Software Reliability viz-a-viz Product Diffusion: Modeling Based Interdisciplinary Research Framework

P.K. Kapur¹, Nitin Sachdeva²

¹Amity Center for Interdisciplinary Research, Amity University, Noida, UP, India
pkkapur1@gmail.com

²Department of Operational Research, University of Delhi, Delhi, India
nitin.sach@gmail.com

Abstract: With the advent of information technology in the past few decades, several areas of research have seen unprecedented growth. Amongst those, the field of interdisciplinary research has received unmatched attention from both academicians and practitioners. Interdisciplinary research focuses on some of the most relevant approaches that have been developed for the purpose of analyzing various engineering and business problems through interdisciplinary work. In this talk, we present six contrasting applications of interdisciplinary work primarily in the areas of innovation diffusion in marketing and software reliability assessment. The idea is to present readers with the understanding of how different areas of research can contribute immensely in solving real world problems related to different areas.

Keywords: Interdisciplinary Research, Software Reliability, Innovation Diffusion, Modeling

I. INTRODUCTION

The merging of optimization and information technologies has seen an exponential growth in the last two decades. This is an area that has sparked as much interest in the academic world as in practical settings. The interdisciplinary research, thus, focuses on some of the most relevant approaches that have been developed for the purpose of analyzing various engineering and business problems through interdisciplinary work. Interdisciplinary Research allows us to answer complex questions, address broad issues, explore disciplinary and professional relations, solve problems that are beyond the scope of any one discipline, achieve unity of knowledge. It also serves as a platform for exchange of original and high quality research between varied areas of research.

With this paper, authors have tried to present interdisciplinary research as a medium to capture and present various techniques and methods applied in several engineering and business disciplines to readers in order to suggest the true application of interdisciplinarity: “Interdisciplinarity is a means of solving problems and answering questions that cannot be satisfactorily addressed using single methods or approaches”. Truly the purpose of such type of research is to combine practical knowledge from engineering, science and management disciplines to advance understanding and make constructive suggestions for solving real world problems.

Importance of interdisciplinary research is impeccable and its impact, truly, beyond comprehension. In this paper, we try and highlight how quantitative modeling framework utilized in one area of research with complete justification on its usage provided by researchers, can be of utmost relevance to researchers and practitioners from a completely different discipline. How the idea of capturing the innovators and imitators adoption behavior based on continuous time, in a social system can be remarkably similar to the detection/correction process followed in software testing for modeling software reliability growth models, is one of the several advantages, interdisciplinary research offers to the world. The underlying behavior analyzed and captured through the two quantitative models is that both the innovation diffusion process and the software reliability growth models follow S-Shaped curve w.r.t. time. Similarly, the idea of disadoption of services proposed by [13] fits very well with the vulnerability discovery modeling framework in the operational phase of the software, as described in this paper. The relevance of such framework in today’s time is immeasurable, since time is changing very fast, information that is accessible to today’s generation is far greater than any of our ancestors and to understand or to comprehend such complex phenomenon, the need of interdisciplinary research is unmatched.

In this talk, we start in Section II with discussion on two very interestingly similar mathematical models finding their existence and relevance in completely distinct areas of research, viz: Bass Model [2] and Kapur & Garg [8]. In Section III, we discuss the true application of interdisciplinary work by citing two more similar models yet again applied in two diagonally opposite fields of research. Here we discuss about disadoption based innovation diffusion model for
service adoption [13] and its relevant application in vulnerability discovery modelling with successful and unsuccessful software patching, [7]. Section IV presents yet another contrasting application of interdisciplinary research in the areas of innovation diffusion in marketing and assessing software reliability with the help of a model which truly combines the two areas [1]. Further, a classical approach to adopter categorization proposed by Rogers [15] is discussed along with yet another similar categorization for classifying faults in software, Kapur et al [11] in Section V. Lastly, we present multidimensional modeling framework, as applied in the areas of marketing ([12];[17]) and software testing ([16]; [9] in Section VI. We finally conclude the talk in Section VII with some insights into the future of interdisciplinary research.

II. BASS V/S KAPUR & GARG MODEL

Bass Model (1969) - Bass model [2] is one of the pioneered models in the history of innovation diffusion mapping. Successive increase in the number of adopters caused by both mass communication and interactions between users and prospective users was successfully depicted by this famous model. Some of the other diffusion models, proposed by ([9];[12]; [17]) underlie the Bass model from both external and internal point of view and have been efficient in modeling customer adoption process but still the Bass model supersedes them.

