On Warranty Cost Analysis For a Software Reliability Model Via Phase Type Distribution

Y. Sarada and R. Shenbagam

•

CEGC, Anna University, Chennai, Tamil Nadu, India – 600 025
sarada@ annauniv.edu, senbagam@ymail.com

Abstract

This research work investigates an optimal software release problem via phase type distribution, warranty and risk cost analysis. The inter arrival time of software failure is assumed to be a phase type distribution. The PH-SRM is one of the most flexible models, which overarches the existing non-homogeneous Poisson process (NHPP) models, and can approximate any type of NHPP-based models with high accuracy. Based on the phase type Non- homogeneous Poisson Process (PHNHPP) formulation and using the renewal reward theorem, the long run average cost rate is obtained. As model parameter estimation is an important issue in developing software reliability models, the software failure parameter has been estimated by the moment matching method. Finally, a numerical example is provided to illustrate the theoretical results therein.

Keywords: phase type distribution, software reliability, method of moments, renewal reward theorem, optimal software release policy

I. Introduction

The most significant feature of commercial software is Reliability, since it quantifies software failures during the growth procedure. Software is examined from an assortment of judgment in the testing procedure. For example, functionality, reliability, usability and the defects located in the procedure should be defined before it is released to the society. Decreasing the development cost and improving the quality of software are notable facts in software testing management. Software managers are confronted with many complicated problems in software testing. The plan for software maintenance (patch schedule) and the decision for the software release (Okumoto and Goel (1980), Pham (1996)) are based on reliability. Therefore, software reliability is an excellent tool to estimate the number of bugs (Musa *et al*. (1987), Musa (1999), Pham (2000)). Previous efforts on software reliability models (SRMs) mainly focused on NHPPs owing to their mathematical tractability and varied modeling situations such as imperfect debugging, change points, testing efforts and fault detection process (Xie *et al*. (2007), Okamura *et al*. (2013)). For example, the debugging process is modelled as a counting process which follows Poisson distribution with a time dependent hazard rate function (Goel and Okumoto (1979); Littlewood (1981), Yamada *et al*. (1983), Goel (1985), Langberg and Singpurwalla (1985), Laprie *et al*. (1991), Gokhale and Trivedi (1998)).

One of the major issues addressed in NHPP based software reliability is that of determining the best model whose solution lies in the statistical methods encompassing fitting of the observed bug data and to decide the model parameters (Langberg and Singpurwalla (1985)). Working in this direction, Okamura and Dohi (2006, 2008) introduced the phase type software reliability model (PHSRM) wherein the fault detection time follows a phase type distribution with the underlying counting process following NHPP.

A warranty is an agreement between a buyer and a seller at the time of product sale. It is a detailed study of the reimbursement type for a given product at the time of occurrence of failures. Also, it plays a significant role to safeguard the customer's interest particularly in the case of complicated products such as automobiles or electric devices. Recently attention has been directed towards warranty policies and warranty cost modelling (Nguyen and Murthy (1984), Blischke and Murthy(1995)). In today's market, many goods like mobile phones, electronic items and home devices are sold with extended warranty policy, in which a few choices are available for the consumer at the expiry time of the free warranty period. Extended warranties present extra security in the event of expensive failures after the initial warranty period and thereby safeguard the buyer against inflation. Also, extended warranty has attracted significant attention among practitioners. Lam and Lam (2001) proposed an extended warranty model with options open to customers to obtain an optimal policy for the consumers.

Furthermore, in the software management scheme the most significant calculation is to find an optimum software release time, referred to as an optimal software release problem. An optimal release problem with warranty cost and reliability requirement was studied by Yamada (1994). Also, Jain and Handa (2001) developed a software reliability model by employing a Hybrid warranty policy. Zhang and Pham (1998) studied a software cost model under warranty with a risk cost due to software failure and a cost to remove each error detected in the software. Prince Williams (2007) derived optimal release time policies to predict the optimal release time of software using imperfect debugging phenomena and warranty. The optimal release problem with simulated cost and reliability requirements was further implemented by Okumoto *et al*. (2013).

