

年齢に依存した費用構造をもつ製品保証の数理モデル

土肥 正[†], 海生 直人[‡], 尾崎 俊治[†]

DOHI Tadashi, KAIO Naoto, OSAKI Shunji

[†]広島大学工学部, [‡]広島修道大学商学部

1. はじめに

コンピュータ社会の到来により, ソフトウェア製品の保守 (software maintenance) 技術を確立することは重要な課題となっている. 事実, ソフトウェアの保守費用は開発費用を大きく上回っており, ソフトウェアのライフサイクル中に機能追加 (version up) や性能向上 (revision up) が頻繁に行われているのが現状である [1]. ソフトウェア保守は予防保守と事後保守に大別される. ソフトウェアの予防保守とは, (i) システム変更時などに予想されるソフトウェア不良 (故障) の予防修正, (ii) ソフトウェア製品の用法についての技術支援, (iii) ソフトウェアの機能追加などによる製品改良等に分類される. 一方, ソフトウェアの事後保守とは, ユーザに提供した製品の不良原因を解明し, 不良を修正することによりユーザのトラブルを解消するサービスのことを意味する.

ソフトウェアの保守サービスシステム (software warranty service system) を設計する場合, その保証期間を如何に設定するかが重要な問題となる. フィールドにおけるソフトウェア不良の発生頻度は不確実であり, 保証期間があまりに長すぎると, ソフトウェア生産者 (software manufacturer: 以下 SM) にとって保守環境を継続的に維持するための費用が増加する. 逆に, ソフトウェアの保証期間が短すぎると, ソフトウェアユーザ (software user: 以下 SU) のリスクが大きくなる. よって, ソフトウェア開発・検査工程で得られた故障データに基づいて, ソフトウェアの保証メカニズムを定量的に評価することが望まれている.

そこで本稿ではまず最初に, ソフトウェアの製品保証 (software product warranty) に対する数理モデルを提案し, ソフトウェアの保証メカニズムが SM ならびに SU に与える影響について調査する. 提案されるモデルはハードウェアに対する従来の保証モデル [2] と類似しているが, (i) 事後保守によるソフトウェア製品の信頼度成長を考慮している点, (ii) ソフトウェアの修理時間に関する仮定を付加している点が特徴となっている. よって, 文献において議論されてきた検査段階におけるソフトウェアの最適リリースモデル (例えば [3]) や, いくつかのソフトウェア保守モデル [4, 5] とは大きく異なることに注意されたい. また, ここで提案されたモデルがソフトウェアの最適リリースモデルを特別な場合として含むことを示す.

続いて, ソフトウェアの製品保証と全く同じ枠組みで, ハードウェアを想定した修理系アイテムの保証モデルを定式化する. アイテムの故障が再生過程に従って生起するならば, 期待保証費用に再生関数が含まれるため, 特別な場合を除いてその解析的な表現を得ることが困難となる. そこで, いくつかの近似手法を用いて期待保証費用を評価することを試みる. 最終的に数値例において, 近似手法の妥当性を精度の点から比較する.

2. ハードウェア保証モデルの概要

ここでは, 非修理系アイテムの保証モデルについて述べる [2]. アイテムの寿命は連続で非負の確率変数であり, 対応する確率分布関数と再生関数を, それぞれ $F(t)$, $M(t)$ とする. 保証サービスの形態は, 無償取替保証サービスとプロレタ取替保証サービスに分類される. 保証期間 $T_F(>0)$ の無償取替保証において, 保証の更新がないものと仮定すれば, 生産者の期待保証費用は

$$C_M = k_s \{1 + M(T_F)\} \quad (1)$$

となる. ここで, $k_s(>0)$ は単位製品当たりの費用 (製造費用, 広告費用, その他) である. また, 無償保証期間中の故障に対して $k_b(>0)$ の単価で払い戻しが行われるアイテムを消費者が購入した場合, 生産者と消費者が被る期待保証費用は, それぞれ

$$C_M = k_s + k_b F(T_F), \quad (2)$$

$$C_C = k_b - k_b F(T_F) \quad (3)$$

となる。

一方，プロレタ取替保証サービスとは，アイテムの故障が発生するまでの使用時間に応じて払い戻しが行われるサービスである．いま，保証期間を $T_P(> 0)$ とすれば，生産者の保証費用は

$$c_M = \begin{cases} k_s + k_b(1 - t/T_P) & 0 \leq t < T_P \\ k_s & \text{その他} \end{cases} \quad (4)$$