Assuming that there are no repeat buyers and purchase volume per buyer is one unit, the model serves the purpose of forecasting the first-purchase sales of innovations. Bass model divides potential adopters into two distinctive groups. One group is called potential innovators who are only affected by mass media communication when making purchase decision, which Bass terms as external influence. The other group is known as potential imitators on whom word-of-mouth communication, called internal influence, has a major influence. The total number of adopters at a certain time is the sum of potential innovators and potential imitators. Mathematically the Bass model is expressed as follows:

\[
\lambda(t) = \frac{f(t)}{1 - F(t)} = p + q \frac{N(t)}{m}
\]

(1)

<table>
<thead>
<tr>
<th>Notations</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\lambda(t))</td>
<td>Hazard rate that gives the conditional probability of a purchase in a small time interval ((t, t+\Delta t)), if the purchase has not occurred till time ‘t’</td>
</tr>
<tr>
<td>(N(t))</td>
<td>Cumulative number of adopters till time t.</td>
</tr>
<tr>
<td>(M)</td>
<td>Initial market size</td>
</tr>
<tr>
<td>(p) and (q)</td>
<td>Innovation and Imitation coefficients respectively</td>
</tr>
<tr>
<td>(f(t))</td>
<td>Likelihood of purchase at time t</td>
</tr>
</tbody>
</table>

\[
F(t) = \int_0^t f(t) \, dt
\]

cumulative likelihood of purchasing the product at time t

Equation (1) can be rewritten as:

\[
f(t) = \left(1 - F(t)\right)\left(p + qF(t)\right)
\]

(2)

Where, \(F(t) = \frac{N(t)}{m}\) equation 2 now becomes,

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

Using the initial condition \(N(0) = 0\), Bass solved the equation (1) as:

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

(3)

Kapur and Garg (KG) model (1992) - Several software reliability growth models (SRGMs) assume that the fault removal phenomenon describes the failure phenomenon as well. In reality this seems untrue. Removing some faults may or may not lead to failure. However, this may result in some extra effort on the part of testers. Fault, removed as a result of a failure is called a leading fault, whereas faults removed due to the removal of these leading faults are known as dependent faults as these are some extra faults removed from the system.

Kapur and Garg [8] described this phenomenon by proposing the following SRGM based on NHPP. The mean value function of this phenomenon describes the removal process. The proposed model is given as

\[
\frac{d}{dt} m(t) = b(a - m(t)) + c \frac{m(t)}{a}(a - m(t))
\]

(4)

Where, \(m(t)\) is number of faults removed by time \(t\); \(a\) represents total number of faults in the software to be removed eventually; \(b\) is failure occurrence rate; and \(c\) represents fault removal rate of additional removed faults. From (4), with initial condition \(m(0) = 0\) we get the solution as

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

(5)

Here, if \(c = 0\), then (5) reduces to nothing but G-O model. It is interesting to note that the two models developed. Of late KG model is being used for different SRGMs as testing model.
III. DIFFUSION OF SERVICES WITH DISADOPTION VS VDM AT THE USER END

Diffusion and Attrition - Services have traits in common with both durable goods and fast-moving consumer goods. For commercial success, repeat purchases are the only way for service providers, exactly similar to the sellers of FMCG products. For their growth, key lies in appropriate and targeted advertising, promotion, and consumer trials; therefore, the relevance of the classical innovation adoption model proposed by Bass [2] as part of the internal communication process governed by word of mouth and imitation. However, a marked difference in services is the outward flow the customer, as customer attrition, wherein a customer terminated the relationship with the service provider. Here in this section, we present attrition or disadoption based model for services that incorporates customer disadoption and analyze how customer disadoption rate affects the growth of new service adoption in the market. Let \( N(t) \) be the users or subscribers, \( m \) the market potential, \( p \) and \( q \) the external and internal parameters, and \( \delta \) the disadoption rate. Thus, the diffusion of the new service is given by the following equation:

\[
\frac{dN(t)}{dt} = p \left( m - N(t) \right) + \frac{q(1 - \delta)N(t)}{m} \left( m - N(t) \right) - \delta N(t)
\]

