Vol.2010-SE-169 No.7
2010/7/23

ソフトウェア信頼性評価のための テスト環境関数に基づいたチェンジポイントモデルに関する一考察

井 上 真 二[†1]　山 田 茂[†1]

ソフトウェア開発工程のテスト工程では，期間中におけるフォールトターゲットの変更等に起因しながらソフトウェア故障発生現象の確率的性質が著しく変化する現象が観測される．特に，その現象が観測されるテスト時刻はチェンジポイントと呼ばれ，その現象を考慮しない従来のモデリング枠組みに基づいて構築されたソフトウェア信頼度成長モデル（SRGM）の評価精度に影響を与えることが知られている．本研究では，このような問題に対処するための１つの回答として，チェンポイント前後におけるソフトウェア故障発生時間間隔の相互の関係性を考慮しながら，ソフトウェア信頼性評価のためのチェンジポイントモデルの構築枠組みについて議論する．

On Change-Point Modeling Based on Testing-Environment Function for Software Reliability Assessment

Shinji Inoue[†1] and Shigeru Yamada[†1]

In a testing-phase of a software development process, we often observe a phenomenon that the stochastic characteristic of the software failure-occurrence phenomenon notably changes due to a change of the testing-environment. Especially, we call such testing-time *change-point*. It is known that the change of the software failure-occurrence phenomenon at change-point influences accuracy of the software reliability assessment based on a software reliability growth model (SRGM) developed under a conventional modeling framework. This research discusses a change-point modeling framework for software reliability assessment by considering with the relationship of the software failure-occurrence phenomenon before and after change-point.

1. はじめに

ソフトウェア信頼度成長モデル（software reliability growth model，以下 SRGM と略す）[1)–5)] は，主に，ソフトウェア開発工程のテスト工程におけるソフトウェア信頼性を定量的に評価するための有用な数理モデルとして知られている．SRGM は，通常，テスト時間上におけるソフトウェア故障発生時間（もしくはソフトウェア故障発生時間間隔）を基本的な確率量として取り扱い，一般的に，ソフトウェア故障発生時間（間隔）の確率的性質がテスト期間中を通じて同一であるという仮定に基づいて構築される．しかしながら，フォールト発見難易度，フォールトの独立性，モジュール毎のフォールト密度の違い，テスト期間中におけるフォールト検出ターゲットの変更などによって，実際のソフトウェア開発のテスト工程では，ソフトウェア故障発生現象もしくはフォールト発見事象の統計的性質が著しく変化する現象がしばしば観測される．このような現象が発生するテスト時刻はチェンジポイント（change-point）[6)] と呼ばれ，これまでに提案されている SRGM に基づいた信頼性評価の精度に影響を与える要因の 1 つとして考えられている．

このような背景の下，上述した要因によるチェンジポイント発生シナリオに基づいた SRGM[6)–12)] に関する議論がなされている．近年では，チェンジポイントでの影響を考慮した一般化 SRGM 構築技術が提案されると共に[13)]，同手法に基づいて構築された SRGM を用いながら，下流工程および運用段階におけるデバッギングコストに関して最適なチェンジポイント発生時期とリリース時期を同時に推定する問題についても議論されており[14)]，実際のテスト環境をより反映したモデリング技術の開発からソフトウェア開発管理に関する問題への応用まで幅広く議論されている．しかし，チェンジポイントにおいて変化するソフトウェア故障発生時間間隔の相互の関係性を考慮したモデリング手法に関する議論は見当たらない．実際のテスト工程では，テスト期間中においてテスト環境が変化しても同一のソフトウェアをテストしているため，チェンジポイント前後におけるソフトウェア故障発生時間（間隔）には何らかの関係性があるものとして議論する方がより現実的であり，チェンジポイントでの影響を定量的に把握する上で極めて有用である．

本研究では，チェンジポイント前後において異なるソフトウェア故障発生時間の関係性をテスト環境関数を用いて与えると共に，チェンジポイント前後におけるソフトウェア故障発

†1 鳥取大学大学院工学研究科社会経営工学講座
Department of Social Management Engineering, Graduate School of Engineering, Tottori University

