Chapter 1

INTRODUCTION

Making decision on issues with important consequences is a complex problem due to the many competing forces under which the world is operating today, and the manual method very often leads to decisions quite far from being optimal. Decision makers are now aware of this problem and make efforts to use mathematical models for decision-making and even spend considerable sums of money to acquire software systems to solve these models. In the light of this problem the scholars and researchers of the scientific discipline, Operational Research targets to make decision with less risk and better outcomes by making mathematical models of the systems and solving them with their knowledge of optimization algorithms.

Mathematical modeling is the art of translating problems from an application area into tractable mathematical formulations where theoretical and numerical analysis provides insight, answers and guidance useful for the originating application. A mathematical model is an abstraction of the assumed real world, expresses in an amenable manner the mathematical functions that represent the behavior of the assumed system (Taha, (2006)). A model can be developed with respect to a system to measure some particular quantity of interest such as a cost model or a profit model or it may represent the assumed system as a whole used to optimize the system performance, the optimization model. The optimization models developed for the engineering and business professional allow them to choose the best course of action and experiment with the various possible alternative decisions. Figure 1.1 demonstrates the distinguished features of mathematical modeling and optimization.

Sophisticated computing systems are the backbone and marketing is the core activity for the efficient working of almost every business and industry. Mathematical modeling and optimization for marketing responsibilities of an organization and for the competent planning and control of the software development process and are the areas of prime interest in the field of Operational Research. This thesis endeavors to apply techniques of Operational Research to model building and optimization in the field of marketing and for building quality in the software.
As illustrated by the figure 1.1 mathematical modeling and optimization problems formulated and solved for one application can be used successfully in many further applications. In this thesis we have shown how modeling and optimization in the two apparently unrelated fields of marketing and quality software development process can enrich each other.

1. MODELING IN MARKETING AND SOFTWARE RELIABILITY

Mathematical Modeling in Marketing

The goal of marketing is to create customer satisfaction profitably by building value-laden relationships with its most important entity – The customers. A marketing department in an organization typically has to perform the functions of product pricing, promotion, distribution, personal planning, etc. working closely with the other departments of the organization. Making marketing decisions in the marketing mix interactions under environmental uncertainties and the challenges of the global economy has become one of the most difficult tasks for the management. Traditionally, managements use their experience to arrive at any decision or they follow their operational doctrines, but decision-making based on judgment method in this formidably complex situation cannot provide the best alternative decision. Sometimes it may lead to the wrong decisions due to the changing marketing environment, which may be a cause of an organization failure in the market. Therefore managements are becoming crucial enough to make their decisions based on scientific knowledge. Firms are increasingly applying modeling approach to marketing decision-making. Mathematical models in marketing are used to understand, predict and optimize processes.
Marketing aims to achieve *profit maximization through creation of value* and a *brand name* for the company, which is impossible to achieve if one wants to survive with its existing products and services. In the face of marketing challenges innovation has become the *path of growth* and new product as the *lifeblood* for the firms fighting for surviving longer and make profit margin in the global market place. The current scenario is that firms are speeding up their new product development process in order to replace their earlier products with the newer one, thereby introducing generations of new products. All the newer generation products are developed with an objective that they suit the latest needs of the consumers, are economic, cost effective, developed on the basis of most recent technology and are able to give competitive advantage to the manufacturer. Development and introduction of any new product demands firms to devote a significantly large percentage of their total profit for the research, development and marketing. The biggest risk associated with an innovation is the failure. A study has revealed that about 33% of the new products and services fail at launch. Reasons of failure may range from ineffective and inefficient promotion to poorly designed product, overestimated market potential, lagging behind technology, unusual costs etc. Another important issue related to the innovation is the reduction of the product life cycle time. In order to fight this issue the firms must find new products to replace old ones. Consequently the managements try to control the associated risks in the way of organization, professional staffing, marketing research, mathematical model construction and their application for the system optimization.

Therefore it has become essential to develop an understanding of the process of diffusion of a product or service and substitution process between the various generations simultaneously. There are four key element of the diffusion process namely - *The Innovation, Channels of communication, Time and the social System*. The process of diffusion initiates the process of adoption. Challenges of global economy require firms to plan and control their innovation diffusion process efficiently so as to sustain their survival in the marketplace and prevent failure of its new products. Controlling the diverse process of adoption in the presence of environmental uncertainties, competition, dynamic adopters and markets etc. is most difficult task of the marketing. The inability to make accurate decisions and measurements relating to the various controlling variables such as adopter behavior, market size and expansion, sale response, promotional effectiveness, immediate response to competitors strategies and environmental changes etc makes a successful innovation a challenge. Mathematical modeling and optimization are the widely applicable and successful tools in that they provide information and measurements related to the various decisions of innovation diffusion process.
Models of *Innovation Diffusion* are used successfully by the various practitioners (Mahajan and Peterson, 1985; Lilien *et al.*, 1997) to evaluate the market response over the product life cycles and make valuable decisions related to product modifications, price differentiation and resource optimization etc. A large number of innovation diffusion models have been developed in literature. An innovation diffusion model produces a life cycle sales curve based on certain parameters such as adopter behavior, promotion, market variables etc of the diffusion process. The purpose of a diffusion model is to *depict* the successive increase in the number of adopters and *predict* the continued development of a diffusion process already in progress. In this thesis we have developed and validated some new models of innovation diffusion for a single product in isolation and multiple generations of the product to measure the substitution effect. We have also formulated and solved models for optimization of resource utilization.

**Modeling for Building Quality in Software**

Computer software is the single most important technology in the human world and an excellent example of the *law of unintended consequences* as it paves the way for creation of the new technologies (e.g., genetic engineering), extension of the existing technologies (e.g., telecommunications) and the demise of the older technologies (e.g., the printing industry). Software plays dual role in our lives. In the first role it is a product, which delivers computing potential embodied by computer hardware (e.g., mathematical software), while in the second role it is a vehicle for delivering a product (e.g., software inside the cellular phones). In its second role, software acts as the basis for the control of system, communication of information, creation and control of other programs [Pressman, 2005]. Now as software have moved to the center of our lives, systems simple or complex are rapidly computerized or updated with latest computing systems due to ease of use and faster performance. Large-scale improvement in hardware performance, profound changes in computing architectures, vast increase in memory and storage capacity, a wide variety of exotic input and output options, has increased demand of software in automation of complex systems, its use as a problem-solving tool for many complex problems of exponential size and to control critical applications. With this size and design complexities of the software has also increased many folds and the trend will certainly continue in future. For instance the NASA Space Shuttle flies with approximately 0.5 million lines of software code on board and 3.5 million lines of code in ground control and processing [Lyu, 1996; Kapur *et al.*, 1999; Pham, 2006].

With the escalation in size, complexity, demand and dependencies on the computer systems the risk of crises due to software failures has also increased. The impact of failures ranges from
inconvenience to economic damage to loss of life. There are numerous reported and unreported
instances when software failures have caused severe loss to the economy and human lives [Pham,
(2006); U2]. Few examples are the crash of Boeing 727 of Mexicana airlines because the
software system didn’t correctly negotiate the mountain position (1986), overdose given to the
cancer patients by the massive Therac-25 radiation machine in Marietta due to flaws in the
computer program controlling the device (1985 and 1986), Explosion of the European Space
Agency’s Ariane 5 rocket, less than 40 seconds after lift-off on 4 June 1996 due to software
design errors and insufficient testing, blackouts in the North-East US during the month August,
2003 due to an error in the AEPR (Alarm and Event Processing Routine) software etc. [Pham
2006]. Even though the ability to design, test and maintain software has grown fairly, lot of
further improvements are desired in the field. As such the software development process has
become a challenging task for the developers. Arguably the most important software
development problem is building software to customer demands so that it be more reliable, built
faster, and built cheaper (in general order of importance). Success in meeting these demands
affects the market share and profitability of a product for the developer. These demands conflict,
causing risk and overwhelming pressure, and hence strong need for a practice that can help them
to have a tight control over the software development process and develop software to the need
of the software market.

In the early seventies, the Software Reliability Engineering (SRE) discipline emerged to establish
and use sound engineering principles in order to economically obtain software systems that are
not only reliable but also work efficiently on real machines, thus bringing the software
development under the engineering umbrella. The immediate concern of software engineering
was aimed at scheduling and systematizing the software development process to have a control
over the various stages of software development using its tools, methods and process to engineer
quality software. It focuses on quantitatively characterizing the distinguished software quality
characteristics such as functionality, usability, reliability, efficiency, maintainability, availability,
portability etc. Software Reliability is the key characteristic of software quality since it quantifies
software failures – the most unwanted events and hence is of major concern to the software
developers and user. For this reason software reliability is considered to be a “must – be quality”
of software. Hence one of the major roles of SRE lies in assuring and measuring the reliability of
the software. The tools of SRE known as software reliability growth models (SRGM) are used
successfully to develop test cases, schedule status, resource optimization, to count the number of
faults remaining in the software and estimate and predict the reliability of the software during
testing and operational environment. In this thesis we have developed and validated some new models for software reliability evaluation for isolated software (commercial as well as fault tolerant) and successive releases of the software. We have also formulated and solved some optimization models to optimize the software release schedule [Musa, (1998)].

1.1 DIFFUSION PROCESS

This topic has been widely studied by researchers from different disciplines, including Sociology, Economics, Psychology and Marketing. Diffusion is the process by which a new idea or new product is accepted by the market. The difference among individuals in their response to new ideas is called their innovativeness; it represents the degree to which an individual is relatively early or late in adopting a new product or services (Midgley and Dowling 1978). Hence the diffusion process is the aggregate of all individual adoptions over time. There have been many studies on how individuals react to new ideas and products. Those studies have identified four crucial elements in the diffusion process. Those are represented in the figure 1.2.

![Diagram of Diffusion Process]

Although diffusion is directly or indirectly affected by the above four elements, it is essentially a form of communication. It is transmission of information related to new ideas, and can be best described by figure 1.3.

In the initial stage of the diffusion process, the new product is adopted by a small group of consumers (known as innovators) who, with time, begin to influence others (known as imitators). This social interaction between the initial adopters and potential adopters can explain the phase of growth in the diffusion process (Rogers 1962).
1.1.1 Innovation

The term innovation may refer to something new - must be substantially different, not an insignificant change. There are three major types of innovations:

- **Continuous Innovation**: It is a simple change or improvement of an existing product where the adopter is expected to use the product in the same fashion as he/she did before.

- **Dynamically Continuous Innovation**: Here the innovations can either be a creation of a new product or a drastic change to an existing one which requires some change in the way the existing one is used.

- **Discontinuous Innovation**: Here the firm introduces a totally new product in the market. As the product or the innovation has never been seen before, consumer's attitude towards it would be different in terms of how they use and buy it.

Rogers (1962) has also discussed that the rate of adoption of an innovation to large extent depend on the innovation’s characteristic (figure 1.4). Each characteristic affects the rate of adoption of an innovation differently.

- **Relative Advantage**: This characteristic expresses to what extent the new product is better than the existing one or the product with which it is competing in the market place. The relative advantage of a new product can depend on factors like price, ease of use and storage as well as uncontrollable factors like war, natural calamities etc.

- **Compatibility**: This characteristic expresses the degree of which an innovation fits into the specific society or the existing values and familiarity of the adopters. The smoother the innovation fits into the culture, the faster the rate of adoption.
- **Complexity:** This characteristic expresses the degree of difficulty for an adopter to understand and use an innovation. It is very logical that the harder an innovation is to use or understand for an adopter, the less likely that an adopter would be make use of it.

- **Divisibility:** This characteristic expresses the ability of the consumer to give the innovation a test run before deciding whether to purchase it or not. As the amount of risk associated with this kind of low investment on trial packages consumers may find it interesting.

- **Communicability:** This characteristic refers to the degree to which the information regarding the utility and benefit of a new product can easily be communicated to the others in the society.

Though these five characteristics affect the rate of adoption, they are not the only ones. Other factors like over-adoption or low-adoption can also affect the rate, where adopters act irrationally without all the information or without full understanding of an innovation that can actually be harmful to the diffusion process. In such situation promotional campaign should be designed in such a way that it can communicate all the attributes (both positive and negative) and complexity of the new product.

![Innovation Characteristics](image)

**Figure 1.4: Characteristics of Innovation**

### 1.1.2 Communication Channel

Communication channel is the primary instrument for activating diffusion network. Mass media are effective source for simple innovations but interpersonal communication that involves human interactions in which one person transmits information regarding an innovation to another person may be essential for complex innovations. Communications that can be transmitted through formal or informal channels are necessary in order for diffusion to take place.
1.1.3 Social System
Social systems refer to the group or groups of people where an innovation diffuses through. Thus the social system for a new product would be those individuals or firms in the social strata who can use the product. When all individuals or units in the targeted area have adopted the innovation, the diffusion process is said to be complete.

1.1.4 Time Dimension
It is very unlikely that all the individuals belonging to a social system will adopt the new innovation at the same time. Thus it is very interesting to know the time of diffusion or in other words when or how much time will be taken by an individual or unit in an social system to decide to adopt an innovation after its introduction.

1.2 The Speed of Diffusion
The speed of the innovation is influenced by
* Competitive Intensity
* Good Supplier Reputation
* Standardization of Technology
* Vertical Channel Coordination
* Resource Commitments

Some cultures tend to adopt new products more quickly than others, based on several factors:

- **Modernity**: The extent to which the culture is receptive to new things. In some countries, such as Britain and Saudi Arabia, tradition is greatly valued—thus, new products often don’t fare too well. The United States, in contrast, tends to value progress.

- **Homophily**: The more similar to each other that members of a culture are, the more likely an innovation is to spread—people are more likely to imitate similar than different models. The two most rapidly adopting countries in the World are the U.S. and Japan. While the U.S. interestingly scores very low, Japan scores high.

- **Physical distance**: The greater the distance between people, the less likely innovation is to spread.

- **Opinion leadership**: The more opinion leaders are valued and respected, the more likely an innovation is to spread. The style of opinion leaders moderates this influence, however. In less innovative countries, opinion leaders tend to be more conservative, i.e., to reflect the local norms of resistance.
1.3 THE ADOPTION PROCESS

Adoption is defined as
"the decision to utilize an innovation as the best course of action available"
whereas the adoption process is defined as
"the steps an individual goes through from the time he hears about an innovation until final adoption and then to use an innovation regularly."

Roger(1983) proposed that consumers go through a sequence of five stages when accepting and adopting a new product.

- **Awareness**: occurs to a consumer when he or she is exposed to the innovation’s existence and general information about the product although consumer has a limited knowledge about its value, passion quality, usefulness and performance etc.
- **Interest**: This is a stage in which the individual is stimulated to seek information about the innovation. It occurs when a consumer forms a favorable or unfavorable attitude towards an innovation knowing more specific values of the product.
- **Evaluation**: occurs when a consumer engages in activities that lead to a choice to adopt or eject the innovation. The individual mentally appraises the innovation to his present and anticipated future situation and then decides whether or not to try it.
- **Trial**: occurs when a consumer puts an innovation in use on a small scale to estimate its utility.
- **Adoption**: It is the final stage of the decision making process. The consumer now decides to adopt this innovation for continued use. He or she may reverse this previous decision if exposed to conflicting messages about the innovation.

The adoption is influenced by the elements of diffusion process.

1.4 THE ADOPTION CATEGORIES FOR NEW PRODUCTS

French sociologist Gabriel Tarde (1903) originally claimed that sociology was based on small psychological interactions among individuals, especially imitation and innovation. This process has been studied extensively in the scholarly literature from a variety of viewpoints, most notably in Everett Rogers' (1962) classic book, *The Diffusion of Innovations*. Rogers proposed that the distribution of adoption of any innovation over time approached normality. Roger on eight different studies found that in the early stage of a particular innovation, growth is relatively slow
as the new product establishes itself. At some point customers begin to demand and the product growth increases more rapidly. New incremental innovations or changes to the product allow growth to continue.

Figure 1.5: Adoption Categorization on the Basis of the Relative Time of Adoption of Innovations (Source: Rogers 1962)

Towards the end of its life cycle growth slows and may even begin to decline. In the later stages, no amount of new investment in that product will yield a normal rate of return. Based on a bell curve Rogers categorized adopters of any new innovation or idea (as given in the figure - 1.5).