(6)

Here, we assume that only those adopters who did not disadopt would spread the positive word-of-mouth amongst the remaining market potential. Thus, the degree of the word-of-mouth promotion by retained customers is the same \( q \), but the effective word-of-mouth impact is reduced as a result of the disadoption rate \( \delta \) from \( \frac{qN(t)}{m} \) to \( \frac{q(1 - \delta)N(t)}{m} \). Because disadopters return to the market potential, the remaining market potential is \( m - N(t) \) and is not affected by the disadoption. Equation (6) is a first-order quadratic differential equation. Using the initial condition \( N(0) = 0 \), we can integrate Equation (6) to arrive at the following solution:

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

(7)

\[- \frac{m}{1 - e^{-\frac{(p+q)}{q}}\Delta t} \]

\[- \frac{m}{1 + e^{-\frac{(p+q)}{q}}\Delta t} \]

\[- \frac{m}{1 - e^{-\frac{(p+q)}{q}}\Delta t} \]

The market penetration curve (Equation 2) has the same functional form as the [2] equation but with different parameters: \( p, q, \) and \( m \) instead of \( p, q, \) and \( m \) respectively. When \( \delta = 0 \), Equation 2 converges to Bass diffusion function.

Vulnerability Removing Model with Successful & Unsuccessful Patch Rate - In software development life cycle, coding and testing is a process which requires developers and tester’s scrutiny. While testing, any carelessness in covering all the paths or statements of software code results in a loop hole or vulnerability. The loop hole unknowingly provides a back gate to the intruder to infringe the software system. These software vulnerabilities are the major cause of security breach such as loss of authenticity, confidentiality, and integrity. Generally, a major amount of these vulnerabilities are discovered by the external users. The external users comprises of multiple users which are categorized as white hat users and black hat users. If vulnerability is found by the white hat users, they will report it to the associated technical team, however, if found by the black hat users, then they tend to use it for personal gain. Vulnerability existence disclosure to black hat user or hacker increases the intrusion rate. Hence the management pressurizes the developers to create and distribute software patch before vulnerability is disclosed. These software patches are a piece of code with the properties capable of fixing the software vulnerability.

Kansal et al [7] proposed a vulnerability patching model to evaluate the amount of successful patch created by the developers. The idea is to help vendors predict the loss incurred to them due to the unsuccessful patches. The proposed model assesses the patched vulnerabilities when the developer is performing the debugging process. The model frames three factors, 1) direct vulnerabilities with successful patch rate, 2) indirect vulnerabilities with successful patch rate, and 3) indirect vulnerabilities with unsuccessful patch rate. The first factor includes vulnerabilities which are primarily discovered by developers and are successfully removed by them. The second factor states that while patching the reported vulnerabilities (vulnerability reported or detected by user) developer has discovered more vulnerability and decides to remove it as well with a successful patch. The third factor includes those vulnerabilities whose patch is unable to remove the vulnerability completely. The successful patch here means those installed patches which succeed in removing the vulnerabilities.

Assumptions:

1. The vulnerability and its associated patches are independent.
2. The patches are already installed by all users.
3. At any given point t, some of the vulnerabilities would always exist that are unsuccessfully patched.

Notations:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Initial number of Patches available</td>
</tr>
<tr>
<td>A</td>
<td>Vulnerability Removal rate</td>
</tr>
<tr>
<td>C</td>
<td>Constant</td>
</tr>
<tr>
<td>δ</td>
<td>Unsuccessful patch rate</td>
</tr>
<tr>
<td>( \hat{\rho} )</td>
<td>Expected number of patches</td>
</tr>
<tr>
<td>( dp/dt )</td>
<td>Rate of change of patched vulnerabilities with respect to time</td>
</tr>
<tr>
<td>( \Delta, \bar{B}, \bar{C} )</td>
<td>Intermediate Variables</td>
</tr>
</tbody>
</table>
\( \dot{\rho}(t) \) Estimated Patch data
\( \rho_i \) Actual Patch data
\( k \) Number of Data Points
\( n \) Number of Parameters

1) Model Development - The proposed model quantitatively measures the success rate of released patch during the operational phase. As vulnerabilities are only observed after release of the software by users, we attrite it via patches in operational phase. According to previous discussion model utilizes three terms with successful and unsuccessful patch rate.