生時間の関係性を陽にモデルへと反映できるようなソフトウェア信頼性評価のためのチェンジポイントモデル構築枠組みについて議論する．また，提案モデルのパラメータ推定手法についても議論しながら，提案モデルに基づいた実測データを用いたソフトウェア信頼性解析例も示す．

2. NHPP モデル構築枠組み

本研究では，まず最初に，SRGM の基本的な構築枠組みついて議論する．テスト工程において検出可能なフォールト数が有限である場合を仮定した SRGM のほとんどは，以下の 3 つの基本的仮定[15)–18)] に基づいて構築することができる．

(A1) ソフトウェア故障が発生した場合，その原因となるフォールトは，直ちにかつ完全に修正・除去される．

(A2) 各ソフトウェア故障は，それぞれ，独立かつ時間に関してランダムに発生して，各ソフトウェア故障発生時刻は，それぞれ，同一の確率分布 $F(t) \equiv \Pr\{T \leq t\} = \int_0^t f(x)dx$ に従う非負の確率変数によって与えられる．ここで，$f(t)$ は確率密度関数を表す．

(A3) 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（初期潜在フォールト数）$N_0(>0)$ は，ある確率分布に従う確率変数とする．

いま，$\{N(t),\, t \geq 0\}$ を任意のテスト時刻 t までに発見された総フォールト数を表す確率過程とする．このとき，(A1) ～ (A3) から，テスト時刻 t までに m 個のフォールトが発見される確率は，次のように定式化される．

$$\Pr\{N(t)=m\} = \sum_n \binom{n}{m} \{F(t)\}^m \{1-F(t)\}^{n-m} \Pr\{N_0=n\} \qquad (m=0,1,2,\cdots). \tag{1}$$

よく知られた結果として，初期潜在フォールト数 N_0 が平均 $\omega(>0)$ のポアソン分布に従うと仮定した場合，式 (1) は，

$$\begin{aligned}\Pr\{N(t)=m\} &= \sum_n \binom{n}{m} \{F(t)\}^m \{1-F(t)\}^{n-m} \frac{\omega^n}{n!} \exp[-\omega] \\ &= \frac{\{\omega F(t)\}^m}{m!} \exp[-\omega F(t)] \qquad (m=0,1,2,\cdots)\end{aligned} \tag{2}$$

となり，平均値関数 $\mathrm{E}[N(t)] = \omega F(t)$ をもつ非同次ポアソン過程（nonhomogeneous Poisson process，以下 NHPP と略す）と本質的に等価となる．ここで，$\mathrm{E}[\cdot]$ は期待値を表す．したがって，当該モデリング枠組みに基づいて，ソフトウェア故障発生現象もしくはフォールト発見事象が NHPP に従い，検出可能なフォールト数が有限な SRGM を構築したい場合，式 (2) において，ある適切なソフトウェア故障発生時間分布 $F(t)$ を与えることで具体的に NHPP モデルを構築することができる．例えば，$F(t)$ がパラメータ λ の指数分布に従うとき $\mathrm{E}[N(t)] = \omega(1-e^{-\lambda t})$ が得られ，代表的な NHPP モデルある指数形 SRGM[20)] と本質的に等価な SRGM を構築することができる．

3. チェンジポイントを考慮した SRGM

2 において議論したモデリングの基本的仮定 (A2) を拡張することによって，本研究で議論する SRGM の構築枠組みを与える．特に，今回の議論では，チェンジポイント前後のソフトウェア故障発生時間間隔の関係性を考慮した SRGM を構築するため，チェンジポイントの発生後，1 番目にソフトウェア故障が発生するまでの時間分布をまず求める必要がある．

まず，ソフトウェア故障発生時間について，今回の議論において必要な確率量を次のように定義する．

X_i : チェンジポイント前における i 番目のソフトウェア故障発生時刻（$X_0=0, \quad i=0,1,2,\cdots$），

S_i : チェンジポイント前における i 番目のソフトウェア故障発生時間間隔（$S_i = X_i - X_{i-1}, \quad S_0=0, \quad i=1,2,\cdots$），

Y_i : チェンジポイント後における i 番目のソフトウェア故障発生時刻（$Y_0=0, \quad i=0,1,2,\cdots$），

T_i : チェンジポイント後における i 番目のソフトウェア故障発生時間間隔（$T_i = Y_i - Y_{i-1}, \quad T_0=0, \quad i=1,2,\cdots$）．

