Journal of the Korean Data & Information Science Society
2012, **23**(4), 815–823

http://dx.doi.org/10.7465/jkdi.2012.23.4.815
한국데이터정보과학회지

Two model comparisons of software reliability analysis for Burr type XII distribution

Jeong Hyang An[1]

[1]Department of Internet Information, Daegu Haany Universityy

Received 14 June 2012, revised 4 July 2012, accepted 14 July 2012

Abstract

In this paper reliability growth model in which the operating time between successive failure is a continuous random variable is proposed. This model is for Burr type XII distribution with two parameters which is discussed in two versions: the order statistics and non-homogeneous Poisson process. The two software reliability measures are obtained. The performance for two versions of the suggested model is tested on real data set by U-plot and Y-plot using Kolmogorov distance.

Keywords: Burr type XII distribution, growth model, Kolmogorov distance, non-homogeneous, Poisson precess, software reliability.

1. Introduction

Software is an important element in computing systems. Different from hardware, the software does not wear-out and it can be easily reproduced. Furthermore, software systems are usually debugged during the testing phase so that their reliability is improving over time as a result of detecting and removing software faults. Many software reliability growth models have been proposed for the study of software reliability, e.g. Pham (2000). Markov models are one of the first types of models proposed in software reliability analysis. In addition, the user of non-homogeneous poisson process (NHPP) models for characterizing software reliability growth has a long history, beginning with Goal and Okumoto (1979). Also see, Goel (1980), Singpurwalla and Wilson (1999), Nalini, Zhaohui and Bonnie (2008), and moon and Lee (2011) for an overview of the use of such models in practice. NHPP models, which are important in software reliability analysis. NHPP models are attractive because of their mathematical tractability.

Methods for software reliability estimation become increasingly relevant with the growing routine use of computers in many safety-critical applications such as nuclear power plants, chemical process plants and aircraft.

The most widely accepted and used models for the purpose of estimating systematic reliability are statistical system testing models. The aim of statistical system testing is the estimation of the system's failure rate which is interpreted as the probability of failure on demand. Estimation is performed based on the results of N independent test runs. A demand

[1] Professor, Department of Internet Information, Daegu Haany University, Gyeongsan 712-715, Korea. E-mail: jhan@dhu.ac.kr

is an input generated by the software's environment. A test input is randomly generated according to the software's operational input distribution.

There are several software reliability of growth models in the literature that estimate the remaining error contained in the software. These models may be classified in two broad types. First, inter-failure time models in which the operating time between successive failures is a continuous random variable. Second, discrete software reliability models where the interest is the number of faults or failures in a specified time interval.

In order statistics (OS) model, the failure times of a software reliability process are modeled as OS of independent non-identically distributed random variables such as: Jelinski and Moranda (1972), Littlewood (1981), Abdel-Ghaly *et al.* (1986, 1990, 1997), Lawless (2003), and Lee and Yoon (2008).

In NHPP model, the testing time used as the unit of error detection period for describing the time dependent behavior of the cumulative number of error detected by the software testing, such as Jelinski and Moranda (1972), and Park (2006).

This paper is organized as follows. Section 2 treats some review of the software reliability model based on OS and in this section, we extend the model of Abdel-Ghaly (1986) to a new model. Section 3 treats some review NHPP of the software reliability model based on Poisson distribution in this section to a new model. For estimation of parameters, we propose the maximum likelihood estimation which can be solved by numerical method such as Newton-Raphson iterative procedure. In Section 4, using a real data in Musa (1987), we investigate the usefulness of proposed model by depicting U-plot Y-plot by Kolmogorov distance. Section 4 devotes some conclusions.

2. The order statistics

Rayleigh model (Lee and Lee, 2010) was introduced by Abdel-Ghaly (1996). He assumed that the system starts its life containing N faults. Each of these faults can cause system failure at any time T. Let X be the total time on test until the fault i is removed. The X 's are assumed to be independent identically distributed (i.i.d.) random variables with Rayleigh probability density function (p.d.f.), given by

$$f(x|\phi) = \begin{cases} 2x\phi\exp(-\phi x^2), & \text{x} > 0 \\ 0, & \text{otherwise}, \end{cases} \tag{2.1}$$

where $\phi > 0$.