2) Direct Vulnerabilities Patching - The fact that software is released after clearing various testing stages, it still contain some vulnerability which are observed later on. The job of discovering vulnerabilities is not only of external user whereas the internal users are also involved in the process. In most of the organizations software screening is an ongoing process even after its release until the software outages. Thus, there is some vulnerability which is discovered by tester independently and is disclosed later with the release of patch. The direct vulnerability does not cost much to the developer as the exploitability score is near to zero of it. As the vulnerability existence is unknown to the external world the developers take advantage of tester’s discovery and gets an enough complimentary resources to create the patch and thus assume to release a patch which works well. The successfully patched direct vulnerability increases with time and thus represented as:

\[ A \cdot (B - \dot{\rho}) \]  

(8)

3) Indirect Vulnerabilities Patching - Vulnerability Patching helps developers to patch more vulnerability. The vulnerabilities which are discovered by users are reported to developers to fix it. While screening the code for debugging purpose, developer has observed some additional vulnerability which was not reported and decides to remove it as well with successful patch rate. Hence, the reported vulnerabilities influence other vulnerabilities to detect. The indirect detected vulnerability is also not known to the external world whose positive effect will make developer to create a patch without any flaw. Therefore, the rate at which these vulnerabilities are detected would be lowered by the extent to which patching is successful. The successfully patched indirect vulnerabilities increases with time and thus represented as:

\[ C \cdot (1-\delta) \cdot \dot{\rho} \cdot (B - \dot{\rho}) \]  

(9)

4) Unsuccessful Patched Vulnerabilities - Hundreds of patches releases every year out of which some proportion of patches fails in removing the vulnerabilities. These unsuccessful patches lead to a great loss and even increase the damage potential. As discussed earlier the reason behind the patch failure is the lack of resource provided by the management to the developers. When no time situation arises where developers have to create the patch as soon as vulnerability is reported, the probability of unsuccessful patch rate increases. The unsuccessful patch rate proves the incompetency of the developer’s in removing all the vulnerabilities. The unsuccessful patched vulnerabilities decrease with time and thus is represented as:

\[ -\delta \cdot \dot{\rho} \]  

(10)

By equation (1), (2) and (3), the rate of change of patched vulnerability with respect to time is represented as:

\[ \frac{d\dot{\rho}}{dt} = A \cdot (B - \dot{\rho}) + C \cdot (1-\delta) \cdot \dot{\rho} \cdot (B - \dot{\rho}) - \delta \cdot \dot{\rho} \]  

(11)

After solving the above equation we get,

\[ \dot{\rho} = \frac{\overline{B}}{1 + \frac{\overline{C}}{\overline{A}}} \left[ 1 - e^{-\left(\frac{\Delta + \beta}{2} + C \cdot (1-\delta)\right) \cdot \frac{\overline{B}}{2}} \right] \]  

(12)

Where \( \overline{B} = B \cdot \frac{\Delta + \beta}{2} \cdot C \cdot (1-\delta), \overline{A} = \frac{\Delta - \beta}{2} \cdot \overline{C} = \frac{\Delta + \beta}{2} \), \( \beta = C \cdot (1-\delta) - A - \delta \) and \( \Delta = \sqrt{\beta^2 + 4 \cdot C \cdot (1-\delta) \cdot A} \).

IV. INTERDISCIPLINARY MODELING IN SOFTWARE RELIABILITY

Several quantitative measures of growth in reliability of software during the testing phase have been proposed in the literature. Many of these can be classified under the title of Non-Homogenous Poisson Process (NHPP) models. As mean value function, \( m(t) \) forms the building block of all the NHPP models existing in the software reliability engineering literature, the models proposed by ([18]; [19]), the effect of intensity of testing effort on the failure phenomenon has been studied. The mean value function for such a situation can be developed based on the following differential equation, [1]

\[ \frac{dm}{dt} = \left( \frac{dm}{dW} \right) \left( \frac{dW}{dt} \right) \]  

(13)