- **Innovators** - are first to buy and typically described as venturesome, younger, well educated, financially stable, and willing to take risks.
- **Early adopters** - are local opinion leaders who read magazines and who are integrate into the social system more than the average consumer.
- **Early majority** - solid, middle-class consumers who are more deliberate and cautious
- **Late majority** - described as older, more conservative, traditional, and skeptical of new products
- **Laggards** - Resist change, Conservative, Like tradition, Often older & lower in socioeconomic status

Rogers further claimed that innovations would spread through society in an S curve (as shown in figure-1.6), and the speed of technology diffusion is influenced by the product's perceived advantage or benefits, riskiness of purchase, ease of product use - complexity of the product,
immediacy of benefits, observability, trialability, price, extent of behavioural changes required, return on investment in the case of industrial products. The s-curve or cumulative-curve maps growth of revenue or productivity in any given potential market against time.

![Cumulative Diffusion Curve](source: rogers 1962)

The s-curve or bell-curve can be derived from half of a normal distribution curve. There is an assumption that new products are likely to have "Product Life". i.e. a start-up phase, a rapid increase in revenue and eventual decline. In fact a great majority among innovations never gets off the bottom of the curve. Innovative companies will typically be working on new innovations that will eventually replace older ones. Successive s-curves will come along to replace older ones and continue to drive growth upwards.

1.5 THEORY AND LITERATURE REVIEW

Research into the diffusion of innovations is historically rooted in several different disciplines, including anthropology, sociology, education, industrial engineering, and advertising. In these different fields, the focal points and specific research questions have varied. For example, anthropological studies have often looked mostly at the changes induced in a society after the introduction of a new technology, while industry-sponsored studies have tended to look at such questions as how advertising can influence adoption of a new product or how to optimize scale-up activities to accommodate an expected diffusion pattern [Rogers, 1962]. One common point among these varied studies, however, is the recognition of the S-shaped curve for adoption of a new technology or innovation [Kemp and Volpi, 2008]. The underlying phenomena to explain this behaviour vary among researchers and academic disciplines [Geroski, 2000]. The two extreme views can be described as social pressure-driven
and individual variety-driven. The first view suggests that the driving force for adoption of an innovation is the slow shift in social pressure as more and more individuals adopt until eventually everyone sees the innovation as the new norm. In this view, the S-curve of adoption of a new technology arises from a combination of the time it takes for individuals to be made aware of a new technology and the manner in which individuals respond with different strength to the adoption of the technology among their peers. Those who adopt early in the curve are generally aware of the existence of the technology earlier and are less reliant on the approval of their peers in making their adoption decisions. The second view, on the other hand, suggests that individual decisions on whether or not to adopt an innovation will be dependent on individual circumstances, such as the perceived value to be derived from the innovation, size of operation (in the case of companies or farms), access to income, or age of current equipment (in the case of replacement technologies). In reality, it is likely that some combination of these two ideas is at work in most diffusion processes.

Research efforts have included extending these models to capture more of the complexity in real systems in which innovation is embedded. For example, topics such as supply-constrained diffusion (Ho et al., 2002), serial diffusion of multiple generations of a technology, generalized models to capture external shocks to the system (Valle and Furlan, 2011), and coupling learning phenomena with standard diffusion models have all been explored in the literature (Bass, 1980). There is, however, debate about whether more complex models give greater precision, and whether this is worth the increased difficulty in using the model that accompanies increased complexity (Makridakis and Hibon, 2000). But even then Bass Model has been used extensively to describe the diffusion pattern. In the current work also, we have used this model and developed some extensions.

The origin of this concept can however be considered with the modelling of Fourt and Woodlook (1960), a pure external influence model. As said, later several models have been proposed to capture the varying aspects of the diffusion process.

**Notations**

\[ N(t) = \text{expected number of adopters by time } t, \quad N(0)=0. \]

\[ n(t) = \text{adoption rate at time } t, \quad n(t) = \frac{dN(t)}{dt}. \]

\[ m = \text{expected number of eventual potential adopters} \]

\[ p = \text{coefficient of external influence} \]

\[ q = \text{coefficient of internal influence} \]
$b(t) =$ time dependent adoption rate per remaining potential adopters.

$\beta, c =$ constants

### 1.5.1 External Influence Models

The basic external influence model is governed by the following differential equation

$$\frac{dN(t)}{dt} = p(m - N(t))$$  \hspace{1cm} (1.1)

Solving the above differential equation under the initial condition $N(0) = 0$ is given as

$$N(t) = m(1 - e^{-pt})$$  \hspace{1cm} (1.2)

The model describes an exponential cumulative adoption curve. Much of the popularity of external influence models is due to the work of Coleman et al (1996) who investigated it on diffusion of a new drug in four mid western communities. One of the earliest external influence model is due to Fourt and Woodlock (1960) given as:

$$N(t) = mp(1 - p)^{t-1}$$  \hspace{1cm} (1.3)

The model explains the successive increments of declining gains.

### 1.5.2 Internal Influence Models

The basic internal influence model is governed by the following differential equation

$$\frac{dN(t)}{dt} = q \frac{N(t)}{m} \left(m - N(t)\right)$$  \hspace{1cm} (1.4)

Solving the above differential equation under the initial condition $N(0) = 0$ is given as:

$$N(t) = \frac{m}{l} \left(1 + e^{-\omega t}\right)$$  \hspace{1cm} (1.5)

The model describes an S-shaped cumulative adoption curve (logistic curve). The most widely cited applications of internal influence model are those of Mansfield (1961) who investigated the diffusion of several innovations such as pallet loaders, diesel locomotives, etc and Griliches (1957) who studied the diffusion of hybrid seed corn in 31 states.

### 1.5.3 Mixed Influence Model-The Bass Model

Like every model is based on certain set of assumptions, The Bass model (1969), one of the most quoted aggregated diffusion model suggested in marketing literature was also developed based on some assumptions;
Assumption 1. The diffusion process is a binary process and population is Homogeneous.

Assumption 2. The population of adopters does not vary.

Assumption 3. The parameters of external and internal influence do not change.

Assumption 4. Only one adoption per adopter is permitted.

Assumption 5. Geographical frontiers do not alter.

Assumption 6. The innovation is diffused in isolation

Assumption 7. The characteristics of an innovation or its perception do not change.

Assumption 8. There are no supply restrictions.

Assumption 9. The impact of marketing strategies is implicitly captured by the model parameters.

The model has large acceptance in the field of innovation diffusion (Mahajan, Muller and Wind 2000). Since its publication, it has become the standard for further development and modifications (Mahajan, Muller and Bass 1990, 1993; Sultan, Farley and Lehmann 1990). The model assumes that there exists a finite population of prospective buyers who with time increasingly adopt the product. The buyers can be categorized as innovators and imitators depending upon the mode through which they receive information about the product. The persons who have already bought do not influence the innovators in their timing of purchase but they may be affected by the steady flow of non-personal promotion. As the process continues the relative number of innovators will diminish monotonically with time. Imitators are however influenced by the number of previous buyers and increase relative to the number of innovators as the process continues. Figure-1.7 gives the conceptual underlying the Bass model.

Figure 1.7: Bass New Product Diffusion Model (Source: Mahajan, Muller and Bass 1990)
Mathematically, the model can be expressed as
\[ \frac{f(t)}{1-F(t)} = p + qF(t) \]  
\[ (1.6) \]
where, \( f(t) \) is the likelihood of purchase at time \( t \), and \( F(t) = \int_0^t f(t) dt \) is the cumulative likelihood of purchasing the product at time \( t \); \( p \) and \( q \) are parameters of the model representing the coefficients of innovation and imitation respectively.

The number of adoptions \( S(t) \) at time \( t \) can be derived by multiplying \( f(t) \) with the market size \( m \) i.e.
\[ S(t) = mf(t) = pm + (q-p)N(t) - \frac{q}{m}[N(t)]^2 \]  
\[ (1.7) \]

The number of adopter in each period will rise due to the increasing impact of social interaction \[ \left[ \frac{q}{m}N(t) \right] \] till the time this effect is surpassed by the reducing the number of people who have not yet adopted i.e. \( (m-N(t)) \). If \( N(t) \) represent the cumulative number of adopters at time \( t \), then the expression for rate of adoption of the product in the market can be given as
\[ \frac{d}{dt} N(t) = p[m - N(t)] + q \frac{N(t)}{m}[m - N(t)] \]  
\[ (1.8) \]
where, \( F(t) = \frac{N(t)}{m} \). The solution of the above differential equation for initial condition \( N(0) = 0 \), is
\[ N(t) = m \frac{1-e^{-(p+q)t}}{1+\left( \frac{q}{p} \right)e^{-(p+q)t}} \]  
\[ (1.9) \]

Figure-1.8 gives the plot of marginal adoptions over time and figure-1.9 gives the plot corresponding of cumulative adoptions over time. In figure-1.8, the non cumulative adopter distribution peaks at time \( T^* \), which is the point of inflection of the S-shaped cumulative adoption curve as shown in the figure-1.9.

Bass suggested to estimate the diffusion parameters is the Ordinary Least Squares (OLS). The OLS procedure estimates parameter by taking the discrete analog of the differential equation of the Bass model that is of the form
\[ N(t-1) - N(t) = pm + (q-p)N(t) - \frac{q}{m}[N(t)]^2 \]  
\[ (1.10) \]
\[ n(t+1) = \alpha_1 + \alpha_2 N(t) + \alpha_3 [N(t)]^2 \]  
\[ (1.11) \]
In equation (1.10), $\alpha_1 = pm$, $\alpha_2 = q - p$ and $\alpha_3 = -q/m$. This regression equation is used to estimate $\alpha_1$, $\alpha_2$, $\alpha_3$, thus once these three parameters are known $p$, $q$ and $m$ can easily be estimated. Some reasons behind success of the Bass model can be attributed to

- The model is uncomplicated
- It is flexible, and describes variety sales growth curves of new products.
- The parameters of the model have well defined meanings and can provide managers with necessary decision making inputs.
The model enables the managers to solve an important concern of new products by determining the time to peak sales ($t^*$) and magnitude ($S(t^*)$). Bass shows that the time to peak and magnitude as

$$
t^* = \frac{1}{(p + q)} \ln \left( \frac{q}{p} \right) \quad \text{and} \quad S(t^*) = m \left( \frac{p + q}{4q} \right)^2
$$

(1.12)

Due to these reasons Bass model has sparked considerable research interest among consumer behavior and marketing scientists. The model doesn’t consider any other demand factors pricing, social acceptability, uncertainty reduction etc., which can affect the diffusion. But even then has acted as a platform for further extensions.

1.5.4 Flexible Models

The point of inflexion of a diffusion curve can occur at any time of the diffusion process and the diffusion process may be unsymmetrical about the inflexion point. Previous models offer limited flexibility to capture these properties. One of the flexible models is due to Jeuland (1981), who assumed a heterogeneous propensity of adopters to adopt varying according to gamma distribution and hence modified the Bass model as:

$$
\frac{dF(t)}{dt} = \left[ p + qF(t) \right] (1 - F(t))^{1/\gamma}
$$

(1.13)

Whereas Easingwood et al (1983) assumed a time varying impact of internal influence. The model can be given as:

$$
\frac{dF(t)}{dt} = \left[ p + q (F(t))^{1/\gamma} \right] (1 - F(t))
$$

(1.14)

Where, $F(t) = N(t) / m$

Some other flexible models are given by Floyd (1968); Sharif and Kabir (1976) etc.

1.5.5 Technological Substitution Models

The models discussed in the previous sections consider a single product in isolation. In the recent years technology is changing at a very higher rate. Technology; we all know can be defined as all the knowledge, products, processes, tools, methods, and systems employed in the creation of goods or in providing services. In simple terms, technology is the way we do things. It is the means by which we accomplish objectives. Technology is the practical implementation of
knowledge, a means of aiding human endeavour. Due to the continuously changing technology, stiff competition, increasing customers wants and their individualistic nature the global era sees more and faster introduction of the newer technology products with enhanced features. The potential buyers do not accept all the new products immediately. A diffusion process is set into the motion. In most of the situations as the time interval between successive products introduction is short, the earlier versions may continue to diffuse even though a newer version is in the market. In this situation a study of the diffusion process of any generational product in isolation to other generations may not provide us with the actual picture of the process. Therefore, it is important to develop an understanding of the process of diffusion and substitution between the various generational simultaneously.

During the last three decades several researchers devoted their research interest to diffusion modeling and contributed significantly in understanding the innovation adoption process. As discussed earlier external and internal influence models were considered as the preliminary attempt in diffusion based modelling until Bass introduced his model in 1969. The research following Bass model was mainly modifying the existing research in the way of incorporating the various aspects of the realistic situations and strategy. Some of the existing innovation diffusion models and concepts that have been used in this thesis are categorized as follows

**Alternative Formulation of Bass Model**
- *Modeling the idea of increased market size*
- *Incorporating change point analysis*
- *The Concept of Stochastic Differential Equations*
- *Modeling Successive Generations of Technologies*
- *Warranty Analysis in Marketing*
- *Two Dimensional Modeling*

In Section 1.6 we briefly discuss some of the most widely known modeling categories.

**1.6 REVISITING VARIOUS CONCEPTS IN DIFFUSION THEORY**
Formulation of Bass Model was based on certain set of assumptions; as discussed earlier. These assumptions were necessary in order to derive the closed form analytical solution; however in certain cases they restrict theoretical as well as practical applications of the model.
It is very difficult to include all the aspects of a diffusion process in a single model. A number of modifications of Bass Model have been carried out in the literature to capture the variability in adopter behaviour and other market conditions (Arndt (1967); Mahajan and Peterson (1978, 1985); Dodson and Muller (1978); Sharif and Ramanathan (1981); Lilien et al (1981); Easingwood et al (1983); Horsky and Leonard (1983); Norton and Bass (1987, 1992); Bass et al (1990), Mahajan and Muller (1979, 1996); Mahajan et al (1990); Putsis (1998); Rogers (2003); Kapur et al (2004), Meade and Islam (2006) etc. Increased use of mathematical models by the firms have engaged in innovation of new products and services, demands more accurate and simple methods. Bass model provides a flexible mathematical form and could describe a number of existing models of innovation diffusion in simpler form. During further modifications and extensions of the Bass model to include more realistic marketing situations such as increasing number of potential adopters over the product life cycle, phenomenon of repeat purchase etc., it is found that the solution gets complicated and sometimes it becomes impossible to arrive at a closed form solution. To look forward in this direction Kapur et al (2004) proposed an alternative formulation of Bass Model, which can be modified easily so as to include more general situations maintaining its flexible structure.

1.6.1 The Alternative Formulation of Bass Model (Kapur et al, 2004)

The Bass model and its revised forms have been used for forecasting innovation diffusion in retail service, industrial technology, agricultural, educational, pharmaceutical, and consumer durables goods markets and successfully predicts the actual shape of the diffusion curve and the timing and magnitude of its peak sales for a number of products (Rogers 1976).

The alternate solution of Bass model proposed by Kapur et al (2004) uses a logistic function to denote the rate of adoption per remaining adopter. In the last decade communication media has become a strong entity and has significant influence on our lives. Also due to faster development of technology such as mobile phones etc. Expression of thoughts and experiences have become easier and faster. Therefore it has become difficult to categorize the adopters as an innovator or imitator. Most of the imitators get exposed to media promoting the product. Moreover an innovator may by chance or by choice be able to get opinion about the product from a purchaser. Hence an individual estimation of innovation and imitation coefficients is very difficult in the present times. The parameters of logistic function describe the adopter’s behavioural aspects prevailing in the market avoiding the distinction between the adopters as innovators and imitators.
The mathematical equation describing Bass Model can be written as:

\[
\frac{dN(t)}{dt} = \left( p + q \frac{N(t)}{m} \right) \left( m - N(t) \right)
\]  
(1.15)

or

\[
\frac{dN(t)}{dt} = b(t) \left( m - N(t) \right)
\]  
(1.16)

Bass model is derived alternatively by changing the mathematical form of the rate of adoption per remaining adopters denoted by \( b(t) \).

Flexibility in Bass Model is captured by proposing a logistic time dependent form for \( b(t) \), given by

\[
b(t) = \frac{b}{(1 + \beta e^{-bt})}
\]  
(1.17)

Consequently, the diffusion model takes the following form

\[
\frac{dN(t)}{dt} = \frac{b}{(1 + \beta e^{-bt})} \left( m - N(t) \right)
\]  
(1.18)

Solving equation (1.18) with the initial boundary condition \( N(0)=0 \) we get

\[
N(t) = m \left( \frac{1 - e^{-bt}}{1 + \beta e^{-bt}} \right)
\]  
(1.19)

Substituting \( \beta = q / p \) and \( b = p + q \) we observe equation (1.19) is identical to the Bass Model (1.19). The S-shapedness in the cumulative adoption curve is created by S-shaped \( b(t) \). Alternative derivation of Bass Model simplifies further extension of the model. How alternative derivation of Bass Model can be used to extend the model by relaxing some simplified assumptions of the Bass Model is discussed in the following section.