のようになる．一方，消費者の費用は

$$c_C = \begin{cases} k_b - k_b(1 - t/T_P) & 0 \leq t < T_P \\ k_b & \text{その他} \end{cases} \quad (5)$$

となり，容易に $C_M = E[c_M]$ ならびに $C_C = E[c_C]$ を評価することができる．ここで，明らかに $C_M + C_C = k_s + k_b$ である．

これに対して，修理系アイテムの保証モデル (例えば [6]) においては，通常，大修理と小修理のみが仮定されており，修理に要する時間は無視できるものとされている．そこで次節では，ソフトウェア製品の保証に着目し，修理の時間が製品の年齢に従属するようなモデルについて考察する．

3. ソフトウェア保証モデル

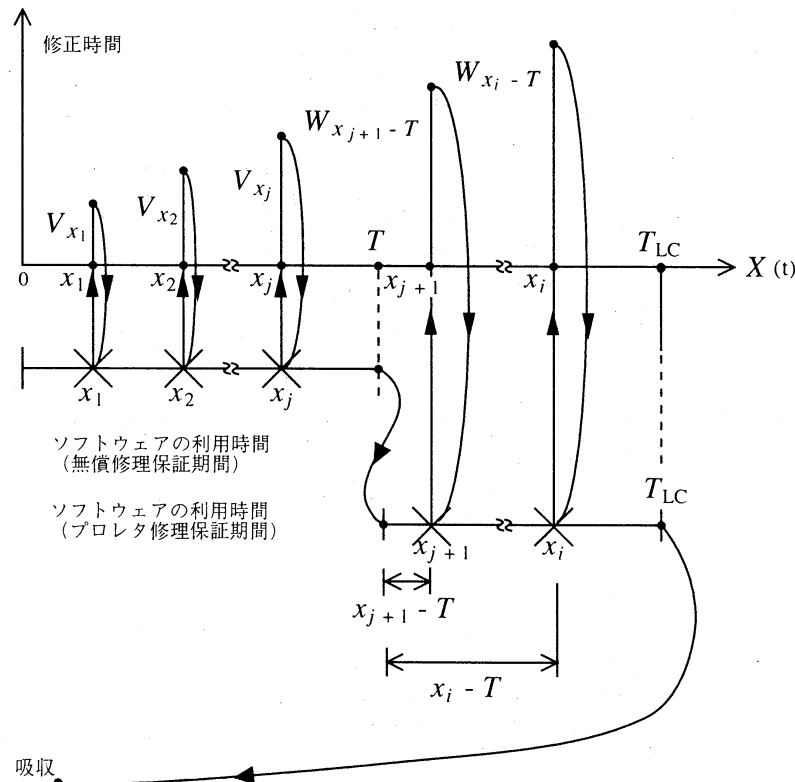


Fig. 1 Possible realization of software warranty period.

ソフトウェアのフィールドにおけるライフサイクル (寿命) $T_{LC}(>0)$ は既知であるとし、無償修理保証サービス (FRWS) とプロレタ修理保証サービス (PRWS) の順でサービスを行うことを考える。特に、PRWS 期間において発生したソフトウェア不良に対しては、全修理費用の $100p\%$ ($0 \leq p \leq 1$) を SU が負担し、残りを SM が負担するとしよう。この仮定は従来の混合型保証モデル [7] と比較して非常に単純であるにもかかわらず、ソフトウェア製品の保証形態を本質的に捕えているものであると考えられる。確率過程 $\{X(t), t \geq 0\}$ は時刻 t までのソフトウェア運用時間 (ユーザがソフトウェアを利用することが可能な時間) であり、FRWS と PRWS の期間は、それぞれ $t \in [0, T]$ ならびに $t \in (T, T_{LC}]$ である。また、FRWS と PRWS において、 $X(t) = x$ ($0 \leq x \leq T$ と $T < x \leq T_{LC}$) でソフトウェア不良が発見されたときの修理時間 (確率変数) をそれぞれ V_x と W_{x-T} のように表記する (Fig.1 を参照)。

いま、ソフトウェアの不良は強度関数 $r(x)$ 、平均値関数 $R(x)$ の非定常ポアソン過程に従って発生すると仮定する。さらに、FRWS 期間と PRWS 期間での単位時間当りの修理費用を $k_R(>0)$ および $k_G(>0)$ とし、FRWS 期間中ソフトウェア保守チームを維持するために必要な単位時間当りの予防保守費用を $k_0(>0)$ とする。

