGMs makes the model selection a tedious task. To reduce this difficulty unified modelling approach has been proposed by many researchers. These schemes have proved to be successful in obtaining several existing SRGMs by following single methodology and thus provide a insightful investigation for the study of general models without making many assumptions. In the work proposed here, we have made use of the unification scheme based on Hazard rate. The approach proves to be fruitful in obtaining several SRGMs by following single methodology and thus present a perspective investigation for study of general models without making any assumptions.
Moving From Single to Multi-Dimensional Framework: Almost all the SRGMs are developed under the assumption that software reliability growth process depends only on testing-time as the software reliability growth factor essentially. Later some testing resource dependent SRGMs were also developed. Also, there exists testing coverage based SRGMs in the literature of the subject. But all these models do not take into account the simultaneous effect of time and resources or fails to consider the concurrent effect of time and coverage on cumulative number of faults removed from software. Therefore such models can be termed as one-dimension software reliability growth models. And in order to capture the mutual effect of testing time and resources or simultaneous effect of testing time and coverage two dimensional software reliability growth model (2-D SRGM) is needed. Therefore, to take into consideration the effect of time and resources together, we have also developed some flexible 2-D SRGMs.

The software industry can be considered as the typical high technology industry where rate of innovation and knowledge creation plays a pivotal role for continued firm growth. In the last few decades it has been observed that the world of software development management has evolved rapidly due to the intensified market competition. In particular the use of feature-addition model of software products in the industry is fast becoming the commonplace. The up-gradation model can be characterized by increasing the number of features in the software that will give the firm competitive edge in the market. The up-gradation of the system is done by extending it through add-ons, interfacing with other applications etc. Continuous up-gradation of software’s also brings complexity in the systems once it failed to work properly. In recent years, there has been a growing interest to predict the link between the rates of failure and the reliability of software. Many software reliability growth models (SRGM) have been proposed over past three decades that estimate the reliability of a software system as it undergoes changes through the removal of failure causing faults. But unfortunately most of the models didn’t consider anything about the increase in failure rate once an up-gradation is made on the software.

Though we have proposed some SRGMs that incorporate the effect of enhancement of features on software during testing and debugging process. From developers point of view it becomes important to focus on utilizing the resources and deliver the product on scheduled time. Hence, in future we would like to work by developing some related Release policies.

Department of Operational Research, University of Delhi
This is an era of globalization and intensive competition. Therefore firms are frequently coming up with successive versions of their software. We shall develop some SRGMs that take into account the earlier discussed concepts along with multi-up gradation phenomenon.