1.6.2 Modeling the Idea of Increased Market Size

One of the simplifying assumptions of the Bass Model is that the number of potential adopters is static at the time of introduction of the product and remains constant over the life cycle of the product. However this might not be the case in most practical situations where the total number of potential adopters changes. A common goal of promotional campaign carried by the firms to promote their product is to make more and more customers aware of the product existence, value
and to persuade them to make a purchase. With an efficient promotional campaign, they are able to stimulate a large proportion of population as compared to the initial estimation made at the time of product introduction. An increasing trend is observed in the population of potential adopters. Increase in population and purchasing power of the population, changes in Govt policies etc. also accounts for increase in market size over time.

Mahajan and Peterson (1978, 1985) proposed dynamic diffusion models in which $m$ is taken as a function of time. They specified $m(t) = f(S(t))$, where $S(t)$ is a vector of relevant exogenous and endogenous variables—controllable as well as uncontrollable—affecting $m(t)$. Sharif and Ramanathan (1981) through their study of diffusion of oral contraceptives in Thailand illustrate the application of dynamic potential adopter diffusion models. They proposed some mathematical forms of $m(t)$ to describe the growth in potential adopter population with respect to time. These forms were substituted in external influence, internal influence and external-internal influence models. A significant improvement in fits for the resultant models was reported. Yet these models are complex and the authors could try only one form. However using the alternative formulation of Bass model it is simple to obtain closed form solution of the models incorporating these forms and several other forms with respect to time.

An important phenomenon observed during diffusion process is the possibility of repeat purchasing. The existing adopters may repurchase a number of products for the second or for more number of times. Therefore increase in number of purchasers of a new product can be due to both first purchase and repeat purchase. Several firms are more interested in estimating the increase in number of adopters due to repeat purchasing since they may become the loyal customers of the product. For consumer durable products, repeat purchasing is observed during the later stages of product life cycle. The Bass Model assumes that each adopter buys only one product, and hence may provide a wrong estimate of the number of potential buyers for the products, which are repurchaseable. Few models have been proposed in the literature incorporating (Mahajan and Peterson (1985)) to forecast sales with repeat purchasing phenomenon. Most of them use consumer purchase or survey data collected at a pretest or test market stage. Using the alternative formulation of the Bass Model, Jha et al (2008) provided some innovation diffusion models including the repeat purchase phenomenon assuming at any given time instant $t$, a proportion of adopters (say $\alpha (0 \leq \alpha \leq 1)$) is susceptible to repeat purchasing and repeat purchasing is influenced by all factors (both internal and external) affecting first purchase. The models were based on the following differential equation:
\[
\frac{dN(t)}{dt} = \frac{b}{1 + \beta e^{-bt}} (m(t) - N(t))
\]  

(1.20)

On solving this with different types of mathematical forms of \(m(t)\) we have expressions as given in table 1.1.

### Table 1.1 Extensions of the Alternative Derivation of Bass Model

<table>
<thead>
<tr>
<th>S. No</th>
<th>(m(t))</th>
<th>(N(t))</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>(m e^{\alpha t})</td>
<td>(\frac{mb \left(e^{\alpha t} - e^{-bt}\right)}{\alpha + b \left(1 + \beta e^{-bt}\right)})</td>
<td>Incorporates increasing market size</td>
</tr>
<tr>
<td>M2</td>
<td>(m(1 + \alpha t))</td>
<td>(\frac{m}{1 + \beta e^{-bt}} \left(\alpha t + \left(1 - e^{-bt}\right)\left(1 - \frac{\alpha}{b}\right)\right))</td>
<td>Incorporates increasing market size</td>
</tr>
<tr>
<td>M3</td>
<td>(m + k(1 - e^{-\theta t}))</td>
<td>(\frac{1}{1 + \beta e^{-bt}} \left(\frac{(m+k)(1 - e^{-bt})}{1 - \frac{kb}{b-\theta}(e^{-\theta t} - e^{-bt})}\right))</td>
<td>Incorporates increasing market size</td>
</tr>
<tr>
<td>M4</td>
<td>(m + \alpha N(t))</td>
<td>(\frac{m}{1 - \alpha} \left(1 - \left(\frac{(1 + \beta)e^{-bt}}{1 + \beta e^{-bt}}\right)^{1-\alpha}\right))</td>
<td>Incorporates repeat purchasing</td>
</tr>
</tbody>
</table>

#### 1.6.3 Incorporating Change point Analysis in Marketing

New products are launched under specific market conditions to meet the demand of a certain section of society, known as Potential adopter population. Under Bass model, it is assumed that each purchase decision is made randomly in time with same distribution. But the adoption process may get affected by many factors such as promotional effort, intensity of advertisement campaign, expectations and satisfaction level of the purchasers, after sales customer care etc. It is also affected by change in market conditions such as entry/exit of competitors from the market, launch of better quality/technology product or just change in the management of the company. There are many more factors which can cause change points in sales that may include changes in packaging (outlook), change in product features, increasing more outlets, availability of pack...
sizes, combo offers and discounts. Thus, the adoption rate may not be smooth and can be changed at some time moment $\tau$, known as change-point.

For the first time, the concept of change point was incorporated in marketing by Kapur et al (2007). In the following section, we discuss this Innovation-Diffusion model with change point.

The authors assumed that due to the phenomenon of change point, the rates of innovation and imitation will also be changed, so the value of $\beta$ i.e. $\beta = q / p$ will also be changed and $b(t)$ can be defined as:

$$b(t) = \begin{cases} 
\frac{b_1}{1 + \beta_1 e^{-\beta_1 t}} & \text{when } 0 \leq t \leq \tau \\
\frac{b_2}{1 + \beta_2 e^{-\beta_2 t}} & \text{when } t > \tau
\end{cases}$$

(1.21)

Where, $\tau$ is the change point.

Under the assumptions described above, the adoption process may be described by the following differential equation:

**Case 1: when $0 \leq t \leq \tau$**

$$\frac{dN(t)}{dt} = \frac{b_1}{1 + \beta_1 e^{-\beta_1 t}} (m - N(t))$$

(1.22)

Solving by assuming $N(0)=0$; we get

$$N(t) = m \left( \frac{1 - e^{-\beta_1 t}}{1 + \beta_1 e^{-\beta_1 t}} \right)$$

(1.23)

**Case 2: when $t > \tau$**

$$\frac{dN(t)}{dt} = \frac{b_2}{1 + \beta_2 e^{-\beta_2 t}} (m - N(t))$$

(1.24)

Solving by assuming $N(t = \tau) = N(\tau)$; we get

$$N(t) = m \left[ 1 - \left( \frac{1 + \beta_1}{1 + \beta_1 e^{-\beta_1 \tau}} \right) \left( \frac{1 + \beta_2 e^{-\beta_2 \tau}}{1 + \beta_2 e^{-\beta_2 t}} \right) e^{-\beta_1 \tau - \beta_2 (t - \tau)} \right]$$

(1.25)

**Stochastic process**: stochastic process \( \{X(t), t \in T\} \) is a collection of random variable i.e. for each \( t \in T \), \( X(t) \) is a random variable. The index \( t \) is often interpreted as time and as a result, we refer to \( X(t) \) as the state of the process at time \( t \).

The set \( T \) is called the index set of the process. When \( T \) is a countable set, the stochastic process is said to be a discrete time process. If \( T \) is an interval of real time, the stochastic process is said to be a continuous-time process. For instance, \( \{X_n, n = 0,1, \ldots\} \) is a discrete-time stochastic process indexed by the nonnegative integers; while \( \{X(t), t \geq 0\} \) is a continuous time stochastic process indexed by the non negative real numbers.

**Stochastic Analog of Classical Differential Equation**

If we allow for some randomness in some of the coefficients of a differential equation, we often obtain a more realistic mathematical model of the situation.

Consider the simple population growth model

\[
\frac{dN(t)}{dt} = a(t)N(t)
\]

\( N(0) = N_0 \)

Where \( N_0 \) is constant and \( N(t) \) is the size of the population at time \( t \) and \( a(t) \) is the relative rate of growth at time \( t \). It might happen that \( a(t) \) is not completely known, but subject to some random environment effect so that we have

\[
a(t) = r(t) + "\text{noise}\" \quad (1.27)
\]

In the above the exact behavior of the noise term is not known. Only its probability distribution is known. The function \( r(t) \) is assumed to be a non random.

The mathematical model for a random quantity is a random variable. As we know a stochastic process is a parameterized collection of random variable \( \{X_t\}_{t \in T} \) defined on a probability space \( (\Omega, F, p) \) and assuming values in \( \mathbb{R}^n \).

\( \sigma \)-Algebra: if \( \Omega \) is a given set, then a \( \sigma \)-algebra \( F \) on \( \Omega \) is a family \( F \) of subset of \( \Omega \) with the following properties

1. \( \phi \in F \)
2. \( F_i \notin F \Rightarrow F_i^c \Rightarrow F, \text{where } F_i^c \text{ is complement of } F \text{ in } \Omega \)
3. \( A_1, A_2, \ldots \in F \) Where \( F_i^c \) is complement of \( F \text{ in } \Omega \)
The pair \((\Omega, P)\) is called a measurable space.

**Probability Measure:**
A probability measure \(p\) on a measurable space \((\Omega, P)\) is function \(p : F \to [0,1]\)
Such that
1. \(p(\emptyset) = 0, p(\Omega) = 1\)
2. if \(A_1, A_2, \ldots \in F\) and \(\{A_i\}_{i=1}^\infty\) is disjoint
   \[ (i.e. A_i \cap A_j = \Omega \text{ if } i \neq j) \text{ then } P \left( \bigcup_{i=1}^\infty A_i \right) = \sum_{i=1}^\infty P(A_i) \]

**Probability Space**
The triple \((\Omega, F, P)\) is called the probability space.

**Brownian Motion**
In 1828 the Scottish botanist Robert Brown observed that Pollen Grains suspended in liquid performed an irregular motion. He and other noted that:
- The path of a given particle is very irregular, having a tangent at no point.
- The motion of two distinct particles appears to be independent.

The motion was later explained by the random collision with the molecules of the liquid. To describe the motion mathematically it is natural to use the concept of a stochastic process \(W(t)\), interpreted as the position at time \(t\).

**Definition:** A real valued stochastic process \(W(.)\) is called a Brownian or Winner process if
1. \(W(0)=0\)
2. \(W(t)-W(s)\) is \(N(0,t-s)\) for \(t \geq s \geq 0\)
3. For all times \(0 < t_1 < t_2 < \ldots < t_n\),

The random variables \(W(t_1), W(t_2) - W(t_1), \ldots, W(t_n) - W(t_{n-1})\) are independent.

In particular
\[ E[W(t)] = 0, E[W^2(t)] = t \text{ for real time } t \geq 0 \]

\[ \text{Var}(W(t)) = t \]

As \(W(t)\) follows normal distribution with Mean 0 and Variance \(t\) so for all \(t > 0\) and \(a \leq b\)
\[ p[a \leq W(t) \leq b] = \frac{1}{\sqrt{2\pi t}} \int_a^b e^{-\frac{1}{2}w^2(t)} \, dw(t) \]

**Construction of the \( \text{i}t \text{o} \) integral**

We now turn to the question of finding a reasonable mathematical interpretation of the “noise” term in equation (1.27)

\[ \frac{dN(t)}{dt} = (r(t) + "noise")N(t), \quad (1.28) \]

Or more generally in equation of the form

\[ \frac{dN(t)}{dt} = b(t, N(t) + \sigma(t, N(t)))noise \quad (1.29) \]

Where \( b \) and \( \sigma \) are some given functions.

Let us first concentrate on case where the noise is 1-dimensional. It is reasonable to look for some stochastic process \( \gamma(t) \) to represent the noise term, so that

\[ \frac{dN(t)}{dt} = b(t, N(t) + \sigma(t, N(t)))\gamma(t) \quad (1.30) \]

Nevertheless it is possible to represent \( \gamma(t) \) as a generalized stochastic process called the white noise process.

That the process is generalized means that it can be constructed as a probability measure on the space \( s' \) of tempered distribution on \([0, \infty]\) and not as probability measure on the much smaller space \( R^{[0,\infty]} \), like an ordinary process.

The time derivative of the Wiener process (or Brownian motion) is white noise

So \( \frac{dW(t)}{dt} = \gamma(t) \)

So equation (1.30) can be rewritten as

\[ dN(t) = b(t, N(t))dt + \sigma(t, N(t))dW(t) \quad (1.31) \]

This is called a stochastic differential equation of \( \text{i}t \text{o} \) type

**Result: The 1-dimensional \( \text{i}t \text{o} \) formula**

Let \( X_t \) be an \( \text{i}t \text{o} \) process given by

\[ dx_t = u dt + v dW(t) \]

Let \( g(t, x) \in C^2((0, \infty) \times R) \) (i.e. \( g \) is twice continuous differentiable on \((0, \infty) \times R)) \)

Then
\[ Y_t = g(t, x_t) \] is again an \( \text{i.o} \) process, and
\[ dY_t = \frac{\partial g}{\partial t} dt + \frac{\partial g}{\partial x_t} dx_t + \frac{1}{2} \frac{\partial^2 g}{\partial x_t^2} (dx_t)^2 \]

Where \((dx_t)^2 = (dx_t, dx_t)\) is computed according to the rules
\[ dt \cdot dt = dt \cdot dW(t) = dW(t) \cdot dt = 0 \quad \& \quad dW(t) \cdot dW(t) = dt \]

Solution of above equation is:
\[
\frac{dN(t)}{dt} = a(t)N(t)
\]
\[ N(0) = N_0 \]

where \( a(t) = b(t) + \sigma \gamma(t) \), \( \sigma \) is a positive constant representing the magnitude of the irregular fluctuation and \( \gamma(t) \) is a Standardized Gaussian white noise.

Let \( b(t) = b \) (constant)
\[
\frac{dN(t)}{dt} = bN(t) + \sigma \gamma(t)N(t) \quad (1.32)
\]
\[ dN(t) = bN(t)dt + \sigma N(t)dW(t) \quad (1.33) \]

Solving above, we have
\[ N(t) = N_0 e^{(b-\frac{1}{2})t+\sigma W(t)} \quad (1.34) \]

From marketing point of view, we consider that if the product under consideration has a considerably large life cycle and caters to the needs of a large section of society, then as the time progresses, the adoption process has smaller increments in total adoption as compared to the potential adopter population. In such cases, adoption process defined by \( \{N(t), t \geq 0\} \) behaves as a stochastic process with continuous state space. The adoption process as defined takes care of irregular fluctuations due to dynamic market conditions and ever changing socio-economic factors. Modeling of time evolution of a process with Stochastic Differential Equation is not new. They have been successfully used in financial engineering, Portfolio Management, Physics and life sciences to name a few (Okasendal 2003). In the thesis, we have developed some innovation diffusion models using this concept.

### 1.6.5 Diffusion Modeling for Successive Generations

An important feature of most modern new technologies is that they come in successive generations. The clearest examples are from the area of information technology and telecommunication industry. Also the time gap between successive generations of products is
reducing and the replacement of earlier technologies with the latest one is occurring quite frequently. With compatibility being no longer a problem area for major product categories, consumer today has more choices and has the opportunity to choose from new as well as older generations of a technology after evaluating the prices, utilities, risk etc. In addition there are repeat purchasers also, who upgrade their older technologies with the latest one, lifting up the potential market size. The factors involved in these two kinds of adoptions can be very different. As a result it has become strategically more important to study the technological changes and the growth rate of consumer preference towards a generation and corresponding market behavior.

![Figure 1.10: Series of Technological Generations (Source: Norton and Bass 1987)](image)

Since the pioneering work of Bass (1969), lots of other models have been developed using the model to study the technological innovations. Norton and Bass (1987) model is a classic example of multiple generation model, which is again built upon the Bass model. In the model it is assumed that the coefficients of innovation and imitation remain unchanged from generation to generation of technology. But many authors have argued against this assumption. Islam and Meade (1997) have tested the hypothesis of coefficient constancy across generation of Norton and Bass model. Their empirical work relaxed the assumption of constant coefficient. They proposed that the coefficients of later generation technology are constant increment/decrement over the coefficients of the first generation. Mahajan and Muller (1996) proposed a model, which is again an extension of Bass model to capture simultaneously both the substitution and diffusion patterns for each successive generation of technological products. Bayus (1992b) developed a model for consumer sales of second-generation product by incorporating replacement behavior of first generation adopters and suggested different dynamic pricing policies of second-generation consumer durables. Speece and MacLachlan (1995); Danaher, Hardie and Putsis (2001),
developed a model in a different way by incorporating price as an explanatory variable. Kim, Chang, and Shocker (2000) in their model have tried to capture intergenerational linkages among successive generations within a product category. The authors proposed that the market potential of a generation of a product category not only affected by the technological substitution from another generation within the category but also affected by the adoption rate of the other categories. Few of the important work in this area discussed below.