Let $X_{(1)}, X_{(2),} \cdots$ be OS of the time of occurrence of the 1st, 2nd, $\cdots$ failures and $T_1, T_2, \cdots$, be the inter-failure times. Then

$$\begin{aligned} T_1 &= X_{(1)}, \\ T_i &= X_{(i)} - X_{(i-1),} \quad i = 2, 3, \cdots . \end{aligned} \tag{2.2}$$

If a total time τ has elapsed and $(i-1)$ faults have been removed, then the p.d.f. of T_i will be given in the following form

$$f(t_i|\lambda_i) = 2\lambda_i(\tau_{i-1} + t_i)\exp\left[-\lambda_i\left\{(\tau_{i-1} + t_i)^2 - \tau_{i-1}^2\right\}\right], \quad t_i > 0, \tag{2.3}$$

where $t_i > 0$, $\lambda_i = (N - i + 1)\phi$ and τ_{i-1} is the total elapsed time at the point $(i - 1)$ which is equal to $\sum t_j$.

This result will be used to extend model which will be presented in this paper and it will be discussed using the OS and NHPP models.

In the OS model, it is assumed that there are N software faults at the start of testing. Each of the N faults in the program will cause a failure after a time which is distributed as in (2.1) and independently of other faults, with a different failure rate ϕ_j. When a failure occurs, there is an instantaneous removal of the faults which caused the failure, and no new faults are introduced during the debugging process.

If total time $\tau_i - 1$ has elapsed, and $(i - 1)$ faults have been removed, the failure rate of the program is given by

$$\lambda_i = \sum_{j=1}^{N-i+1} \phi_j, \tag{2.4}$$

where $\phi_1 + \phi_2 + \cdots + \phi_{N-i+1}$ are the non-identical rates of the remaining faults. The ϕ_i are regarded as realizations of random variables. Thus the occurrence of the N faults with which the program begins life can be regarded as i.i.d. random variables.

When debugging starts (i.e. $T = 0$) each ϕ has gamma distribution with shape parameter α, i.e.

$$f(\phi|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\phi^{\alpha-1}\exp(-\beta\phi), \quad \alpha > 0. \tag{2.5}$$

Bayes' theorem can be used to describe how uncertainty about the occurrence rate of the faults changes with time. Consider the occurrence rate, ϕ, of one of the $N - i + 1$ remaining faults at the period "now", then

$$\begin{aligned} & f(\phi|\text{this fault is fixed}\in(0,\tau_{i-1})) \\ & \quad = f(\phi|\text{no failure is casused by this fault}\in(0,\tau_{i-1})) \\ & \quad = C\exp\left(-\phi\tau_{i-1}^2\right)\frac{\beta^\alpha}{\Gamma(\alpha)}\phi^{\alpha-1}\exp(-\phi\beta) \\ & \quad = \frac{[\phi(\beta+\tau_{i-1}^2)]^{\alpha-1}\exp(-\phi(\beta+\tau_{i-1}^2))}{\Gamma(\alpha)}, \end{aligned} \tag{2.6}$$

That is, ϕ distributes as gamma distribution (Rahman and Muraduzzaman, 2010) with parameters α and $\beta + \tau_{i-1}^2$, which we denote as $\phi \sim \Gamma(\alpha, (\beta + \tau_{i-1}^2))$, where

$$\begin{aligned} C^{-1} &= \int_o^\infty \frac{\beta^\alpha}{\Gamma(\alpha)}\phi^{\alpha-1}\exp(-\phi(\beta+\tau_{i-1}^2))d\phi \\ &= \frac{\beta^\alpha}{(\beta+\tau_{i-1}^2)^{\alpha-1}} \end{aligned}$$

and

$$\Pr(\text{no failure due to this fault in the interval}\in(0,\tau_{i-1})) = \exp(-\phi\tau_{i-1}^2).$$

From (2.4), the failure rate of program λ_i is a sum of the $(N-i+1)$ i.i.d. gamma distributed random variable with shape parameter $(\alpha(N-i+1))$ and scale parameter $(\beta+\tau_i-1^2)$, with p.d.f. is given by

$$f(t_i|\alpha,\beta,N,\tau_{i-1}) = \frac{(\beta+\tau_{i-1}^2)^{\alpha(N-i+1)-1}}{\Gamma(\alpha(N-i+1))}\lambda_i^{\alpha(n-i+1)-1}\exp-\left\{(\beta+\tau_{i-1}^2)\lambda_i\right\},\lambda_i>0. \quad (2.7)$$

Multiplying (2.7) by (2.3), and integrating with respect to λ_i we get the p.d.f. of T_i given τ_i-1 as

$$f(t_i|\alpha,\beta,N,\tau_{i-1}) = \frac{2(\beta+\tau_{i-1}^2)^{\alpha(N-i+1)}\alpha(N-i+1)(\tau_{i-1}+t_i)}{\{\beta+(\tau_{i-1}+t_i)^2\}^{\alpha(N-i+1)+1}}, \quad (2.8)$$

This form can be obtained from the OS formulation of the model; if the detection times (and removal) of the fault are independent random variables, X_i, with density function given by

$$f(x|\alpha,\beta) = \frac{2\alpha\beta x}{(\beta+x^2)^{\alpha+1}}, \qquad x>0, \quad (2.9)$$

which is a special case Burr type XII distribution.