このとき、FRWS 期間において SM に必要な期待保守費用は

$$V_{FRWS}^{SM}(T) = k_0T + k_R \int_0^T E[V_x]r(x)dx \quad (6)$$

であり、PRWS 期間において SU と SM が被る期待保守費用は、それぞれ

$$V_{PRWS}^{SU}(T) = pk_G \int_T^{T_{LC}} E[W_{x-T}]r(x)dx, \quad (7)$$

$$V_{PRWS}^{SM}(T) = (1-p)k_G \int_T^{T_{LC}} E[W_{x-T}]r(x)dx \quad (8)$$

となる。よって、保証期間を通じて SM が必要とする総期待保守費用は

$$C(T) = V_{FRWS}^{SM}(T) + V_{PRWS}^{SM}(T). \quad (9)$$

次に、SM の総期待保守費用を最小にする保証切り替え期間 T^* を求めるために次の仮定をおく。

(A.1) 期待修理時間は、FRWS, PRWS それぞれの開始時刻からソフトウェア不良発生までの年齢に比例する。つまり、

$$E[V_x] = \alpha_1 x + \beta_1, \quad (0 \leq x \leq T, \alpha_1 \geq 0, \beta_1 \geq 0).$$

$$E[W_{x-T}] = \alpha_2(x - T) + \beta_2, \quad (T < x \leq T_{LC}, \alpha_2 \geq 0, \beta_2 \geq 0).$$

(A.2) ソフトウェアの不良は指数型ソフトウェア信頼度成長モデル [3] に従って発生すると仮定する。つまり、

$$r(x) = ab \exp(-bx),$$

$$R(x) = a \{1 - \exp(-bx)\}.$$

ここで、パラメータ $a(>0)$ と $b(>0)$ は、ソフトウェア内に潜在する総期待フォールト数および残存フォールト 1 個当りの不良発見率をそれぞれ表す。仮定 (A.1) と (A.2) の下で、SM と SU の総期待費用は、

$$\begin{aligned} C(T) = & k_0T + k_R \left\{ a \left(\frac{\alpha_1}{b} + \beta_1 \right) (1 - \exp(-bT)) - \alpha_1 a T \exp(-bT) \right\} \\ & + (1-p)k_G \alpha_2 a (T \exp(-bT) - T_{LC} \exp(-bT_{LC})) \\ & + a \left(\frac{\alpha_2}{b} + \beta_2 - \alpha_2 T \right) (\exp(-bT) - \exp(-bT_{LC})), \end{aligned} \quad (10)$$

$$V_{PRWS}^{SU}(T) = pk_G\alpha_2a(T\exp(-bT) - T_{LC}\exp(-bT_{LC})) \\ + a\left(\frac{\alpha_2}{b} + \beta_2 - \alpha_2T\right)(\exp(-bT) - \exp(-bT_{LC})) \quad (11)$$

となる。

仮定 (A.1) におけるパラメータ $\alpha_i (\geq 0)$ ($i = 1, 2$) は次のように解釈することが可能である。ソフトウェアの使用回数がパラメータ $\lambda_i (> 0)$ のポアソン過程に従って生じ、ソフトウェアの使用時間がパラメータ $\mu_i (> 0)$ の指数分布に従うと仮定すれば、 (λ_i, μ_i) は SU がソフトウェアを使用する際の到着率と処理率を意味する。従って、ソフトウェアの使用時間を基本とした修理時間の変動部分に対する比例定数を $m_i (> 0)$ とすれば、

$$\alpha_i = m_i\lambda_i/\mu_i, \quad (i = 1, 2) \quad (12)$$

と考えることができる。また、パラメータ $\beta_i (> 0)$ は修理に要する固定時間を意味する。

仮定 (A.1) と (A.2) の下で、次のような非線形関数を定義する。

$$q(T) = k_0 + k_RE[V_T]r(T) + (1-p)k_G \\ \times \left\{ \int_T^{T_{LC}} \frac{\partial E[W_{x-T}]}{\partial T} r(x)dx - E[W_0]r(T) \right\}. \quad (13)$$

また、以下のような定数を定義する。

$$D = k_R\alpha_1 + (1-p)k_G\alpha_2 + b((1-p)k_G\beta_2 - k_R\beta_1). \quad (14)$$

