Communications of the Korean Statistical Society
2011, Vol. 18, No. 5, 667–674

DOI: http://dx.doi.org/10.5351/CKSS.2011.18.5.667

Estimation of Coverage Growth Functions

Joong-Yang Park[a], Gyemin Lee[1,a], Seo Yeong Kim[b]

[a]Department of Information and Statistics, RINS and RICI, Gyeongsang National University
[b]Department of Information and Statistics, Graduate School, Gyeongsang National University

Abstract

A recent trend in software reliability engineering accounts for the coverage growth behavior during testing. The coverage growth function (representing the coverage growth behavior) has become an essential component of software reliability models. Application of a coverage growth function requires the estimation of the coverage growth function. This paper considers the problem of estimating the coverage growth function. The existing maximum likelihood method is reviewed and corrected. A method of minimizing the sum of squares of the standardized prediction error is proposed for situations where the maximum likelihood method is not applicable.

Keywords: Construct, coverage, coverage growth function, maximum likelihood, prediction error, software testing, software reliability.

1. Introduction

Software systems have become critical components of computer systems. Failures in a software system can cause severe consequences; therefore, software developers and users are concerned about the quality of software systems, especially reliability. The reliability of a software system is improved only when faults resident in the software system are detected and removed. Software systems are usually tested for fault detection and removal before they are released because software testing is a key activity to improve software reliability.

Theoretically it is impossible to execute all the possible inputs of a software system under testing. Consequently, it is nearly impossible to detect and remove all the faults in a software system. The developed software system is to be released or delivered at an appropriate time. This demands that software testers perform testing activity for a reasonable amount of time. Software developers usually determine when to stop testing based on the estimates of reliability measures.

Many software reliability growth models(SRGMs) have been proposed and applied in practice for the estimation of software reliability measures (Musa *et al.*, 1987; Lyu, 1996; Musa, 1999). Most SRGMs describe the relationship between a reliability measure and testing time. Such a relationship is obtained by modeling the fault detection and removal process during testing; however, it was recognized that the testing time was not enough to express the fault detection and removal process. Attempts to integrate coverage information into SRGMs have been made by Gokhale *et al.* (1996), Malaiya *et al.* (2002), Pham and Zhang (2003), Park and Fujiwara (2006), Crespo *et al.* (2008, 2009), and Park *et al.* (2008b). Each coverage-based SRGM involves a coverage growth function(CGF), that describes the coverage growth process during testing. The performance of such SRGMs depends on how closely its CGF represents the actual coverage growth phenomenon. Recently, a class of CGFs

[1] Corresponding author: Professor, Department of Information and Statistics, RINS and RICI, Gyeongsang National University, 900, Gazwa-dong, Jinju 660-701, Korea. E-mail: gyemin@gnu.ac.kr

for software reliability modeling was proposed by Park *et al.* (2007, 2008a). Three specific CGFs of the class have been empirically validated.

Application of CGFs requires the estimation of the CGFs. This paper considers the problem of estimating the CGFs. Section 2 briefly reviews the CGFs proposed by Park *et al.* (2007, 2008a). The existing maximum likelihood(ML) method is reviewed and corrected in Section 3. Since the coverage data sets reported by the previous studies do not provide all the information necessary for the ML method, the method minimizing the sum of squares of the standardized prediction error is proposed as an alternative in Section 4. Section 5 presents numerical examples of the alternative estimation method.

2. Coverage and Coverage Growth Function

Let us begin by defining the coverage and CGF. A software system can be considered as a collection of constructs, where a construct is a basic building element of a software system. Some usual constructs are statements, blocks, branches, c-uses and p-uses. Let M be the set of constructs of the software system under testing. Then M is the software system itself. The set of constructs executed up to t testing time is denoted by $M(t)$. One metric for measuring the thoroughness and/or the progress of the testing is the coverage defined as $C(t) = |\,M(t)\,|\,/\,|\,M\,|$, where $|\cdot|$ is the cardinality of a set. Since test cases are selected randomly from the inputs domain according to the given testing profile, $M(t)$ and $C(t)$ are stochastic processes. CGF $c(t)$ is defined as the expected value of $C(t)$, *i.e.*, the expected proportion of constructs executed by t.

Park *et al.* (2007) considered the case where the testing time is discrete and modeled the coverage growth process under the following assumptions:

(i) Constructs in M are executed independently.

