

ソフトウェア保守コストモデルにおける 最適リリース問題

山田 茂, 木村 光宏

1. まえがき

われわれの社会生活を支える重要なインフラストラクチャーとして、コンピュータシステムの果たす役割はますます重要になりつつある。また、このようなシステムを構成するソフトウェアも大規模化、複雑化、多様化してきている。一方、ソフトウェア開発は依然として手工業的な人的作業によってなされるため、多くの人的誤りや欠陥、いわゆるソフトウェアフォールト（以下、フォールトと略す）が潜入してしまうことは避けられない。ここで、ソフトウェア内に潜在するフォールトにより、期待どおりにソフトウェアが動作しないことをソフトウェア故障と定義する。そこで、効率的に高品質ソフトウェア製品を開発するために、開発プロセスの各工程を、品質、コスト、納期の観点から、科学的に管理していく必要性がでてきた。特に、ソフトウェア開発の最終工程であるテスト工程は、開発プロセスで潜入したフォールトの発見・修正を行なうとともに、ソフトウェアの出荷品質を評価するために重要な工程となっている。従来、テスト工程の進捗状況の把握や出荷時期の見積りは、あらかじめ準備されたテスト用入力データやテスト資源の消化具合や当該ソフトウェアの納期などの制約を考慮して、ソフトウェア開発管理者の勘や経験などから行なわれることも多かった。これに対して、近年ではテスト工程で観測されるフォールト発見事象をソフトウェア信頼度成長モデル (software reliability growth model) のような数理モデルにより記述し、このモデルから導出される定量的な信頼性評価尺度（たとえばソフトウェア信頼度、平均ソフトウェア故障時間間隔、および期

待残存フォールト数など）を予測することにより、ソフトウェア開発管理における意思決定が行なわれることが多くなってきた [1-5]。これにより、テスト工程の進捗状況をより正確に把握することができ、ソフトウェアをユーザに引き渡すのに適切な時期を見積ることも可能となる。このように、ソフトウェアの信頼性評価尺度を用いて、テスト期間とテストにより確認される信頼性との関係から、テスト工程を終了して実際の運用段階に移行するのに最適な時期、すなわち総テスト時間である最適リリース時刻を決定する問題は、ソフトウェアの最適リリース問題 (optimal software release problem) と呼ばれている [6-8]。

本稿では、最適リリース時刻の決定に関する評価基準に総期待ソフトウェアコストを採用し、第3章および第4章において、それぞれ異なる考え方にもとづく最適リリース方策について議論する。前者の最適リリース問題においては、総期待ソフトウェアコストの定式化に際して、ソフトウェア開発が数年間のような長期にわたることと、ソフトウェアの運用期間全体にわたって（すなわち、ソフトウェアが廃棄されるまで）ソフトウェアの保守を行なうことを想定している [9]。これに対して後者では、ソフトウェアのリリース後、所定の期間に対しては検出されたフォールトに対する保守コストを開発者側が負担するような保証期間を設けることを想定した問題について議論する [10]。第5章ではそれぞれの最適方策にもとづく数値例を示し、各コストパラメータの変化と最適リリース時刻との間の関係について考察する。

2. ソフトウェア信頼度成長モデル

ソフトウェアは人間の作った知的生産物であるため、その開発工程内において人為的誤りや欠陥などのフォールトが潜入することは避けられない。そこで、ソフトウェア開発の最終段階におけるテスト工程や実際の

