

離散時間ソフトウェア信頼性モデルに基づいた最適リリース時刻の 区間推定に関する一考察

鳥取大学・大学院工学研究科 井上 真二, 山田 茂

Shinji Inoue and Shigeru Yamada

Graduate School of Engineering,

Tottori University

1 はじめに

ソフトウェア開発プロセスの最終工程であるテスト工程から運用段階へ移行する時刻（ソフトウェアの出荷時刻）を推定することは、開発計画を立てる上で有用な情報である。通常、テスト工程に費やすテストコストと運用段階におけるソフトウェアの保守コストはトレードオフの関係にあることが知られており [1]、このような観点から最適リリース（出荷）時期を推定することは、開発管理面からの興味ある問題の 1 つである。これは一般的に、コスト最小化問題として捉えられ、広くソフトウェア最適リリース問題として知られている。当問題では、従来より、その目的関数を構成するソフトウェア信頼度成長モデル [2,3] に含まれるパラメータが、点推定によって推定されるため、最適リリース時刻も点推定に基づいて推定されていた。

しかしながら、実際のソフトウェア開発プロジェクトでは、信頼度成長モデルのパラメータの点推定に十分なフォールトデータが得られない場合が多く、点推定に基づいた信頼性評価結果を含め、それに基づいて推定された最適リリース時刻に対する信ぴょう性が薄いという現実的問題がある。一般的にこの場合、区間推定手法を適用すべきであり、点推定よりも不確実性を考慮した情報量が比較的多いため、信頼性評価や開発計画を立てる際にも有用であることが考えられる。しかしながら、特に、広く実用に供されている非同次ポアソン過程 (NHPP) に従った信頼度成長モデルを適用した場合、近似的なアプローチを用いても複雑な解析的アプローチを用いる必要があると共に、現実を反映した有効な推定結果が得られない場合も少なくないことが知られている [4]。本研究では、その 1 つの解決方法として、離散化 NHPP モデル [5] を用いたソフトウェア最適リリース時刻の区間推定のためのノンパラメトリックブートストラップ法について議論する共に、テストにおいて達成された信頼度目標も同時に考慮した最適リリース時刻の区間推定手法について議論する。また、3 つのブートストラップ信頼区間を取り上げ、実測データを用いた提案手法の適用例を示す。

2 離散化指数形 SRGM

本研究では、離散化 NHPP モデル [5] の中でも最もシンプルな構造をもつ離散化指数形 SRGM を取り上げて議論する。まず、離散化 NHPP モデルとは、テスト開始後 n 期目までに発見された総フォールト数を表す離散時間計数過程 $\{N_n; n = 0, 1, 2, \dots\}$ が、

$$\Pr\{N_n = x \mid N_0 = 0\} = \frac{\{\Lambda_n\}^x}{x!} \exp[-\Lambda_n] \quad (x = 0, 1, \dots), \quad (1)$$

に従うものと仮定する SRGM である。式 (1) において、 Λ_n は離散時間 NHPP の平均値関数と呼ばれ、テスト開始後 n 期目までに発見された総期待フォールト数を表す。式 (1) より、離散時間計数過程 $\{N_n; n = 0, 1, 2, \dots\}$ はその平均値関数 Λ_n によって具体的に特徴付けられることがわかる。

離散化指数形 SRGM は、微分方程式で記述される従来の連続時間指数形 SRGM [2] の基本的仮定に基づいて導出される $\Lambda_n \equiv H_n$ に関する差分方程式：

$$H_{n+1} - H_n = \delta\beta(\omega - H_n) \quad (\omega > 0, \beta > 0), \quad (2)$$

によってソフトウェア故障発生現象の平均的挙動が特徴づけられる。ここで、 H_n はテスト開始後 n 期目までに発見された総期待フォールト数、 ω はテスト開始前にソフトウェア内に潜在する総フォールト数の期待値、 β は 1 個当りのフォールト発見率を表す。また、式 (2) は厳密解 H_n をもち、

$$H_n = \omega[1 - (1 - \delta\beta)^n], \quad (3)$$

のように導出される。なお、 $\delta \rightarrow 0$ のとき、式 (2) と式 (3) はそれぞれ、従来の連続時間指数形 SRGM [2] の基本的仮定を表現する微分方程式およびその厳密解へ収束する。