(ii) The execution probability p of a construct follows a distribution with cdf $F(p)$ and pdf $f(p)$.

Assumption (ii) reflects that constructs in M may have different execution probabilities. Let T denote the time to execution of a construct. Due to Assumption (i), the time to execution of a construct with execution probability p follows a geometric distribution. Therefore the probability that a construct is executed up to t testing time is obtained as

$$\pi(t) = \Pr(T \le t) = \int_0^1 \Pr(T \le t\,|\,p)\, f(p)\, dp = \int_0^1 \left[1 - (1-p)^t\right] f(p)\, dp. \tag{2.1}$$

Specifically, if $F(p)$ is a beta distribution with parameters α and β,

$$\pi_{beta}(t) = 1 - \frac{B(\alpha,\beta+t)}{B(\alpha,\beta)}, \tag{2.2}$$

where $B(\alpha,\beta)$ is the beta function.

Park *et al.* (2008a) dervied $\pi(t)$ for the continuous testing time by replacing Assumptions (ii) with

(ii)$'$ The execution rate λ of a construct follows a distribution with cdf $G(\lambda)$ and pdf $g(\lambda)$.

Then the time to execution of a construct with execution rate λ is exponentially distributed with parameter λ. Therefore,

$$\pi(t) = \Pr(T \le t) = \int_0^\infty \Pr(T \le t|\,\lambda)\, g(p)\, dp = \int_0^\infty \left[1 - e^{-\lambda t}\right] g(p)\, dp. \tag{2.3}$$

The gamma and lognormal distributions have been considered for $G(\lambda)$. The corresponding $\pi(t)$'s are obtained as

$$\pi_{gamma}(t) = 1 - (1 + \beta t)^{-\alpha} \tag{2.4}$$

for the gamma distribution with parameters α and β and

$$\pi_{lognormal}(t) = 1 - \int_0^\infty e^{-\lambda t} \frac{e^{-\frac{(\ln\lambda-\mu)^2}{2\sigma^2}}}{\lambda\sigma\sqrt{2\pi}} \, d\lambda \tag{2.5}$$

for the lognormal distribution with parameters μ and σ.

The probability that a construct is executed up to t, $\pi(t)$, can be interpreted as the expected proportion of constructs executed up to t. Thus $\pi(t)$'s were proposed as plausible CGFs. It was further shown that $\pi(t)$'s could be used irrespective of the continuity of the testing time.

3. Maximum Likelihood Estimation of CGF

This section considers ML estimation of the CGFs reviewed in the previous section. Let us first present the existing ML estimation procedure. Suppose that a coverage growth process was observed at t_i for $i = 1, 2, \ldots, n$. Let m_{t_i} and c_{t_i} be values of $|M(t)|$ and $C(t)$ measured at t_i. The observed increments of $|M(t)|$ and $C(t)$ during $(t_{i-1}, t_i]$ are then expressed as $x_{t_i} = m_{t_i} - m_{t_{i-1}}$ and $c_{t_i} - c_{t_{i-1}} = x_{t_i}|M|^{-1}$, where $m_{t_0} = 0$, $c_{t_0} = 0$ and $t_0 = 0$.

Now suppose that we want to predict the coverage at some $t' > t_n$ when the corresponding CGF is given by $\pi(t)$. Coverage growth occurs only when some constructs in $M - M(t_n)$ are executed. It should be noted that the distribution of execution probability over $M - M(t_n)$ is not $F(p)$ any more. In order to describe the coverage growth process after t_n, we need to derive the distribution of the execution probability of the constructs in $M - M(t_n)$. A similar argument can be made for the execution rate. First, let us consider the case where the testing time is discrete. Since the distribution of the execution probability of a construct in $M - M(t_n)$ is

$$f(p \,|\, T > t_n) = \frac{[1 - \Pr(T \le t_n \,|\, p)]\, f(p)}{1 - \pi(t_n)}, \tag{3.1}$$

the probability that a construct in $M - M(t_n)$ is executed by t additional testing time after t_n is obtained as

$$\pi(t \,|\, t_n) = \int_0^1 \Pr(T \le t \,|\, p)\, f(p \,|\, T > t_n)\, dp = \frac{\pi(t_n + t) - \pi(t_n)}{1 - \pi(t_n)}. \tag{3.2}$$

It can be verified that

$$\pi(t \,|\, t_n) = \frac{[\pi(t_i) - \pi(t_{i-1})]}{[1 - \pi(t_{i-1})]} \tag{3.3}$$

also holds for the continuous testing time.