From a thorough review of the existing literature done, a combination of Phase type distribution and warranty (fixed / extended) has not been employed anywhere in the literature, in the analysis of software reliability. Motivated by this and in order to fill the gap in the literature, following Okamura and Dohi (2006,2008), the two new features attempted in this research article are the parameter estimation through moment matching method and the long run average cost rate analysis under the combination of phase type distribution and extended warranty in the context of software reliability models.

The structure of the paper is organized as follows: Section II furnishes the basics for the related work. Section III gives a detailed problem description and model assumptions. An explicit expression for the long run average cost rate is obtained in Section IV, while parameter estimation is discussed in Section V. Further, Section VI provides the numerical illustrations. Finally Section VII presents the concluding remarks.

II. Basics

Software reliability model based on Non-Homogeneous Poisson process:

The NHPP modelling in the SRMs essentially treats a counting process of software failures / faults / bugs in software system testing. It is virtually similar to a functioning profile of released software (Musa 1999), which provides information regarding the number of software failures in the system testing vis-à-vis the software reliability in the working phase. Further, existing NHPP-based SRMs are categorized into finite and infinite models. In the finite models, the detection slowly reduces with testing time and ultimately becomes zero, while in the infinite model, it does not become zero, that is, the number of software faults infinitely increases with time.

Specifically, if M(t) represents the number of software faults by time t with F(t) as the cumulative distribution function(c.d.f) of the detection times of software faults while the random variable N is the total number of software faults with mean m, then the probability mass function (p.m.f) of M(t) is given by

$$P\{M(t)=k\}=P_k(t)=\frac{(m\,F(t))^k}{k!}e^{-m\,F(t)},k=0,1,2,... \tag{1}$$

(Refer Langberg and Singpurwalla (1985)).

2.1 PH-SRM

Okamura and Dohi (2006, 2008) introduced phase type distribution in SRMs in which the fault detection time is a PH distribution in the NHPP- based model, referred to as PH-SRM.

A PH distribution is defined as the time to absorption in a continuous-time Markov chain (CTMC) with one absorbing state. Let Q denote the infinitesimal generator matrix of the CTMC with one absorbing state. Without loss of generality, Q is assumed to be partitioned as follows:

$$Q=\begin{pmatrix} U & U^0 \\ 0 & 0 \end{pmatrix}$$

where U and U^0 correspond to transition rates of transient states and exit rates from transient states to the absorbing state, respectively and a probability vector (α,α_{m+1}) exists such that $F_{PH}(x)=1-\alpha.\exp(U.x).e$ for x≥0. Here e denotes a column vector of ones with an appropriate dimension (Neuts (1981)). Note that the exit vector U^0 is given by $U^0=-U$e. The transient states of PH distribution are often called phases. PH distribution is proved to be dense, so that it can approximate any probability distribution with any precision as the size of U (the number of phases) increases (Asmussen and Koole (1993)).

By substituting the c.d.f. of PH distribution into (1), the p.m.f. of PH-SRM is given by

$$P\{M(t)=k\}=P_k(t)=\frac{(m\,F_{PH}(t))^k}{k!}e^{-m\,F_{PH}(t)},k=0,1,2,... \tag{2}$$

Let

$$\Lambda(t)=mF_{PH}(t)=\left(1-\pi\exp(Ut)e\right)m \tag{3}$$

so that

$$P\{M(t)=k\}=P_k(t)=\frac{(\Lambda(t))^k}{k!}e^{-\Lambda(t)},k=0,1,2,... \tag{4}$$

Thus PH-SRM exactly comprises of NHPP-based SRMs whose fault detection time distributions are exponential (Goel & Okumoto 1979), *k*-stage Erlang (Yamada et al. 1983; Khoshgoftaar 1988; Zhao & Xie 1996), hyperexponential (Laprie et al. 1991) and hypoexponential distributions (Fujiwara & Yamada 2001).

III. Model assumptions

We make the following assumptions about software reliability model for a phase type distribution in the context of warranty (fixed and an extended) modelling.

i. C_1: The set up cost of software development process is a constant.
ii. C_2: The cost to remove errors during debugging period is proportional to the total time of removing all errors detected during this period.
iii. C_3: The cost to remove errors during fixed warranty period (T_W) is proportional to the total time of removing all errors detected in the time interval [T, T+T_W].
iv. C_4: The cost to remove errors during extended warranty period (T_E) is proportional to the total time of removing all errors detected in the time interval [T+T_W, T+ T_W+T_E].
v. C_5: Risk cost due to the software failure after its release.
vi. It takes random time to remove errors and hence it is assumed that the time to remove each error follows a phase type distribution.