このとき、SM と SU にとって最適なソフトウェア保証切り替え期間に関する以下の結果を得る。

定理 1 (SM): (I) $D > 0$ であるとき次のことが成立する。

- (1) $q(0) < 0$ かつ $q(T_{LC}) > 0$ ならば、非線形方程式 $q(T) = 0$ を満たす有限で唯一の最適解 T_{SM}^* が存在し、 $T_{LC} \leq D/(k_R\alpha_1b)$ ならば $0 < T_{SM}^* < T_{LC}$ であり、そうでないならば $0 < T_{SM}^* < D/(k_R\alpha_1b)$ である。
- (2) もし $q(0) \geq 0$ ならば、 $T_{SM}^* = 0$ となり、SM にとって FRWS を行わないことが最適となる。
- (3) もし $q(T_{LC}) \leq 0$ ならば、 $T_{SM}^* = T_{LC}$ となり、SM にとって PRWS を行わないことが最適となる。

(II) $D \leq 0$ であるとき、 $T_{SM}^* = 0$ となる。

定理 2 (SU): 総期待保守費用 $V_{PRWS}^{SU}(T)$ は T の減少関数となり、SU にとって最適な保証期間は $T_{SU}^* = T_{LC}$ となる。

上述の結果より、SU が要求する期待保証費用の上限 $C_{SU} (> 0)$ が与えられた場合、SM の設定する保証期間は以下の問題の解となる。

$$\min_{0 \leq T \leq T_{LC}} C(T) \quad (15)$$

$$\text{subject to } \{T \mid V_{PRWS}^{SU}(T) \leq C_{SU}\}.$$

さらに、本稿で議論したソフトウェア保証モデルが従来のソフトウェア最適リリース問題を特別な場合として含むことを示す。

系： $\alpha_i = 0, \beta_i = 1, p = 0$ ($i = 1, 2$) とすれば、SM の総期待保証費用は文献 [3] で議論されたソフトウェアリリース問題の期待費用と一致する。

4. 修理系アイテムの保証モデル

前節で仮定した非定常ポアソン過程を通常の再生過程に置き換えることによって、従来から議論されてきた修理系アイテムの保証モデルを修理の観点から一般化することが可能となる。いま、アイテムの故障時間間隔は独立で同一な確率変数であり、その確率分布関数ならびに n 次モーメントを、それぞれ $F(t)$, γ_n ($n=1, 2, 3, \dots$) とする。このとき、アイテムの故障数は再生過程に従い、対応する再生関数および密度を $M(t)$, $m(t)$ と定義する。その他の記号は 3 節で用いたものをそのまま使用する。以上の仮定の下で、SM と SU に対する総期待保守費用は

$$C(T) = k_0 T + k_R \int_0^T E[V_x] m(x) dx + (1-p) k_G \int_T^{T_{LC}} E[W_{x-T}] m(x) dx, \quad (16)$$

$$V_{PRWS}^{SU}(T) = p k_G \int_T^{T_{LC}} E[W_{x-T}] m(x) dx \quad (17)$$

となる。よって、仮定 (A.1) の下で、それぞれ

$$\begin{aligned} C(T) = & k_0 T + \{k_R(\alpha_1 T + \beta_1) - (1-p)k_G\beta_2\}M(T) \\ & + \{(1-p)k_G\alpha_2 - k_R\alpha_1\} \int_0^T M(x) dx \\ & + (1-p)k_G\{\alpha_2(T_{LC} - T) + \beta_2\}M(T_{LC}) \\ & - (1-p)k_G\alpha_2 \int_0^{T_{LC}} M(x) dx, \end{aligned} \quad (18)$$

$$\begin{aligned} V_{PRWS}^{SU}(T) = & -p k_G \beta_2 M(T) + p k_G \alpha_2 \int_0^T M(x) dx \\ & + p k_G (\beta_2 - \alpha_2 T + \alpha_2 T_{LC}) M(T_{LC}) - p k_G \alpha_2 \int_0^{T_{LC}} M(x) dx. \end{aligned} \quad (19)$$

一般に、 $F(t)$ が特殊な場合を除いて再生関数 $M(t)$ の解析的な表現は知られていない。よって、以下では最適保証方策を簡便に計算するための近似手法について述べる。

(4.1) 直線近似

再生関数について、以下のような漸近的性質が知られている。

$$\lim_{t \rightarrow \infty} \left\{ M(t) - \frac{t}{\gamma_1} \right\} = \frac{\gamma_2}{2\gamma_1^2} - 1. \quad (20)$$

これより、 $M(t)$ とその定積分の近似式は以下の式で与えられる [8]。

$$M(t) \approx \frac{t}{\gamma_1} + \frac{\gamma_2}{2\gamma_1^2} - 1, \quad (21)$$

$$\begin{aligned} \int_0^t M(x) dx \approx & \frac{t^2}{2\gamma_1} + \left(\frac{\gamma_2}{2\gamma_1^2} - 1 \right) t \\ & + \frac{\gamma_2^2}{4\gamma_1^3} - \frac{\gamma_3}{6\gamma_1^2}. \end{aligned} \quad (22)$$