やまだ しげる, きむら みつひろ
鳥取大学工学部
〒680 鳥取市湖山町南 4-101

運用・保守段階において、フォールトの発見・修正が行なわれる。このようなフォールト発見事象を記述し、テスト工程や運用・保守段階におけるソフトウェアの信頼性評価を行なうために、時刻 t までに発見された総フォールト数 $N(t)$ を確率変数として導入し、これを計数過程 (counting process) とみなして $\{N(t), t \geq 0\}$ とする。この $N(t)$ に対し、非同次ポアソン過程 (nonhomogeneous Poisson process, 以下 NHPP と略す) [2-5] を導入し、 $N(t)$ の期待値を $m(t)$ とすれば、NHPP にもとづくソフトウェア信頼度成長モデルは、

$$\Pr\{N(t) = n\} = \frac{\{m(t)\}^n}{n!} \exp[-m(t)]$$

$$(n = 0, 1, 2, \dots), \quad (1)$$

$$m(t) = \int_0^t h_m(t') dt', \quad (2)$$

と定式化され、ソフトウェアの信頼性評価において特に実用性の高さから有望視されている。ここで $m(t)$ は、NHPP の平均値関数 (mean value function) と呼ばれ、時刻 t までに発見される総期待フォールト数を表わす。また、 $h_m(t)$ は瞬間フォールト発見率を表わし、NHPP の強度関数 (intensity function) と呼ばれる。ここで、テスト工程のフォールト発見事象に対して、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を a とすると、 $m(\infty) = a$ を仮定すれば、式(1)より $N(t)$ の極限分布は

$$\lim_{t \rightarrow \infty} \Pr\{N(t) = n\} = \frac{a^n}{n!} e^{-a}, \quad (3)$$

となり、ポアソン分布に従うことになる。

本稿では、単位テスト時間当りに発見されるフォールト数が、その時点において残存するフォールト数に比例するものと仮定して構築された指数形ソフトウェア信頼度成長モデル (exponential software reliability growth model) [11] を考える。すなわち、式(2)において

$$m(t) = a(1 - e^{-bt}) \quad (a > 0, b > 0), \quad (4)$$

$$h_m(t) = abe^{-bt}, \quad (5)$$

である。ここで、パラメータ b は任意のテスト時刻における 1 個当りのフォールト発見率を表わす。このモデルは、テスト時間を CPU 時間 (実行時間) で計測するようなソフトウェア故障発生現象を記述したり、大規模ソフトウェアシステムの信頼性評価を行なうときによく適用される。

3. ソフトウェアのライフサイクルを考慮した最適リリース問題

本章では、コストの時間的価値を導入した最適リリース問題について議論する。この問題は、ソフトウェアがリリースされてから廃棄されるまでの期間 (本稿ではライフサイクルと呼ぶ) に一定値の仮定をおくことが多い従来の研究よりも進めて、確率変数とする場合の検討であり、大規模なソフトウェアシステムの開発・運用を対象とした場合に特に有効となる。

以下の諸量を定義する。

c_1 : テスト工程における 1 個当りのフォールト修正コスト。

c_2 : 運用段階における 1 個当りのフォールト修正 (保守) コスト ($c_2 > c_1 > 0$)。

c_3 : 単位テスト時間当りのテストコスト ($c_3 > 0$)。

T : テスト時間の長さ ($T \geq 0$)。

$C(T)$: 総テスト時間が T のときの総期待ソフトウェアコスト。

$G(t)$: テスト終了後のライフサイクルの長さ t に対する分布関数 ($t \geq 0$)。

α : コストの割引率。

ここでは、ソフトウェア開発の最終段階におけるテストを開始してからリリース後のソフトウェアが廃棄されるまでの時間が長期にわたるような大規模ソフトウェアを想定し、各コストパラメータの時間的価値を考慮している。式(4)および式(5)を用いることにより、総期待ソフトウェアコストは

$$C(T; \alpha) = \int_0^T [c_1 h_m(t) + c_3] e^{-\alpha t} dt$$

$$+ c_2 \int_T^{T+t} h_m(x) e^{-\alpha x} dx dG(t) \quad (6)$$

と表わされる。式(6)において、 $\alpha \rightarrow 0$ とすれば、従来より考察されている最適リリース問題の総期待ソフトウェアコストの定式化に帰着する。[7]