Since a coverage growth process is repeatedly observed at different times, the observed values m_{t_i}'s and c_{t_i}'s are not independent. That is, the coverage growth behavior during $(t_{i-1}, t_i]$ depends on $m_{t_{i-1}}$ and the conditional probability that a construct not executed up to t_{i-1} is executed up to t_i. The increment of $|M(t)|$ during $(t_{i-1}, t_i]$ has the binomial distribution with parameters $|M| - m_{t_{i-1}}$ and

$\pi\,(t_i - t_{i-1}\,|\,t_i)$. Therefore, for given c_{t_i}'s and t_i's, the likelihood function and log likelihood function are obtained as

$$\begin{aligned} L &= \prod_{i=1}^{n} \binom{|M| - m_{t_{i-1}}}{x_{t_i}} \left[\frac{\pi\,(t_i) - \pi\,(t_{i-1})}{1 - \pi\,(t_{i-1})}\right]^{x_{t_i}} \left[\frac{1 - \pi\,(t_i)}{1 - \pi\,(t_{i-1})}\right]^{|M|-m_{t_i}} \\ &= \frac{|M|!}{x_{t_i}! \cdots x_{t_n}!\,(|M| - m_{t_n})!} \cdot \prod_{i=1}^{n} [\pi\,(t_i) - \pi\,(t_{i-1})]^{x_{t_i}} \cdot [1 - \pi\,(t_n)]^{|M|-m_{t_n}} \end{aligned}$$

and

$$\begin{aligned} \ln L &= K + \sum_{i=1}^{n} x_{t_i} \ln\,(\pi\,(t_i) - \pi\,(t_{i-1})) + (|M| - m_{t_n}) \ln\,(1 - \pi\,(t_n)) \\ &= K + |M| \cdot \left[\sum_{i=1}^{n} (c_{t_i} - c_{t_{i-1}}) \ln\,(\pi\,(t_i) - \pi\,(t_{i-1})) + (1 - c_{t_n}) \ln\,(1 - \pi\,(t_n))\right] \\ &= K + |M| \cdot L^*, \end{aligned} \tag{3.4}$$

where K is a constant independent of parameters to be estimated.

However, a slight modification is made for a practical reason. Generally, 100% coverage can rarely be achieved because of the presence of infeasible constructs and constructs with an extremely small execution probability or execution rate. The upper bound for each coverage metric is imposed on $\pi(t)$, that is,

$$\widetilde{\pi}(t) = c_{max}\,\pi(t), \tag{3.5}$$

where c_{max} is the maximum achievable coverage. It was shown that $\widetilde{\pi}(t)$ worked well for various real data sets. ML estimates of $\widetilde{\pi}(t)$ have been obtained by maximizing L^* after replacing $\pi(t)$ of Equation (3.4) with $\widetilde{\pi}(t)$.

The ML estimation procedure briefly reviewed above is to be corrected. First, note that c_{max} was introduced due to the infeasible constructs in M. Denote the set of all the feasible constructs in M by M_F. Coverage growth occurs only when constructs in M_F are executed during testing. This implies that $F(p)$ and $G(\lambda)$ are to be considered as distributions over M_F, not over M. Therefore, $\pi(t)$ is the probability that a construct in M_F, not in M, is executed up to t.

Second, coverages are measured relative to $|M|$, not $|M_F|$. Since $c_{t_i} = |M\,(t_i)\,|/|M|$, it is not reasonable to relate c_{t_i}'s to $\pi(t)$. c_{t_i}/c_{max}'s should be related to $\pi(t)$, where $c_{max} = |M_F|/|M|$. Practically, M and $|M|$ are well defined and known at the beginning of testing, but M_F is not. Thus c_{max} is to be estimated.