上述の近似式を用いれば、総期待保証費用 $C(T)$ は T に関する 2 次関数

$$C(T) \approx a_0 \left(T - \frac{a_1}{2a_0} \right)^2 + a_3 - \frac{a_1^2}{4a_0}, \quad (23)$$

$$a_0 = \frac{1}{2\gamma_1} \{ k_R \alpha_1 + (1-p) k_G \alpha_2 \}, \quad (24)$$

$$a_1 = \frac{1}{\gamma_1} \{ (1-p) k_G (\beta_2 + \alpha_2 T_{LC} - k_R \beta_1) \} - k_0, \quad (25)$$

$$\begin{aligned} a_3 = & \left(\frac{\gamma_2}{2\gamma_1^2} - 1 \right) \{ k_R \beta_1 + \beta_2 - (1-p) k_G \beta_2 \} \\ & - k_R \alpha_1 \left(\frac{\gamma_2^2}{4\gamma_1^3} - \frac{\gamma_3}{6\gamma_1^2} \right) + \frac{1}{2\gamma_1} (1-p) k_G \alpha_2 T_{LC}^2 \\ & + \frac{1}{\gamma_1} \beta_2 T_{LC} \end{aligned} \quad (26)$$

となる。これより、最適解を $T_{SM}^* = a_1/(2a_0)$ と近似する。SU の期待保証費用 $V_{PRWS}^{SU}(T)$ に対しても同様な近似表現を得ることができる。

(4.2) 均衡分布による近似

確率分布 $F(t)$ に対する均衡分布を次のように定義する。

$$F_e(t) \equiv \frac{1}{\gamma_1} \int_0^t \bar{F}(\tau) d\tau. \quad (27)$$

均衡分布を用いた再生関数の近似式として、Ozbaykal [9] は次のような表現を得ている。

$$M(t) \approx \frac{t}{\gamma_1} - F_e(t) + \int_0^t [1 - F_e(t - \tau)] d\tau. \quad (28)$$

上式で与えられた近似式は、Batholomew [10] の近似よりも再生関数の漸近的性質をよく捕えており、Deligönül [11] の近似式よりも解析的な取り扱いが容易である。

(4.3) ガンマ近似

再生関数

$$M(t) = \sum_{k=1}^{\infty} F^{(k)}(t) \quad (29)$$

の計算が困難であることは、たたみこみの項 $(F^{(k)}(t), k \geq 2)$ を計算することが困難であることに起因する。よって、確率分布のあてはめと同様な発想から、分布関数 $F(t)$ と同じ 2 つのモーメントを有するガンマ分布により $F^{(k)}(t), k \geq 2$ の項を近似することを考える。このことは Smeitink and Dekker [12] が Coxian-2 分布を用いて行った近似計算と類似している。パラメータ (α, λ) のガンマ分布 $F_G(t)$ を以下の式で定義する。

$$F_G(t) = \frac{1}{\Gamma(\alpha)} \int_0^t e^{-\lambda\tau} \lambda^\alpha \tau^{\alpha-1} d\tau, \quad t \geq 0. \quad (30)$$

ここで、 $\Gamma(\cdot)$ は標準ガンマ関数である。ガンマ分布の n 重たたみこみ $F_G^{(n)}(t)$ はパラメータ $(n\alpha, \lambda)$ のガンマ分布に従うので、近似再生関数は以下の式で与えられる。

$$M(t) \approx F(t) + \sum_{n=2}^{n_G} F_G^{(n)}(t). \quad (31)$$

ガンマ分布のパラメータ (α, λ) を決定する際には、モーメントマッチによる法を用いる。すなわち、

$$\alpha = \frac{\gamma_1^2}{\sigma^2}, \quad (32)$$

$$\lambda = \frac{\gamma_1}{\sigma^2}. \quad (33)$$

ここで σ^2 は分布関数 $F(t)$ の分散を示す。(31) 式の n_G は打ち切り誤差 ε を用いて、以下の条件が成立しない最小の n として定める。

$$\frac{F_G^{(n)}(t)F_G(t)}{1 - F_G(t)} \leq \varepsilon. \quad (34)$$

よって、上式を用いることで、 $F^{(n)}(t)$ を逐次的に評価することが可能となる。

5. 数値例

(5.1) ソフトウェア保証モデル

ここでは実際にソフトウェアのテスト行程で観測された 136 個の故障データ [13] を用いて、最適な保証切り替え時間 T_{SM}^* を求める。仮定 (A.2) の指数型ソフトウェア信頼度成長モデルに基づいて未知パラメータ a と b を最尤法を用いて推定し、その結果は以下のようになった。