離散化 NHPP モデルに含まれるパラメータは、差分方程式から導出される回帰式に基づいて最小二乗法を用いて推定される。いま、フォールト発見数データ： $(n, y_n) (n = 1, 2, \dots, N)$ が観測されたとき、離散化指数形 SRGM を適用した場合、式 (2) の差分方程式から単回帰式：

$$C_n = \alpha_0 + \alpha_1 D_n, \quad (4)$$

を得る。ここで、 y_n はテスト開始後 n 期目までに発見される総フォールト数を表し、

$$\begin{cases} C_n = H_{n+1} - H_n \equiv y_{n+1} - y_n, \\ D_n = H_n \equiv y_n, \\ \alpha_0 = \delta\omega\beta, \\ \alpha_1 = -\delta\beta, \end{cases} \quad (5)$$

である。式 (5) より、離散化指数形 SRGM のパラメータは、それぞれ、推定された回帰パラメータ $\hat{\alpha}_0$ および $\hat{\alpha}_1$ から、

$$\begin{cases} \hat{\omega} = -\hat{\alpha}_0/\hat{\alpha}_1, \\ \hat{\delta}\beta = -\hat{\alpha}_1, \end{cases} \quad (6)$$

のように導出される。

また、式 (1) と式 (3) から、定量的なソフトウェア信頼性評価に有効な種々の信頼性評価尺度を導出することができる。その代表的な尺度として、離散時間ソフトウェア信頼度関数 $R(n, h)$ は、テストが開始後 n 期目まで進行しているとき、その後の時間区間 $(n, n+h] (h = 1, 2, \dots)$ においてソフトウェア故障がしない確率として定義され、

$$\begin{aligned} R(n, h) &\equiv \Pr\{N_{n+h} - N_n = 0 \mid N_n = x\} \\ &= \exp[-\{\Lambda_{n+h} - \Lambda_n\}] \\ &= \exp[-H_h(1 - \delta\beta)^n], \end{aligned} \quad (7)$$

のように導出される。

3 ソフトウェアの最適リリース問題

テスト工程および運用段階においてフォールト修正作業に必要なコストは、総期待ソフトウェアコストと呼ばれ、離散化指数形 SRGM を適用した場合、次のように与えられる：

$$C_Z = (c_1 - c_2)H_Z + c_2H_{Z_{LC}} + c_3Z. \quad (8)$$

ここで、 c_1 および c_2 はそれぞれテスト工程および運用段階において発見されたフォールト 1 個当りの修正コスト、 c_3 は単位時間当りのテストコスト、 Z はテスト終了時刻、および Z_{LC} はソフトウェアのライフサイクルをそれぞれ表す。式 (8) から、コスト評価基準の下では、その必要条件としての一階条件より、最適リリース時刻 Z_C^* は、

$$Z_C^* = \frac{\log c_3 - \log \{(c_2 - c_1)\omega\delta\beta\}}{\log(1 - \delta\beta)}, \quad (9)$$

のように求められる。また、ソフトウェア信頼度達成目標を R_0 を考慮したとき、最適リリース時刻 Z_R^*

$$Z_R^* = \frac{\log(-\log R_0) - \log[\omega\{1 - (1 - \delta\beta)^h\}]}{\log(1 - \delta\beta)}, \quad (10)$$

と求められる。

4 ブートストラップ法の適用

離散化 NHPP モデルに基づいたソフトウェア信頼性解析へのブートストラップ法の適用について、前述した離散化指数形 SRGM を取り上げ議論する。離散化指数形 SRGM を適用した場合におけるブートストラップ推定手順を以下に示す：