From the above discussion the increment of $|M(t)|$ during $(t_{i-1}, t_i]$, $M\,(t_i) - M\,(t_{i-1}) = |M|\,(c_{t_i} - c_{t_{i-1}})$, follows the binomial distribution with parameters $|M_F| - m_{t_{i-1}}$ and $\pi\,(t_i - t_{i-1}\,|\,t_i)$. The likelihood function is thus obtained as

$$\begin{aligned} L &= \prod_{i=1}^{n} \binom{|M_F| - m_{t_{i-1}}}{x_{t_i}} \left[\frac{\pi\,(t_i) - \pi\,(t_{i-1})}{1 - \pi\,(t_{i-1})}\right]^{x_{t_i}} \left[\frac{1 - \pi\,(t_i)}{1 - \pi\,(t_{i-1})}\right]^{|M_F|-m_{t_i}} \\ &= \frac{|M_F|!}{x_{t_1}! \cdots x_{t_n}!\,(|M_F| - m_{t_n})!} \cdot \prod_{i=1}^{n} [\pi\,(t_i) - \pi\,(t_{i-1})]^{x_{t_i}} \cdot [1 - \pi\,(t_n)]^{|M_F|-m_{t_n}}\,, \end{aligned} \tag{3.6}$$

where $x_{t_i} = |M| \, (c_{t_i} - c_{t_{i-1}})$, $|M_F| - m_{t_{i-1}} = |M| \, (c_{max} - c_{t_{i-1}})$ and $|M_F| - m_{t_n} = |M| \, (c_{max} - c_{t_n})$. ML estimates of c_{max} and the parameters of $\pi(t)$ are to be computed by maximizing the likelihood function (3.6). However, maximization of the likelihood function (3.6) is numerically very complex.

4. Minimum Sum of Squares of Prediction Error Estimation of CGF

As mentioned in the previous section, M is well defined at the beginning of testing. If $|M|$ is available, $\pi(t)$'s can be fitted to the observed coverage values c_{t_i}'s by ML method. Most studies on the coverage growth process report the values of c_{t_i} and t_i but $|M|$. ML estimates cannot be computed when $|M|$ is not available. Alternative estimation methods are necessary for such situations. In this section we define the prediction error and propose the estimation method optimizing the sum of squares of the standardized prediction error.

Execution of the software system without the coverage increase does not help tester detect the remaining faults. This results in the overestimation of reliability. In order for testers to decide whether to stop testing, the coverage expected from additional testing should be predicted. Accurate prediction of the future coverage is therefore an important problem. Suppose that testing was performed up to t_{i-1} and the coverage was observed as $c_{t_{i-1}}$. Since $\pi\,(t_i - t_{i-1}|t_{i-1})$ is the probability that a construct in $M_F - M\,(t_{i-1})$ is executed up tp t_i, the coverage at t_i is predicted as

$$c_{t_{i-1}} + (c_{max} - c_{t_{i-1}})\,\pi\,(t_i - t_{i-1}\,|\,t_{i-1})\,. \tag{4.1}$$

Therefore, the corresponding prediction error is obtained as

$$c_{t_i} - c_{t_{i-1}} - (c_{max} - c_{t_{i-1}})\,\pi\,(t_i - t_{i-1}\,|\,t_{i-1})\,. \tag{4.2}$$

Since the distribution of $M\,(t_i) - M\,(t_{i-1}) = |M|\,(c_{t_i} - c_{t_{i-1}})$ is the binomial distribution with parameters $M_F - M\,(t_{i-1})$ and $\pi\,(t_i - t_{i-1}\,|\,t_{i-1})$, the sum of squares of the standardized prediction error divided by $|M|$ (SSPE)

$$\sum_{i=1}^{n} \frac{[c_{t_i} - c_{t_{i-1}} - (c_{max} - c_{t_{i-1}})\,\pi\,(t_i - t_{i-1}\,|\,t_{i-1})]^2}{(c_{max} - c_{t_{i-1}})\,\pi\,(t_i - t_{i-1}\,|\,t_{i-1})\,[1 - \pi\,(t_i - t_{i-1}\,|\,t_{i-1})]} \tag{4.3}$$

is a reasonable measure for the predictive ability of CGF $\widetilde{\pi}(t)$. An alternative to ML estimation we suggest minimization of SSPE , which is especially useful when $|M|$ is not available.

5. Numerical Examples

The first data set, DS1, was collected by Pasquini *et al.* (1996) from a configuration software for an array of antennas developed by European Space Agency. It consists of 29 observations on the number of detected faults, the number of executed test cases and 4 coverages. The 4 coverages are block, branch, c-use, and p-use coverages. For the sake of brevity, we fit the three CGFs to block and p-use coverages of DS1. Maximum likelihood estimates of CGFs are summarized in Table 1. The fitted CGFs are plotted in Figure 1.The fitted $\widetilde{\pi}^{d}_{beta}(t)$ and $\widetilde{\pi}^{c}_{gamma}(t)$ are so close that they are indistinguishable in Figure 1. All the three CGFs work well for DS2. The value of L^* suggest as the best model $\widetilde{\pi}^{c}_{gamma}(t)$ for block coverage and $\widetilde{\pi}^{c}_{lognormal}(t)$ for the p-use coverage.