1.6.5.1. Norton and Bass model (1987)

The Norton and Bass (1987) model was the first model used explicitly to describe the diffusion process of the sequence in the diffusion of separate generations. For a sequence of technological generations described through Norton and Bass model Jaakkola, Gabbouj and Neuvo (1998) have observed a certain diffusion pattern and have depicted the following observations:

- The rapid adoption phase or growth phase (i.e. the change point of lower strata of adoption to the higher strata) of a new technology generation is shorter than that of the earlier generation.
- Any variation in the diffusion of a new technology is more noticeable than that of the old technology.
- The market potential is monotonically increasing from generation to generation These changes can be well explained by the word-of-mouth communication effect, better performance level of the latest generations, etc.

The original Bass model is for single generation and it assumes that a fixed adopter population can be divided into two groups: innovators and imitators. Norton and Bass extended this result for multiple generations of products. The Norton–Bass model equations for two generations are:

\[
N_1(t) = F_1(t)m_1[1 - F_2(t - \tau_2)] \\
N_2(t) = F_2(t - \tau_2)[m_2 + F_1(t)m_1]
\]

where, equation (1.35) represent the diffusion equation of the first generation product and equation (1.36) that of the second generation product. \(N_i(t)\) is the shipments of \(i^{th}\) generation product at time ‘\(t\)’, \(F_i(t)\) represent the fraction of adoption for \(i^{th}\) generation product at time ‘\(t\)’, which is a cumulative Bass distribution of the form

\[
F_i(t) = \left[ \frac{1 - e^{-(p+q)t}}{1 + \left( \frac{q}{p} \right) e^{-(p+q)t}} \right]
\]
and \( m_i \) is the population served by the \( i^{th} \) generation product. The coefficients \( p \) and \( q \) determine the rate of adoption or the speed of diffusion of an innovation and these coefficients remain constant over sequence of generations. The author validated the model in the semiconductor industry later on they extended this model to cover several other industries.

1.6.5.2 Mahajan and Muller Model (1996)

Mahajan and Muller (1996) developed a multigenerational diffusion and substitution model for IBM mainframe case, which is again based on the Bass model (1969). They defined leapfrogging as skipping to the latest generation from any previously adopted generations by completely ignoring the intermediate generational products.

Mahajan and Muller validated their model with the data of four-generation IBM product families and estimated the diffusion parameters with reasonably good fit; the model can be represented as:

**Period of first generation** (\( t_1 \leq t < t_2 \)): In this period there is only one generation in the market, hence the Bass model governs the sale, given as

\[
\frac{d}{dt} N_1(t) = \left( a_1 + \frac{b_1 N_1(t)}{m_1} \right) \left( m_1 - N(t) \right) \quad (1.38)
\]

**Period of second generation** (\( t_2 \leq t < t_3 \)):

\[
\frac{d}{dt} N_2(t) = \alpha_2 \left( a_2 + \sum_{i=1}^{2} \frac{b_i N_i(t)}{m_2} \right) \left( m_2 - N(t) \right) + \alpha_2 \left( a_2' + \frac{b_2 N_2(t)}{m_2} \right) N_1(t) \quad (1.39)
\]

\[
\frac{d}{dt} N_1(t) = (1 - \alpha_2) \left( a_2 + \sum_{i=1}^{2} \frac{b_i N_i(t)}{m_2} \right) \left( m_2 - N(t) \right) - \alpha_2 \left( a_2' + \frac{b_2 N_2(t)}{m_2} \right) N_1(t) \quad (1.40)
\]

The first term in equations (1.39) and (1.40) is due to adoptions from the untapped market potential. Out of these adoptions, a fraction, \( \alpha_2 \) adopt the second generation and the remainder (1 - \( \alpha_2 \)) still purchases the first generation. The second term is due to adoptions from upgraders, who adopt with a different innovation and imitation parameters.

**Period of third generation** (\( t_3 \leq t < t_4 \)):

\[
\frac{d}{dt} N_3(t) = \alpha_3 \left( a_3 + \sum_{i=1}^{3} \frac{b_i N_i(t)}{m_3} \right) \left( m_3 - N(t) \right) + \alpha_3' \beta_3 \left( a_3' + \frac{b_3 N_3(t)}{m_3} \right) N_1(t) + \alpha_3 \left( a_3 + \frac{b_3 N_3(t)}{m_3} \right) N_2(t)
\]
A Study of Innovation Adoption & Warranty Analysis in Marketing and Successive Software Releases

\[
\frac{d}{dt} N_2(t) = (1 - \alpha_3) \beta_3 \left( a_3 + \sum_{i=1}^{3} b_i N_i(t) \right) \left( m_3 - N(t) \right) - \alpha_3 \left( a_3 + b_3 \frac{N_3(t)}{m_3} \right) N_2(t) + \alpha_3 (1 - \beta_3) \left( a_3 + b_3 \frac{N_3(t)}{m_3} \right) N_1(t)
\]

\[
\frac{d}{dt} N_1(t) = (1 - \alpha_3)(1 - \beta_3) \left( a_3 + \sum_{i=1}^{3} b_i N_i(t) \right) \left( m_3 - N(t) \right) - \alpha_3 \left( a_3 + b_3 \frac{N_3(t)}{m_3} \right) N_1(t)
\]

In the three-generation case, leapfrogging phenomenon is incorporated. The models considers the proportion of customers who own a first generation product and decide to upgrade is given by the parameter \( \alpha_3 \), then the proportion who upgrade to the next available generation is \( 1 - \beta_3 \), and the proportion who leapfrogs to the latest available generation is \( \beta_3 \). Where \( m_i \) is the current market potential on the introduction of \( n^{th} \) generation, \( N_n(t) \) is the number of systems in use of \( n^{th} \) generation, \( N(t) = \sum N_n(t) \), where \( n^{th} \) generation is the latest available generation at time \( t \), \( t_n \) is the time of introduction of generation \( n \).

Here \( N_n(t) \) be the number of systems-in-use of the \( n^{th} \) generation, \( a \) and \( b \) are coefficients due to innovation and imitation respectively. \( \alpha \) and \( \beta \) are leapfrogging parameters. \( m_i \) is the market potential of the \( n^{th} \) generational technology product.

If \( T_n \) is the introduction time of \( n^{th} \) generation product then the conceptualization of the Mahajan and Muller model can be given as in the figure-1.11.

![Figure 1.11: Relation between generations](image)

**Figure 1.11: Relation between generations**

1.6.5.3. An extended multi-generation diffusion model (Jiang 2010)

After a new product generation becomes available, some first-time adopters who otherwise would adopt the old generation may decide to purchase the new generation instead, we call this
type of adopter behavior first purchase substitution (or substitution for short). Besides the first-time adopters, some existing adopters of the old generation may be willing to upgrade to the new generation, if they believe that the improvements in the new generation are worth the investment. We call this behavior generation switching (or switching for short). To a software firm, the primary difference between switching and substitution is that the former leads to repeat purchases from the same potential adopter, while latter does not. In their study, Norton and Bass acknowledge the existence of substitution and switching, but do not differentiate the two. Jiang (2010) in their study, differentiated first-purchase substitution and generation switching, because their economic implications are different.

The Mahajan and Muller model provides a basis for extending the Norton and Bass model to address potential adopters' generation switching behavior. In their model, Mahajan and Muller assume that product upgrades also follow a Bass-type diffusion process, with the potential market size for upgrade at any given time \( t \) equal to the number of existing adopters for each previous generation right before \( t \). In this research, we restrict the potential adopters for switching to those who have adopted before the new generation is released. This restriction is appropriate because software products, unlike physical products, do not wear out by usage. If an adopter decides to purchase the older version even after the new version is released, the adopter is highly unlikely to purchase an upgrade to the new version again. Regarding the speed of diffusion for a new software version, we assume that the coefficient of innovation (\( p \)) is the same for new purchases and upgrades, so is the coefficient of imitation (\( q \)). We have used these existing models and the aforementioned factors to develop new ideas in this research work.

![Figure. 1.12: Diffusion curves for two successive generations](image_url)

After generation 2 is introduced to the market at time \( \tau \), a portion \( (\alpha_2) \) of those who otherwise would have adopted generation 1 would substitute it with generation 2 instead. Hence, the
cumulative number of substitutions from generation 1 to generation 2 by time $t$, denoted by $B_2(t)$, equals

$$B_2(t) = \begin{cases} 0, & t < \tau \\ \alpha_2 m_1 [F_1(t) - F_1(\tau)], & t \geq \tau \end{cases}$$  \hspace{1cm} (1.41)$$

We refer to $\alpha_2$ as the substitution ratio, representing the proportion of potential adopters of generation 1 who, after being informed of generation 2, will substitute generation 1 with generation 2. Along with the first-time purchasers, those who have adopted generation 1 before time $\theta$ may switch to generation 2 after it is available. The speed of switching depends on the progress of the diffusion of information about the newer generation. The cumulative number of switching adoptions by time $t$, denoted by $W_2(t)$, is

$$W_2(t) = \begin{cases} 0, & t \leq \tau \\ m_1 [F_1(\tau) - F_2(t - \tau)], & t > \tau \end{cases}$$  \hspace{1cm} (1.42)$$

Due to the first-purchase substitutions, the cumulative number of adoptions of generation 1 by time $t$, denoted by $Y_1(t)$, is

$$Y_1(t) = \begin{cases} m_1 F_1(t), & t \leq \tau \\ m_1 F_1(t) - B_2(t) = m_1 F_1(t) - \alpha_2 m_1 [F_1(t) - F_1(\tau)], & t > \tau \end{cases}$$  \hspace{1cm} (1.43)$$

Note that a switching from generation 1 to generation 2 does not lead to a reduction in $Y_1(t)$; hence $Y_1(t)$ is an increasing function of time. The cumulative number of adoptions of generation 2, denoted by $Y_2(t)$, equals

$$Y_2(t) = \begin{cases} 0, & t \leq \tau \\ m_2 F_2(t - \theta) + B_2(t) + W_2(t) = m_2 F_2(t - \tau) - \alpha_2 m_2 [F_2(t) - F_2(\tau)] + m_2 F_2(\tau) F_2(t - \tau), & t > \tau \end{cases}$$  \hspace{1cm} (1.44)$$

Jiang called this modified model the Extended Multi-Generation diffusion model (EMG).

1.6.6 Warranty Analysis in Marketing

Modern manufacturing is characterized by (i) rapidly changing technologies, (ii) global markets, (iii) fierce competition, (iv) often nearly identical products due to common components and technology being used and, (v) better educated and more demanding customers. This has posed serious challenges for buyers, manufacturers and policy makers at national and regional levels. In the purchase decision of a product, buyers typically compare characteristics of comparable models of competing brands. When competing brands are nearly identical, it is very difficult, in
many instances, to choose a particular product solely on the basis of the product related characteristics such as product price, special features, perceived product quality and reliability, financing offered by the manufacturer and so on. In such situations, post-sale factors – warranty, parts availability and cost, service, maintenance, and so forth – take on added importance in product choice (Lele (1983A), Lele and Karmarkar (1983B), Ives and Vitale (1988A, 1988B), and Ritchken et al. (1989). Of these, warranty is one that is known (or at least potentially known) to the buyer at the time of purchase. In the case of new products, another feature is that each new generation is more complex than the earlier generation it replaces. Often customers are uncertain about new product performance. Here warranties play an important role in providing product assurance to customers and different types of warranties are offered depending on the product and the buyer.

The notion of post-sale support is becoming an important feature of any product sale. In this context, warranty (and extended warranty) is an element of post-sale support and manufacturing business needs to view it as part of the post-sale service strategy. A warranty of any type, since it involves an additional service associated with a product, will lead to potential costs beyond those associated with the design, manufacture and sale of the product. These costs, in fact, are unpredictable future costs, which typically range from 2% to as much as 15% of net sales McGuire (1980). As a result, warranty has a significant impact on the total profits for a manufacturing business. Similarly, for businesses where new products purchased constitute a major component of their total operating budget ineffective management of warranties can have a significant impact on the total operating costs. As a result, product warranty plays an increasingly important role in consumer and commercial transactions. The use of warranties is widespread and they serve many purposes. These include protection for manufacturer and buyer, signaling of product quality, an important element of marketing strategy, assuring buyers against items which do not perform as promised and play an important role in the dispute resolution between buyer and manufacturer. The concept of warranty has been around for almost as long as there has been trade and there have been many representations of warranty throughout history. It has existed in some form or another from the early civilizations (Babylonian, Assyrian, and Egyptian Eras, Ancient Hindu and early Islamic periods), through the European Period (Roman Era, Germanic, Jewish, and early English periods, and the early Russian Era), the Middle Ages, the Industrial Revolution and beyond. Until the sixteenth century, the general purpose of warranty was to protect the buyer from fraud and faulty workmanship. When trade policy reversed around the dawn of the industrial revolution to favor the manufacturer, it was not a pressing issue since products were still produced locally by people known personally to buyers. Products were still
relatively simple and easily evaluated, and any dissatisfaction was addressed directly to the manufacturer, with word of mouth travelling fast in local and tight knit communities. As communities grew, so did the acceptance of caveat emptor or “let the buyer beware”. Late in the nineteenth century, standardized product warranties became more common, although many were extremely limited in coverage. As deceit became more widespread, consumers began to see warranties as indicators of poor quality, with manufacturers offering contracts with no intention of honoring them, and no legal incentive to do so. This was the basis of the exploitation theory of warranty. According to this theory, the warranty terms are developed for the manufacturer’s benefit, while the consumer has few rights and bears the risks. Buyers who believe this theory often feel that if a product is sold, it should last a certain amount of time, and the warranty is seen to serve the manufacturer by adding to the price of the product; i.e., by offering a service which should be provided anyway. Because a warranty is offered, it is reasoned, these buyers feel that the manufacturer does not have confidence in its own product. In order to counter this trend, many commissions were established, which set forth codes to govern the sale of goods.

Before 1975, consumers were still at the mercy of manufacturers for several reasons. Warranties did not provide notice of consumer rights, administration of warranty was confusing and ineffective, remedies were impractical for defective items and excessive and unjustified claims often resulted from consumer frustration and hostility (Burton, PWH, chapter 28).

The product performance and the warranty terms together determine the costs incurred by the manufacturer, so it follows that a longer warranty period will result in more costs unless the product performance is of a correspondingly higher quality. This theory proposes that if a manufacturer offers a better warranty than a competitor, then the reliability of the product should also be better to reduce costs associated with warranty claims. Due to this signaling characteristic, warranty is an important product feature and can be used by marketing to promote sales.

**Warranties: An Overview**

The consumer buying process (for both consumer durables and industrial and commercial products) is a multi-stage process. In a simplified characterization it involves the following stages:

- Recognition of the need for the product.
- Obtaining information regarding the different product brands.
• Evaluating the different brands to decide on the brand to purchase in terms of price, performance, assurance, post-sale support etc.
• Final purchase.

To help the process, manufacturers need to promote their brand and provide the information to help consumers in the decision-making process.

As discussed earlier, warranties (and extended warranties) serve both promotional and protective roles.

The literature on warranty and marketing can be grouped into several categories as indicated below:

(1) Warranty and consumer behavior
(2) Product price and sales for warranted products
(3) Warranty and market outcome
(4) Warranty as a marketing strategy

Warranty and consumer behavior

The behavior of consumers depends on the type of product since the purchase process for consumer durables is totally different from that for industrial and commercial products. Of particular relevance are the empirical studies that show that warranties have a significant impact on consumer product choice and that they would pay a premium for products with better warranty terms. Boulding and Kirmani (1993) carry out an experimental investigation to answer the question – do consumers perceive warranties as signals of quality? The results indicate that consumer responses to warranties are consistent with the behavioral assumptions of signaling theory of warranty. Agrawal et al. (1996) study warranty as a source of information regarding product reliability in the context of household appliances.

Sale price and sales for warranted products

The price for a product sold with warranty needs to take into account the cost of servicing the warranty. The sales for a product sold with warranty needs to take into account the negative impact of warranty on the sale price and the positive impact as a promotional tool. Blischke and Murthy discuss one of the earliest models (due to Glickman and Berger (1976)) which use a Cobb–Douglas function with sales being a function of price and warranty duration. Menezes and Currim (1992) model the total sales by a more general formulation that is a function of several variables that include price, warranty period, quality, advertising and competitor’s actions. They
derive expressions for the optimal price and warranty as functions of the price and warranty elasticities. They illustrate it with an application to historical data on automobiles. Mesak (1996) models sales through a continuous time diffusion model. The demand rate is modeled as a function of price and warranty length that can vary with time. Mesak examines different structures for the diffusion model and characterizes the optimal price and warranty length. Chun and Tang (1995) deal with a model where the buyer has the choice of buying a product with or without warranty.