- S1 観測データ $(n, y_n) (n = 1, 2, \dots, N)$ に基づいて回帰パラメータ α_0 と α_1 を推定し、それら推定結果を、それぞれ、 $\hat{\alpha}_{0(0)}$ と $\hat{\alpha}_{1(0)}$ と表記する。
- S2 各観測時点における残差 \hat{d}_i を、 $\hat{d}_i = C_i - (\hat{\alpha}_{0(0)} + \hat{\alpha}_{1(0)} D_i) (i = 1, 2, \dots, N-1)$ に基づいて算出する。
- S3 残差 $\{\hat{d}_1, \hat{d}_2, \dots, \hat{d}_{N-1}\}$ がそれぞれ独立かつ同一の分布に従うものと仮定すると共に、これらを昇順に並べ、経験分布関数 \hat{F} を生成する。
- S4 ブートストラップ反復回数の上限值 B を設定する。また、 $b (b = 1, 2, \dots, B)$ を反復回数とする。
- S5 経験分布関数 \hat{F} から重複を許しながら残差に関するブートストラップ標本（ブートストラップ残差）、 $\hat{\mathbf{d}}_{(b)}^* = \{\hat{d}_{b,1}^*, \hat{d}_{b,2}^*, \dots, \hat{d}_{b,N-1}^*\}$ を生成する。
- S6 ブートストラップ残差から、 $C_{b,i}^* = \hat{\alpha}_{0(0)} + \hat{\alpha}_{1(0)} D_i + \hat{d}_{b,i}^*$ に基づいて回帰分析に必要なブートストラップ標本 $\mathbf{z}_{(b)}^* = \{(y_1, C_{b,1}^*), (y_2, C_{b,2}^*), \dots, (y_{N-1}, C_{b,N-1}^*)\}$ を生成する。
- S7 ブートストラップ標本 $\mathbf{z}_{(b)}^*$ から $\alpha_{0(b)}^*$ と $\alpha_{1(b)}^*$ を推定する。
- S8 $\hat{\alpha}_{0(b)}^*$ と $\hat{\alpha}_{1(b)}^*$ を用いて、 $\hat{\omega}_{(b)}^* = -\hat{\alpha}_{0(b)}^* / \hat{\alpha}_{1(b)}^*$, $\hat{\delta}\beta_{(b)}^* = -\hat{\alpha}_{1(b)}^*$ を求める。
- S9 各種ソフトウェア信頼性評価尺度を推定する。
- S10 $b < B$ ならば $b = b + 1$ として、S5 へ移動する。
- S11 $\hat{\omega}$, $\hat{\delta}\beta$, および各種ソフトウェア信頼性評価尺度に関する B 組のデータを得る。

5 ブートストラップ信頼区間

今回議論した手法によって得られたブートストラップ分布に基づいて、モデルパラメータや各種ソフトウェア信頼性評価尺度の区間推定を行うことができる。なお、ここでは、 θ を興味のある一次元パラメータ、 $\hat{\theta}$ を θ の推定量として、以下の3つのブートストラップ信頼区間を取り上げて議論する。

まず、基本ブートストラップ信頼区間 [6] は、 $\hat{\theta} - \theta$ の分布をブートストラップ統計量 $\hat{\theta}^* - \hat{\theta}$ の分布で近似する手法である。すなわち、 $\hat{\theta} - \theta$ の分布における α および $(1 - \alpha)$ 分位点 v_α および $v_{1-\alpha}$ を、それぞれ $\hat{\theta}_{[B\alpha]}^* - \hat{\theta}$ および $\hat{\theta}_{[B(1-\alpha)]}^* - \hat{\theta}$ で近似すると、

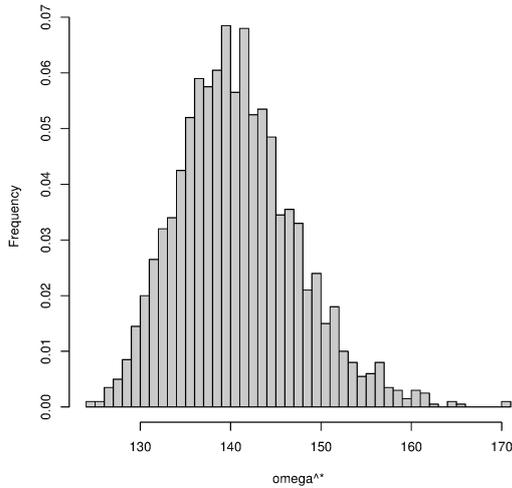
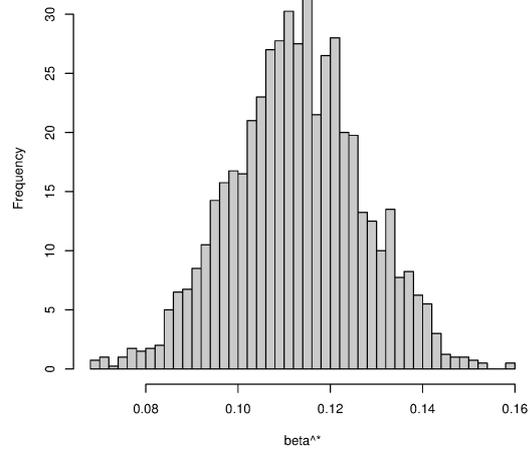
$$\begin{aligned} 1 - 2\alpha &= \Pr\{v_\alpha \leq \hat{\theta} - \theta \leq v_{1-\alpha}\} \\ &= \Pr\{\hat{\theta} - v_{1-\alpha} \leq \theta \leq \hat{\theta} - v_\alpha\} \\ &= \Pr\{2\hat{\theta} - \hat{\theta}_{[B(1-\alpha)]}^* \leq \theta \leq 2\hat{\theta} - \hat{\theta}_{[B\alpha]}^*\}, \end{aligned}$$