According to the above mentioned equation, the fault identification is dependent upon the testing effort that includes manpower and computer time. It is assumed that the consumption of testing resources follow a definite pattern with respect to time and exponential, Rayleigh and Weibull curves have been used to describe it. Several authors have extended such models to count the number of faults identified in operational phase. Though the dependence of fault exposure on
usage rate cannot be disputed, extending the model to operational phase can be contested, as the intensity of use of commercial software at the user end doesn't follow any particular explicit time dependent pattern. This rate at which the software is executed is dependent upon the number of users of software. Hence, models for operational phase should contain the effect of growth in number of users of the software over time. Therefore the last equation can be modified as follows:

$$\frac{dm}{dt} = \left(\frac{dm}{dX}\right) \left(\frac{dX}{dS}\right) \left(\frac{dS}{dt}\right)$$

(14)

In the above expression, m, X, and S are differentiable functions of time but are explicitly dependent upon X, S and t respectively. The proposed model gives functional form to each term as follows:

$$\frac{dS}{dt} = \left(\alpha + \beta \frac{S}{S} \left(\frac{S}{S}\right) - X \right)$$

$$\frac{dS}{dt} = \alpha \left(\frac{S}{S} - X \right) + \beta \left(\frac{S}{S} - X \right)$$

$$\frac{dm}{dX} = k_1 \left(\frac{p + q m}{a}\right)\left(a - m\right)$$

$$\frac{dX}{dS} = k_2$$

The solution at $S(t = 0)$ is

$$S = S \left(\frac{1 - e^{-\left(\alpha + \beta\right)t}}{1 + \frac{\beta e^{-\left(\alpha + \beta\right)t}}{\alpha}}\right)$$

$$dS = \frac{\beta \left(1 + \frac{\beta e^{-\left(\alpha + \beta\right)t}}{\alpha}\right)}{\left(1 + \frac{\beta e^{-\left(\alpha + \beta\right)t}}{\alpha}\right)^2}$$

Using the three equations mentioned above, in Equation (14), we get

$$\frac{dm}{dt} = \eta \left(\frac{p + q m}{a}\right)\left(a - m\right) \left(\frac{e^{-\left(\alpha + \beta\right)t}}{1 + \frac{\beta e^{-\left(\alpha + \beta\right)t}}{\alpha}}\right)^2$$

(15)

V. CATEGORIZATION OF ADOPTERS V/S CATEGORIZATION OF FAULTS

Adopter Categorization - Tarde appealed that social systems are based of psychological interactions among people, especially imitation and innovation. Rogers [15] classic book, _The Diffusion of Innovations_, extensively studied this process and proposed that adoption distribution of any innovation follows normal distribution over time. He suggested that during the early stage of innovation adoption, the rate of growth is relatively slower as compared to the time till the new product stabilizes itself. Later on, the product growth increases rapidly with more customers beginning to demand the product. When the product reaches near to the end of its life cycle, growth slows down and sometime begins to decline too. Rogers, on the basis of Normal Curve, categorized the adopters of any innovation as shown here: Innovators: 2.5%; Early Adopters: 13.5%; Early majority: 34%; Late majority 34%; Laggards 16%

Mahajan & Muller [14] proposed the adopter categorization using the famous Bass model and suggested the categorization as presented in Table 1 here:

<table>
<thead>
<tr>
<th>Adopter Category</th>
<th>Time Interval Covered</th>
<th>Expression for the Time Interval</th>
<th>Expression for the adopter category size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Innovators</td>
<td>Initiation of the diffusion process</td>
<td>$P$</td>
<td></td>
</tr>
<tr>
<td>Early Adopters</td>
<td>Up to $T_1$</td>
<td>$\frac{1}{p + q} \ln \left(\frac{2 + \sqrt{3} p}{q}\right) - \frac{1}{2} \left(1 - \frac{p}{q}\right) - \frac{1}{\sqrt{12}} \left(1 + \frac{p}{q}\right)^{-p}$</td>
<td></td>
</tr>
<tr>
<td>Early Majority</td>
<td>$T_1$ to $T_*$</td>
<td>$\frac{1}{p + q} \ln \left(\frac{2 + \sqrt{3}}{q}\right)$</td>
<td></td>
</tr>
<tr>
<td>Late Majority</td>
<td>$T_*$ to $T_2$</td>
<td>$\frac{1}{p + q} \ln \left(\frac{2 + \sqrt{3}}{q}\right) - \frac{1}{\sqrt{12}} \left(1 + \frac{p}{q}\right)^{-p}$</td>
<td></td>
</tr>
<tr>
<td>Laggards</td>
<td>Beyond $T_2$</td>
<td>$\frac{1}{2} \left(1 + \frac{p}{q}\right) - \frac{1}{\sqrt{12}} \left(1 + \frac{p}{q}\right)^{-p}$</td>
<td></td>
</tr>
</tbody>
</table>