図 1 に，テスト時間軸上において定義されたこれらの確率量をそれぞれ図示する．本研究では，上述したチェンジポイント前後の確率量が，それぞれ，

$$\left.\begin{aligned} Y_i &= \alpha(X_i), \\ T_i &= \alpha(S_i), \\ J_i(t) &= K_i(\alpha(t)), \end{aligned}\right\} \tag{3}$$

のような関係にあるものと仮定する．ここで，$\alpha(\cdot)$ はチェンジポイント前後におけるソフトウェア故障発生時刻もしくはソフトウェア故障発生時間間隔の関係性を表すテスト環境関数，$J_i(t)$ および $K_i(t)$ はそれぞれ確率変数 S_i および T_i に対応する確率分布関数を表す．

本研究では，テスト環境関数が $\alpha(t) = \alpha t(\alpha > 0)$ の場合[19] を考える．ここで，α は比例定数であり，チェンジポイントにおけるテスト環境要因の変化がチェンジポイント前後のソフトウェア故障発生時間間隔の変化に与える影響の相対的な大きさを表すパラメータである．いま，チェンジポイントまでに n 個のフォールトが発見され，それぞれのソフトウェア故障発生時刻が $0 < x_1 < x_2 < \cdots < x_n \leq \tau$ であるとする．ここで，τ はチェンジポイントを表す．このとき，チェンジポイントから $n+1$ 番目のフォールトが発見されるまでの時間間隔 T_1 の確率分布関数は，

$$\begin{aligned}\bar{J}_1(t) &\equiv \Pr\{T_1 > t\} \\ &= \frac{\Pr\{S_{n+1} > \tau - x_n + t/\alpha\}}{\Pr\{S_{n+1} > \tau - x_n\}} \\ &= \frac{\exp[-\{M_B(\tau + t/\alpha) - M_B(x_n)\}]}{\exp[-M_B(\tau) - M_B(x_n)]}\end{aligned} \tag{4}$$

と求められる．ここで，$\bar{J}_1(t)$ は確率分布関数 $J_1(t) \equiv \Pr\{T_1 \leq t\}$ の余関数であり $\bar{J}_1(t) \equiv 1 - J_1(t)$ である．また，$M_B(t)(\equiv \omega K(t))$ は NHPP の平均値関数であり，チェンジポイント前において発見された総期待フォールト数を表す．式 (4) より，チェンジポイント後における任意のテスト時刻 $t \in (\tau, \infty]$ において発見される総期待フォールト数を表す

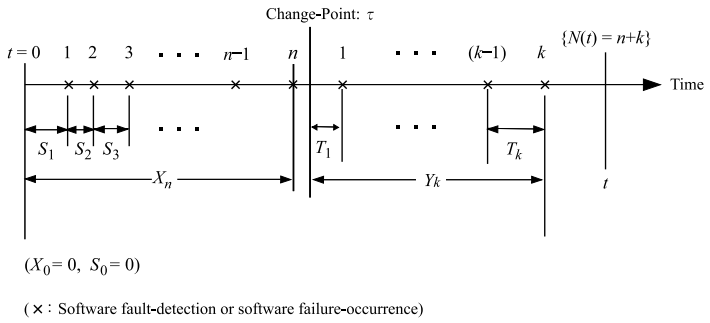

図 1 チェンジポイントを考慮したソフトウェア故障発生現象およびフォールト発見事象に関する確率諸量．
Fig. 1 Stochastic quantities for the software failure-occurrence or the fault-detection phenomenon with change-point.

NHPP の平均値関数は，

$$\begin{aligned}M_A(t) &= -\log \Pr\{T_1 > t - \tau\} \\ &= -\log \bar{J}_1(t - \tau) \\ &= M_B\left(\tau + \frac{t-\tau}{\alpha}\right) - M_B(\tau)\end{aligned} \tag{5}$$

と導出される．以上より，CP を考慮した NHPP の平均値関数は，

$$\Lambda(t) = \begin{cases} \Lambda_1(t) = M_B(t) & (0 \leq t \leq \tau) \\ \Lambda_2(t) = M_B(\tau) + M_A(t) & \\ \quad = M_B\left(\tau + \frac{t-\tau}{\alpha}\right) & (t > \tau) \end{cases} \tag{6}$$