The parameters α, β and N are estimated by maximizing the likelihood function to obtain their maximum likelihood estimates (MLEs) of α, β and N. Such MLEs are used to predict the p.d.f., reliability function and its measures for the next failure time. The likelihood function based on the data of the first n failures is given

$$\begin{aligned} L(\alpha,\beta,N|t_1,t_2,\cdots,t_n) &= \prod_{i=1}^{n} f(t_i|\tau_{i-1},\alpha,\beta,N) \quad (2.10) \\ &= 2^n\alpha^n\left[\prod_{i=1}^{n}\left\{(N-i+1)\frac{\tau_i+t_i}{\beta+(\tau_{i-1}+t_i)^2}\right\}\right]\exp\left[\alpha\sum_{i=1}^{n}(N-i+1)\log\left\{\frac{\beta+\tau_{i-1}^2}{\beta+(\tau_{i-1}+t_i)^2}\right\}\right]. \end{aligned}$$

Then we have the logarithm in the above likelihood function of (2.10) is given by

$$\begin{aligned} &\ell(\alpha,\beta,N|t_1,t_2,\cdots,t_n) \quad (2.11) \\ &= n\log\alpha+\sum_{i=1}^{n}\log\left\{\frac{(N-i+1)(\tau_{i-1}+t_i)}{\beta+(\tau_{i-1}+t_i)^2}\right\}+\alpha\sum_{i=1}^{n}(N-i+1)\log\left\{\frac{\beta+\tau_{i-1}^2}{\beta+(\tau_{i-1}+t_i)^2}\right\}. \end{aligned}$$

Now, we take the partial derivative of the log-likelihood function with respect to α, N and β, respectively. Then we have the normal equation are given by

$$\begin{aligned} \frac{\partial\ell}{\partial\alpha} &= \frac{n}{\alpha}+\sum_{i=1}^{n}(N-i+1)\log\left\{\frac{\beta+\tau_{i-1}^2}{\beta+(\tau_{i-1}+t_i)^2}\right\}, \\ \frac{\partial\ell}{\partial N} &= \sum_{i=1}^{n}\frac{1}{(N-i+1)}+\alpha\sum_{i=1}^{n}(N-i+1)\log\left\{\frac{\beta+\tau_{i-1}^2}{\beta+(\tau_{i-1}+t_i)^2}\right\}, \quad (2.12) \\ \frac{\partial\ell}{\partial\beta} &= -\sum_{i=1}^{n}\frac{1}{\beta+(\tau_{i-1}+t_i)^2}+\alpha\sum_{i=1}^{n}(N-i+1)\log\left\{\frac{t_i^2-2\tau_{i-1}t_i}{(\beta+(\tau_{i-1}+t_i)^2)(\beta+\tau_{i-1}^2)}\right\}. \end{aligned}$$

Then we can simplify the above equations as follow,

$$\widehat{\alpha} = -\frac{n}{\sum_{i=1}^{n}(N-i+1)\log\left\{\dfrac{\beta+\tau_{i-1}}{\beta+(\tau_{i-1}+t_i)^2}\right\}},$$

$$\widehat{N} = -\frac{2\alpha\sum_{i=1}^{n}\log\dfrac{\beta+\tau_{i-1}}{\beta+(\tau_{i-1}+t_i)^2}}{n\left\{2+n(n-1)\alpha\sum_{i=1}^{n}\left\{\dfrac{\beta+\tau_{i-1}}{\beta+(\tau_{i-1}+t_i)^2}\right\}\right\}},$$

and

$$\sum_{i=1}^{n}\frac{1}{\beta+(\tau_{i-1}+t_i)^2} = \alpha\sum_{i=1}^{n}(N-i+1)\frac{t_i^2+2\tau_i+t_i}{(\beta+\tau_{i-1})(\beta+(\tau_{i-1}+t_i))^2}.$$