Categorization of Faults - Faults can be categorized according to their severity in terms of impact, complexity in removal or the relative ease with which they are identified while testing. Here the last case is exclusively taken up. It is quite interesting to note that the Roger’s idea of adopter categorization based on adoption time is quite intuitively utilized here. In software reliability area, faults are now categorized based on their detection time. During testing, while checking the codes for possible failure cause, newer unaddressed faults get detected.
These faults can provide insightful details regarding the type of faults detected and remaining in the system. In real testing scenario, large number of simple (trivial) fault gets easily detected at an early stage of testing with fault removal process becoming increasingly difficult at the later stages. Thereby the value being different for detection of these faults based on whether they are detected early or late.

Based on [8] model, the instantaneous fault detection function at any instant \( t' \), is given by the derivative of Equation (5) as:

\[
m'(t) = \frac{d}{dt} m(t) = \frac{b(b+c)}{b+c} e^{-(b+c)t} \frac{1}{b+c} \]

The curve for \( m'(t) \) is depicted in Figure-1. The peak of the curve is at \( T^* \), where

\[
t^* = \frac{-1}{(b+c) \ln(b/c)}
\]

![Fig. 1. Noncumulative failure curve](image)

The trends for the SRGM based on equation (16) are summarized in Table 2 [11]

**TABLE 2: CATEGORIZATION OF FAULTS**

<table>
<thead>
<tr>
<th>Time interval</th>
<th>Trend in ( m'(t) )</th>
<th>Type of Faults</th>
<th>Number of faults in each category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero to ( T_1 )</td>
<td>Increasing at an increasing rate</td>
<td>Easy Faults</td>
<td>( m(T_1) )</td>
</tr>
<tr>
<td>( T_1 ) to ( T^* )</td>
<td>Increasing at a decreasing rate</td>
<td>Difficult Faults</td>
<td>( m(T^*) - m(T_1) )</td>
</tr>
<tr>
<td>( T^* ) to ( T_2 )</td>
<td>Decreasing at an increasing rate</td>
<td>Hard Faults</td>
<td>( m(T_2) - m(T^*) )</td>
</tr>
<tr>
<td>Beyond ( T_2 )</td>
<td>Decreasing at an increasing rate</td>
<td>Complex Faults</td>
<td>( a - m(T_2) )</td>
</tr>
</tbody>
</table>

The points of inflection \( T_1 \) and \( T_2 \) can be found by taking the second derivative of \( m'(t) \) given in equation (11) with respect to time and equating to zero as:

\[
T_1 = -\frac{1}{b+c} \ln \left( \frac{2 + \sqrt{3}}{c} \right) \quad \text{and} \quad T_2 = -\frac{1}{b+c} \ln \left( \frac{2 - \sqrt{3}}{c} \right)
\]

Using \( T_1 \) and \( T_2 \) the number of faults of each category present in the software can be calculated.

**VI. TWO DIMENSIONAL MODELING USING – COBB DOUGLAS FUNCTION**

Modeling Innovation Diffusion Process in Marketing - To incorporate the effect of two factors in the consumer decision making process, [12] proposed a two dimension innovation diffusion model based on the famous Cobb-Douglas [4] production function. Using the Cobb-Douglas functional form, value of the product can be given as:

\[
x(t, P) = \alpha \cdot P^{\alpha - \alpha}\quad (17)
\]

where, \( x \) is the value of the product, \( t \) is the continuation time of the product in the market, \( P \) is the price of the product, \( \alpha \) is the output elasticity which represents the effect of product valuation to the innovation diffusion model.