のようになる．式 (6) から，チェンジポイント前において観測されたデータに適したソフトウェア故障発生時間分布を与えることによって，チェンジポイントを考慮した SRGM を構築することができる．

4. 信頼性評価尺度

ソフトウェア信頼性評価尺度は，定量的な信頼性評価を行う上で有用な評価尺度として知られている．ここでは，代表的な以下の 3 つの信頼性評価尺度について簡単に議論する．まず，期待残存フォールト数は，任意のテスト時刻 t においてソフトウェア内に残存しているフォールト数の期待値を表し，確率過程 $\{N(t), t \leq 0\}$ が式 (2) の NHPP に従う場合，一般的に次のように導出される．

$$\begin{aligned}M(t) &\equiv \mathrm{E}[\bar{N}(t)] = \mathrm{E}[N(\infty) - N(t)] \\ &= \omega - \Lambda(t).\end{aligned} \tag{7}$$

ここで，$\Lambda(t)$ は NHPP の平均値関数を表す．また，ソフトウェア信頼度関数は，任意のテスト時刻 t までテストが進行しているという条件の下で，その後の時間区間 $(t, t+x](t \geq 0, x \geq 0)$ においてソフトウェア故障が発生しない条件付き確率と定義される．したがって，ソフトウェア信頼度関数 $R(x \mid t)$ は，

$$\begin{aligned}R(x \mid t) &\equiv \sum_k \Pr\{N(t+x) = k \mid N(t) = k\} \Pr\{N(t) = k\} \\ &= \exp[-\{\Lambda(t+x) - \Lambda(t)\}]\end{aligned} \tag{8}$$

のように導出できる．最後に，累積 MTBF（mean time between software failures）は，ソ

フトウェア故障発生時間間隔分布が通常の確率分布関数の性質を満たさない場合における平均ソフトウェア故障発生時間間隔の代替的尺度の 1 つであり，

$$MTBF_C(t) = \frac{t}{\Lambda(t)} \tag{9}$$

として求められる．

5. 適 用 例

実際のテスト工程において観測されたフォールト発見数データを用いて，今回提案したモデルに基づく信頼性解析例を示す．本研究で用いる実測データは，WEB システムの開発プロジェクトにおいて得られた 26 組のフォールト発見数データ $(t_k, y_k)(k = 0, 1, 2, \cdots 26;\ t_{26} = 26(\text{日}), y_{26} = 34)$ であり，$\tau = 17$ 日目から強化テストを実施したものである．また，実測データの挙動に基づきながら，チェンジポイント前におけるソフトウェア故障発生時間分布がパラメータ b の指数分布に従うものと仮定する．このとき，今回議論したモデリング枠組みに基づいて構築されるモデルは，

$$\Lambda(t) = \begin{cases} \Lambda_1(t)=\omega\left\{1-\exp\left[-bt\right]\right\} & (0 \le t \le \tau), \\ \Lambda_2(t)=\omega\left\{1-\exp\left[-b\left(\tau+\frac{t-\tau}{\alpha}\right)\right]\right\} & (t > \tau), \end{cases} \tag{10}$$

と求められる．式 (10) に含まれるパラメータ ω, b, および α の推定値 $\widehat{\omega}$, $\widehat{b}$, および $\widehat{\alpha}$ は最尤法を用いて推定する．まず，確率過程 $\{N(t), t \ge 0\}$ に関する対数ゆう度関数 $\ln L(\boldsymbol{\theta}, \alpha \mid \tau)$ は，

$$\ln L(\boldsymbol{\theta}, \alpha \mid \tau) = \sum_{k=1}^{K} (y_k - y_{k-1}) \ln[\Lambda(t_k; \boldsymbol{\theta}, \alpha \mid \tau) - \Lambda(t_{k-1}; \boldsymbol{\theta}, \alpha \mid \tau)]$$
$$-\Lambda_i(t_K; \boldsymbol{\theta}, \alpha \mid \tau) - \sum_{k=1}^{K} \ln[(y_k - y_{k-1})!], \tag{11}$$