The second data set, DS2, was collected, as a simulation of testing, by Gokhale and Mullen (2004) with SHARPE(Symbolic Hierarchical Automated Reliability and Performance Evaluator) that solves stochastic models of reliability. SHARPE contains 35,081 lines of C code and has a total of 373

Table 1: Maximum likelihood estimates of CGFs fitted to 2 coverages of DS1

	parameter	$\widetilde{\pi}^{d}_{beta}(t)$	$\widetilde{\pi}^{c}_{gamma}(t)$	$\widetilde{\pi}^{c}_{lognormal}(t)$
block coverage	c_{max}	0.8239	0.8205	0.8239
	$\alpha\ (\mu)$	0.5917	0.6596	−1.0189
	$\beta\ (\sigma)$	0.9186	1.0839	2.3032
	SSPE	0.2520	0.2215	0.2566
p-use coverage	c_{max}	0.6727	0.6723	0.6700
	$\alpha\ (\mu)$	0.7097	0.8385	−1.5880
	$\beta\ (\sigma)$	1.7664	0.4536	1.9965
	SSPE	0.4145	0.3387	0.3452

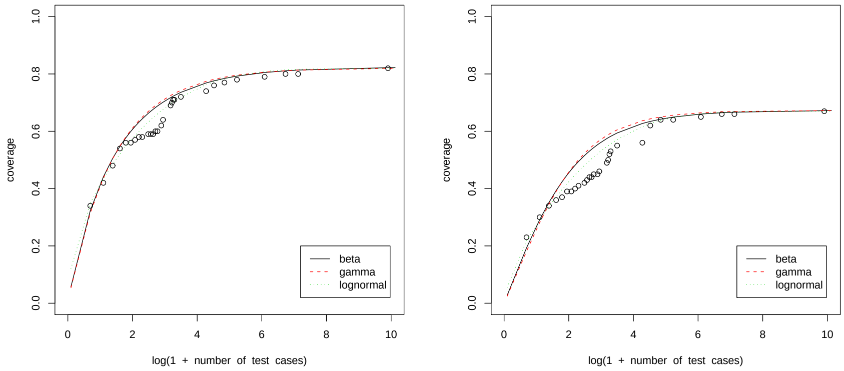

Figure 1: *CGFs fitted to DS1: (left) block coverage, (right) p-use coverage*

Table 2: Maximum likelihood estimates of CGFs for DS2

parameter	$\widetilde{\pi}^{d}_{beta}(t)$	$\widetilde{\pi}^{c}_{gamma}(t)$	$\widetilde{\pi}^{c}_{lognormal}(t)$
c_{max}	1.0000	1.0000	0.9961
$\alpha\ (\mu)$	0.4662	0.4660	−3.2271
$\beta\ (\sigma)$	3.6027	0.2991	2.2470
SSPE	0.1033	0.1031	0.0350

functions. DS2 consists of block coverage values measured by applying 10 different sequences of 735 test cases; in addition, an average of 10 sequences is also reported. In this paper we fit the three CGFs to the average block coverage of DS3. Maximum likelihood estimates are tabulated in Table 2. The fitted CGFs are shown in Figure 2. It is apparent that $\widetilde{\pi}^{c}_{lognormal}(t)$ is the best model for DS2.