**Warranty and market outcome**

The market outcome is determined by the interaction between individual consumers and manufacturers. Microeconomics deals with this topic and examines the effect of the decisions made by individual consumers and manufacturers and other market related factors on the market outcome. The literature on warranties and market outcome is extensive and is reviewed under Warranty as a marketing strategy in the next Section.

**Warranty as a marketing strategy**

As a marketing tool, warranty plays many roles. One of the most important of these is that it provides the buyer with a degree of assurance against uncertainty. Warranty decisions as a marketing tool depend on whether the manufacturer is a leader or a follower with the product being introduced and the warranty being offered, i.e., whether an offensive or a defensive warranty strategy, as defined by Menezes and Quelch (1990), is being pursued. Warranty is viewed as an offensive tool when it is used as a signal of reliability. Better warranty terms imply higher product quality and the manufacturer will stand behind the product. A longer warranty requires higher reliability for cost control and as such higher reliability gives a competitive advantage (as long as the customer is convinced that the product is, in fact, more reliable). Producers of lower quality products cannot meet the competition or can do so only at substantially higher cost. Warranty in this situation is an offensive marketing tool in that the manufacturer is able to take a pro-active stance in setting warranty terms. If, instead, the manufacturer is reacting to the competition, warranty will be used as a defensive tool. The objective here is (i) to meet competition to avoid losing sales; (ii) correct possible consumer misperceptions concerning the quality of the item; and (iii) limit liability. The FRW is sometimes thought of as an offensive strategy, while the PRW is defensive Menezes and Quelch (1990). In this context, a combination FRW/PRW would be a reasonable compromise between these two strategies. Menezes and Quelch focus mainly on warranty as a marketing strategy. There are
many other issues involved in the strategic management of warranty and these are discussed in the next section. Mitra and Patankar (1997) examine market share as a function of warranty and the option of extending the warranty at the end of the base warranty should the item not fail in the base warranty period. They study the effect of warranty decisions on the market share.

More recently, warranty has been viewed as both an insurance policy and a repair contract. This has given rise to a third theory of warranty, the investment theory. Under this theory, the warranty is seen as an investment by the buyer to reduce the risk of early failure. Manufacturers are insured against having to rectify problems caused by inappropriate use while the buyer is covered for repair costs of premature failures. The aim is to extend the useful life of the product by specifying responsibilities of the manufacturer and the buyer. Blischke’s (1990) was the first review paper on warranties and it dealt with mathematical models for warranty cost analysis. Since then the literature on warranties (for both new and used products) has grown considerably.

1.6.7 Two Dimensional Modeling

In economics, The Cobb Douglas functional form of production functions is extensively used to characterize the relationship of an output to input. It was proposed by Knut Wicksell (1851–1926), and tested against statistical evidence by Charles Cobb and Paul Douglas in 1900–1928. The function of Cobb Douglas present a simplified outlook of the economy in which production output is obtained by the amount of labor occupied and the amount of capital invested. While there are many factors influencing economic performance, their model demonstrated remarkable accuracy. The mathematical form of the production function is specified as:

\[ Y = A L^v K^{-v} \]

Where:

- \( Y \) = total production (the monetary value of all goods produced in a year),
- \( L \) = labor input
- \( K \) = capital input,
- \( A \) = total factor productivity \( v \) is elasticity of labor.

This value is constant and determined by available technology.

The assumptions made by Cobb and Douglas can be stated as follows:

1. If either labor or capital vanishes, then so will production.
2. The marginal productivity of labor is proportional to the amount of production per unit of labor.
3. The marginal productivity of capital is proportional to the amount of production per unit of capital.

The Cobb-Douglas function based on the above assumptions is very appealing. The basic characteristic of this function is linearly homogeneous with constant return to scale i.e. a proportion increase in all inputs leads to same proportion increase in output.

Studying the dynamics of the technology diffusions under the key determinants that influence the adoption of a technology across time and/or space into the market is crucial to assess the business case for new technologies. The topic diffusion has been widely studied by researchers from different disciplines, including Sociology, Economics, Psychology and Marketing. However a substantial amount of research has been focused on one dimension: either to examine the individual's adoption of an innovation or to explain the time path of adoption of technologies typically following an S-shaped curve. The other dimensions of the diffusion of an innovation, has gained less attention. Therefore it’s imperative to study a two-dimensional technology innovation diffusion model which combines the adoption time of technological diffusion and price of the technological product. In this context Kapur et. al (2010) proposed a two dimensional innovation diffusion model, as represented below:

\[
T(x) = \frac{dN(x)}{dx} = p(m-N(x)) + q \frac{N(x)}{m} (m-N(x)) = (p+q) \frac{N(x)}{m} (m-N(x))
\]  

(1.45)

Using the Cobb–Douglas functional form, the value of the product can be given as:

\[x(t, p) = t^\alpha . p^{1-\alpha}\]

where, \(\alpha\) represents the effect of product valuation to the innovation diffusion model. The physical interpretation of the above model is, an adopter will purchase a product due to the product valuation \(x\) is a function of the adoption variation (due to the product value) and the
number of remaining purchasers. With the initial condition \( x(0,0)=0 \Rightarrow N(0,0)=0 \), on solving Eq. (1.45) we have

\[
N(t, p) = m \left( \frac{1 - e^{-(p+q)t^\alpha \cdot p^{-\alpha}}} {1 + \frac{q}{p} e^{-(p+q)t^\alpha \cdot p^{-\alpha}}} \right)
\]  

(1.46)

It is this function that we have utilised in framing some of the problems in the thesis. Talking in terms of diffusion, time and price are the two factors which are very important for any organization, company and individual for optimization. The rationale being, both of these are limited and precious too.

1.7 SOFTWARE RELIABILITY MODELING

Software engineering is relatively a young disciple and was first proposed in 1968 at a conference held to discuss the problem known at that time as software crisis. Software crisis was the result of introduction of the powerful, third generation computer hardware. Many software projects run over budget and schedule and were unreliable, difficult to maintain and performed poorly. The software crisis was originally defined in terms of productivity, but evolved to emphasize quality. New techniques and methods were needed to control the complexities of the large software projects and the techniques developed and adopted lead to the foundation of SRE. The SRE technologies were mainly inherent (such as specification, design, coding, testing and maintenance techniques) that aid in software development directly and management technologies (such as quality and performance assessment and project management) that support the development process indirectly. Our focus in this thesis lies on the management technologies. The IEEE (Musa, (1998)) society has defined SRE as widely applicable, standard and proven practice that apply systematic, disciplined, quantifiable approach to the software development, test, operation, maintenance and evolution with emphasis on reliability.

**SRE Management Techniques Works by Applying Two Fundamental Ideas**

- Deliver the desired functionality more efficiently by *quantitatively characterizing* the expected use, use this information to optimize the resource usage focusing on the most used and/or critical functions and make testing environment representative of operational environment.
- Balances *customer needs* for reliability, time and cost effectiveness. It works by setting quantitative reliability, schedule and cost objectives and engineers strategies to meet these objectives.

### 1.7.1 Software Development Life Cycle

The generic process framework applicable to the vast majority of software projects includes

- *Analysis and specification*
- *Software development*
- *Verification and validation*
- *Maintenance and evolution*

Each framework activity is populated by a set of software engineering actions such as software project tracking and control, risk management, quality assurance and measurement, technical reviews, reusability measurement etc. Following the generic framework activities every software development and engineering organization describe a unique set of activities it adopts with the complemented set of engineering actions in terms of a task set that identifies the work to be accomplished known as the process model for the SDLC. Many life cycle models have been proposed, based on the tasks involved in developing and maintaining software, but they all consist of the following stages and faults can be introduced during any of the stages. The framework activities of the SDLC model are shown in figure 1.14.

![Figure 1.14: Various Steps involved in Software Development Life Cycle](image)

**Activity 1: Requirement Analysis**

This phase is also known as feasibility study phase. In this phase, the development team visits the customer and studies their system. It involves heavy communication with the user, investigation of the need for possible software automation in the given system. By the end of the feasibility study, the team furnishes a document that holds the different specific recommendations for the candidate system. It also includes the personnel assignments, costs, project schedule and target dates. To understand the nature of the programs to be built, the
system engineer must understand the information domain for the software, as well as required function, behavior, performance and interfacing. The essential purpose of this phase is to find the need and to define the problem that needs to be solved. The final outcome of this phase is the document called as requirement specifications.

**Activity 2: System Analysis and Design**
The basic task of this phase is to transform the requirement specifications into design specifications. In this phase, the software's overall structure and its nuances are defined. In terms of the client/server technology, the number of tiers needed for the package architecture, the database design, the data structure designs etc. are all defined in this phase. A software development model is created. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. Much care is taken during this phase. The logical system of the product is developed in this phase.

**Activity 3: Program Design and Implementation**
The design specifications are translated into the program design document, which consist of definitions of the various modules of the software and the basic structure of the programs to be coded in some programming language (a machine-readable form). The code generation step performs this task. If the program design is performed in a detailed manner, code implementation can be accomplished without much complication. Programming tools like compilers, interpreters; debuggers are used to generate the code. Different high level programming languages like C, C++, Visual basic and Java are used for coding. With respect to the type of application, the right programming language is chosen. Once the independent programs are implemented they are linked to form the modular structure of the software.

**Activity 4: Integration and Testing**
Once the code is generated, the software program testing begins. Testing is the key method for dynamic verification and validation of a system. Its objective is to uncover as many faults as possible with a minimum cost. Testing is generally focused on two areas: *internal efficiency* and *external effectiveness*. The goal of internal testing is to make sure that the computer code is efficient, standardized and well documented. The goal of external effectiveness testing is to verify that the software is functioning according to system design and that it is performing all necessary functions or sub-functions. Initially testing begins with unit testing of independent
modules then the modules are integrated and system testing is performed followed by acceptance testing.

**Activity 5: Operation and Maintenance**

The system or system modifications are installed and made operational in the operational environment. The phase is initiated after the system has been tested and accepted by the user. This phase continues until the system is operating in accordance with the defined user requirements. Inevitably the system will need maintenance during its operational use. During this period the software is maintained by the developer to conquer all the faults that remain in it at its release time. Software will definitely undergo change once it is delivered to the customer. There are many reasons for a potential change. Change could happen because of some unexpected input values into the system. Changes in the system could directly affect the software operation. The software should be developed to accommodate changes that could happen during the post implementation period.

1.7.2 **Significance of Software Testing**

Despite using the best engineering methods and tools during each stage of the software development the software is subject to testing in order to verify and validate it. The previous discussion on the importance of computing systems and human dependence on them clarifies the need of software testing. Bugs if appear during software operation in user environment can be fatal to the users in terms of loss of time, money and even lives depending on criticality of the function as well as to the developers in terms of cost of debugging, risk cost of failure and goodwill loss. The bugs in the software can be manifested in each stage of its development. Figure 1.15 shows how the bugs get manifested in the various stages of SDLC and the contributing factors.

The aim of software testing is nothing other than *quality improvement* in order to achieve the necessary reliability. Although defined in various ways basically software quality is defined as the attribute measuring how well the software product meets the stated user functions and requirements. Software testing involves checking processes such as inspections and reviews at each stage of the SDLC start form the requirement specification to coding. Ideally the test cases that are executed on the software to test the software are designed throughout its development lifecycle. Testing is inherent to every phase of the SDLC. Software testing in the testing phase is a three stage process in which first the systems individual components, programs and modules
are tested called *unit testing*, followed by *integration testing* at subsystem and system level which includes top-down and bottom-up testing, interface testing and stress testing and conclude with the *acceptance testing*.

![Figure 1.15: Sources of Faults in Each Phase of SDLC](image)

There is plethora of testing methods and testing techniques which can serve multiple purposes in different phases of SDLC. Testing is basically of four types: *defect testing, performance testing, security testing* and *statistical testing* (Sommerville, 1995).

### 1.7.3 Measuring Software Reliability

Reliability Assessment during the different phases of the software development is an attractive approach to the developer as it provides a quantitative measure of what is most important to them *software quality*. However computing an appropriate measure of reliability is difficult (Sommerville, 1995); it is associated with many difficulties such as

- **Operational profile uncertainty**: the simulated operational profile can’t reflect the real user operational profile accurately.

- **High costs of test data generation**: defining the large set of test data that covers each program statement, path and functions etc is very costly as it requires long time, expert experience.


- **Statistically uncertainty**: statistically significant number of failure data is required to allow accurate reliability measurement; measurements made with insufficient data involve huge uncertainty. With this choice of the appropriate metric and model used adds to the uncertainty of the reliability measurement.

Despite all these challenges to reliability measurement, reliability of the software is assessed during the different phases of the software development and is used for practical decision-making. Before we discuss how the reliability measure is actually made we must clearly understand the difference between the software failures, faults and errors [Kapur et al., (1999); Pham, (2006)]. A **software failure** is a software functional imperfection resulting from the occurrence(s) of defects or faults. A **software fault** or bug is an unintentional software condition that causes a functional unit to fail to perform its required function due to some error made by the software designers. An **error** is a mistake that unintentionally deviates from what is correct, right, or true; a human action that results in the creation of a defect or fault.

Reliability assessment typically involves two basic activities – **reliability estimation** and **reliability prediction**. Estimation activity is usually retrospective and determines achieved reliability from a point in the past to present using the failure data obtained during system test or operation. The prediction activity usually involves future reliability forecast based on available software metrics and measures. Depending on the software development stage this activity involves either early reliability prediction using characteristics of the software and software development process (case when failure data is unavailable) or parameterization of the reliability model used for estimation and utilizes this information for reliability prediction (case when failure data is available. In either activity reliability models are applied on the collected information, and using statistical inference techniques reliability assessment is carried out. In the view of theorists software reliability is a concept borrowed from hardware reliability. Next we specify the important similarities and dissimilarities.

### 1.7.4 Software vs Hardware Reliability

According to ANSI, Software Reliability is defined as: the probability of failure-free software operation for a specified period of time in a specified environment. Although Software Reliability is defined as a probabilistic function but it must be noted that it is different from traditional Hardware Reliability.
Software reliability strives systematically to reduce or eliminate system failures which adversely affect performance of a software program. Software systems do not degrade over time unless modified. There are many differences between the reliability and testing concepts and techniques of hardware and software. Therefore, a comparison of software and hardware reliability would be useful in developing software reliability modeling.

**Distinct Characteristics of Software and Hardware**

- **Faults**: Software faults are mainly design faults, whereas hardware faults are mostly physical faults.
- **Wear-out**: Software does not have energy related wear-out phase and thus errors can occur without any warning. On the other hand, hardware may become “old” and wear out with time and usage.
- **Requirable system concept**: Periodic restarts can help fix software problems but the same may not be applicable to hardware.
- **Reliability prediction**: Software reliability cannot be predicted from any physical basis, since it depends completely on human factors in design. On the other hand, hardware reliability may be predicted from any physical basis.
- **Testing**: Software essentially requires infinite testing, while hardware can usually be tested exhaustively.
- **Failure rate curves**: Over time, hardware exhibits the failure characteristics shown in Figure 1.16, known as the bathtub curve with three phases- burn-in phase, useful life phase and wear out (end-of-life) phase. Software reliability, however, does not show the same characteristics similar as hardware (shown in Figure 1.17).

Hardware reliability theory relies on the analysis of stationary processes, because only physical faults are considered. However, with the increase of systems complexity and the introduction of design faults in software, reliability theory based on stationary process becomes unsuitable to address non-stationary phenomena such as reliability growth or reliability decrease experienced in software. This makes software reliability a challenging problem, which requires an employment of several methods to attack (Lyu 1996). Because of this difference in the effect of faults, software reliability must be defined differently from hardware reliability.
1.7.5 Software Reliability Model Classification

Reliability models are powerful tools of SRE for estimating, predicting, controlling, and assessing software reliability. A software reliability model specifies the general form of dependence of the failure process/ reliability metrics and measurements on some of the principle factors that affect it: software and development process characteristics, fault introduction, fault removal, testing efficiency and resources, and the operational environment. Software reliability modeling has been a topic of practical and academic interest since the 1970s. Today the number of existing models exceeds hundred with more models developed every year. It is important to classify the existing models in the literature into different categories to as to simplify the model selection by the practitioners and further enhancement of the field. There have been various attempts in the literature to classify the existing models according to various criteria. Goel, (1985) classified reliability models into four categories namely, time between failure models, error count models, error seeding models and input domain models. Classification due to Musa et al, (1989) is according to time domain, category and the type of probabilistic failure distribution. Some other classifications are given by Ramamoorthy and Bastani, (1982); Xie, (1990); Popstojanova and Trivedi, (2001). A recent study due to Asad et al, (2004) classified software reliability models according to their application to the phases of SDLC into six categories. (Refer Figure 1.18).