のように求められる．ここで，$L(\boldsymbol{\theta} \mid \alpha, \tau)$ は確率過程 $\{N(t), t \ge 0\}$ に関する同時確率関数もしくは同時ゆう度関数を表しており，$\boldsymbol{\theta}$ は今回のパラメータ推定において推定するパラメータの集合を表す．次に，式 (11) から導出できる同次対数ゆう度方程式：

$$\frac{\partial \ln L(\boldsymbol{\theta}, \alpha \mid \tau)}{\partial \boldsymbol{\theta}} = \frac{\partial \ln L(\boldsymbol{\theta}, \alpha \mid \tau)}{\partial \alpha} = 0 \tag{12}$$

を，それぞれ，モデル内部に含まれるパラメータに対して数値的に解くことによって，パラメータ推定値を得ることができる．

図 2 に，上記の方法に基づいて推定された平均値関数とその 95% 信頼限界を示す．ただし，推定された発見フォールト数の期待値 $\widehat{\mathrm{E}}[N(t)] \equiv \widehat{H}(t)$ の $100\gamma\%$ 信頼限界は，$\{N(t), t \ge 0\}$

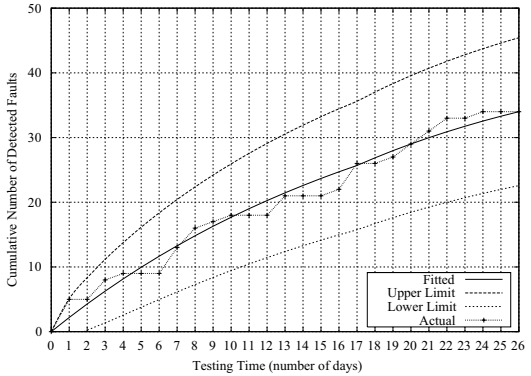

図 2 推定された平均値関数とその 95% 信頼限界（$\tau = 17$）．
Fig. 2 Estimated mean value function and its 95% confidence limits. ($\tau = 17$)

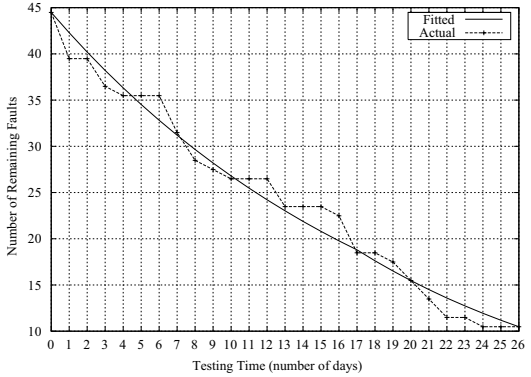

図 3 推定された期待残存フォールト数（$\tau = 17$）．
Fig. 3 Estimated expected number of remaining faults. ($\tau = 17$)

が NHPP に従う場合，

$$\widehat{\Lambda}(t) \pm K_\gamma \sqrt{\widehat{\Lambda}[(t)]} \tag{13}$$

として計算される[3]．ここで，K_γ は標準正規分布の $100(1+\gamma)/2$ パーセント点を表す．このような信頼限界を求めておくことは，テスト工程管理上，極めて有効である．図 2 から，

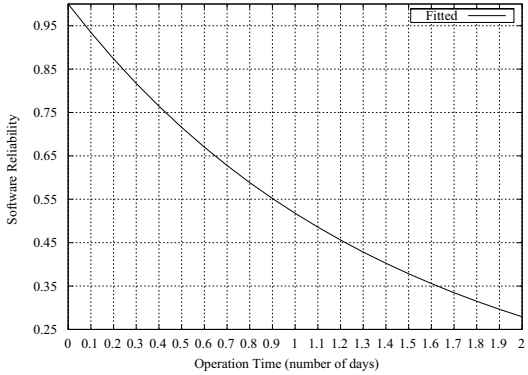

図 4 推定されたソフトウェア信頼度関数（$\tau = 17$）.
Fig. 4 Estimated software reliability function. ($\tau = 17$)

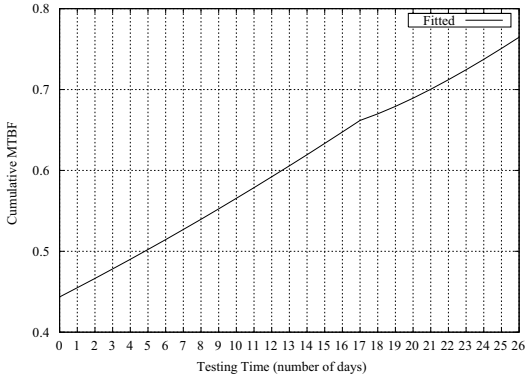

図 5 推定された累積 MTBF（$\tau = 17$）.
Fig. 5 Estimated cumulative MTBF. ($\tau = 17$)

表 1 MSE に基づくモデルの適合性比較結果.
Table 1 Results of model comparisons based on MSE.