$$a = 106.548, \quad b = 0.67211 \times 10^{-4}. \quad (35)$$

使用するモデルパラメータは、 $k_0 = 0.01$, $k_R = 2.0$, $k_G = 3.5$, $\alpha_1 = 0$, $\alpha_2 = 0.5\rho$, $\rho = 0.11$, $\beta_1 = 1.0$, $\beta_2 = 1$, $p = 0.5$, $T_{LC} = 10.0$ である。

Fig. 2 は SM と SU に対する総期待保証費用のふるまいをそれぞれ調べたものである。これより、SU が要求する総期待費用がある上限を下回るように最適保証切り替え時間を決定すればよい。Table 1 はパラメータ ρ に対する最適方策 T_{SM}^* の従属性を示している。パラメータ ρ が増加する、すなわちユーザの到着率と処理率の割合が増加するにつれて、最適切り替え時間と総期待保証費用 $C(T_{SM}^*)$ は減少することがわかる。

(5.2) 修理系アイテムの保証モデル

確率分布 $F(t)$ の n 重たたみこみ $F^{(n)}(t)$ の値を計算するアルゴリズムとして、3 次スプラインアルゴリズム [14] を用いて再生関数を計算する。時間区間 $[0, t_0]$ を刻み幅 d により n 等分し、その時点における $F^{(k)}(t)$ 上の点列 $(0, F^{(k)}(0))$, $(d, F^{(k)}(d))$, $(2d, F^{(k)}(2d))$, $\dots$, $(nd, F^{(k)}(nd))$ が与えられるものとする。このとき、これらの点列 $(jd, F^{(k)}(jd))$ を用いて区間 $[jd, jd+d]$ 内で以下の $y_j(t)$ で表される区分的な多項式補間を行う。

$$y_j(t) = \frac{(jd+d-t)^3}{6d}c_j + \frac{(t-jd)^3}{6d}c_{j+1} + b_j(jd+d-t) + b_{j+1}(t-jd). \quad (36)$$

ここで、 $jd \leq t \leq jd+d$ ($j = 0, 1, 2, \dots, n-1$) であり、

$$b_j = F^{(k)}(jd)/d - dc_j/6. \quad (37)$$

c_j ($j = 0, 1, 2, \dots, n$) はスプライン係数と呼ばれており, 補間を行う点列 $(jd, F^{(k)}(jd))$ ($j = 0, 1, 2, \dots, n$) を用いた 3 重対角方程式を解くことによって得られる. 最終的に, 得られた補間関数に Robatto の積分公式を適用し, たたみこみを計算する.

いま, 故障時間分布関数として次のようなワイブル分布を仮定する.

$$F(t) = 1 - \exp\left\{-\left(\frac{t}{\theta}\right)^\beta\right\}. \quad (38)$$

ワイブル分布の形状パラメータと尺度パラメータをそれぞれ $\beta = 1.8$, $\theta = 2.4$ のように設定し, その他のパラメータは Table 1 と同じものを用いた. この場合の分布のモーメントは, $\gamma_1 = 2.13429$, $\gamma_2 = 6.06058$, $\gamma_3 = 20.7993$ となる. Fig. 3 は SM と SU に対する総期待保証費用のふるまいをそれぞれ調べたものである. ソフトウェア保証モデルと同じように, SU が要求する総期待費用がある上限を下回るように最適保証切り替え時間を決定すればよいことがわかる. Table 2-4 は線形近似, Ozbaykal 近似, ガンマ近似を用いたときの最適解 T_{SM}^* と総期待保証費用の値を示したものである. 近似を用いた場合は共通して, 最適切り替え時刻の値は ρ が増加するに従い真の値に近づき, 近似に対する相対誤差も比較的小さい. しかしながら, 線形近似と Ozbaykal の近似は期待保証費用をそれぞれ過小評価ならびに過大評価する傾向にあり, 期待保証費用を計算するためにはガンマ近似が最も良好であるという結果を得た.

