CHAPTER 7

SUMMARY & CONCLUSION

7.1 SUMMARY

The probability that software faults do not cause a failure during a specified exposure period in a specified environment is called software reliability. In this thesis, we prepare and analyze an array of different procedures related to software reliability based on SPC & SPRT methods. Software reliability growth model can be provide a good prediction from the number of failure at time(t) and the number of remaining failure can be found out. Reliability models are a powerful tool for predict controlling and assessing software reliability. In this thesis we discuss concepts related to the areas of Non-Homogeneous Poisson Process (NHPP), Software Reliability Growth models (SRGM), Statistical Process Control (SPC), Maximum Likelihood Estimation (MLE) and Sequential Probability Ratio Test (SPRT) which are extensively used in carrying out this thesis.

A perfect program is said to be reliable while an imperfect one is said to be unreliable. Many models have been developed with different mathematical techniques to adapt to different testing environments. Generally these models can be categorized into two groups: data-driven software reliability models and analytical software reliability models. Data-driven models focus on the failure data generated through the software testing process. They consider software reliability prediction as a time-series analysis problem. These models are developed from past software failure data and have less restrictive assumptions. Analytical software reliability models describe the software failure behaviour during the software testing process and take this process as a stochastic process. For these models, some restrictive assumptions are made such as perfect and immediate fault correction. NHPP models are applied extensively, which are the main concern for the present thesis.

A reliability growth model is needed to estimate the current reliability level, the time and resources required to achieve the objective reliability level. During this phase, reliability estimation is based on the analysis of failure data. The number of failures experienced can be denoted as a stochastic counting process characterized by its mean value function. This process can be represented by a Poisson model.
Research activities in software reliability engineering have been conducted and a number of NHPP software reliability growth models have been proposed to assess the reliability of software. Software reliability can be estimated once the mean value function is determined. The technique of control chart has been used in the software engineering so as to improve the quality of software products. The software failure process which follows a Non-Homogenous Poisson is monitored on Time domain data.

The data analysis can be done depending on the format in which test data are available, there are two common types of failure data: time-domain (i.e., ungrouped) data and interval domain (i.e., grouped) data. The time-domain approach involves recording the individual times at which failure occurred. The interval-domain approach is characterized by counting the number of failures occurring during a fixed period (e.g., hour, week, day). These data are usually used by practitioners when analyzing, assessing and predicting reliability applications. Some software reliability models can handle both types of data.

Data forms are the basis for analysis, decision and action. Their form and presentation will obviously differ from process to process. After the data is collected, they are analyzed and useful information is extracted through the use of statistical methods. If data is not carefully and systematically recorded, especially at the point of operation, they cannot be analyzed and put to use. The control limits are utilized to monitor the failure times If the plotted point falls between the calculated control limits, it indicates that the process is in the state of statistical control and no action is warranted. If the point falls above the UCL, it indicates that the failure occurrence rate, may have decreased which results in the increase of TBFs. This is an important indication of possible process improvement. If the plotted point falls below the LCL, it indicates that the failure occurrence rate, may have increased which results in the decrease of failure time. This means that process may have deteriorated. This has been discussed in Chapter 5.

Software reliability assessment is increasingly important in developing and testing new software products. Software reliability is the commonly used quality metric. This is estimated by the use of an analytical model whose unknown parameters are estimated from the available failure data. There are a number of views to quantify
software reliability. Parameter estimation is of primary importance in software reliability estimation. In this thesis, the parameters are estimated by MLE technique for the Time domain data. The idea behind maximum likelihood parameter estimation is to determine the parameters that maximize the probability of the sample data. A MLE method is versatile and applies to many models and to different types of data. This is discussed in Chapter 4 with a Numerical illustration.

The SPRT was initially developed for quality control problems to reformulate it as a sequential analysis problem. SPRT is used as a statistical device to decide which of two simple hypotheses is more correct. It has been formulated for use in the computerized testing of human examinees as a termination criterion. This is been discussed with numerical illustration in Chapter 6.

7.2 CONCLUSIONS

Chapters 1, 2 and 3 of this thesis are usual format of introduction, focuses problems, review of literature and methodology to be adopted to the proposed research problems. The results of each investigation are given in the form of conclusion at the end of Chapters 4, 5 and 6. The broad conclusion arrived at in Chapters 4 to 6 are briefly listed below.

Chapter 4 Software reliability is an important quality measure that quantifies the operational profile of computer systems. This model is primarily useful in estimating and monitoring software reliability, viewed as a measure of software quality. In this thesis we have discussed the performances of 6 time domain datasets by using our new Burr type XII SRGM. The experiment result shows that the s2 system test data (Table 4.5) can provide a better goodness-of-fit compared with other datasets. Since, it is having the highest negative value for the log likelihood. This is a simple method for model validation and is very convenient for practitioners of software reliability. The reliability of all the data sets are given in Table 4.8. The reliability of the model over SONATA data is high among the data sets which were considered.

Chapter 5 The failure control chart of Figures 5.1 to 5.6 exemplifies that the first out of control situation is observed at the $9^{th}, 5^{th}, 3^{rd}, 4^{th}, 10^{th}$ and 4th failures with the corresponding successive differences of $m(t)$ falling below the LCL, which results
that the failures were detected at early stages and hence it can be said that the
proposed method is giving positive recommendations for their use in finding out
whether the process is in control or out of control signal. The early detection of
software failures will significantly improve the software reliability. When the
successive differences of $m(t)$ falls below the LCL, the assignable causes for the
process deterioration should be investigated. When the successive differences of
$m(t)$ have exceeded the UCL, the reasons should be identified.

Chapter 6 The SPRT methodology for the proposed software reliability growth model
Burr type XII is applied for the software failure data sets IBM (2002), NTDS (2005),
Xie (2002), SONATA (2010), S2 (1996) and AT &T (2005) is illustrated in Table
6.4.2.Hence, it is observed that we are able to come to a conclusion in less time
regarding the reliability or unreliability of a software product. The results exemplifies
that the model has given a decision of rejection for all the data sets at various time
instant of the data. Therefore, we may conclude that, applying SPRT on different data
sets we can come to an early conclusion of reliable / unreliable of software.