1. Two-Dimensional Demand Model [12] - Similar to the Bass model, the two dimensional model also assumes that there are two major sources of adopters: innovators and imitators and rate of adoption is dependent upon the behavior of these two groups. Let us say, \( N(\psi) \) represents the cumulative number of adopters through \( \psi \) value of the product; \( n(\psi) \) as the number of adopters at \( \psi \) value of the product; \( m \) is the market potential; \( \rho \) & \( q \) be the innovation and imitation coefficients respectively; while, \( P \) is the price of \( N \) products, when \( \psi \) amount of value is generated by a product. Assuming \( \frac{N(\psi)}{m} \) as the adoption distribution function, the mean value function can be written as:

\[
\overline{N}(x) = E[N(x)] = mF(x). 
\]

The related demand model is given as:

\[
\frac{dN(\psi)}{d\psi} = \rho \left( m - \overline{N}(\psi) \right) + \frac{q \overline{N}(\psi)}{m} \left( m - \overline{N}(\psi) \right)
\]

\[
= \left( \rho + \frac{q \overline{N}(\psi)}{m} \right) \left( m - \overline{N}(\psi) \right) 
\]

(18)

Now, with the initial condition of \( \psi(0,0) = 0 \Rightarrow N(0,0) = 0 \), we have
Therefore (22) can be rewritten as follows:

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

(19)

Modeling SRGMs in Assessing Software Reliability- Similar to two dimensional modeling used to model innovation adoption process in marketing, ([6]; [9]), proposed two dimensional models to measure software reliability. Quite interestingly, the same Cobb-Douglas functional form of production functions is used wherein testing time, testing coverage or testing efforts are used as input parameters which affect the overall testing resource output. Here we present a two dimensional software reliability growth model to measure number of faults removed based on the testing resources \( \tau \) is presented. We define the testing resources as follows:

\[
\tau \geq s^\alpha u^{1-\alpha} \quad 0 \leq \alpha \leq 1
\]  

(20)

Where, \( \tau \) : testing resources, \( S \) : testing time, \( U \) : testing effort, \( \alpha \) : Output elasticity of testing time. Let us assume that \( \{N(s, u), s \geq 0, u \geq 0\} \) be a two-dimensional stochastic process which represents cumulative number of software failures by time \( s \) and testing effort \( u \). The mean value function \( m(s, u) \) for this two dimensional NHPP model is given as:

\[
Pr(N(s, u) = n) = \frac{(m(s, u))^n}{n!} \exp(-m(s, u)), \quad n = 0, 1, 2, \ldots
\]  

(21)

The mean value function with respect to testing resources is given as follows:

\[
m(\tau) = aF(\tau)
\]  

(22)

As discussed, \( \tau \) is a two dimensional variable, with testing time and testing effort as its dimensions. These two dimensional models are useful as they have the ability to show the simultaneous effect of two variables on dependent variable. Therefore (22) can be rewritten as follows:

\[
m(\tau) = a F(s^\alpha u^{1-\alpha}) \quad 0 \leq \alpha \leq 1
\]  

(23)

Very recently, Sachdeva et al. [17] extended the study of two dimensions to model innovation diffusion process to a three dimensional model. The three dimensional model includes time, price and marketing effort as three parameters to model the innovation diffusion process. It is quite interesting to note that a similar study in the area of software reliability can be conducted provided the data for the three dimensions is available.

**VII. CONCLUSION AND FUTURE DIRECTIONS**

Importance of interdisciplinary research is impeccable and its impact, truly, beyond comprehension. In this paper, we try and highlight how quantitative modeling framework utilized in one area of research with complete justification on its usage provided by researchers, can be of utmost relevance to researchers and practitioners from a completely different discipline. How the idea of capturing the innovators and imitators adoption behavior based on continuous time, in a social system can be remarkably similar to the detection/correction process followed in software testing for modeling software reliability growth models, is one of the several advantages, interdisciplinary research offers to the world. The underlying behavior analyzed and captured through the two quantitative models is that both the innovation diffusion process and the software reliability growth models follow S-Shaped curve w.r.t. time. Similarly, the idea of disadoption of services proposed by [13] fits very well with the vulnerability discovery modeling framework in the operational phase of the software, as described in this paper. The relevance of such framework in today's time is immeasurable, since time is changing very fast, information that is accessible to today's generation is far greater than any of our ancestors and to understand or to comprehend such complex phenomenon, the need of interdisciplinary research is unmatched.

**ACKNOWLEDGEMENT**

Authors express their deep sense of gratitude to the founder president of Amity Universe, Dr. Ashok K. Chauhan for his keen interest in promoting research in the Amity Universe and for always inspiring us for achieving greater heights.

**REFERENCES**