IV. Cost analysis

No efforts were made previously to carry out the cost analysis under warranty (fixed / extended) and phase type modeling in SRMs. Thus the goal of cost analysis is to obtain the long-run average cost rate for the proposed warranty model. By applying the standard result based on the renewal reward theorem, the long-run average cost per unit time is given by, (Ross, 1996):

$$\text{C(T)} = \frac{\text{Expected cost incurred in a cycle}}{\text{Expected length of a cycle}} = \frac{E(C)}{E(L)} \tag{5}$$

Here, the expected length of the cycle E(L) is given by,

$$E(L) = T + T_W + T_E \tag{6}$$

Next, the expected cost in a cycle E(C) can be expressed as,

$$E(C) = E(C_1) + E(C_2) + E(C_3) + E(C_4) + E(C_5) \tag{7}$$

In what follows, the calculations pertaining to the cost analysis are presented.

- $E(C_2)$:

$$y_i = y_0 + (i-1)Y, \quad \text{i} = 1,2,3,....$$

Let y_i be the cost of fixing i^{th} software fault. It consists of a deterministic part y_0 and an incremental random part (i-1) Y, where Y is a phase type random variable with mean $\mu_Y = -\chi S^{-1} e_2$. Note that the cost of fixing a fault is increasing as the number of faults removed is increasing. This is reasonable because it may become difficult to identify and fix a fault that occurs in the later testing phases.

Let N(T) denote the number of detected errors removed by time T, so that:

$$E(C_2) = C_2 \left\{ E\left[\sum_{i=1}^{N(T)} \left(y_0 + (i-1)Y \right) \right] \right\}$$

$$= C_2 \left\{ \sum_{n=1}^{\infty} E\left[\sum_{i=1}^{N(T)} \left(y_0 + (i-1)Y \mid N(T) = n \right) P[N(T) = n] \right] \right\}$$

$$= C_2 \left\{ \frac{1}{2} \left\{ \sum_{n=1}^{\infty} n[2y_0 + \mu_Y] P_n(T) + \sum_{n=1}^{\infty} n^2 \mu_Y P_n(T) \right\} \right\}$$

Employing (3) we have,

$$E(C_2) = C_2\left\{\frac{1}{2}\left[2y_0\Lambda(T) + \mu_Y\left(\Lambda(t)\right)^2\right]\right\} \tag{8}$$

- $E(C_3)$:

$$E(C_3) = C_3\left\{E\left[\sum_{N(T)}^{N(T+T_W)} W_i\right]\right\}.$$

Using $E\left[\sum_{N(T)}^{N(T+T_W)} W_i\right] = \mu_W\left\{\Lambda(T+T_W) - \Lambda(T)\right\}$, we have,

$$E(C_3) = C_3\left\{\mu_W\left[\Lambda(T+T_W) - \Lambda(T)\right]\right\} \tag{9}$$

- $E(C_4)$:

$$E(C_4) = C_4\left\{\mu_E\left[\Lambda(T+T_W+T_E) - \Lambda(T+T_W)\right]\right\} \tag{10}$$

- $E(C_5)$:

$$E(C_5) = C_5\left\{1 - R(x/T)\right\}$$

Here R(x/T) is the software reliability expressed as:

$$R(x/T) = e^{-[\Lambda(T+x)-\Lambda(T)]} = e^{-m[\alpha e^{UT}e_1 - \alpha e^{U(T+x)}e_1]},$$

so that

$$E(C_5) = C_5\left\{1 - e^{-m[\alpha e^{UT}e_1 - \alpha e^{U(T+x)}e_1]}\right\} \tag{11}$$

Now, employing (8) – (11), the expected total software cost E[C] can be expressed as in (12).

$$\begin{aligned} E(C) = {} & C_1 + C_2\left[\frac{1}{2}\left[2y_0\Lambda(T) + \mu_Y\left(\Lambda(t)\right)^2\right]\right] + C_3\left[\mu_W\left(\Lambda(T+T_W) - \Lambda(T)\right]\right. \\ & + \mathrm{C}_4\left[\mu_E\left(\Lambda(T+T_W+T_E\right) - \Lambda(T+T_W)\right] \\ & + C_5\left[1 - e^{-m\left[\alpha e^{UT}e_1 - \alpha e^{U(T+x)}e_1\right]}\right] \end{aligned} \tag{12}$$

The optimal T* can be determined from (5), by using numerical or analytical methods.

