

# 短納期開発におけるソフトウェア信頼性評価技術とその実現のための評価シミュレータの開発（継続）

研究代表者 井上真二 鳥取大学大学院工学研究科社会経営工学講座

## 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[Yamada and Osaki (1985)], 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[Musa et al. (1987), Yamada (2013)]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[Kaneishi and Dohi (2010)]. 本研究では、その1つの解決方法として、離散化 NHPP モデル[Inoue and Yamada (2006)]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 計算統計手法に基づくソフトウェア最適リリース時刻の区間推定

### 2-1 離散化指数形ソフトウェア信頼度成長モデル

本研究では、ソフトウェア信頼度成長モデルの中でも広く実用に供されている非同次ポアソン過程モデルを取り上げ、テスト工程で観測されるソフトウェア信頼度成長過程を記述する。特に、適合性、予測妥当性、およびモデル構造上観点から、今回は、離散化指数形ソフトウェア信頼度成長モデルを採用した。離散化指数形ソフトウェア信頼度成長モデルは、離散型非同次ポアソン過程の平均値関数（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）が、 $H_n = \omega[1 - (1 - \delta b)^n]$  ( $\omega > 0, b > 0$ ) に従うモデルである。ここで、 $\omega$  はテスト開始前にソフトウェア内に潜在する総期待フォールト数、 $\delta$  は一定時間間隔、および  $b$  は 1 個当りのフォールト発見率を表す。さらに、非同次ポアソン過程の性質から、総期待残存フォールト数やソフトウェア信頼度関数などの定量的なソフトウェア信頼性評価に有用な尺度（例えば、ソフトウェア信頼性評価尺度）を導出できる。ソフトウェア信頼度関数  $R(n, h)$  は、 $n$  期目までテストが実施された条件の下で、その後のテスト時間区間  $(n, h)$  ( $h=1, 2, \dots$ ) においてソフトウェア故障が発生しない条件付確率として定義される。モデルに含まれるパラメータの推定は、上記の離散化指数形ソフトウェア信頼度成長モデルによって記述されるソフトウェア故障発見数の平均的挙動に対する基本的仮定を表した差分方程式から回帰式が得られるため、通常、回帰分析を行うことでパラメータの推定値をそれぞれ得ることができる。

### 2-2 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、離散化指数形ソフトウェア信頼度成長モデルを適用した場合、次のように与えられる：

$$C_Z = (c_1 - c_2)H_Z + c_2H_{ZLC} + c_3Z. \quad (1)$$

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

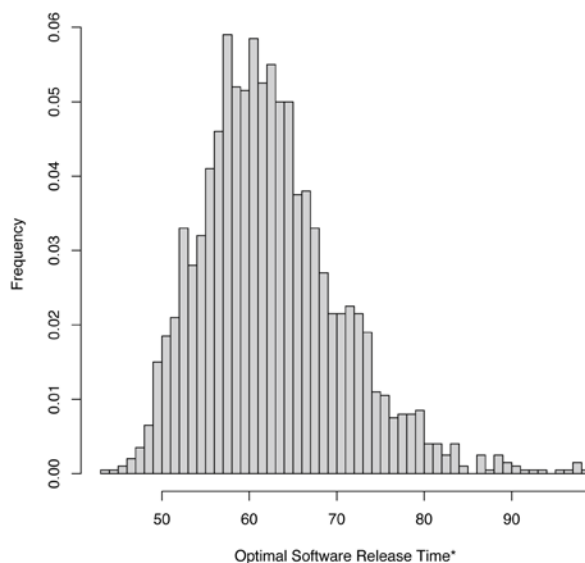


図 1:  $c_1 = 1, c_2 = 2$ , および  $c_3 = 0.01$  における  $Z_C^*$  のブートストラップ分布.