The above equations can be solved by numerical method. Once the estimates $\widehat{\alpha}$, $\widehat{\beta}$ and $\widehat{N}$ of the parameters are obtained, we will use them to predictive p.d.f.. of T_{n+1} by substituting them in (2.8).

$$\widehat{f}(t_{n+1}|\tau_n) \\ = 2\widehat{\alpha}(N-n)(\tau_n+t_{n+1})\frac{(\widehat{\beta}+\tau_n^2)^{\widehat{\alpha}(N-n)}}{\left\{\widehat{\beta}+(\tau_n+t_{n+1})^2\right\}^{\widehat{\alpha}(N-n)+1}}. \tag{2.13}$$

Now, one wants to obtain the probability that the system will be running without failure up to time T_n+1, this probability is the reliability function of T_{n+1} is given by

$$\widehat{R}(t_n+1|\tau_n) = \left\{\frac{\widehat{\beta}+\tau_n^2}{\widehat{\beta}+(\tau_n+t_n+1)^2}\right\}^{(\widehat{N}-n)\widehat{\alpha}}. \tag{2.14}$$

The failure rate(hazard) function of T_n+1 is as follows

$$\widehat{\alpha}(t_{n+1}|\tau_n) = 2(\widehat{N}-n)\widehat{\alpha}\frac{(\tau_n+t_{n+1})}{(\widehat{\beta}+(\tau_n+t_{n+1})^2)}. \tag{2.15}$$

3. Non-homogeneous Poisson process

NHPP models are used to predict future failures. This approach is one of the best ways to measure software reliability and other measures. NHPP are also called counting models because the mean value function $H(t)$ represents the cumulative number of faults exposed by time t. These models also have a failure rate, λ, which is a function of time.

The first model discussed in Section 2 is the OS model where N is the initial number of faults in the system or program. Suppose that α is of known value, and N has a Poisson distribution with parameter μ and probability mass function is as follows

$$P(N) = \frac{\mu^N\exp(-\mu)}{N!}, N = 0,1,\cdots. \tag{3.1}$$

Miller (1984) suggested that any OS mixture model mixed over the Poisson N, will be a NHPP. The posterior mass function of N can be obtained from (2.7) and (3.1) as follows

$$P^*(N|t_1, t_{2,} \cdots, t_n, \alpha, \beta) = C \cdot L(N, \alpha, \beta | t_1, t_{2,} \cdots, t_n) P(N),$$

where

$$\begin{aligned} C^{-1} &= (2\alpha\mu)^n \left[\prod_{i=1}^{n} \frac{\tau_{i-1} + t_i}{\beta + (\tau_{i-1} + t_i)^2} \right] \\ &\times \exp\left[-\mu - \alpha \sum_{i=1}^{n} (n-i+1) \log\left(\frac{\beta + (\tau_{i-1} + t_i)^2}{\beta + \tau_{i-1}^2} \right) \right. \\ &\left. + \mu \exp\left\{ \sum_{i=1}^{n} (n-i+1) \log\left(\frac{\beta + (\tau_{i-1} + t_i)^2}{\beta + \tau_{i-1}^2} \right) \right\} \right], \end{aligned} \tag{3.2}$$

since, the last equation follows by the fact that

$$\sum_{N=n}^{\infty} \frac{\mu^{N-n}}{(N-n)!} \exp\left\{ \alpha \sum_{i=1}^{n} (N-n) \log\left(\frac{\beta + \tau_{i-1}}{\beta + (\tau_{i-1} + t_i)^2} \right) \right\} = \mu \exp\left[\alpha \sum_{i=1}^{n} \log\left(\frac{\beta + \tau_{i-1}}{\beta + (\tau_{i-1} + t_i)^2} \right) \right].$$

Therefore, the posterior distribution of the number of remaining faults is Poisson distribution with probability mass function is given by

$$\begin{aligned} &P^*(N|t_{1,}t_2, \cdots, t_n, \alpha, \beta) \\ &= \frac{\left[\mu \exp\left(-\alpha \sum_{i=1}^{n} \log\left(\frac{\beta + (\tau_{i-1} + t_i)^2}{\beta + \tau_{i-1}^2} \right) \right) \right]^{N-n}}{(N-n)!} \\ &\times \exp\left[-\mu \exp\left\{ -\alpha \sum_{i=1}^{n} \log\left(\frac{\beta + (\tau_{i-1} + t_i)^2}{\beta + \tau_{i-1}^2} \right) \right\} \right]. \end{aligned} \tag{3.3}$$