を得る。これより、 $100(1 - 2\alpha)\%$ 信頼区間は、

$$[2\hat{\theta} - \hat{\theta}_{[B(1-\alpha)]}^*, \quad 2\hat{\theta} - \hat{\theta}_{[B\alpha]}^*], \quad (11)$$

のように求められる。ここで、 $\hat{\theta}_{[B\alpha]}^*$ は、昇順に並べた B 個の $\hat{\theta}_{(b)}^*$ の中で $B\alpha$ 番目の値を示す。次に、標準正規ブートストラップ信頼区間 [6] は、 $\hat{\theta} - \theta$ の分布を $N(0, \text{SD}[\hat{\theta}])$ で正規近似して得られる信頼区間であり、

$$1 - 2\alpha = \Pr\left\{z_\alpha \leq \frac{\hat{\theta}^* - \hat{\theta}}{\text{SD}[\hat{\theta}]} \leq z_{1-\alpha}\right\},$$

図 1: $\hat{\omega}$ に対するブートストラップ分布.図 2: $\hat{\beta}$ に対するブートストラップ分布.

として表される. したがって, $100(1-2\alpha)\%$ 信頼区間は,

$$[\hat{\theta} - w_{1-\alpha}SD[\hat{\theta}], \hat{\theta} - w_{\alpha}SD[\hat{\theta}]], \quad (12)$$

として与えられる. ここで, $w_{\alpha} = \Phi^{-1}(\alpha)$ であり, $\Phi^{-1}(\cdot)$ は標準正規分布の分布関数の逆関数を表す. パーセントイル信頼区間 [6] は, $\hat{\theta}^*$ の反復値: $\hat{\theta}_{(1)}^*, \hat{\theta}_{(2)}^*, \dots, \hat{\theta}_{(B)}^*$ に基づいて得られる信頼区間であり, $100(1-2\alpha)\%$ 信頼区間は,

$$[\hat{\theta}_{[B\alpha]}^*, \hat{\theta}_{[B(1-\alpha)]}^*], \quad (13)$$

として求められる.

6 適用例

実際の観測されたフォールト発見数データ $(n, y_n)(n = 1, 2, \dots, 25; y_{25} = 136)$ [2] を用いて, 離散化指数形ソフトウェア信頼度成長モデルに基づいたブートストラップ標本の生成とブートストラップ信頼区間に基づいた最適リリース時刻の区間推定に関する適用例を示す. まず, 式 (4) および (5) から回帰分析に基づいて, $\hat{\alpha}_{0(0)} = 15.8586$ および $\hat{\alpha}_{1(0)} = -0.1133109$ を得る. さらに, 今回示した手順に基づいてブートストラップ標本 $\{z_{(1)}^*, z_{(2)}^*, \dots, z_{(2000)}^*\}$ が得られ, モデルに含まれる各パラメータに関するブートストラップ分布を得る. 図 1 および図 2 に, パラメータ $\hat{\omega}$ および $\hat{\beta}$ に対して得られたブートストラップ分布をそれぞれ示す.