The models of early prediction and architecture based category are together known as called as white box models which regard software as consisting of multiple components, while those in categories of hybrid white box models and black box models known as black box models which regard software as a single unit. Black box models are studied widely by many eminent research scholars and engineers.
Popstojanova and Trivedi, (2001) classified black box models as failure rate models, failure count models, static models, Bayesian models and Markov models. Most of the research work in software reliability modeling is done on failure count models, Bayesian models and Markov models. We give below a brief description of these categories.

**Fault Counting Models** A fault counting model describes the number of times software fails in a specified time interval. Modes in this category are assumed to describe the failure phenomenon by stochastic processes in discrete and continuous time space like Homogeneous Poisson Process (HPP), Non-Homogeneous Poisson Process (NHPP), and Compound Poisson Process etc. The majority of these failure count models are based upon the NHPP. The pioneering attempt in NHPP based software reliability has been made by Goel and Okumoto, (1979). In the section 1.8 of this chapter we discuss the NHPP based SRGM in detail.

**Markovian Models** A Markov process represents the number of detected faults in the software system by a Markov process. The state of the process at time $t$ is the number of faults remaining
at that time. The Markov assumption implies the memory less property of the process, which is a helpful simplification of many stochastic processes and is associated with the exponential property. Jelinski and Moranda (JM), (1972); Schick & Wolverton, (1978); Cheung, (1980); Goel, (1985); Littlewood, (1987) are examples of some Markov models.

Models based on Bayesian Analysis In the previous two categories the unknown parameters of the models are estimated either by the least squares method or by the maximum likelihood method (later in this chapter both these methods are briefly discussed). But in this category of models, the Bayesian analysis technique is used to estimate the unknown parameters of the models. This technique facilitates the use of information obtained by developing similar software projects. Based on this information the parameters of given model are assumed to follow some distribution (known as priori distribution). Littlewood and Verral, (1979) proposed the first software reliability model based on Bayesian Analysis.

1.7.6 Software Reliability Model Selection
A very important aspect of software reliability modeling and application of the models to the reliability measurement is to determine which model to be use for a particular situation. Models that are good in general are not always the best choice for a particular data set, and it is not possible to know in advance which model should be used in any particular case. We do not have a guideline with high confidence level, which can be followed to choose any particular model. No one has succeeded in identifying a priori the characteristics of software that will ensure that a particular model can be trusted for reliability predictions [Asad et al, (2004)]. Previously most of the tools and techniques used trend exhibited by the data criterion for model selection. Among the tools that rank models is AT&T software reliability engineering toolkit. This tool can be used for only few software reliability growth models. Asad et al, (2004) discussed various criteria to be used to in the order of their importance to select a model for a particular situation. Following are the criteria specified by them.

*Life cycle phase; Output desired by the user; Input required by model; Trend exhibited by the data; Validity of assumptions according to data; Nature of project; Structure of project; Test process; Development process.*

The author suggests, choosing the best model to apply to a particular reliability measurement situation, first select the life cycle phase and then find the existing reliability models applicable to that phase. Define deciding criteria, their order of importance and assign weights to each criterion. For each criterion give applicability weights to each model, multiplying the criterion
and applicability weights one obtains the models with high scores, which can be use to measure the reliability for that case.

**1.7.7 NHPP Based Software Reliability Growth Modeling**

Among the class of discrete process, counting process in reliability engineering widely used to describe the occurrence of an event of time (e.g. failure, repair, etc.). Poisson process is used most widely to describe a counting process as in reliability engineering. NHPP has been used successfully in hardware reliability analysis to describe the reliability growth and deteriorating trends. Following the trends in hardware reliability analysis many researchers proposed and validated several NHPP based SRGM. SRGM describe the failure occurrence and/or failure removal phenomenon with respect to time (CPU time, calendar time or execution time or test cases as unit of time) and/or resources spent on testing and debugging during testing and operational phases of the software development.

NHPP based SRGM are broadly classified into two categories first – *continuous time models*, which uses time (CPU time, calendar time or execution time) as a unit of fault detection period and second – *Discrete time models*, which adopt the number of test occasions/cases as a unit of fault detection period. Models can also be categorized as *concave* and *S-shaped* depending upon the shape of the failure curve described by them. Concave models describe an exponential failure curve while second category of models describes an S-shaped failure curve (Kapur et al (1999), Pham (2006)). The two types of failure growth curves are shown in figure 1.19 and 1.20. The most important property of these models is that they have the same asymptotic behavior in the sense that the fault detection rate decreases as the number of detected defects increases and approaches a finite value asymptotically. The S-shaped curve describes the early testing process to be less efficient as compared to the later testing i.e. it depicts the learning phenomenon observed during testing and debugging process.

During the last three decades several researchers devoted their research interest to NHPP based software reliability modeling and contributed significantly in understanding the testing and debugging process and developing quality software. Schneidewind (1975) made the preliminary attempt in NHPP based software reliability modeling. He assumed exponentially decaying failure intensity and rate of fault correction proportional to the number of faults to be corrected.
Goel and Okumoto (1979) presented a reliability model where intensity function is proportional to the remaining number of faults in the software. Their research was a pioneering attempt in the field of software reliability growth modeling and paved the way for research on NHPP based software reliability modeling. The model describes the failure occurrence phenomenon by an exponential curve. The research following GO model was mainly modifying the existing research in the way of incorporating the various aspects of the real testing environment and strategy. Some of the existing NHPP based SRGM that have been used in this thesis are; *Modeling under perfect debugging environment; Modeling fault severity; Software reliability assessment using SDE, Modeling using Unified Approach.*

In the later part of the chapter i.e. in Section 1.8 we briefly discuss these widely known modeling categories.

### 1.7.8 General Description of a Non-Homogeneous Poisson Process

**(NHPP in Continuous Time Space)**

A counting process \((N(t), t \geq 0)\) is said to be an NHPP with mean value function \(m(t)\), if it satisfies the following conditions:

1. There are no failures experienced at \(t=0\), that is, \(N(0) = 0\).
2. The counting process has independent increments, i.e. for any finite collection of times \(t_1 < t_2 < \ldots < t_k\), the \(k\) random variables \(N(t_1), N(t_2)-N(t_1), \ldots, N(t_k)-N(t_{k-1})\) are independent.
3. \(\Pr(\text{exactly one failure in } (t, t + \Delta t)) = \lambda(t) + o(\Delta t)\)
4. \(\Pr(\text{two or more failures in } (t, t + \Delta t)) = o(\Delta t)\)

where \(\lambda(t)\) is intensity function of \(N(t)\). If we let \(m(t) = \int_0^t \lambda(x)dx\) then \(m(t)\) is a non-decreasing, bounded function representing the mean of number of faults removed in the time interval \((0,t]\) (Kapur *et al*, (1999,2011)). It can be shown that...
\[ \Pr[N(t) = k] = \frac{(m(t))^k e^{-m(t)}}{k!}, \quad n = 0, 1, 2, \ldots \]

i.e. \( N(t) \) has a Poisson distribution with expected value \( E[N(t)] = m(t) \) for \( t > 0 \).

and the reliability of the software in the time interval of length \( x \) is given as

\[ R(x | t) = e^{-(m(t+x)-m(t))} \]

In case we have \( n \), the number of test cases as the unit of time then \( t \) is replaced by \( n \) and \( \Delta t \) is replaced by 1, and describes NHPP based SRGM in discrete time space.

**1.8 NHPP BASED SRGMs: LITERATURE SURVEY**

A large number of SRGMs have been developed in the literature to describe the failure and fault removal phenomenon of software in continuous and discrete time space. In this section we introduce briefly some of the existing SRGM in the literature.

**Notations**

- \( m(t) \): expected number of failure/removal by time \( t \), \( m(0) = 0 \).
- \( m_f(t) \): expected number of failure by time \( t \), \( m_f(0) = 0 \).
- \( m_r(t) \): expected number of removals by time \( t \), \( m_r(0) = 0 \).
- \( a(a_{i=ap_i}) \): number of fault in the software at the time of start of software testing.
- \( a(t) \): expected initial fault content at time \( t \), \( a > 0 \).
- \( p \): probability of perfect debugging of a fault, \( 0 < p < 1 \).
- \( \alpha \): constant rate of error generation, \( 0 < \alpha < 1 \).
- \( b(t) \): time dependent rate of fault removal per remaining faults.
- \( \beta \): constants

**1.8.1 SRGMs under perfect debugging environment**

The earlier attempts in NHPP based reliability growth modeling assumed a perfect debugging environment, which means whenever an attempt is made to remove a detected fault it is removed perfectly and no new faults are generated. Various models have been proposed in this category. The earliest model is due to Goel and Okumoto (1979).

This model is based on NHPP and assumes that failure intensity is proportional to the remaining number of faults in the software. So the corresponding differential given by:
The above first order linear differential equation when solved with the initial condition 
\( m(0) = 0 \) gives the following mean value function for NHPP (1.47)

\[ m(t) = a(1 - e^{-bt}) \]  

(1.48)

The mean value function is exponential in nature and doesn't provide a good fit to the S-Shaped growth curves that generally occur in Software Reliability. But the model is popular due to its simplicity. Yamada et al (1983) refined the GO model describing testing as a two-stage process namely fault detection and removal. The model describes an S-shaped failure curve. SRGM proposed by Obha (1984a), Bittanti et al (1988) and Kapur and Garg (1992) are similar to Yamada et al (1983) but are developed under different set of assumptions. These models can describe both exponential and S-shaped growth curves depending on the parameter values and therefore are termed as flexible models or S-shaped models. Some of these models are discussed here.

**Delayed S-Shaped SRGM (Yamada, Ohba and Osaki 1983)**

This model is assumes to be a two-phase testing process consisting of failure detection and its eventual removal by isolation. It takes into account the time taken to isolate and remove a fault and so it is important that the data to be used here should be that of fault isolation. It is further assumed that the number of faults isolated at any time instant is proportional to the number of faults remaining in the software. Failure rate and isolation rate per fault are assumed to be same and equal to \( b \).

Thus

\[ \frac{d}{dt} m_f(t) = b[a - m_f(t)] \]  

(1.49)

\[ \frac{d}{dt} m(t) = b[m_f(t) - m(t)] \]  

(1.50)

\( m_f(t) \) is the expected number of failures in \((0, t]\). Solving (1.49) and (1.50), we get the mean value function as

\[ m(t) = a\left[1 - \left(1 + bt\right)e^{-bt}\right] \]  

(1.51)

Alternately the model can also be formulated as one stage process directly as follows:

\[ \frac{d}{dt} m(t) = \left(\frac{b^2t}{1+bt}\right)(a - m(t)) \]  

(1.52)
It is observed that $\frac{b^2 t}{1 + bt} \rightarrow b$ as $t \rightarrow \infty$. This model was specifically developed to account for lag in the failure observation and its subsequent removal. Xie and Zhao (1992) have also proposed alternative ways of deriving the above model.

**Inflection S-Shaped SRGM (Obha 1984)**

The model attributes S-Shapedness to the mutual dependency between software faults. Here apart from the usual assumptions it is also assumed that the software contains two types of faults, namely mutually dependent and mutually independent. The mutually independent faults are those located on different execution paths of the software, therefore they are equally likely to be detected and removed. The mutually dependent faults are those faults located on the same execution path. According to the order of the software execution, some faults in the execution path will not be removed until their preceding faults are removed.

Let $r$ denote the ratio of independent faults to the total number of faults in the software. This ratio is called the inflection parameter ($0 < r \leq 1$). If all faults in the software system are mutually independent ($r = 1$) then the faults are randomly removed and the growth curve is exponential. According to the assumptions of the model, the fault removal intensity per unit time can be written as

$$\frac{d}{dt} m(t) = b(t) [a - m(t)]$$  \hspace{1cm} (1.53)

$b(t)$, the fault removal rate at time $t$ is defined as

$$b(t) = b \phi(t)$$

Where, $\phi(t)$ the inflection function is defined as

$$\phi(t) = r + (1 - r) \frac{m(t)}{a}, \quad \phi(0) = 0 \text{ and } \phi(\infty) = 1$$  \hspace{1cm} (1.54)

$b$ is the fault removal rate in the steady state. Solving (1.53) under the initial condition $m(0) = 0$ we get

$$m(t) = a \left[ \frac{1 - e^{-bt}}{1 + \left( \frac{1 - r}{r} \right) e^{-bt}} \right]$$  \hspace{1cm} (1.55)
If $r = 1$, the model reduces to the Goel-Okumoto model (1979). For different values of $r$ different growth curves can be obtained and in that sense the model is flexible.

SRGM for an Error Removal Phenomenon (Kapur and Garg 1992)

This model is based upon the following additional assumption: On a failure observation, the fault removal phenomenon also removes some additional faults, without causing any failure.

Based on the assumption the fault removal intensity per unit time can be written as

$$\frac{d}{dt} m(t) = p[a - m(t)] + q \frac{m(t)}{a}[a - m(t)]$$  \hspace{1cm} (1.56)

Solving equation (1.56) with the usual initial condition, the expected number of faults detected in $[0, t]$ is given as

$$m(t) = a \left[ \frac{1 - e^{-(p+q)t}}{1 + \left( \frac{q}{p} \right) e^{-(p+q)t}} \right]$$  \hspace{1cm} (1.57)

Which is similar to equation (1.57) derived under different assumptions.

Alternately the model can also be formulated as one stage process using a logistic rate function directly (Kapur et al (1999, 2011))

$$\frac{d}{dt} m(t) = \left( \frac{b}{1 + e^{-\beta}} \right) (a - m(t))$$  \hspace{1cm} (1.58)

Where, $\beta = q/p$ and $b = p + q$ gives the same result as by equation (1.57).

1.8.2 Modeling Related to Faults Severity

Different faults may require different amount of testing efforts and testing strategy for their removal from the system. In the literature to incorporate this phenomenon faults are categorized as of different types and are analyzed separately. The first attempt in this category was due to Yamada and Osaki (1985) who modified G-O exponential SRGM assuming that there are two types of faults in the software. Pham (2006) incorporated the effect of error generation in GO model to analyze the reliability growth considering three level complexities of the faults. Both models assume different fault detection rate for each type of faults. Kapur et al (1995b)
addressed three level complexities of faults considering the time delay between the failure observation and its subsequent removal.

**Generalized Erlang SRGM (Kapur et al (1995b))**

Assuming $a_1$, $a_2$, $a_3$ to be simple, hard and complex faults in a software system ($a_1 + a_2 + a_3 = a$), The simple fault removal process is modeled as the following:

$$\frac{d}{dt} m(t) = b_1[a_1 - m(t)] \quad (1.59)$$

Solving, We get

$$m(t) = a_1(1 - e^{-bt}) \quad (1.60)$$

The hard fault removal process is modeled as a two stage process

$$\frac{d}{dt} m_2(t) = b_2[m_2(t) - m(t)] \quad (1.62)$$

Solving, We get

$$m_2(t) = a_2(1 - (1 + b_2t)e^{-bt}) \quad (1.63)$$

The Complex fault removal process is modeled as a three stage process

$$\frac{d}{dt} m_3(t) = b_3[m_3(t) - m(t)] \quad (1.65)$$

Solving, We get

$$m_3(t) = a_3\left[1 - (1 + b_2t + \frac{b_3^2t^2}{2})e^{-bt}\right] \quad (1.67)$$

The mean value function of the proposed SRGM is

$$m(t) = m_1(t) + m_2(t) + m_3(t) \quad (1.68)$$

Assuming $a_2 = p.a$, $a_3 = q.a$ and $a_1 = (1 - p - q)a$,
\[ m(t) = a[(1 - e^{-bt})(1 - p - q) - p(1 + b_2t)e^{-bt} - q(1 + b_3t + \frac{b_3^2t^2}{2})e^{-bt}] \]  

(1.69)

Assuming \( b_1 = b_2 = b_3 = b \), we have

\[ m(t) = a[1 - e^{-bt}(1 + (p + q)bt + q \frac{b^2t^2}{2})] \]  

(1.70)

The mean value function obtained in (1.70) can be generalized to include \( n \) different types of faults depending upon their severity. We may write

\[ m(t) = \sum a_i[1 - e^{-b_{ir}t}\sum_{j=0}^{i-1}\frac{(bt)^j}{j!}] \]  

(1.71)

**1.8.3 SDE Based Modeling**

We have already discussed about the concept behind stochastic differential equation based modeling earlier in section 1.6.4. Here, we describe a SRGM depicting the use of irregular fluctuations in the fault detection rate.