The predictive p.d.f. of T_{n+1} can be obtained from equations (2.10) and (3.3) as follows

$$\begin{aligned} f^*(t_{n+1}|\mu, \alpha, \beta, \tau_n) &= \sum_{N=n}^{\infty} \widehat{f}(t_{n+1}|\tau_n) P^*(N-n|t_1, t_{2,} \cdots, t_n, \alpha, \beta) \\ &= 2\mu\alpha \frac{\tau_n + t_{n+1}}{(\beta + (\tau_n + t_{n+1})^2)^{\alpha+1}} \\ &\times \exp\left[\frac{-\mu}{(\beta + \tau_n)^{\alpha}} \left\{ 1 - \left(\frac{\beta + \tau_n^2}{\beta + (\tau_n + \tau_{n+1})^2} \right)^{\alpha} \right\} \right]. \end{aligned} \tag{3.4}$$

By integrating (3.4) with respect to t_{n+1} we get the predictive c.d.f. of T_{n+1} as follows

$$F^*(t_{n+1}|\mu, \alpha, \beta, \tau_n) = 1 - \exp\left[-\mu \frac{1}{(\beta + \tau_n)^{\alpha}} \left\{ 1 - \left(\frac{\beta + \tau_n^2}{\beta + (\tau_n + \tau_{n+1})^2} \right)^{\alpha} \right\} \right]. \tag{3.5}$$

It follows, from (3.5), that the predictive reliability function is given by

$$R^*(t_{n+1}|\mu,\alpha,\beta) = \exp\left[-\mu\frac{1}{(\beta+\tau_n)^\alpha}\left\{1-\left(\frac{\beta+\tau_n^2}{\beta+(\tau_n+\tau_{n+1})^2}\right)^\alpha\right\}\right], \tag{3.6}$$

and the failure rate function is

$$\lambda(t_{n+1}) = 2\mu\alpha\frac{\tau_n+t_{n+1}}{(\beta+(\tau_n+t_{n+1})^2)^{\alpha+1}}. \tag{3.7}$$

4. Conclusions

The new software reliability growth model has been considered to measure and predict the reliability of software which has been studied in the two previous versions. The user usually needs to know which version would be most suitable in a particular context. Such comparison may be carried out with the assistance of many different statistical tools. Among these tools are the U-plot, Y-plot, measures of variability, the U-plot (Brocklehurst and Littlewood, 1992) is used to determine if the postulated c.d.f., $\widehat{F}(t)$ is close to true distribution $F(t)$. This can be developed into an operational measure by finding the Kolmogorov distance. In the meantime, the Y-plot (Karanta, 2006) measures the consistency of a model's bias; a model might be initially too pessimistic and eventually too optimistic concerning the number of faults in the software. On the other hand, any systematic departure from unit slop indicates step a mis-specification of the probability distribution (that is, a reasonably consistent bias). The performance of the two versions of the new model can be achieved for real data set by using both U-plot and Y-plot are measure by finding the Kolmogorov distance.

The Table 4.1 is real data set, The reason for using data sets is for comparison purposes and decisions regarding the version's preference for the new model. These analysis has been carried out for inter-failure time data. The following method will be used to accomplish the calculations for a data set: assuming that each data set contains n inter-failure times, $m(\leq n-1)$ is chosen as a starting sample size. Using the results of Section 2 and 3, the MLEs are obtained for the sample and then used to direct in p.d.f. and c.d.f., for the next failure-time. The sample is then increased by the observed $T_m + 1$, the process repeated for $T_m + 2$, and so on.

This data set is shown in Table 4.1 which contains 136 observations of the execution time in seconds between successive failures. The sample size at the beginning of the analysis is 35 observations to estimate the models parameters and to predict the 36th failure. The process is repeated for the next observation and so on.

Table 4.2 summarizes the results concerning the quality of performance of two versions of the new model.

Table 4.2 show the U-Plot for distances are 0.2934 and 0.3024, respectively. The Kolmogorov distances of the U-Plot of the OS and NHPP are significant at 0.01 level. That is, there is evidence of optimistic prediction for the two versions of the new model, and show the Y-Plot for the OS and NHPP, respectively. The Kolmogorov distances are: 0.1027 and 0.0729, respectively.

The OS model, the Kolmogorov distance is significant for the 0.20 level. However, NHPP model, the Kolmogorov distances are nonsignificant at the 0.20 level.

