SUMMARY AND CONCLUSION

The present thesis mainly focused on application of process control method for software product, more specifically on the software failure time data. The problem is modeled using statistical techniques and research investigations are made. The results are spread over the four chapters of the thesis (Chapters 2 to 5). Chapter 1 is introduction. The conclusions consolidated in the four chapters are narrated below:

The Failures Control Chart of Figure 2.4 exemplifies that, the first out–of–control situation is noticed at the 10th failure with the corresponding successive differences of m(t) falling below the LCL. It results in an earlier and hence preferable out-of-control for the product. The assignable cause for this is to be investigated and analysed. In comparison, the Time Control Chart for the same data given in Xie et al.,(2002) reveals an out - of - control for the first time above the UCL at 23rd failure. Since the data of the time-control chart are inter-failure times, a point above UCL for time-control chart is also a preferable criterion for the product. The Time Control chart gives the first out-of-control signal in a positive way, but at the 23rd failure. Hence it is claimed that the proposed Failures Control Chart detects out-of-control in a positive way much earlier than the Time-Control Chart, of Xie et al., (2002) indicating a preferability for the proposed Failures Control Chart. On similar lines comparing our Failures Control Chart with the Time Control Chart of Xie et al.,(2002) for the remaining data sets, we have developed the respective Time Control Charts also. However, the Time Control Charts are not separately presented in this thesis in view of their simple calculations. But the results of the comparison reveal that our Failures Control Chart has an early detection of out-of-control signal with regard to the data set of Table 2.7. Similar detection with regard to the data set of table 2.3 is observed. The comparison of other data sets of Table 2.1, Table 2.2, Table 2.4 and Table 2.5 shows the process in control. We therefore say that the Failures Control Chart proposed by us based on HLD in Chapter 2 will have a reasonable role in monitoring software failure process control.
The Failures Control Charts in Figure 3.1 to Figure 3.7 of Chapter 3 developed for the data sets of Table 2.1 to Table 2.7 using Modified Maximum Likelihood Estimation Method shows similar results in comparison with the Failures Control Charts (Figure 2.1 to Figure 2.7 of Chapter 2) developed for same data sets using MLE. We therefore conclude that adopting a modification to the likelihood method does not alter the situation, but simplifies the procedure of getting the estimates of parameters, thus resulting in a preference of the control mechanism of Chapter 3 to the one described in Chapter 2.

The Failures Control Charts of Figure 4.1 to 4.10 have shown out-of-control signals i.e. below LCL. By observing Failures Control Charts, we identified that failure situation is detected at an early stage. The early detection of software failure will improve the software reliability. When the control signals are below LCL, it is likely that there are assignable causes leading to significant process deterioration and it should be investigated. Hence, we conclude that our control mechanism proposed in chapter 4 gives a positive recommendation for its use to assess whether the process is in control or out of control.

The Mean Value charts shown in Figure 2.1, Figure 2.2 and Figure 2.4 and Figure 5.4, Figure 5.5 and Figure 5.6 and the Failure charts shown in Figure 5.1 to Figure 5.3 have shown the similar results. Since both control mechanisms are making the similar detections, any mechanism based on Exponential distribution presented in Chapter 2 or Half Logistic Distribution or Weibull Distribution (Satyaprasad et al., 2011a), (Satyaprasad et al., 2011b) is preferable. Specifically from the statistics point of view, Exponential is a Constant Failure Rate (CFR) model and HLD is an Increasing Failure Rate (IFR) model. If a software product associate with constant failure is developed anywhere, then NHPP based Exponential Distribution is an applicable model.