Conclusions and Future Research Directions

Ever since human evolution took place, technology had a profound effect on human development and on the progress of civilization as a whole. Yet there has been no other time in history when technology has been as pervasive in human lives as it is in today’s time. It has invaded every aspect of human endeavor to an ever increasing extent. Multi-national corporations, government actions, private ventures and individuals are highly dependent on technology for their success. Society’s dependence on technology only promises to intensify as the world moves into the twenty first century.

At present software technology industries are facing a number of challenges. To have a competitive edge, these organizations must meet two major challenging requirements at the same time, (1) be predictable to meet market windows and (2) provide nearly fault-free software.

The study in the thesis discusses various aspects in multi up-gradation SRGMs and successive generations of technologies. We have proposed mathematical models to manage the testing processes for building reliable software and predicting the adoption process of new technology products. This thesis is divided into six chapters. The first chapter provides the impetus behind the work done in the thesis and briefly discusses the fundamentals of software reliability engineering and successive generations of technologies models, optimal release policy and data analysis techniques related to the understanding of the research work.

In chapter 2 we have proposed three SRGMs for software with add on features. As the software firm introduces new add-ons or features, software experiences a drastic increase in failure rate. Due to functionalities enhancements to the feature up-gradations, the complexity of software is likely to increase. Even fixing bugs may induce more software failures due to other defects into software. It is usually seen in practice that no software is bug free. We assume that when the testing of a newly developed code is in progress there
are chances that some faults of the previously released software may be detected and removed. In Section 1 of the chapter, the modeling of multi up-gradation software development process under imperfect debugging environments is discussed. The model is based on the assumption that the overall fault removal of the new release depends on the faults generated in that release and on the leftover undetected faults of just previous release.

Section 2 of the chapter presents the effect of severity of faults on the FRP of software system. The faults have been categorized as simple and hard with respect to the time taken and effort utilized to remove these faults. Here we have incorporated the concept of change in nature of fault during successive release of software. Section 3 discusses a unified modeling framework based on hazard rate function to develop various multi up-gradation model using different failure distribution functions under imperfect debugging phenomenon. We have derived a number of SRGMs using proposed modeling framework. After the model validation, it was observed that model corresponding to GO and Yamada fails to give good result but logistic and Normal distribution functions based models provide better goodness of fit. Parameters of all models have been estimated on four release software fault data set and goodness of fit criteria have been evaluated. The research in this chapter may be extended by using different distribution functions for modeling different releases of the software. Further, more realistic situations may be modeled by taking into consideration the randomness factor associated with the fault detection rate by using stochastic differential equations. The phenomenon of change point can also be used to model a new framework that takes into account the sudden changes in detection/correction rate after certain time point.

In chapter 3, we have first reviewed fault detection-correction process. It has been observed that time between detection and correction is not negligible in practical software testing process. Here a unified framework for Fault Detection and Correction Processes for Successive release of software is proposed. It may be noted that all multi up-gradation model discussed in the thesis is categorized as continuous MUSRGMs. In future, various
multi up-gradation SRGMs can be developed in discrete environment. Also we can use delayed fault detection models by using a debugging time lag function in modeling of MUSRGM.

In section 2, we considered the combined effect of bugs encountered during testing of present release and user reported bugs from operational phase. We assumed that during testing of multi release software, it is quite possible that some of faults of old code are removed directly by testing team of new release (without using any bug reports of operational phase of the previous release of software) and some others are removed on the basis of bugs reported during operational phase. The proposed model takes into consideration the testing and the operational phase as two separate processes which follows Kapur-Garg model and Weibull-model, respectively. In future, this type of modeling framework can be extended for formulating an optimal release policy by taking into consideration the effect of reported bugs on the testing process.

In chapter 4, we have proposed a two dimensional framework to develop a flexible SRGM / MUSRGM using Cobb Douglas production function. This technique captures the combined effect of ‘testing time and testing coverage’ or ‘testing time and testing effort’. In first section we have used a unified framework to develop various two dimensional SRGMs using different failure distributions with change point. The models developed have been validated on real datasets. In future, we may consider various other factors apart from time and coverage, which affect the cumulative number of faults removed in the software.

Later in the Chapter, a two-dimensional multi release SRGM has been formulated. We have developed a cost model with two dimensional modeling technique for deriving the optimal release time and effort before the release of the software. In order to determine optimal time and effort for the release of a new version of the software, multi attribute utility theory has been used. Numerical illustrations are given to justify the release time problems and finally sensitivity analysis is done for few parameters based on time and effort, separately. In our decision problem we have used cost and reliability attributes.
For future we may consider the effect of other attribute like risk and use different optimization techniques like genetic algorithm or ant colony and compare the optimal result with current cost model.

In chapter 5, we have discussed about successive generations of technologies. In first section of chapter a mathematical model has been proposed for number of System-in-Use of successive generation of high technology product. The model proposed in this chapter decomposes the total system in use into (i) Those who are switching from the earlier generation, (ii) those who adopt the new technology instead of the earlier one, (iii) potential customers who would only adopt the new technology and (iv) adoptions for each generation may also come from adopters of earlier generations and make decision to upgrade the product not exactly in the next generation of the product. Moreover, we categorize switching adopters in two groups, i.e. those who previously purchased earlier generation products and proceed to the adoption of the new generation without discarding the earlier ones, or those who discard the earlier generation product and adopt only the new product. We also present a relationship between our model and Norton-Bass model. Empirical implications of the proposed model have been validated on data collected from two industries (Semiconductor Industry DRAM shipments of six generations, Mainframe Industry (USA)). The model describes the growth of these generations quite effectively.

With this model we are able to calculate (1) cumulative number of Systems-In-Use for each generation, and (2) cumulative number of Systems-In-Use related to leapfrogging, switching and substitution for each generation.

By incorporating leapfrogging and distinguishing between substitutions and switching, the proposed model estimation results on IBM framework and DRAM data set are better as compared to some existing models.

Diffusion of product in the market can exhibit traditional symmetric and asymmetric bell shaped curve around the peak. But there may exist a dip/chasm between early and later parts of diffusion curve as shown by adoption of high technology product. It happens
because the early market for such products consist of technology enthusiasts who are quick to adopt the innovation, where as the main market consists of risk-averse decision makers. In future research, we aim to include the effect of dip/chasm in our modeling framework.

In section 2, we have discussed about the determination of optimal introduction time, for high-technology products. The timing decision depends on whether companies invest more time for product design or push the product to market before maturity. The study identifies two attributes such as Customer’s Adoption Indicator and Cost that affect the introduction time of new generation. To trade-off between two decision factors, MAUT is applied in our decision space. Empirical implications of the proposed model have been validated on data collected from two industries. In future, we can discuss the effect of warranty period and price into optimal introduction time.

In chapter 6 a mathematical model has been proposed for studying diffusion of sales of high technology product in two-dimensional framework as time and price. The model proposed in this chapter is for successive generations of product. It decomposes the total sales into first time purchasers, switchers and substituters. We have shown the importance of considering market component for understanding adoption behavior under successive product generation. The proposed model has been compared with same model in one dimensional structure in generations 1 to 3 and 3 to 4 of Semiconductor Industry DRAM shipments. Results are encouraging and the findings are consistent with the idea that attitudes of purchasers towards new generations change from the previous one and it is imperative to identify a trend. The adoptions might decide to upgrade not exactly in the next generation of the product (Leapfrogging). Furthermore, there may exist some class of adopters who are resistant in accepting a new product at beginning launch of product (opponent). In future research, we aim to include these effects in our modelling framework.