得られた各パラメータのブートストラップ分布に基づいて, コストのみを考慮した場合およびコストと信頼度目標を同時に考慮した場合における最適リリース時刻のブートストラップ推定量を求める. ここで, 式 (9) および式 (10) より, それぞれの最適リリース時刻に関する反復回数 b 回目のブートストラップ推定量は, 次のように得られる:

$$Z_{C^{(b)}}^* = \frac{\log c_3 - \log \left\{ (c_2 - c_1) \hat{\omega}_{(b)}^* \delta \hat{\beta}_{(b)}^* \right\}}{\log(1 - \delta \hat{\beta}_{(b)}^*)}, \quad (14)$$

$$Z_{R^{(b)}}^* = \frac{\log(-\log R_0) - \log \left[\hat{\omega}_{(b)}^* \left\{ 1 - (1 - \delta \hat{\beta}_{(b)}^*)^h \right\} \right]}{\log(1 - \delta \hat{\beta}_{(b)}^*)}. \quad (15)$$

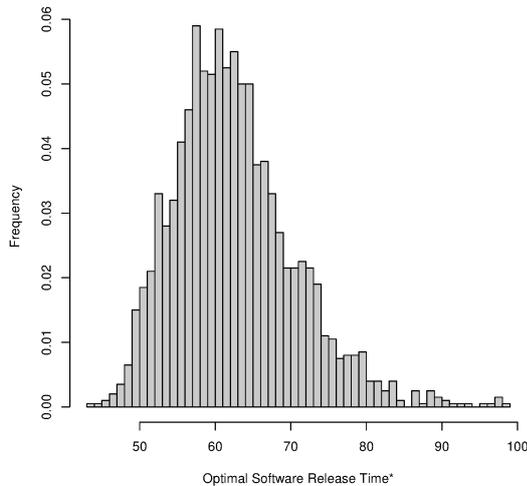


図 3: $c_1 = 1$, $c_2 = 2$, および $c_3 = 0.01$ における \hat{Z}_C^* のブートストラップ分布.

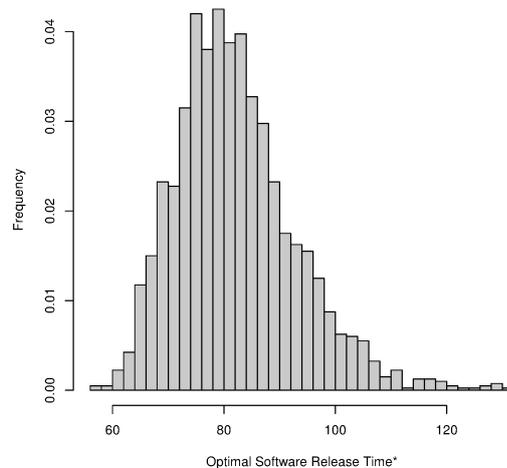


図 4: $c_1 = 1$, $c_2 = 2$, $c_3 = 0.01$, $h = 1.0$, および $R_0 = 0.9990$ における \hat{Z}_R^* のブートストラップ分布.

表 1: コスト評価基準による最適リリース時刻 \hat{Z}_C^* の区間推定結果 ($\alpha = 0.05$).

c_1	c_2	c_3	基本 BCI		標準正規 BCI		パーセンタイル BCI	
			下限値	上限値	下限値	上限値	下限値	上限値
1	2	0.001	54.79084	95.88359	59.27501	101.5664	64.95781	106.0506
1	4	0.001	60.75098	106.8938	65.81631	113.2956	72.21811	118.3609
1	8	0.001	65.3477	115.3857	70.86116	122.3417	77.81719	127.8552
1	16	0.001	69.48243	123.0241	75.39893	130.4787	82.85354	136.3952
1	2	0.01	42.16082	72.81781	45.56433	76.98391	49.73043	80.38742
1	4	0.01	48.19938	83.82055	52.10614	88.71257	56.99817	92.61934
1	8	0.01	52.85582	92.30993	57.15127	97.75844	62.59978	102.0539
1	16	0.01	56.99055	99.9468	61.68923	105.8953	67.63768	110.5939

(BCI: ブートストラップ信頼区間)