Optimization:

This section is devoted to obtain the optimal software release time T* using the concept of pseudo-convexity of a function.

Lemma:

Using the convexity property of C(T), the optimal software release time T* is determined by solving the following equation.

$$\left\{[T+T_W+T_E]\left[\begin{array}{l} C_2\left(y_0\Lambda'(T)+\mu_Y\Lambda(T)\Lambda'(T)\right)+C_3\mu_W\left(\Lambda'(T+T_W)-\Lambda'(T)\right) \\ +C_4\left[\mu_E\left(\Lambda'(T+T_W+T_E)-\Lambda'(T+T_W)\right)\right] \\ +C_5m\left(\left(\alpha e^{UT}Ue_1-\alpha e^{U(T+x)}Ue_1\right)e^{-m[\alpha e^{UT}e_1-\alpha e^{U(T+x)}e_1]}\right) \end{array}\right]\right\}- \tag{13}$$

$$\left\{\begin{array}{l} C_2\left[\frac{1}{2}\left[2y_0\Lambda(T)+\mu_Y(\Lambda(t))^2\right]\right]+C_3\left[\mu_W\left(\Lambda(T+T_W)-\Lambda(T)\right]\right. \\ +C_4\left[\mu_E\left(\Lambda(T+T_W+T_E)-\Lambda(T+T_W)\right]\right. \\ +C_5\left[1-e^{-m\left[\alpha e^{UT}e_1-\alpha e^{U(T+x)}e_1\right]}\right] \end{array}\right\}=C_1+C_5$$

Proof

Consider the derivative of C(T) with respect to T as given below:

$$\left\{[T+T_W+T_E]\left[\begin{array}{l} C_2\left(y_0\Lambda'(T)+\mu_Y\Lambda(T)\Lambda'(T)\right)+C_3\mu_W\left(\Lambda'(T+T_W)-\Lambda'(T)\right) \\ +C_4\left[\mu_E\left(\Lambda'(T+T_W+T_E)-\Lambda'(T+T_W)\right)\right] \\ +C_5m\left(\left(\alpha e^{UT}Ue_1-\alpha e^{U(T+x)}Ue_1\right)e^{-m[\alpha e^{UT}e_1-\alpha e^{U(T+x)}e_1]}\right) \end{array}\right]\right\}-$$

$$\left\{\begin{array}{l} C_2\left[\frac{1}{2}\left[2y_0\Lambda(T)+\mu_Y(\Lambda(t))^2\right]\right]+C_3\left[\mu_W\left(\Lambda(T+T_W)-\Lambda(T)\right]\right. \\ +C_4\left[\mu_E\left(\Lambda(T+T_W+T_E)-\Lambda(T+T_W)\right]\right. \\ +C_5\left[1-e^{-m\left[\alpha e^{UT}e_1-\alpha e^{U(T+x)}e_1\right]}\right] \end{array}\right\}=C_1+C_5$$

If the cost function given in (5) is pseudo-convex, then it has only one local minimum and thus there will exist a unique global minimum. Now consider (12):

$$\frac{d}{dT}E(C)=\left[\begin{array}{l} C_2\left(y_0\Lambda'(T)+\mu_Y\Lambda(T)\Lambda'(T)\right)+C_3\mu_W\left(\Lambda'(T+T_W)-\Lambda'(T)\right) \\ +C_4\left[\mu_E\left(\Lambda'(T+T_W+T_E)-\Lambda'(T+T_W)\right)\right] \\ +C_5m\left(\left(\alpha e^{UT}Ue_1-\alpha e^{U(T+x)}Ue_1\right)e^{-m[\alpha e^{UT}e_1-\alpha e^{U(T+x)}e_1]}\right) \end{array}\right]>0$$