Most of the existing SRGM assume the software fault detection as the continuous process. However, if the size of the software is large, number of software faults detected during testing phase becomes large and the change of the faults which are detected and removed through debugging activities becomes sufficiently small compared with the initial fault content at the beginning of the testing phase. In such a situation, we can model the software fault detection process as a stochastic process with continuous state space. So, its equation will be given by

\[ \frac{dN(t)}{dt} = r(t)[a - N(t)] \]  

(1.72)

Where \( N(t) \) be a random variable which represents the number of faults detected in the software up to time \( t \). here we assume that \( r(t) \) has irregular fluctuations i.e. \( r(t) = b(t) + \sigma \gamma(t) \), so above equation can be written as

\[ \frac{dN(t)}{dt} = [b(t) + \sigma \gamma(t)][a - N(t)] \]  

(1.73)

Where \( \sigma \) is positive constant representing magnitude of the irregular fluctuations and \( \gamma(t) \) a standardized Gaussian white noise.
By using the Ito formula, we can obtain the solution process as follows:

\[
N(t) = a \left[ 1 - \exp \left( -\int_0^t b(x) dx - \sigma W(t) \right) \right]
\]  

(1.74)

As we know that the Brownian motion or Wiener process follows normal distribution. The density function of \( w(t) \) is given by:

\[
f(w(t)) = \frac{1}{\sqrt{2\pi t}} \exp \left( -\frac{(w(t))^2}{2t} \right).
\]

Thus the mean number of detected fault is given as:

\[
m(t) = E[N(t)] = a \left[ 1 - \exp \left( -\int_0^t b(x) dx + \frac{1}{2} \sigma^2 t \right) \right]
\]

(1.75)

We put different values for \( b(x) \) and obtain the required model. i.e. if we put \( b(x) = b \); we obtain \( m(t) \) for GO model as:

\[
m(t) = E[N(t)] = a \left[ 1 - \exp \left( -bt + \frac{\sigma^2 t}{2} \right) \right]
\]

(1.76)

1.8.4 Software Reliability Growth Model Using Unified Approach

Unified modeling approach is one of the recent topics of research in software reliability. Two research directions for SRGM are usually considered: unification and parameter estimation of SRGM. In fact, a unified modeling framework comprising some typical reliability growth patterns should be developed for robust software reliability assessment. There are some, but only a few, model unification schemes in the literature. Langberg and Singpurwalla (1995) showed that several SRGM can be comprehensively described by adopting a Bayesian point of view. Miller (1986) and Thompson, Jr. (1988) extended the Langberg and Singpurwalla’s idea and developed a generalized Order Statistic models (GOS). More precisely, they showed that almost all SRGM can be explained within the framework of GOS and record value and claim especially for the NHPP models that the model selection problem is reduced to a simple selection problem of fault detection time distribution. Based on their result the mean value function in NHPP models that can be characterized by theoretical probability distribution function of fault detection time. Chen and Singpurwalla (1997) proved that all SRMs as well as the NHPP models developed in the past literature can be unified by self-exciting point processes. Apart from the probabilistic approach, Huang et al. (2003) explained the deterministic behavior of the NHPP models, namely mean value function of time by introducing several kinds of mean operation.
Pham et al. (1997) solved a generalized differential equation by which the mean value function in the NHPP model is governed and proposed an NHPP with a generalized mean value parameter.

Therefore, we see that during last three decades many SRGM have been proposed in the literature (Musa et. Al 1987, Kapur et. al 1999, Pham 2006), which relates the number of failure (faults identified/corrected) and execution time. These SRGMs assumes diverse testing and debugging (T&D) environments like distinction between failure and correction processes, learning of the testing personnel, possibility of imperfect debugging and fault generation, constant or monotonically increasing/decreasing fault detection rate (FDR) or randomness in the growth curve. These SRGMs have been applied successfully in many real life software projects but no SRGM can claim to be the best in general as the physical interpretation of the testing and debugging changes due to numerous factors e.g., design of the test cases, defect density, skills and efficiency of the testing team, availability of testing resources etc. The plethora of SRGMs makes the model selection a tedious task. As stated earlier, to reduce this difficulty unified modeling approach have 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.

From the above literature it is clear that the work in this area started quite early. Infact we say it started as early as in 1980s with Shantikumar (1981) proposing a generalized birth process model. Gokhale and Trivedi (1996) used Testing coverage function to present a unified framework and showed how NHPP based models can be represented by probability distribution functions of fault–detection times. Dohi et al (2004) proposed a unification method for NHPP models describing test input and program path searching times stochastically by an infinite server queuing theory. Inoue (2006) applied infinite server queuing theory to the basic assumptions of delayed S-shaped SRGM (1983) i.e. fault correction phenomenon consists of successive failure observation and detection/correction processes and obtained several NHPP models describing fault correction as a two stage process.

Another unification methodology is based on a systematic study of Fault detection process (FDP) and Fault correction process (FCP) where FCPs are described by detection process with time delay. The idea of modelling FCP as a separate process following the FDP was first used by Schneidewind (1975). More general treatment of this concept is due to Xie et. al (1992) who suggested modelling of Fault detection process as a NHPP based SRGM followed by Fault
correction process as a delayed detection process with random time lag. The recent unification scheme (due to Kapur et al. (2004)) is based on Cumulative Distribution Function for the detection/correction times. The authors have used the concept of hazard rate in order to study fault detection rate. We are aware that the rate at which the failure occurs in a certain time interval \([t, t + \Delta t]\) is called failure rate. Thus hazard rate is defined as the probability that a failure per unit time occurs in the interval, given that a failure has not occurred prior to \(t\), the beginning of the interval. Thus the failure rate is \(\frac{R(t) - R(t + \Delta t)}{\Delta t R(t)}\).

The hazard rate/function is defined as the limit of the failure rate as the interval approaches to zero. Thus the hazard rate function \(h(t)\) is instantaneous failure rate, and is defined by

\[
h(t) = \lim_{\Delta t \to 0} \frac{R(t) - R(t + \Delta t)}{\Delta t R(t)}
\]

(1.77)

\[
= \frac{1}{R(t)} \left[ -\frac{d}{dt} R(t) \right]
\]

(1.78)

\[
= \frac{f(t)}{R(t)} = \frac{f(t)}{\int_{t}^{\infty} f(t)\, dt} = \frac{f(t)}{1 - F(t)}
\]

(1.79)

In addition to above literature one more unification scheme has been discussed based on Queuing theory approach. Moreover, the hazard rate approach proved to be fruitful in obtaining several SRGMs by following single methodology (Kapur et al 2011) and thus present a perspective investigation for study of general models without making any assumptions. We have therefore used this approach in modeling some new SRGMs in the present thesis.

1.9 SUCCESSIVE SOFTWARE RELEASES

The present software development environment is very competitive and advanced. Many independent and well established developing organizations are competing in the market with similar products and capabilities to attain the maximum market share and brand value. As such software delivered with full functionalities and very high reliability built over a period of time may turn out to be unsuccessful due to technological obsolescence. Therefore now a day’s the software are rather developed in multiple releases where the latest releases might be developed by improving the existing functionality and revisions, increasing the functionality, a combination of both or improving the quality of the software in terms of reliability etc. For example we can see the various software in the market named as Windows 98, Windows 2000, Windows ME,
Windows XP, Windows Vista, Windows 7 etc. For another illustration consider a development firm developing antivirus software. Such a firm can begin with releasing the product that detects and remove viruses and spywares from the computer system. In their second release they can provide the feature of protecting the system from virus infected emails. Next, they can add the trait of blocking spyware automatically for the next release. Finally, the fourth release can provide the root kit protection along with removing hidden threats in the operating system.

This step by step release (base software with features enhancement) is advantageous for the developing firms in various contexts. Firstly, if a firm implements the complete characteristic capabilities in first release, than that would delay the product release in the market in the desired time window. Secondly, launching of new software product may bring the developing firm in limelight, but the stream of subsequent product releases is the source of their bread and butter. Moreover releasing different versions of the product lengthen the market life of product, protect competitive advantages and sustain crucial maintenance revenue streams.

Software products aren’t static and each release has a limited existence. As soon as a software product reaches the market, a variety of factors begin creating demand for changes (Figure 1.21). Defects require repairs. Competitors offer software with added features and functions. Evolving technology requires upgrades to support new hardware and updated versions of operating software. Sales demands new features to win over prospects. Customers want new functionality to justify maintenance expenditures. These demands accumulate over time, eventually reaching the point when the software product must be upgraded with a new version to remain viable in the market. As soon as the new version is released, the cycle begins again.

![Figure 1.21: Necessity for successive releases](image-url)
For software developing organizations it is not an easy task to design software in isolation. Developing reliable software is one of the most difficult problems faced by them. Timely development, resource limitations, and unrealistic requirements can all negatively impact software reliability. Moreover, there is some interdependence between their developments. The interdependence between their developments exists in many ways, which also affects their reliability. A new release (an upgraded version of the base software) may come into existence even during its development, at the time of release or during its operation. The code and other documents related to a release may be some modification of the existing code and documents and/or addition of new modules and related modification in the documents. The dependence of the development process of successive releases necessitates considering this dependence in the reliability growth analysis.

The traditional software reliability growth models fail to capture the error growth due to the software enhancements in user-end (figure 1.22). In the useful-life phase as the software firm introduced new add-ons or features on the basis of the user needs, software will experience an increase in failure rate each time an upgrade is made. The failure rate levels off gradually, partly because of the defects found and fixed after the upgrades. Due to the feature upgrades, the complexity of software is likely to be increased as the functionality of software is enhanced (figure 1.23). Even fixing bugs may induce more software failures by fetching other defects into software. But if the goal of the firm is to upgrade the software by enhancing its reliability then it is possible to incur a drop in software failure rate that can be done by redesigning or re-implementing some modules using better engineering approaches.

![Figure 1.22 Traditional failure rate curve for Software Systems](image)
Not much work has been done in the modeling framework for multi releases failure process. In the earlier work Kapur et al (2007) proposed initial development in this field by proposing multiple releases of a software in operational phase for product and project type software. Later Kapur et al (2010) also described the phenomenon specially for testing phase by reformulating modeling methodology that shows the dependence of operational phase of the previous release with testing phase of the current release. In this thesis we have formulated SRGMs applicable to describe the failure\removal phenomenon of multiple releases of software in testing phase and also considered its dependency with operational phase wherein the expected number of failure and consequently the faults removed in testing phase is modeled by the joint effect of failure\removal phenomenon of successive releases.

1.10 MODEL APPLICATION
Mathematical models are applied on the actual data and model parameters are estimated. These estimates are used to generate information useful to the practitioners or fed to the optimization models to make decisions. It is a known fact that most of the times success of a model in a particular situation is due to judicious choice of situation, process, environment and time frame for evaluating the data. Generation of useful information depends heavily upon quality of collected data and selection of a model relevant to the process and environment. Usually there are two types of collected data time domain data and interval domain data [Mahajan et al, (1986); Pham, (2006). The time domain data is record of the individual occurrence time of the events (adoption/failure) while the second type of data is characterized by counting the number of events occurring during a fixed time period for example hours, weeks etc. Most diffusion/ reliability models can handle both types of data as both consider the cumulative occurrence time of the event. Time domain data usually provides higher accuracy and require more data collection efforts whereas interval domain data is more readily available and used for practical purpose.
1.11 PARAMETER ESTIMATION

Most of the innovation diffusion models as well as NHPP based SRGMs are described by the non-linear functions. Method of Non-linear Least Square (NLLS) and Maximum Likelihood Estimate (MLE) [Schmittlein and Mahajan, (1982); Putsis, (1998); Kapur et al, (1999,2011); Hardie et al, (1998); Meade and Islam, (2006); Pham, (2006)] are the two widely used estimation techniques for non-linear models. Unlike traditional linear regression, which is restricted to estimating linear models, nonlinear regression methods can estimate models with arbitrary relationships between independent and dependent variables.

1.11.1 Non-Linear Least Square Method

Consider a set of observed data points \( (t_i, y_i); \ i = 1, 2, \ldots n \), where \( t_i \) is the observation time and \( y_i \) is the observed sample value. A mathematical model of the form \( m(x,t) \) is fitted on this data set. The model depends on the parameters \( x = \{ x_i; \ i = 1, 2, \ldots m \} \), for some \( \bar{x} \) we can compute the residuals,

\[
 f_i(\bar{x}) = y_i - m(\bar{x}, t_i) \tag{1.80}
\]

The method of least square determines the unknown parameters of the model by minimizing the sum square of these residuals for the observed data values.

1.11.2 Maximum likelihood Estimation Method

For the interval domain data points \( (t_i, y_i); \ i = 1, 2, \ldots n \), where \( t_i \) is the observation time and \( y_i \) is the cumulative observed sample value by the time \( t_i \), based on the NHPP assumptions the likelihood function is defined as

\[
 L = \prod_{i=1}^{n} \left[ \frac{m(t_i) - m(t_{i-1})}{(y_i - y_{i-1})!} (y_i - y_{i-1}) e^{-\{m(t_i) - m(t_{i-1})\}} \right] \tag{1.81}
\]

If the data set is time domain the likelihood function is defined as

\[
 L = \prod_{i=1}^{n} \frac{\int_{0}^{t_i} \lambda(x) dx}{\lambda(t_i)} \tag{1.82}
\]

Maximum likelihood estimation method yields the unknown estimates of the parameters maximizing the likelihood function. In most of the cases Log of the likelihood function is maximized as log function is monotonic and provides easy computations as compared to the actual likelihood function.
It requires numerical methods and huge computation time to solve the problem, which is not favored by the management and software engineering practitioners. Statistical software packages such as SPSS, SAS, Maple etc. help to overcome this problem in which we can use the inbuilt software functions to solve these kinds of optimization problems to find the estimates of nonlinear models. In our study we have used the Statistical Package for Social Sciences (SPSS), SAS and Maple to solve the problems. They are comprehensive and flexible statistical analysis and data management system. They take data from almost any type of file and use them to generate tabulated reports, charts, and plots of distributions and trends, descriptive statistics, and conduct complex statistical analysis.

1.12 COMPARISON CRITERIA FOR IDMs/SRGMs

The performance of SRGMs are judged by their ability to fit the past software fault data (goodness of fit) and to predict satisfactorily the future behaviour of the software fault removal process (predictive validity). Kapur, Garg and Kumar (1999), Musa (1987, 1999) have suggested the following attributes for choosing an SRGM.

**Capability:** The model should possess the ability to estimate with satisfactory accuracy metrics needed by the software managers.

**Quality of assumptions:** The assumptions should be plausible and must depict the testing environment.

**Applicability:** A model can be adjudged as the better one if it can be applied across software products of different sizes, structures, platforms and functionalities.

**Simplicity:** The data required for an ideal SRGM should be simple and inexpensive to collect. The parameter estimation should not be too complex and is easy to understand and apply even for persons without extensive mathematical background.

Other than the above qualitative aspects the following indices help in comparing Diffusion models/SRGMs. The term goodness of fit is used in two different contexts. In one context, it denotes the question if a sample of data came from a population with a specific distribution. In another context, it denotes the question of “How good does a mathematical model (for example a linear regression model) fit to the data”?
1. **The Mean Square Fitting Error (MSE):** The model under comparison is used to simulate the fault data, the difference between the expected values, $\hat{m}(t_i)$ and the observed data $y_i$ is measured by MSE as follows.

$$MSE = \frac{1}{k} \sum_{i=1}^{k} (\hat{m}(t_i) - y_i)^2$$  \hspace{1cm} (1.83)

where $k$ is the number of observations. The lower MSE indicates less fitting error, thus better goodness of fit (Kapur, Garg and Kumar 1999).

2. **Coefficient of Multiple Determination ($R^2$)** This Goodness-of-fit measure can be used to investigate whether a significant trend exists in the observed failure intensity. We define this coefficient as the ratio of the Sum of Squares (SS) resulting from the trend model to that from a constant model subtracted from 1, that is

$$R^2 = 1 - \frac{\text{residual SS}}{\text{corrected SS}}$$  \hspace{1cm} (1.84)

$R^2$ measures the percentage of the total variation about the mean accounted for by the fitted curve. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well. The larger, the better the model explains the variation in the data (Kapur, Garg and Kumar 1999).

3. **Prediction Error (PE):** The difference between the observation and prediction of number of failures at any instant of time $i$ is known as $PE_i$. Lower the value of Prediction Error better is the goodness of fit (Pillai and Nair 1997).

4. **Bias:** The average of PEs is known as bias. Lower the value of Bias better is the goodness of fit (Pillai and Nair 1997).