式(6)を最小にする $T = T^*$ を求めることにより、コスト評価基準による最適リリース時刻を決定することができる。これに対して、テスト工程におけるソフトウェアに対してあらかじめ設定された目標信頼度を満足しながら、総期待ソフトウェアコストを最小化する最適リリース問題について考察することも可能である。これは、コスト評価基準のみを考える場合に比べて、より実際的なリリース時刻の決定要因を考慮しており、また、信頼度成長に対するテストコストの影響度合い

を確認することにおいても有意義である。式(1)および式(4)の指数形ソフトウェア信頼度成長モデルでは、テストを T 時間実施した後の時間区間 $(T, T+x]$ においてソフトウェア故障の起こらない確率は

$$R(x|T) = \exp[-\{m(T+x) - m(T)\}] \quad (x \geq 0), \quad (7)$$

により与えられ、ソフトウェア信頼度 (software reliability) と呼ばれる。したがって、目標信頼度を R_0 ($0 < R_0 \leq 1$) とし、式(7)が R_0 を満足しながら、式(6)を最小化する最適リリース時刻 $T = T^*$ を求めればよい。

すなわち、最適リリース問題は所定の x の値に対して minimize $C(T; \alpha)$ subject to $R(x|T) \geq R_0$, (8) と定式化される。

3.1 コスト評価基準にもとづく場合

ライフサイクルの長さが次に示す確率密度関数をもつ切断正規分布 (truncated normal distribution) に従うものと仮定する。

$$\frac{dG(t)}{dt} = \frac{1}{A\sqrt{2}\pi\sigma} \exp\left[-\frac{(t-\mu)^2}{2\sigma^2}\right] \quad (t \geq 0), \quad A = \frac{1}{\sqrt{2}\pi\sigma} \int_0^\infty \exp\left[-\frac{(t-\mu)^2}{2\sigma^2}\right] dt. \quad (9)$$

式(9)を使って、第2章で議論した指数形ソフトウェア信頼度成長モデルを用いた場合の最適リリース方策を議論する。式(6)の $C(T; \alpha)$ を最小にする $T = T^*$ を求めるために、式(6)を時間 T について微分して 0 とおくことにより

$$h_m(T) = \frac{c_3}{c_2\gamma - c_1}, \quad (10)$$

なる関係式を得る。ここで、

$$\gamma = 1 - \frac{\Phi\left(\frac{\mu - \sigma^2(a+b)}{\sigma}\right)}{\Phi\left(-\frac{\mu}{\sigma}\right)} \exp\left[-\frac{2\mu(a+b) - (a+b)^2\sigma^2}{2}\right] \quad (11)$$

$$\Phi(x) = \frac{1}{\sqrt{2}\pi} \int_{-\infty}^x e^{-\frac{y^2}{2}} dy, \quad (12)$$

である。式(10)が有限かつ唯一の解をもつとき、その解を T_{ce} とすると

$$T_{ce} = \frac{1}{b} \ln\left[\frac{ab(c_2\gamma - c_1)}{c_3}\right], \quad (13)$$

である。したがって、最適リリース時刻 T^* に関して以下の方策を得る。

[方策 I - 1]

- (1) $h_m(0) > c_3 / (c_2\gamma - c_1)$ ならば、最適リリース時刻は $T^* = T_{ce}$ である。
- (2) $h_m(0) \leq c_3 / (c_2\gamma - c_1)$ ならば、 $T^* = 0$ である。

3.2 信頼度要求を考慮する場合

式(7)は、総テスト時間 T に関して単調増加関数であるので、 $R(x|0) < R_0 < 1$ のとき、目標信頼度 R_0 を満たすテスト時刻 $T = T_{re}$ が存在して、

$$T_{re} = \frac{1}{b} \ln\left[-\frac{m(x)}{\ln R_0}\right], \quad (14)$$

となる。したがって、式(8)の最適リリース問題に対して、以下の方策を得る。

[方策 I - 2]