$$\frac{d^2}{dT^2}E(C)=\left[\begin{array}{l} C_2\left(y_0\Lambda''(T)+\mu_Y(\Lambda'(T))^2+\mu_Y\Lambda(T)\Lambda''(T)\right)+C_3\mu_W\left(\Lambda''(T+T_W)-\Lambda''(T)\right) \\ +C_4\left[\mu_E\left(\Lambda''(T+T_W+T_E)-\Lambda''(T+T_W)\right)\right] \\ +C_5m\left(\begin{array}{l}\left(\alpha e^{UT}U^2e_1-\alpha e^{U(T+x)}U^2e_1-m\left(\alpha e^{UT}Ue_1-\alpha e^{U(T+x)}Ue_1\right)^2\right) \\ \left(e^{-m[\alpha e^{UT}e_1-\alpha e^{U(T+x)}e_1]}\right)\end{array}\right) \end{array}\right]$$

$$\frac{d^2}{dT^2}E(C)>0$$

Hence E(C) is positive and convex. Since C(T) is linear in T and positive, C(T) is pseudo- convex in E(C). Hence the lemma.

V. Parameter estimation

In this section statistical estimation of the model parameters is to be performed which will take the modelling closer to the realm of applicability. Methods based on the estimation of the parameters for a phase type distribution, can be classified into three types such as (i) moment matching method (MM), (ii) maximum likelihood (ML) estimation and (iii) Bayes estimation. In this section, the method of moments (MM) is mainly used for the PH distribution. The concept of MM method is to find PH parameters so that the population moments can fit the moments derived from samples or probability density function of the true distribution. As mentioned before, the accuracy of MM method depends on the number of moments used.

Consider the Hyper - exponential distribution of order 2, that can be expressed as, $\Lambda(t) = mF(t) = (1 - \pi \exp(Ut)e)m$, so that

$$\pi = [p \quad 1-p],\ \mathrm{U} = \begin{bmatrix} -\frac{1}{\lambda_1} & 0 \\ 0 & -\frac{1}{\lambda_2} \end{bmatrix}, \mathrm{e} = [1 \quad 1]`.$$

Therefore, the p.d.f and the first four moments are given by:

$$f(t) = \Lambda'(t) = m\left[\frac{p}{\lambda_1} e^{-\frac{t}{\lambda_1}} + \frac{(1-p)}{\lambda_2} e^{-\frac{t}{\lambda_2}}\right], t \ge 0, 0 \le \mathrm{p} < 1\,.$$

$$m_1 = m[p\lambda_1 + (1-p)\lambda_2] \tag{14}$$

$$m_2 = m[2p\lambda_1^2 + 2(1-p)\lambda_2^2] \tag{15}$$

$$m_3 = m[6p\lambda_1^3 + 6(1-p)\lambda_2^3] \tag{16}$$

$$m_4 = m[24p\lambda_1^4 + 24(1-p)\lambda_2^4] \tag{17}$$

From (14) we have,

$$\lambda_2 = \frac{m_1 - pw\lambda_1}{m(1-p)} \tag{18}$$

Substituting (18) in (15), and after some simple calculations,

$$\lambda_1 = \frac{m_1}{w}\left\{1 \pm \sqrt{\frac{(m_2 m - 2m_1^2)(1-p)}{2pm_1^2}}\right\} \tag{19}$$

We have two cases:

Case (i)

When

$$\lambda_1 = \frac{m_1}{m}\left\{1 - \sqrt{\frac{(m_2 m - 2m_1^2)(1-p)}{2pm_1^2}}\right\} \tag{20}$$

Substituting (20) in (18) we have,

$$\lambda_2 = \frac{m_1}{m}\left\{1+\sqrt{\frac{(m_2 m - 2m_1^2)(p)}{2(1-p)m_1^2}}\right\} \tag{21}$$