In this study a new reliability growth model in which the operating time between successive failures is a continuous random variable is considered. This model has been discussed in two versions, the order statistics and the non-homogeneous Poisson process. The Kolmogorov distances for software reliability measurements have been derived from the two versions of the new model. The OS seems less optimistic than the NHPP versions on the real data set.

Table 4.1 Execution times in s between successive failures, Musa(1981)

3	30	113	81	115	9	2	91	112	15
138	50	77	24	108	88	670	120	26	114
325	55	242	68	422	180	10	1146	600	15
36	4	0	8	227	65	176	58	457	300
97	263	452	255	197	193	6	79	816	1351
148	21	233	134	357	193	236	31	369	748
0	232	330	365	1222	543	10	16	529	379
44	129	810	290	300	529	281	160	828	1011
445	296	1755	1064	1783	860	983	707	33	868
724	2323	2930	1461	843	12	261	1800	865	1435
30	143	108	0	3110	1247	943	700	875	245
729	1897	447	386	446	122	990	948	1082	22
75	482	5509	100	10	1071	371	790	6150	3321
1045	648	5485	1160	1864	4116				

Table 4.2 The analysis of data in Table 4.1

Version	U-Plot	Y-Plot
OS	0.2934	0.1027
NHPP	0.3024	0.0729

References

Abdel-Ghaly, A. A. (1996). A new model for software and hardware faults's removel process. *Economic and Business Review*, Faculty of Commerce, Ain-shams University, **1**, 23-47 .

Abdel-Ghaly, A. A., Al-Dayian, G. R. and Al-Kashkari, F. H. (1997). The use of Burr type XII distribution on software modelling. *Microelectro Reliability*, **37**, 305-313.

Abdel-Ghaly, A. A. and Attiah, A. F. (1990). A Bayesian analysis of the Weibull model of software reliability. *Proceedings of 25th Annual Conference Institute of Statistical Studies and Research*, **25**, 15-34.

Abdel-Ghaly, A. A., Chan, P. Y. and Littlewood, B. (1986). Evaluation of computing software reliability predictions. *IEEE Transaction Software Engineering*, **12**, 950-967.

Brocklehurst, S. and Littlewood, B. (1992). New ways to get accurate reliability measures. *IEEE Software*, **9**, 34-42.

Goel, A. L. (1980). Software error detecting model with applications. *Journal of System Software*, **1**, 243-249.

Goal, A. L., and Okumoto, K. (1979). Time-dependent error-detection rate model for software reliability and other performance measure. *IEEE Transactions on Reliability*, **28**, 206-211.

Jelinski, K. and Moranda, P. (1972). Software reliability research. In *Statistical Computer Performance Evaluation*, edited by W. Freiberger, Academic Press, London, 465-484.

Karanta, I. (2006). *Methods and problems of software reliability estimation*, VTT working papers 63, VTT Technical Research Center of Finland.

Lawless, J. F. (2003). *Statistical models and methods for lifetime data*, 2nd edition, John Wily and Sons, New York.

Lee, J. and Lee, C. (2010). Reliability and ratio in aright truncated Rayleigh distribution. *Journal of the Korean Data & Information Science Society*, **21**, 195-200.

Lee, J. and Yoon, S. C. (2008). Software reliability for order statistic of Burr XII Distribution. *Journal of the Korean Data & Information Science Society*, **19**, 1361-1369.

Littlewood, B. (1981). Stochastic reliability growth: A model for fault removal in computer programs and hardware design. *IEEE Transactions on Reliability*, **30**, 13-320.

Miller, D. R. (1984). *Exponential order statistics models of software reliability growth*, Technical Report T-496184, George Washington University, Washington, U.S.A..

Moon, Y. and Lee, C. (2011). Estimating reliability in discrete distributions. *Journal of the Korean Data & Information Science Society*, **22**, 811-817.

Nalini, R., Zhaohui, L., and Bonnie, K. R. (2008). NHPP models with Markov switching for software reliability. *Computational Statistics and Data Analysis*, **52**, 3988-3999.

Pham, H. (2000). *Software reliability*, New York, Berlin.

Park, J. Y., Park, J. H. and Fujiwara, T. (2006). Frameworks for NHPP software reliability growth models. *International Journal of Reliability and Applications*, **7**, 155-166.

Rahman, M. and Muraduzzaman, S. M. (2010). Likelihood ratio in estimating gamma distribution parameters. *Journal of the Korean Data & Information Science Society*, **21**, 345-354.

Singpurwalla, N. and Wilson, S. (1999). *Statistical methods in software engineering : Reliability and risk*, Springer, New York.