図 3 および図 4 に、 $c_1 = 1$, $c_2 = 2$, $c_3 = 0.01$ におけるコスト評価基準による最適リリース時刻 \hat{Z}_C^* , およびその時に $h = 1.0$, $R_0 = 0.9990$ とした場合の信頼度目標も同時に考慮した最適リリース時刻 \hat{Z}_R^* のブートストラップ分布をそれぞれ示す。また、表 1 に、前述した 3 種類のブートストラップ信頼区間に基づいたコスト評価基準による最適リリース時刻 \hat{Z}_C^* の区間推定結果 (有意水準 $\alpha = 5\%$) を示す。さらに、表 2 に、 \hat{Z}_R^* に関して運用期間 h および目標信頼度 R_0 と最適リリース時刻の区間推定結果 (有意水準 $\alpha = 5\%$) との関係に関する感度分析結果を示す。表 1 および表 2 より、運用段階におけるフォールト 1 個当りの修正コストが相対的に上昇すればするほど、最適リリース時刻 \hat{Z}_C^* は遅くなる傾向があることがわかる。また、 \hat{Z}_R^* においては、リリース後の運用期間が長くなったり、目標信頼度が高くなる程、最適リリース時刻は遅くなる傾向にあることがわかる。一方、信頼区間の適用については、信頼区間の種類によって区間推定結果も異なっているが、各最適リリース時刻の確率分布の形状を考慮しつつ、特に、それらが非対称な分布であればパーセンタイルブートストラップ信頼区間を適用する方が適当であろう。ただし、これらの信頼区間は、分布のバイアス等を陽に考慮したものではないため、今後は、新たなブートストラップ信頼区間も導入しながら、どのブートストラップ信頼区間を適用すべきか議論する必要がある。

表 2: コストおよび信頼度両評価基準による最適リリース時刻 \hat{Z}_R^* の区間推定結果 ($\alpha = 0.05$) .

h	R_0	基本 BCI		標準正規 Normal BCI		パーセントイル BCI	
		Lower	Upper	Lower	Upper	Lower	Upper
1.0	0.9990	54.78813	95.87858	59.27203	101.561	64.9545	106.0449
2.0	0.9990	58.06745	102.3362	62.91802	108.4725	69.05432	113.3231
3.0	0.9990	59.79862	105.9185	64.86361	112.32	71.26515	117.385
4.0	0.9990	60.9033	108.3277	66.12038	114.9175	72.71013	120.1346
3.0	0.9995	63.56041	112.8679	68.99214	119.723	75.84723	125.1548
3.0	0.9999	72.29294	129.0003	78.57585	136.9084	86.48398	143.1913

(BCI: ブートストラップ信頼区間)

7 おわりに

本研究では、ソフトウェア信頼度成長モデルに基づいたソフトウェアの最適リリース時刻の区間推定手法として、コスト評価基準およびコストと信頼度目標の両評価基準を適用した場合について、離散化 NHPP モデルを用いたノンパラメトリックブートストラップ法に関する議論を行った。また、実測データを適用した提案手法の適用例として、基本、標準正規、パーセントイルの3つのブートストラップ信頼区間を取り上げ、ソフトウェアの最適リリース時刻の区間推定に関する適用例を示した。今回の手法の特長としては、離散化 NHPP モデルを利用することによって回帰分析に対するノンパラメトリックブートストラップ法が適用でき、NHPP に従うソフトウェア故障発生時間データを生成することなくブートストラップに基づいた区間推定が比較的容易に行える点が挙げられる。しかしながら、その一方で、ブートストラップ標本を生成する際に NHPP の性質をほとんど考慮していないため、理論上では問題が残る手法である。今後は、NHPP の性質に基づいたブートストラップ標本の生成を行い、今回の手法に基づいた結果と照らし合わせながら、今回提案した手法の有用性について検証する必要がある。また、ブートストラップ信頼区間についても、より適切な信頼区間を導入しながら、最終的に有用なブートストラップ信頼区間を選定する必要がある。

謝辞

本研究の一部は、科学研究費補助金 基盤研究 (C) (課題番号 25330081 および 25350445)、および電気通信普及財団研究調査助成の援助を受けたことを付記する。

参考文献

- [1] S. Yamada and S. Osaki, "Cost-reliability optimal release policies for software systems," *IEEE Transactions on Reliability*, Vol. R-34, No. 5, pp. 422–424, 1985.
- [2] J.D. Musa, D. Iannio, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*. McGraw-Hill, New York, 1987.
- [3] S. Yamada, *Software Reliability Modeling — Fundamentals and Applications —*, Springer-Verlag, Tokyo, 2013.
- [4] T. Kaneishi and T. Dohi, "Parametric bootstrapping for assessing software reliability measures," *Proceedings of the 17th IEEE Pacific Rim International Symposium on Dependable Computing*, 2010, pp. 1–9.
- [5] S. Inoue and S. Yamada, "Discrete software reliability assessment with discretized NHPP models," *Computers & Mathematics with Applications: An International Journal*, Vol. 51, Issue 2, pp. 161–170, 2006.
- [6] M.L. Rizzo, *Statistical Computing with R*, Chapman and Hall/CRC, Boca Raton, 2008.