5. **Variation:** The standard deviation of PE is known as variation.

$$Variation = \sqrt{\frac{1}{N-1} \sum (PE_i - Bias)^2}$$  \hspace{1cm} (1.85)

Lower the value of Variation better is the goodness of fit (Pillai and Nair 1997).

6. **Root Mean Square Prediction Error:** It is a measure of closeness with which a model predicts the observation.

$$RMSPE = \sqrt{Bias^2 + Variation^2}$$  \hspace{1cm} (1.86)
Lower the value of Root Mean Square Prediction Error better is the goodness of fit (Pillai and Nair 1997).

### 1.12.1 Predictive Validity Criterion
Predictive validity is defined as the ability of the model to determine the future failure behavior from present and past failure behavior. This criterion was proposed by Musa et al (1987). Suppose \( t_k \) be the time, \( x_k \) is number of faults detected during the interval \((0, t_k]\), and \( \hat{m}(t_k) \) is the estimated value of the mean value function \( m_r(t) \) at \( t_k \), which is determined using the actually observed data up to an arbitrary time \( t_e(0 < t_e \leq t_k) \), in which \((t_e / t_k)\) denotes the testing progress ratio. In other words, the number of failures by \( t_k \) can be predicted by the SRGM and then compared with the actually observed number \( x_k \). The difference between the predicted value \( \hat{m}(t_k) \) and the reported value \( x_k \) measures the prediction fault. The ratio \( \left\{ \left( \hat{m}(t_k) - x_k \right) / x_k \right\} \) is called Relative Prediction Error (RPE). If the RPE value is negative / positive the SRGM is said to underestimate / overestimate the future failure phenomenon. A value close to zero for RPE indicates more accurate prediction, thus more confidence in the model and better predictive validity. The value of RPE is said to be acceptable if it is within \( \pm (10\%) \) Kapur et al (1999), Musa et al (1987).

### 1.13 GENETIC ALGORITHM
To solve the optimization problems of resource allocation and release planning decisions optimization techniques are applied to obtain the best possible course of action for the process being studied. The use of soft computing techniques in place of the traditional mathematical techniques has shown a noteworthy improvement in the predictions in the recent years. Unlike hard computing, soft computing techniques are tolerant of ambiguity, vagueness, and approximation. Soft computing techniques comprise of computational techniques like Genetic Algorithms, Genetic Programming, Ant Colony Optimization, Swarm Particle Optimization etc. These soft computing techniques are applicable to different aspects of software reliability which includes estimating the reliability of software system and solving various optimization problems. Genetic Algorithms are the most popular type of Evolutionary Algorithm (EA).
1.13.1. Biological Background for Genetic Algorithm

**Genetics:** The science that deals with the mechanism responsible for similarities and differences in a species is called genetics.

**Cell:** Every animal/human cell is a complex of many “small” factories that work together. The centre of all this is the cell nucleus. The genetic information is contained in the cell nucleus.

**Chromosomes:** All the genetic information gets stored in the chromosomes.

In the nucleus of each cell, the DNA molecule is packaged into thread-like structures called chromosomes. Each chromosome is build of “Dioxy Ribo Nucleic Acid” (DNA). Humans have 23 pairs of chromosomes. The chromosome is divided into several characteristics called genes. In genetic Algorithms, all the genes are usually stored on the same chromosomes. The chromosomes and genomes are synonyms with one other. Each chromosome has a constriction point called the centromere, which divides the chromosome into two sections, or “arms.” The short arm of the chromosome is labeled the “p arm.” The long arm of the chromosome is labeled the “q arm.” The location of the centromere on each chromosome gives the chromosome its characteristic shape, and can be used to help describe the location of specific genes. The structure of the chromosome is shown in Figure 1.24.

![Figure 1.24: Chromosome Structure](image)

**Reproduction:** Reproduction of species via genetic information is carried out by:

• **Mitosis:** The same genetic information is copied to new offspring.

• **Meiosis:** It form basis of sexual reproduction.

1.13.2 Principle of GA and its Working

Genetic Algorithm (GA) is a method that imitate biological evolution as a problem-solving strategy (Goldberg [1989]). Table 1.2 shows the comparison of natural evolution and genetic
algorithm terminology. As the name implies GA is based on the model of evolution, in which a population evolves towards overall fitness, even though individuals perish. Evolution dictates that superior individuals have a better chance of reproducing than inferior individuals, and thus are more likely to pass their superior traits on to the next generation. This “survival of the fittest” criterion was first converted to an optimization algorithm by Holland in 1975, and is today a major optimization technique for complex, nonlinear problems.

Table 1.2 Comparisons of Natural Evolution and GA

<table>
<thead>
<tr>
<th>Natural Evolution</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromosome</td>
<td>Individual String</td>
</tr>
<tr>
<td>Gene</td>
<td>Character</td>
</tr>
<tr>
<td>Allele</td>
<td>Feature Position</td>
</tr>
<tr>
<td>Locus</td>
<td>String Position</td>
</tr>
</tbody>
</table>

Genetic algorithms (GAs) are described by the natural process of evolution, which is a rapidly growing field of artificial intelligence.

GA is inspired by the theory of Charles Darwin about the natural evolution in the origin of species. Goldberg (1989) gave the introduction of GA.

The basic genetic algorithm is described in detail as follows (Sastry (2007)):

Step 1: Start

Step 2: Generate random population of chromosomes

Step 3: Evaluate the fitness of each chromosome in the population

Step 4: Create a new population by repeating following steps until the new population is complete:

- [Selection] Select two parent chromosomes from a population according to their fitness.
- [Crossover] With a crossover probability, cross over the parents to form new offspring (children). If no crossover is performed, offspring is the exact copy of parents.
- [Mutation] With a mutation probability, mutate offspring at each locus (position in chromosome).
- [Accepting] Place new offspring in the new population.
• [Replace] Use new generated population for further sum of the algorithm.
• [Test] If the end condition is satisfied, stop and return the best solution in the current population.
• [Loop] Go to step 3 for fitness evaluation.

**Step 1: Chromosome Representation**
Genetic Algorithm starts with the initial population of solutions represented as chromosomes. A chromosome consists of genes where each gene represents a specific attribute of the solution. In our problem each chromosome of length 2N is taken (Figure 1.25). It is divided into two parts. First N genes corresponds to time \( t_i, i=1,2,...,N \) and last N to Price \( u_i, i=1,2,...,N \).

![Chromosome Representation for two dimensional problem](image)

**Step 2: Initial Population**
GA generates the initial population randomly. It initialize to random values within the limits of each variable. Here, we for the first N variables the limit is with respect to testing time and for the last N variables the limit is with respect to manpower resources.

**Step 3: Fitness of a Chromosome**
The fitness is a measure of the quality of the solution represented in terms of various optimization parameters of the solution. A fit chromosome suggests a better solution. In our profit maximization (Problem 1), the fitness function is the objective of optimization problem along with the penalties of the constraints that are not met.

**Step 4: Selection**
Selection is the process of choosing two parents from the population for crossover. The higher the fitness function, the more chance an individual has to be selected. The selection pressure drives the GA to improve the population fitness over the successive generations. Selection has to be balanced with variation form crossover and mutation. Too strong selection means sub optimal
highly fit individuals, will take over the population, reducing the diversity needed for change and progress. Too weak selection will result in very slow evolution. We use Tournament selection without replacement as selection method for solving Problem 1 using GA.

**Step 5: Crossover**
Crossover is the process of taking two parent solutions and producing two similar chromosomes by swapping sets of genes, hoping that at least one child will have genes that improve its fitness. In our two dimensional problem the first N genes of a chromosome are crossed over with the first N genes of other selected chromosome. The crossover of the last N genes of the same chromosome takes place with the last N genes of other selected same chromosome.

**Step 6: Mutation**
Mutation prevents the algorithm to be trapped in a local minimum. Mutation plays the role of recovering the lost genetic materials as well as for randomly disturbing genetic information. Mutation in our case is done for first N testing time genes and then of the last N genes but on the same selected chromosome.

1.14 **OPTIMIZATION PROBLEMS IN MARKETING & SOFTWARE ENGINEERING**
Optimization models are of immense importance to the marketing managers as well as to software engineers as they deal with many sophisticated decision-making activities. Optimization models are the constructs used to optimize a decision and provide the alternative courses of action. The ultimate goal of all optimization problems is to either maximize benefits and/or minimize efforts. Optimization is not a single method rather different approaches are used to solve different class of optimization problems. The optimum seeking methods are known as mathematical programming techniques and are generally studied as a part of Operational Research. Some of the well-known techniques of optimization are differential calculus, linear programming, non-linear programming, dynamic programming, network methods, game theory, multi objective programming etc. Apart from mathematical programming methods some other well-known techniques are fuzzy systems, evolutionary computation, neural networks, and probabilistic reasoning known as soft computing algorithms. Apart from classifying the optimization problems on the basis of the technique used to solve the problem often people classify them as unconstrained and constrained optimization. In an unconstrained optimization usually some measure of effectiveness are optimized under the boundary conditions whereas in constrained optimization, constraints are imposed on the measure of effectiveness.
In optimization problems values of the decision variables produce the optimal result. The major steps required to perform for solving a typical optimization problem are as follows

- The system-variables interaction must be known accurately and quantitatively.
- A single or multiple measure of effectiveness must be expressed in terms of system variables.
- The choice of these values of the system variables must yield optimum solution.

The term optimization is believed to be coined by the German mathematical philosopher Labnitz in his book “Essay on goodness of God, the freedom of man and the origin of evil” (Leibniz et al, (1985)). However the optimization problems and their solutions have made their footmarks in the literature as old as 1826 due to the work of Fourier (Fourier, (1831)). Now days various optimization models and techniques are used successfully in almost every industry, engineering system etc. Research in this field has grown enormously. Various new techniques and algorithms have been proposed by many scholars in the field of mathematics, operational research and engineering in the literature over the years.

Classical mathematical programming models require well-defined criterion and activity constant coefficients, resource, requirement and structural conditional constants together with well-defined inequalities. The coefficient and conditional constants are calculated based on the system information where it is assumed that the system behavior and environment are deterministic in nature, which is not true in general. In most real life problems it is observed that some (or all in some cases) of the model constants can only be computed roughly due to their dependence on various non-deterministic factors.

Optimizations to marketing managers and software engineers includes optimal scheduling of the development project, cost determination, optimal allocation of development resources to the various stages of PDLC/SDLC, etc. In this thesis we have discussed and solved some profit maximization problems under two dimensional framework and software release time decision problems formulated under set of budgetary constraint. These problems and approach adopted to solve are discussed in details in chapter 3 and 6.

1.15 STRUCTURE OF THE THESIS

The work presented in this thesis focuses on some contributions to modeling and optimization in innovation diffusion and software reliability. The performance of the proposed models is shown
with the parameter estimation and model validations on real life data sets existing in software reliability and marketing literature. The results obtained are encouraging and close to the actual values. Various optimization problems and their solution methodology are presented with numerical examples.

**Chapter 1** is introductory and explains the basic concept of modeling and optimization in marketing and software engineering. The existing research and applicability of the research carried in the thesis in innovation diffusion modeling, software reliability growth modeling, innovation diffusion modeling and optimization is highlighted in this chapter. Rest of the thesis is organized in 5 chapters as follows:

**Chapter 2** is based on studying the extension of Bass Model and relaxing one its assumption of constant market size. Bass Innovation and Diffusion model and many of its extended forms have been reported in marketing literature and applied successfully for depicting and predicting adoption curve for products from different sectors of economy. All these models assume the adoption process as a discrete counting process and that an adopter buys the product only once in his lifetime, however, this is not true, because a consumer may buy the product more than once for his utility (repeat purchasing). Also, there can occur a situation that the consumer leaves the system without buying the product (balking). Therefore, the behavior of adopters plays an important role in calculating actual number of units sold. These two aspects form the crux of the chapter. Furthermore, if the potential adopter population is large and product is in the market with greater life cycle length; it is quite likely that adoption process is a stochastic process with continuous state space. The chapter is divided into two parts where we propose two new innovation diffusion model based on \( \text{Ito} \) type of stochastic differential equation with repeat purchasing and balking. The later part of the chapter also incorporates the change-point concept, where the rate of product adoption per remaining potential adopter might change due shift in marketing/promotional strategy, entry/exit of some of the competitors in the market. The applicability and accuracy of the proposed models is illustrated using new product sales data. Predictive validity and Mean-squared error have been used to check the validity of the model.

**Chapter 3.** Warranty is seller assurance to a buyer that a product will carry out as stated and this promise works as a confidence for the buyer in purchasing the product. Warranties are also an effective means of promoting a product in the market when the company or product is not well known. Or in other words warranty serves as a source for spread of an innovation in the market.
But servicing a warranty engages additional costs to the manufacturer and this has an effect on the profit levels. The present chapter deals in studying this warranty analysis. It is divided into two sections. In **Sec.1**, we formulate an optimization problem that determines the optimal adoption time and sale price of product that taking into consideration the cost of warranty. The factors like fixed cost, production cost and inventory cost also considered in the problem of maximization of the profit. In the proposed optimization problem, a two-dimensional technology diffusion innovation model which combines the adoption time of technological diffusion and price of the technology product has been considered using Cobb-Douglas production function. The profit maximization problem is solved using genetic algorithm. In **Sec.2** A scientific approach to meticulously examine the significance of warranty as a key marketing tool to promote sales of goods by virtue of warranty length optimization, which presumably results into an increase in the overall profit for an organization is proposed. The two models presented in this paper cater to two different scenarios: (1) longer warranty period to increase marketability and in creating goodwill for the manufacturer and sacrificing on the overall profit, (2) shorter warranty period ensuring improved profitability for the manufacturer. Exponential distribution is used to represent the lifetime of a product. Two numerical illustrations demonstrate applications of the models and their comparison.

**Chapter 4** This chapter is divided into two sections. In **Sec.1**, we have proposed an innovation diffusion model that takes into consideration the increased market size because of factors like increase in population, marketing mix effects, cross country effect etc. With the increase in population, purchasing power and promotional efforts, there can occur a change in the adoption rate. Hence, the concept of change point is also incorporated in the later part of the section. Results are illustrated with numerical examples. In **Sec. 2** we have focused on the concern of marketing managers of improving their existing products and coming up with their newer versions quickly in order to sustain themselves. The study in this section develops a mathematical model to find out the time of launching of next generation when the overall cost (development & promotion) is minimized. We have discussed the problem with the help of numerical illustration for a case of two generational products.

**Chapter 5** discusses important management problems related to software consisting of successive releases. This chapter is divided into two sections. In order to incorporate the irregular fluctuations in fault detection process due to various uncertainty factors during testing phase, we
have used stochastic differential equations of \( it^o \) type in the chapter. In Sec. 1 we develop a mathematical model for multiple generations of software to forecast the bug removal pattern in the software system due to multiple add-ons. The model consider that during the testing process of new code (due to enhancement), the faults which were present in the previous release are also removed with new fault detection rate. Thus we can say that when we are in testing phase of new release, we are also in the operational phase for the previous release simultaneously. The proposed model is estimated on real failure data set of the software which is released in the market four times. In Sec. 2 we have formulated a successive software release model with severity of faults. Moreover, we discuss the identification of the faults left in the software when it is in operational phase during the testing of the new code i.e. developed while adding new features to the existing software. We examine the case where there exists two types of faults in the software; simple and hard and during testing the simple faults are removed by exponential rate whereas hard faults are removed by two stage Yamada function. Results are supplemented by a numerical example.

Chapter 6 deals with the optimal software release time decision problem in software engineering. The chapter is again divided in two sections. In Sec. 1 we have first discussed a Successive Software Release model where faults are removed by logistic rate function using the concept of SDE. We have further investigated an optimal bi-criterion release planning for this multi-upgraded software that maximizes the reliability and minimizes the cost of testing of the release under the dual constraints of budget and achieving a desired level of reliability. In Sec. 2 in order to determine optimal release time of a new version of the software, Multi Attribute Utility Theory (MAUT) has been used. This technique attempts to identify relevant objectives for any given decision making problem, where a decision is typified by multiple objectives. It can be difficult to quantitatively compare the objectives like cost, reliability, failure intensity etc. one against another. In order to provide insight into this problem, a utility function is assessed for each of the relevant objectives. This allows for an appropriate multiple-objective utility function that is used to identify trade-offs and compare the various objectives in a consistent manner. Numerical illustrations are given to justify the release time problem.

Conclusion of the work done, scope of future research and a reference list is given at the end of the thesis.
This thesis is based on the contents of the following research papers which have been published/communicated/presented at International Conferences.


A Study of Innovation Adoption & Warranty Analysis in Marketing and Successive Software Releases


Some other publications not included in the thesis