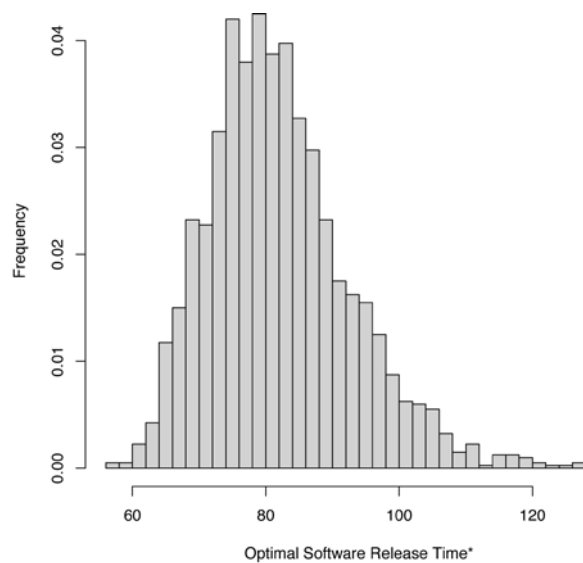


図 4:  $c_1 = 1, c_2 = 2, c_3 = 0.01, h = 1.0$  および  $R_0 = 0.9990$  における  $Z_R^*$  のブートストラップ分布.

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  $ZLC$  はソフトウェアのライフサイクルをそれぞれ表す. 式(1)から, コスト評価基準の下では, その必要条件としての 1 階条件より, 最適リリース時刻  $Z_C^*$  は,

$$Z_C^* = \frac{\log c_3 - \log\{(c_2 - c_1)\omega\delta b\}}{\log(1 - \delta b)}, \quad (2)$$

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

$$Z_R^* = \frac{\log(-\log R_0) - \log[\omega\{1 - (1 - \delta b)^h\}]}{\log(1 - \delta b)}, \quad (3)$$

と求められる.

### 2-3 ブートストラップ法を用いた最適リリース時刻の区間推定

離散化指数形ソフトウェア信頼度成長モデルに含まれるパラメータおよびそれに基づいて導出される種々のソフトウェア信頼性評価尺度は, 前年度の研究調査において開発したブートストラップ法に基づく区間推定手法によって推定される. また, 式(2)および(3)は, 離散化指数形ソフトウェア信頼度成長モデルに基づいて導出された最適リリース時刻であるため, 離散化指数形ソフトウェア信頼度成長モデルが有するパラメータが含まれている. したがって, 前年度に開発したブートストラップ法によって得られたブートストラップ標本に基づいて, 式(2)および式(3)に対するブートストラップ分布をそれぞれ得ることができる.

### 2-4 ブートストラップ信頼区間

上述した手順に基づいて得られたソフトウェアの最適リリース時刻に関するブートストラップ分布に基づいて, それらを区間推定するため, ブートストラップ信頼区間[Efron (1979)]の適用を行った. 具体的には, 基本, 標準正規, パーセンタイルの各種ブートストラップ信頼区間を適用した.

## 3 適用例

実際に観測されたフォールト発見数データ:  $(n, y_n) (n = 1, 2, \dots, 25; y_{25} = 136)$  を用いて, 離散化指数形ソフトウェア信頼度成長モデルを用いながら, 今回議論したソフトウェア最適リリース時刻のブートストラップ法に基づいた区間推定手法の適用例を示す. また, 本研究では, 実測データを用いた適用例を示すため, フリー (無料) の統計解析環境である R を用いて, 今回議論した手法を実現するためのシミュレータを開発した.

実測データに基づいて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(2)および(3)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

表 1：コスト評価基準による最適リリース時刻 $Z_C^*$ の区間推定結果（危険率 5%）.