参考文献

- [1] 石井康雄 (編), ソフトウェアの検査と品質保証, 日科技連, 東京 (1986).
- [2] W. R. Blischke and D. N. P. Murthy, *Mathematical Models for Warranty Cost Analysis*, Mercel Dekker, N.Y. (1993).
- [3] K. Okumoto and A. L. Goel, "Optimum release time for software systems based on reliability and cost criterion", *J. Sys. Soft.*, vol. 1, pp. 315-318 (1980).
- [4] N. S. Wee, "Optimal maintenance schedules of computer software", *Prob. Eng. Inf. Sci.*, vol. 4, pp. 243-255 (1990).
- [5] F. J. Radermacher, S. M. Ross and N. S. Wee, "An optimal software debugging model", *Ann. Ope. Res.*, vol. 32, pp. 141-163 (1991).
- [6] D. G. Nguyen and D. N. P. Murthy, "A general model for estimating warranty costs for repairable products", *IIE Trans.*, vol. 16, pp. 379-386 (1984).
- [7] D. G. Nguyen and D. N. P. Murthy, "Cost analysis of warranty policies", *Naval Res. Logst. Quart.*, vol. 31, pp. 525-541 (1984).
- [8] H. C. Tijms, *Stochastic Models: An Algorithmic Approach*, John Wiley & Sons, N.Y. 1994.
- [9] T. Ozbaykal, "Bounds and approximations of the renewal function", unpublished M.S. thesis, Naval Postgraduate School, Department of Operations Research and Administrative Science, Monterey, California, 1971.
- [10] D. J. Bartholomew, "An approximate solution of the integral equation of renewal theory", *J. Roy. Stat. Soc.*, vol. 25B, pp. 432-441 (1963).
- [11] Z. Deligönül, "An approximate solution of the integral equation of renewal theory", *J. Appl. Prob.*, Vol. 22, pp. 926-931 (1985).
- [12] E. Smeitink and R. Dekker, "A simple approximation to the renewal function", *IEEE Trans. Reli.*, vol. R-39, pp. 71-75 (1990).
- [13] A. A. Abdel-Ghaly, P. Y. Chan and B. Littlewood, "Evaluation of competing software reliability prediction", *IEEE Trans. Soft. Eng.*, vol. SE-12, pp. 950-967 (1986).
- [14] R. Cléroutx and D. J. McConalogue, "A numerical algorithm for recursively-defined convolution integrals involving distribution functions", *Management Sci.*, vol. 22, pp. 1138-1146 (1976).

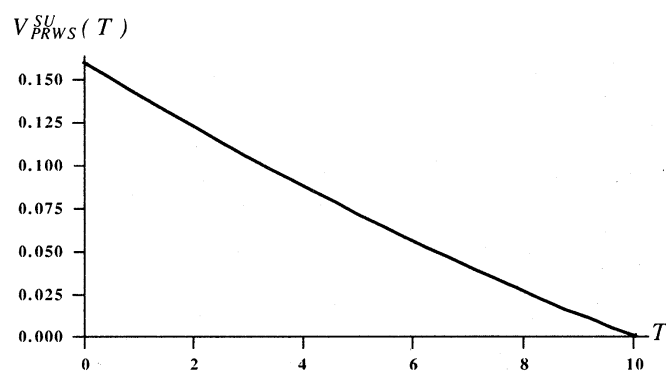
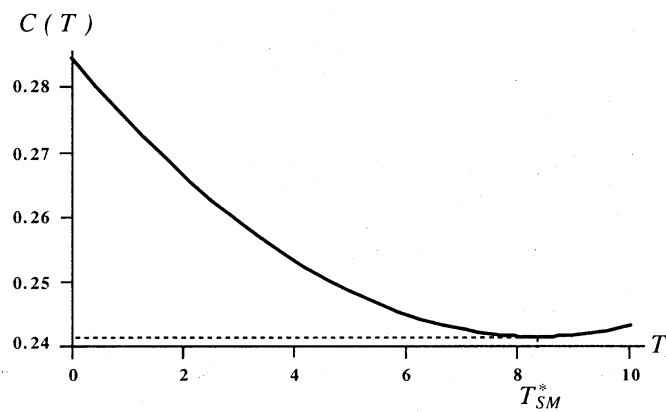


Fig. 2 Behaviour of the expected total warranty cost.
(NHPP case)