	DS1	DS2	DS3	DS4	DS5
CPM	**18.4005**	**24.6323**	**2.1926**	**0.3311**	**2.4723**
EXP	16.085	32.0834	2.4009	0.3489	2.4724

CPM: change-point model (our model)
EXP: exponential SRGM

発見フォールト数の実測データに沿って，チェンジポイント前後における発見フォールト数の平均的挙動は変化していることがわかる．また，このとき，パラメータ α の推定値は $\widehat{\alpha} = 0.781$ であり，チェンジポイント前よりもチェンジポイント後の方が厳しい環境でテストが実施されたことがわかる．次に，図 3 に，推定された期待残存フォールト数の時間的挙動を表す．図 3 より，そのテスト終了時にソフトウェア内に残存する総期待フォールト数は $\widehat{M}(26) \approx 10$ と推定される．さらに，図 4 および図 5 に，推定されたソフトウェア信頼度関数 $\widehat{R}(x \mid 26)$ および累積 MTBF をそれぞれ示す．図 4 より，当該ソフトウェアがテスト開始後 26 週目にリリースされ，運用段階においてもチェンジポイント以降のテスト工程と同様の環境で使用した場合，リリース後 1 週目におけるソフトウェア信頼度は約 0.518 と推定される．また，図 5 より，テスト終了時刻における累積 MTBF は $\widehat{MTBF}_C(26) \approx 0.76$ (日) と推定される．

6. モデルの適合性比較

実測データを用いながら，チェンジポイントの影響を考慮しない従来の SRGM との適合性比較を行い，ソフトウェア信頼性評価に関する提案手法の有効性を検証する．適合性比較基準としては，平均偏差平方和（mean square errors，以下 MSE と略す）[3] を用いる．なお，比較する従来モデルは指数形 SRGM[20]：

$$\Lambda(t) = a(1 - e^{-bt}) \quad (a > 0, b > 0) \tag{14}$$

であり，チェンジポイントを考慮していないモデルとして式 (10) に対応する SRGM である．なお，適用する実測データは，**5** において適用したデータを含めた 5 つのデータ（DS1∼DS5）であり，それぞれ，チェンジポイントが明らかなデータである．

MSE は，K 組のフォールト発見数データが観測された場合，ある一定のテスト時刻 $t_k (k = 1, 2, \cdots, K)$ までに発見された総フォールト数の実測値 y_k と推定値 $\widehat{y}(t_k)$ との

偏差 2 乗和をデータ数で平均化したものであり，

$$\mathrm{MSE} = \frac{1}{K}\sum_{k=1}^{K}\{y_k - \widehat{y}(t_k)\}^2, \tag{15}$$

として算出される．したがって，小さな MSE 値を示すモデルほど，適用した実測データに対して適合性が高いモデルであると判断される．表 1 に，MSE に基づいたモデルの適合性比較結果を示す．表 1 より，今回適用したすべての実測データにおいて従来モデルよりも良い適合性を示しており，MSE の観点から，チェンジポイントを考慮した SRGM の有効性を確認することができる．

7. おわりに

本研究では，テスト期間中に観測されるチェンジポイントを考慮した新たなソフトウェア信頼度成長モデリング手法について議論した．本研究において議論した SRGM は，チェンジポイント前後におけるソフトウェア故障発生時間の変化とその関係性をテスト環境関数を導入しながらモデル化を行っており，これまでに提案されているチェンジポイントモデルよりも理論的な整合性を有するものと思われる．なお，今回議論したモデルは，チェンジポイントを考慮した NHPP モデルの構築枠組みを与えるものであり，チェンジポイント前に対する平均値関数を与えることで容易にチェンジポイントを考慮した NHPP モデルを構築することができる．今後は，数多くの実測データを適用しながら今回議論したモデリング枠組みの有効性をさらに検証する必要があると共に，複数のチェンジポイントが観測される場合における SRGM の開発を行う必要もある．