Since (20) and (21) involve λ_1 and λ_2 expressed in terms of the unknown parameters p and m only. Using (16) and (17) the following system of equations are obtained:

$$m_3 - \frac{6m_1^3}{m^2}\left\{\frac{3m_2 m}{2m_1^2} - 2 + \left(\frac{m_2 m - 2m_1^2}{2m_1^2}\right)^{\frac{3}{2}} \frac{2p-1}{\sqrt{p(1-p)}}\right\} = 0 \tag{22}$$

$$m_4 - \frac{24m_1^4}{m^3}\left\{\frac{m^2 m_3}{3m_1^3} - 1 + \left[\left(\frac{m_2 m - 2m_1^2}{2m_1^2}\right)^{\frac{3}{2}} \frac{2(2p-1)}{\sqrt{p(1-p)}}\right] + \left[\left(\frac{m_2 m - 2m_1^2}{2m_1^2}\right)^{2}\left(\frac{3p^2 - 3p + 1}{p(1-p)}\right)\right]\right\} = 0 \tag{23}$$

Thus, (22) and (23) are to be solved to get the parameters m and p. Subsequently the remaining parameters are obtained by the method of back substitution.

Case (ii)

Proceeding similarly as in case (i), we have,

$$\lambda_1 = \frac{m_1}{m}\left\{1+\sqrt{\frac{(m_2 m - 2m_1^2)(1-p)}{2pm_1^2}}\right\} \tag{24}$$

$$\lambda_2 = \frac{m_1}{m}\left\{1-\sqrt{\frac{(m_2 m - 2m_1^2)p}{2(1-p)m_1^2}}\right\} \tag{25}$$

$$m_3 - \frac{6m_1^3}{m^2}\left\{\frac{3m_2 m}{2m_1^2} - 2 + \left(\frac{m_2 m - 2m_1^2}{2m_1^2}\right)^{\frac{3}{2}} \frac{1-2p}{\sqrt{p(1-p)}}\right\} = 0 \tag{26}$$

$$m_4 - \frac{24m_1^4}{m^3}\left\{\frac{m^2 m_3}{3m_1^3} - 1 + \left[\left(\frac{m_2 m - 2m_1^2}{2m_1^2}\right)^{\frac{3}{2}} \frac{2(1-2p)}{\sqrt{p(1-p)}}\right] + \left[\left(\frac{m_2 m - 2m_1^2}{2m_1^2}\right)^{2}\left(\frac{3p^2 - 3p + 1}{p(1-p)}\right)\right]\right\} = 0 \tag{27}$$

It is easily seen that both cases lead to the same result, but with the roles of p and 1-p interchanged. Hence, the parameter estimation carried out may help the practitioners in making flexible decisions for the proposed software reliability model via phase type distribution.

Additionally a numerical example is provided to demonstrate the applicability of the proposed software reliability model with warranty in Section III.

VI. Numerical illustration and sensitivity analysis

In this section, a numerical example is given to illustrate the impact of combining the phase type distribution and an extended warranty in the field of software reliability. The first four moments are assumed to be:

$$m_1 = 2.0125\times10^4, m_2 = 1.62125\times10^7, m_3 = 1.9607\times10^{10}, m_4 = 3.1644\times10^{13}.$$

The inter arrival times of software faults are assumed to be hyper exponential. Employing the moment matching method and using the above moments, the estimated values are obtained as:

$$w = 50,\ \pi = [0.95 \quad 0.05],\ \mathrm{U} = \begin{bmatrix} -\frac{1}{400} & 0 \\ 0 & -\frac{1}{450} \end{bmatrix}, \mathrm{e} = [1 \quad 1]`.$$

Also, assume that,

$$C_1 = 5000, C_2 = 50, y_0 = 0.5, \mu_y = 0.9, \mu_W = 0.95, \mu_E = 0.85, T_W = 500, T_E = 400, x = 1.5, C_3 = 360,$$
$$C_4 = 200, C_5 = 500$$

Utilizing the above parameter values in (5), the result has been presented in Table 1. The optimal software release time T*=67 and the corresponding cost 10.0249 is depicted in Table 1. Additionally, the sensitivity analysis of a proposed software reliability model using C(T) is analyzed. Tables 2 and 3 illustrate the sensitiveness of the long run average cost rate. Table 2 shows that as C_1, C_2 , C_3 and C_5 increase respectively, the long run average cost rate increases while an increase in C_4 results in a decrease in C(T). In a similar manner, Table 3 indicates that the long run average cost rate increases as x, μ_Y, μ_W and y_0 increase and decreases as the parameter μ_E increases. Further, Tables 2 and 3 illustrate that the optimal software release time T* increases with an increase in the parameters C_1, C_3, x and μ_W while it decreases as the parameters C_2, C_4, μ_Y, μ_E and y_0 increase. Also, the optimal software release time T* is unchanged as the parameter C_5 increases. Thus, the sensitivity analysis of such parameters may aid the software system manager in making decisions to model the software system testing.