$c_1$	$c_2$	$c_3$	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	54.791	95.884	59.27501	101.566	64.958	106.05
1	4	0.001	60.751	106.89	65.81631	113.30	72.218	118.36
1	8	0.001	65.348	115.39	70.86116	122.34	77.817	127.86
1	16	0.001	69.482	123.02	75.39893	130.48	82.854	136.40
1	2	0.01	42.161	72.818	45.56433	76.984	49.730	80.387
1	4	0.01	48.199	83.821	52.10614	88.713	56.998	92.619
1	8	0.01	52.856	92.310	57.15127	97.758	62.600	102.05
1	16	0.01	56.991	99.947	61.68923	105.90	67.638	110.59

(BCI：ブートストラップ信頼区間)

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 $Z_R^*$ の区間推定結果（危険率 5%）.

$h$	$R_0$	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
1.0	0.9990	54.788	95.879	59.272	101.56	64.955	106.04
2.0	0.9990	58.067	102.34	62.918	108.47	69.054	113.32
3.0	0.9990	59.799	105.92	64.864	112.32	71.265	117.39
4.0	0.9990	60.903	108.33	66.120	114.92	72.710	120.13
3.0	0.9995	63.560	112.87	68.992	119.72	75.847	125.15
3.0	0.9995	72.293	129.00	78.576	136.91	86.484	143.19

(BCI：ブートストラップ信頼区間)

ラップ推定量は、次のように得られる：

$$Z_{C(b)}^* = \frac{\log c_3 - \log\{(c_2 - c_1)\hat{\omega}_{(b)}^* \delta b_{(b)}^*\}}{\log(1 - \delta b_{(b)}^*)}, \quad (4)$$

$$Z_{R(b)}^* = \frac{\log(-\log R_0) - \log[\hat{\omega}_{(b)}^* \{1 - (1 - \delta b_{(b)}^*)^h\}]}{\log(1 - \delta b_{(b)}^*)}, \quad (5)$$

図 1 および図 2 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 $Z_C^*$ 、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 $Z_R^*$ のブートストラップ分布をそれぞれ示す。また、表 1 に、基本、標準正規、およびパーセンタイルの 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 $Z_C^*$ の区間推定結果（有意水準 $\alpha = 5\%$ ）を示す。さらに、表 2 に、 $Z_R^*$ 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 $Z_C^*$ は遅くなる傾向があることがわかる。また、 $Z_R^*$ においては、リリース後の運用期間が長く、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。

#### 4 おわりに

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、パーセンタイルの 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[Efron (1987)]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。

【参考文献】

S. Yamada and S. Oasaki, “Cost-reliability optimal release policies for software systems”. IEEE Transactions on Reliability, Vol. R-34, No. 5, pp. 422-424, 1985.

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

S. Yamada, *Software Reliability Modeling –Fundamentals and Applications–*, Springer-Verlag, Tokyo, 2013.

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.

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

B. Efron, “Bootstrap methods: another look at the jackknife,” *The Annals of Statistics*, Vol. 7, No. 1, pp. 1-26, 1979.

B. Efron, “Better bootstrap confidence intervals,” *Journal of the American Statistical Association*, Vol. 82, No. 397, pp. 171-185, 1987.

〈発表資料〉

題名	掲載誌・学会名等	発表年月
A nonparametric bootstrapping approach for optimal software release planning	presented at the 2014 Spring Conference on the Korea Reliability Society, Jeju KAL Hotel, Jeju, Republic of Korea, 30 May, 2014, pp. 401--416.	2014年5月
ソフトウェア信頼性評価のための初期潜在フォールト数の切断分布に基づいたチェンジポイントモデルに関する一考察	電子情報通信学会技術研究報告 [信頼性], Vol. 114, No. 47, pp. 35--40, 金城学院大学, 名古屋市守山区, 2014年5月.	2014年5月
Optimal software shipping time estimation based on a change-point hazard rate model	International Journal of Reliability, Quality and Safety Engineering, Vol. 21, No. 2, pp. 1450007 (14 pages), April 2014. (DOI: 10.1142/S0218539314500077)	2014年6月
A modified change-point software reliability growth model	Proceedings of the 2014 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE) and the 5th International Conference on Maintenance Engineering (ICME 2014), Dalian, Liaoning, China, July 22--25, 2014, CD-ROM (IEEE Catalog Number: CFP1374P-CDR, ISBN: 978-1-4799-1016-8), 6 pages.	2014年6月
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	京都大学数理解析研究所講究録 1912「不確実性の下での数理的意思決定法の理論と応用」, pp.	2014年8月