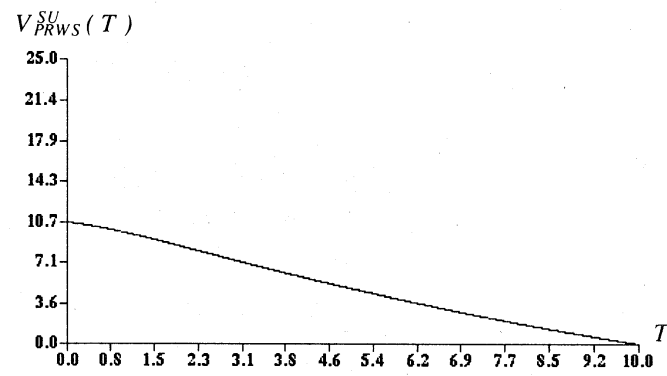
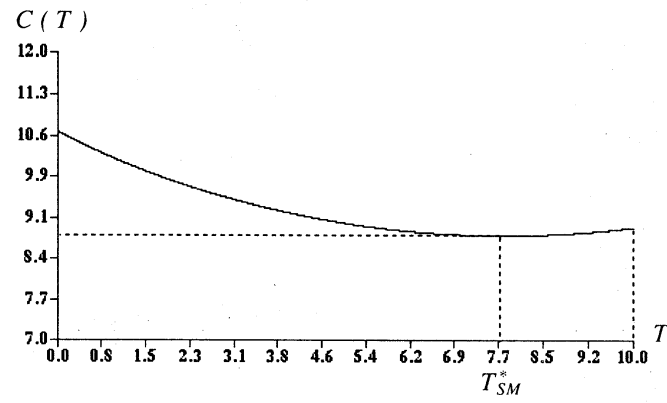


Fig. 3 Behaviour of the expected total warranty cost.
(renewal case)

Table 1 Software warranty based on the exponential software growth model.

NHPP		
ρ	T_{SM}^*	$C(T_{SM}^*)$
0.110	8.353	0.242
0.120	8.284	0.241
0.130	8.220	0.241
0.140	8.162	0.241
0.150	8.108	0.240
0.160	8.056	0.240
0.170	8.007	0.240
0.180	7.959	0.239
0.190	7.913	0.239
0.200	7.867	0.239

Table 2 Linear approximation.

linear approximation						
ρ	T_{SM}^*	relative error	$C(T_{SM}^*)$	relative error	exact T_{SM}^*	exact $C(T_{SM}^*)$
0.110	7.181	0.574	5.359	38.980	7.140	8.782
0.120	7.416	0.474	5.374	38.914	7.381	8.798
0.130	7.615	0.396	5.387	38.855	7.585	8.810
0.140	7.785	0.335	5.397	38.818	7.759	8.821
0.150	7.933	0.291	5.407	38.770	7.910	8.831
0.160	8.062	0.249	5.415	38.736	8.042	8.839
0.170	8.176	0.221	5.422	38.708	8.158	8.846
0.180	8.277	0.194	5.429	38.673	8.261	8.853
0.190	8.368	0.180	5.435	38.645	8.353	8.858
0.200	8.450	0.166	5.440	38.625	8.436	8.864

Table 3 Ozbaykal approximation.

Ozbaykal approximation						
ρ	T_{SM}^*	relative error	$C(T_{SM}^*)$	relative error	exact T_{SM}^*	exact $C(T_{SM}^*)$
0.110	7.182	0.583	10.131	15.359	7.140	8.782
0.120	7.416	0.477	10.146	15.330	7.381	8.798
0.130	7.615	0.393	10.159	15.306	7.585	8.810
0.140	7.785	0.338	10.170	15.287	7.759	8.821
0.150	7.933	0.288	10.179	15.270	7.910	8.831
0.160	8.062	0.249	10.187	15.255	8.042	8.839
0.170	8.176	0.221	10.194	15.242	8.158	8.846
0.180	8.277	0.196	10.201	15.231	8.261	8.853
0.190	8.368	0.180	10.207	15.221	8.353	8.858
0.200	8.450	0.161	10.212	15.212	8.436	8.864

Table 4 Gamma approximation.

gamma approximation						
ρ	T_{SM}^*	relative error	$C(T_{SM}^*)$	relative error	exact T_{SM}^*	exact $C(T_{SM}^*)$
0.110	7.186	0.644	8.623	1.820	7.140	8.782
0.120	7.420	0.528	8.637	1.820	7.381	8.798
0.130	7.618	0.435	8.650	1.819	7.585	8.810
0.140	7.787	0.361	8.661	1.818	7.759	8.821
0.150	7.934	0.303	8.670	1.817	7.910	8.831
0.160	8.063	0.261	8.678	1.816	8.042	8.839
0.170	8.177	0.233	8.686	1.815	8.158	8.846
0.180	8.278	0.206	8.692	1.814	8.261	8.853
0.190	8.369	0.192	8.698	1.813	8.353	8.858
0.200	8.450	0.166	8.703	1.812	8.436	8.864