6. Concluding Remarks

In this paper we proposed a class of continuous-time CGFs and showed that it is identical to the class of discrete-time CGFs suggest by Park *et al.* (2007). Specifically, three CGFs of the class, $\widetilde{\pi}^{d}_{beta}(t)$, $\widetilde{\pi}^{c}_{gamma}(t)$ and $\widetilde{\pi}^{c}_{lognormal}(t)$, have been investigated by applying them to real data sets. In general, coverage is assumed to grow exponentially. That is, coverage grows fast in the early phase of testing and the coverage growth rate reduces as fast as the testing progresses. The data sets analyzed in the previous section comply with this assumption. Fujiwara *et al.* (2005) and Fujiwara and Yamada (2002) reported 4 data sets not supporting this assumption, These data sets, referred to as DS4-DS7 respectively, are plotted in Figure 1. Although ti is not described in detail in this paper, $\widetilde{\pi}^{d}_{beta}(t)$, $\widetilde{\pi}^{c}_{gamma}(t)$ and $\widetilde{\pi}^{c}_{lognormal}(t)$ does not work well for these data sets. That is, the CGFs proposed in this

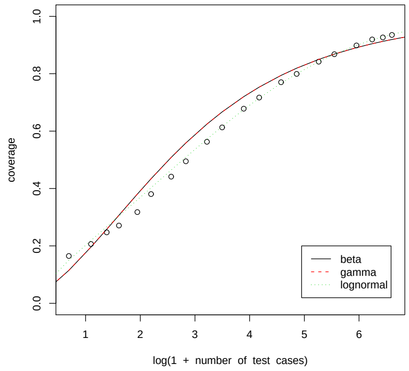

Figure 2: *CGFs fitted to DS2.*

paper are not appropriate for the data sets with non-exponential growth. It is therefore necessary to develop new CGFs for the coverage growth phenomenon with non-exponential growth. Our future research will be directed to this problem.

References

Crespo, A. N., Pasquini, A., Jino, M. and Maldonado, J. C. (2008). A binomial software reliability model based on coverage of structural testing criteria, *Empirical Software Engineering,* **13**, 185–209.

Crespo, A. N., Pasquini, A., Jino, M. and Maldonado, J. C. (2009). Applying code coverage approach to an infinite failure software reliability model, In *Proceedings of 23rd Brazilian Symposium on Software Reliability Engineering,* 216–226.

Fujiwara, T., Inoue, S. and Yamada, S. (2005). A software reliability growth model with module composition, In *Proceedings of 11th International Society of Science and Applied Technologies*, 266–270.

Fujiwara, T. and Yamada, S. (2002). Co coverage-measure and testing-domain metrics based on a software reliability growth model, *International Journal of Reliability, Quality and Safety Engineering*, **9**, 329–340.

Gokhale, S. S. and Mullen, R. E. (2004). From test count to code coverage using the lognormal failure rate, In *Proceedings of 15th IEEE International Symposium on Software Reliability Engineering*, 295–305.

Gokhale, S. S., Philip, T., Marinos, P. N. and Trivedi, K. S. (1996). Unification of finite failure non-homogeneous Poisson process models through test coverage, *Proceedings of 7th IEEE International Symposium on Software Reliability Engineering,* 299–307.

Lyu, M. R. (1996). *Handbook of Software Reliability Engineering,* McGraw-Hill, New York.

Malaiya, Y. K., Li, M. N., Bieman, J. M. and Karcich, R. (2002). Software reliability growth and test coverage, *IEEE Transactions on Reliability,* **51**, 420–426.

Musa, J. D. (1999). *Software Reliability Engineering: More Reliable Faster Development and Testing,* McGraw-Hill, New York.

Musa, J. D., Iannino, A. and Okumoto, K. (1987). *Software Reliability: Measurement, Prediction, Application,* McGraw-Hill, New York.

Park, J.-Y. and Fujiwara, T. (2006). Coverage growth functions for software reliability modeling, *Proceedings of 2nd Asian International Workshop on Advanced Reliability Modeling*, 435–442.

Park, J.-Y., Lee, G. and Park, J. H. (2007). A class of discrete time coverage growth functions for software reliability engineering, *Communications of the Korean Statistical Society,* **14**, 497–506.

Park, J.-Y., Lee, G. and Park, J. H. (2008a). A class of coverage growth functions and its practical application, *Journal of the Korean Statistical Society,* **37**, 241–247.

Park, J.-Y., Lee, G. and Park, J. H. (2008b). A general coverage-based NHPP SRGM framework, *Communications of the Korean Statistical Society,* **15**, 875–881.

Pasquini, A., Crespo, A. N. and Martella, P. (1996). Sensitivity of reliability-growth models to operational profile errors vs testing accuracy, *IEEE Transactions on Reliability,* **45**, 531–540.

Pham, H. and Zhang, X. (2003). NHPP software reliability and cost models with testing coverage, *European Journal of Operational Research,* **145**, 443–454.

Received May 2011; Accepted August 2011