	59—64.	
複数回のチェンジポイントを考慮したソフトウェア信頼性評価のためのハザードレートモデル	電子情報通信学会技術研究報告 [信頼性], Vol. 114, No. 170, pp. 1--6, スマイルホテル函館, 北海道函館市, 2014年8月.	2014年8月
Mending a change-point model for software reliability assessment	Proceedings of the Twentieth ISSAT International Conference on Reliability and Quality in Design, Seattle, Washington, U.S.A., August 7--9, 2014, pp. 119--123.	2014年8月
Hazard rate modeling for software reliability assessment with multiple change-point occurrences	Proceedings of the Twentieth ISSAT International Conference on Reliability and Quality in Design, Seattle, Washington, U.S.A., August 7--9, 2014, pp. 178--182.	2014年8月
Software reliability assessment using exponential-type change-point hazard rate models	International Journal of Reliability, Quality and Safety Engineering, Vol. 21, No. 4, pp. 1450019 (12 pages), August 2014. (DOI: 10.1142/S0218539314500193)	2014年8月
Change-point modeling and detection methods for software reliability assessment based on hazard rate models	Proceedings of the 6th Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling (APARM 2014) (Advanced Reliability and Maintenance Modeling VI --- Recent Developments on Reliability, Maintainability and Dependability, S.J. Bae, Y. Tsujimura, L. Cui, Eds., McGraw-Hill, Taiwan, 2014), Sapporo, Japan, August 21--23, 2014, pp. 184--191.	2014年8月
Extension of a hazard rate modeling approach for software reliability assessment with change-point	日本オペレーションズ・リサーチ学会 2014 年秋季研究発表会アブストラクト集, 2014年9月, 北海道科学大学, 札幌市手稲区, pp. 104--105.	2014年9月
Simulation-based interval estimation of optimal software release time	Proceedings of the Twelfth International Conference on Industrial Management, Chengdu, China, September 3--5, 2014, pp. 74--78.	2014年9月
Bootstrapping interval estimation of software reliability based on discretized NHPP model	Proceedings of the Twelfth International Conference on Industrial Management, Chengdu, China, September 3--5, 2014, pp. 255--259.	2014年9月
Change-point modeling and detection method for software reliability assessment	Proceedings of the Twelfth International Conference on Industrial Management, Chengdu, China, September 3--5, 2014, pp. 266--270.	2014年9月
Optimal testing-effort expending policies based on a two-dimensional software reliability growth model	International Journal of Reliability and Quality Performance, Vol. 6, No. 2, pp. 65--77, July--December 2014.	2014年12月
Software hazard rate modeling with multiple change-point occurrences	Proceedings of the 2014 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM 2014), Selangor, Malaysia, December 9--12, 2014, IEEE Catalog Number: CFP14IEI-USB (ISBN: 978-1-4799-6409-3), Reliability & Maintenance Engineering, 5 pages.	2014年12月

ソフトウェア信頼性評価のためのチェンジポイントを考慮したハザードレートモデルとチェンジポイント検出手法に関する一考察	統計数理研究所共同研究レポート 347「最適化：モデリングとアルゴリズム 27」, 統計数理研究所, pp. 81—86.	2015 年 3 月
A zero-truncated change-point model for software reliability assessment	日本オペレーションズ・リサーチ学会 2015 年春季研究発表会アブストラクト集, 2015 年 3 月, 東京理科大学神楽坂キャンパス, 東京都新宿区, pp. 298--299.	2015 年 3 月