謝辞 本研究の一部は，文部科学省科学研究費補助金 基盤研究 (C)（課題番号 22510150）の援助を受けたことを付記する．

参考文献

1) Musa, J.D., Iannio, D. and Okumoto, K.: *Software Reliability*：*Measurement, Prediction, Application*. McGraw-Hill, New York (1987).
2) Yamada, S. and Osaki, S.: Software reliability growth modeling：Models and applications, *IEEE Trans. Soft. Eng.*, Vol. SE-11, No. 12, pp. 1431–1437 (1985).
3) 山田茂: ソフトウェア信頼性モデル —基礎と応用，日科技連出版社, 東京 (1994).
4) Pham, H.: *Software Reliability*. Springer-Verlag, Singapore (2000).
5) 山田茂，福島利彦: 品質指向ソフトウェアマネジメント，森北出版，東京 (2007).
6) Zhao, M.: Change-point problems in software and hardware reliability, *Commun. Statist. — Theory Meth.*, Vol. 22, No. 3, pp. 757–768, (1993).
7) 大寺浩志，山田茂，成久洋之: ソフトウェア信頼度成長モデルによるテスト工程管理, 電子情報通信学会論文誌, Vol. J70-D, No. 5, pp. 889–895 (1987).
8) Ohtera, H. and Yamada, S.: Optimal allocation & control problems for software-testing resources, *IEEE Trans. Reliab.*, Vol. 39, No. 2, pp. 171–176 (1990).
9) Huang, C.Y.: Performance analysis of software reliability growth models with testing-effort and change-point, *J. Sys. Soft.*, Vol. 76, No. 2, pp. 181–194 (2005).
10) Huang, C.Y.: Cost-reliability-optimal release policy for software reliability models incorporating improvements in testing efficiency, *J. Sys. Soft.*, Vol. 77, No. 2, pp. 139–155 (2005).
11) Zhao, J., Liu, H.W., Cui, G., and Yang, X.Z.: Software reliability growth model with change-point and environmental function, *J. Sys. Soft.*, Vol. 79, No. 11, pp. 1578–1587 (2006).
12) Zou, F.Z.: A change-point perspective on the software failure process, *Softw. Test., Verif. Reliab.*, Vol. 13, No. 2, pp. 85–93 (2003).
13) Inoue, S. and Yamada, S.: Software reliability measurement with change-point, *Proc. Intern. Conf. Qual. Reliab.* (ICQR 2007), pp. 170–175 (2007).
14) Inoue, S. and Yamada, S.: Optimal software release policy with change-point, *Proc. IEEE Intern. Conf. Indust. Eng. Eng. Mana.* (IEEM 2008), pp. 531–535 (2008).
15) Langberg, N. and Singpurwalla, N.D.: A unification of some software reliability models, *SIAM J. Scien. Comput.*, Vol. 6, No. 3, pp. 781–790 (1985).
16) Miller, D.S.: Exponential order statistic models of software reliability growth, *IEEE Trans. Soft. Eng.*, Vol. SE-12, No. 1, pp. 12–24 (1986).
17) Raftery, A.E.: Inference and prediction for a general order statistic model with unknown population size, *J. ASA*, Vol. 82, No. 400, pp. 1163–1168 (1987).
18) Joe, H.: Statistical inference for general-order-statistics and nonhomogeneous-Poisson-process software reliability models, *IEEE Trans. Soft. Eng.*, Vol. 15, No. 11, pp. 1485–1490 (1989).
19) 岡村寛之，土肥正，尾崎俊治: 運用段階におけるソフトウェア製品の信頼性評価手法 —加速寿命試験モデルの提案—，電子情報通信学会論文誌, Vol. J83-A, No. 3, pp. 294–301 (2000).
20) Goel, A.L. and Okumoto, K.: Time-dependent error-detection rate model for software reliability and other performance measures, *IEEE Trans Reliab.*, Vol. R-28, No. 3, pp. 206–211 (1979).