Table 1: *C(T) versus T*

T	C(T)	T	C(T)	T	C(T)	T	C(T)	T	C(T)
20	10.6718	45	10.1554	62	10.0316	67	***10.0249***	80	10.0631
25	10.5325	50	10.1018	63	10.0292	68	10.0250	85	10.0975
30	10.4121	55	10.0629	64	10.0274	69	10.0257	90	10.1420
35	10.3097	60	10.0379	65	10.0261	70	10.0268	95	10.1959
40	10.2244	61	10.0345	66	10.0252	75	10.0393	100	10.2587

Table 2 : *Sensitivity analysis for the parameters C_1, C_2, C_3, C_4 and C_5 in C(T)*

C_1	T*	C(T*)	C_2	T*	C(T*)	C_3	T*	C(T*)	C_4	T*	C(T*)	C_5	T*	C(T*)
4000	65	8.9898	40	88	9.6337	**360**	**67**	**10.024**	150	69	11.882	400	67	10.0098
5000	**67**	**10.024**	**50**	**67**	**10.024**	400	76	11.189	**200**	**67**	**10.024**	**500**	**67**	**10.0240**
6000	69	11.057	60	54	10.288	450	87	12.595	250	65	8.1645	600	67	10.0400
7000	72	12.088	70	45	10.478	500	98	13.947	300	63	6.3018	700	67	10.0550

Table 3: *Sensitivity analysis for the parameters x, μ_y, μ_W, μ_E and y_0 in C(T)*

X	T*	C(T*)	μ_y	T*	C(T*)	μ_W	T*	C(T*)	μ_E	T*	C(T*)	y_0	T*	C(T*)
0.5	67	9.9760	**0.9**	**67**	**10.024**	.75	52	7.7208	.75	68	10.899	**.5**	**67**	**10.024**
1.5	**67**	**10.024**	1.1	54	10.276	.85	59	8.8888	**.85**	**67**	**10.024**	1.5	57	10.3964
2.5	67	10.068	1.3	45	10.451	**.95**	**67**	**10.024**	.95	66	9.1497	2.5	47	10.7144
3.5	68	10.108	1.5	39	10.581	1.05	76	11.129	1.05	65	8.2740	3.5	37	10.9770

VII. Conclusion

In this study, a phase type Non – homogeneous Poisson process software reliability model that incorporates warranty (fixed and an extended) via long run average cost rate in order to determine the optimal software release time is developed. Hence, PH-SRM is a promising tool to reduce the effort to select the best models in software reliability assessment. Also, the parameter estimation was carried out using moment matching method. The graphical illustrations conform to the theoretical observations made earlier. Additionally, a sensitivity analysis has been carried out for all the parameters, to exemplify the optimal software release policy (T*) and the corresponding long run average cost rate C(T*). To the best of authors' knowledge, it is observed that, phase type distribution in the cost analysis has not been studied from the view point of software reliability systems with fixed and extended warranty model. As a final remark, the extended warranty model enables the software manger to decide on whether the software is sufficiently tested to allow its release or unrestricted use. Such predictions provide a quantitative basis for achieving reliability, risk and cost goals.

References

[1] Asmussen, S. and Koole, G. (1993). Marked point processes as limits of Markovian arrival streams. *Journal of Applied Probability,* 30: 365–372.

[2] Blischke, W. and Murthy, D. Product Warranty Handbook, Marcel Dekker, New York, 1995.

[3] Fujiwara, T. and Yamada, S. (2001). Testing - domain dependent software reliability growth models and their comparisons of goodness-of-fit. *In Proceedings of the 7th ISSAT international conference on reliability and quality in design* (pp. 36–40).

[4] Goel, A. L. and Okumoto, K. (1979). Time – dependent error-detection rate model for software reliability and other performance measures. *IEEE Transactions on Reliability, R–28*: 206–211.

[5] Goel, A. L. (1985). Software reliability models: Assumptions, limitations and applicability. *IEEE Transactions on Software Engineering, SE–11*: 1411–1423.

[6] Gokhale, S. S. and Trivedi, K. S. (1998). Log-logistic software reliability growth model. *In Proceedings of the 3rd IEEE international high – assurance systems engineering symposium (HASE 1998)* (pp. 34–41).

[7] Jain, M. and Handa, B. R. (2001). Cost analysis for repairable units under hybrid warranty. Recent developments in Operational Research. (ed. Manju Lata Agarwal and Kanwar Sen), New Delhi, Narosa Publishing House, 149-165.

[8] Khoshgoftaar, T. M. (1988). Non - Homogeneous Poisson processes for software reliability growth. *In Proceedings of the international conference on computational statistics (COMPSTAT)* (pp. 13–14).

[9] Lam, Y. and Lam, P. (2001). An extended warranty policy with options open to consumers. *European Journal of Operational Research,* 131: 514-529.

[10] Langberg, N. and Singpurwalla, N. D. (1985). Unification of some software reliability models. SIAM Journal on Scientific Computing 6 (3): 781–790.

[11] Laprie, J. C., Kanoun, K., Béounes, C. and Kaâniche, M. (1991). The KAT (knowledge-action - transformation) approach to the modelling and evaluation of reliability and availability growth. *IEEE Transactions on Software Engineering,* 17(4): 370–382.

[12] Littlewood, B. (1981). Stochastic reliability growth: A model for fault removal in computer programs and hardware designs. *IEEE Transactions on Reliability, 30* (10):313–320.

[13] Musa, J. D., Iannino, A. and Okumoto, K. *Software reliability, measurement, prediction application*. New York: McGraw-Hill, 1987.

[14] Musa, J. D. *Software reliability engineering*. New York: McGraw-Hill, 1999.

[15] Nguyen, D. G. and Murthy, D. (1984). Cost analysis of warranty policies. *Naval Research Logistics Quaterly*, 31:525-541.

[16] Neuts, M. F. *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach*. The John Hopkins University Press, Baltimore, 1981.

[17] Okamura, H. and Dohi, T. (2006). Building phase - type software reliability model. In *Proceedings of the 17th international symposium on software reliability engineering (ISSRE 2006)*, 289–298.

[18] Okamura, H. and Dohi, T. (2008). Hyper-Erlang software reliability model. *Proceedings of the 14th Pacific Rim international symposium on dependable computing (PRDC 2008),* 232–239.

[19] Okumoto, K. and Goel, L. (1980). Optimum release time for software systems based on reliability and cost criteria. *Journal of Systems and Software*, 1: 315–318.

[20] Okamura, H., Dohi, T. and Osaki, S. (2013). Software reliability growth models with normal failure time distributions. *Reliability Engineering and System Safety,* 116:135–141.

[21] Prince Williams, D. R. (2007). Study of the Warranty Cost Model for Software Reliability with an Imperfect Debugging Phenomenon. *Turkish Journal of Electrical Engineering & Computer Sciences,* 15 (3): 369-381.

[22] Pham, H. (1996). A software cost model with imperfect debugging, random life cycle and penalty cost. *International Journal of System Science,* 27(5): 455–463.

[23] Pham, H. *Software Reliability*. Springer, Singapore, 2000.

[24] Ross, S. M. *Stochastic Processes*. Second edition. Wiley, New York, 1996.

[25] Xie, M., Hu, Q. P., Wu, Y. P. and Ng, S. H. (2007). A study of the modeling and analysis of software fault detection and fault - correction processes. *Quality and Reliability Engineering International,* 23:459–470.

[26] Yamada, S., Ohba, M. and Osaki, S. (1983). S-shaped reliability growth modeling for software error detection. *IEEE Transactions on Reliability,* 32: 475–478.

[27] Yamada, S. (1994). Optimal release problems with warranty period based on a software maintenance cost model. *Transactions of Information Processing Society of Japan,* 35 (9), 2197-2202.

[28] Zhao, M. and Xie, M. (1996). On maximum likelihood estimation for a general non-homogeneous Poisson process. *Scandinavian Journal of Statistics, 23:*597–607.

[29] Zhang, X. and Pham, H. (1998). A software cost model with warranty cost, error removal times and risk costs. *IIE Transactions*, 30 (12): 1135-1142.