- (1) $h_m(0) > c_3 / (c_2\gamma - c_1)$ かつ $R(x|0) < R_0$ ならば、最適リリース時刻は $T^* = \max\{T_{ce}, T_{re}\}$ である。
- (2) $h_m(0) > c_3 / (c_2\gamma - c_1)$ かつ $R(x|0) \geq R_0$ ならば、 $T^* = T_{ce}$ である。
- (3) $h_m(0) \leq c_3 / (c_2\gamma - c_1)$ かつ $R(x|0) < R_0$ ならば、 $T^* = T_{re}$ である。
- (4) $h_m(0) \leq c_3 / (c_2\gamma - c_1)$ かつ $R(x|0) \geq R_0$ ならば、 $T^* = 0$ である。

4. 保証期間を考慮した最適リリース問題

前章では、運用段階で発生する期待コストを、当該ソフトウェアのライフサイクル期間に対して算出した。しかし、運用段階では、ソフトウェアの保守活動として修正・改良・適用保守などが実施され、ソフトウェア自体が変化することになり潜在フォールト数も変化していく。したがって、最適リリース問題をソフトウェア開発の最終段階で考えるのであれば、ライフサイクルのような長期間を考えず、ソフトウェア自体が大きく変化しない所定の期間に対して、運用段階で発生する期待コストを算出するというアプローチも有効である。そこで本章では、開発者側がリリース後の保守コストを負担するような所定の保証期間 (warranty period) を設定する場合を考える。定式化のために、以下の諸量を定義する。

- c_0 : 最小限必要なテストコスト ($c_0 > 0$).
- c_t : 単位時間当たりのテストコスト ($c_t > 0$).
- c_w : 保証期間中に発見・修正されるフォールト 1 個当たりの保守コスト ($c_w > 0$).
- T : ソフトウェアの総テスト時間 ($T \geq 0$).

$C_w(T)$: 総テスト時間 T に関係したリリース後の保証期間に対するフォールト処理コスト。

$W(t)$: テスト終了時点から測定した保証期間の長さに対する分布関数 ($t \geq 0$)。

テスト終了時点以降のソフトウェアの保証期間が分布関数 $W(t)$ に従うことから、ソフトウェアのテスト開始後より保証期間が終了するまでに要する総期待ソフトウェアコストは、

$$WC(T) \equiv c_0 + c_t T + C_w(T), \quad (15)$$

により与えられる。したがって、テスト工程および運用段階におけるフォールト発見事象が式(4)の平均値関数 $m(t)$ をもつ NHPP モデルにより記述されるものと仮定すると、式(15)の $C_w(T)$ は次の3つの場合が考えられる (図1)。

[ケース1]: 保証期間は T_w (一定), すなわち $W(t) = U(t - T_w)$, かつテスト終了時点で信頼度成長が停止する。ここで、 $U(t) = 1 (t \geq 0)$, $U(t) = 0 (t < 0)$ とする。このとき、

$$C_w(T) = c_w h_m(T) T_w, \quad (16)$$

である。

[ケース2]: 保証期間は T_w (一定) かつテスト終了後も信頼度成長が継続する。このとき、

$$C_w(T) = c_w [m(T + T_w) - m(T)], \quad (17)$$

である。

[ケース3]: 保証期間は一般分布 $W(t)$ に従い、かつテスト終了後も信頼度成長が継続する。このとき、

$$C_w(T) = c_w \int_0^\infty [m(T + T_w) - m(T)] dW(T_w), \quad (18)$$

である。

ここでは、保証期間の分布関数に式(9)の切断正規分布を仮定する。なお、保証期間が一般分布 $W(t)$ に従い、かつテスト終了時点で信頼度成長が停止する場合は、[ケース1]と同等となる。

上の3つのケースについて、式(15)の総期待ソフトウェアコスト $WC(T)$ が最小となるような最適リリース時刻 $T = T^*$ を求めると、容易に各ケースに対して以下のような最適方策を得る。なお、コスト評価基準だけでなく、信頼度要求を考慮する場合も、第3章と同様に議論できる。

[方策II-1 (ケース1)]

(1) $h_m(0) > c_t / (c_w b T_w)$ ならば、最適リリース時刻は

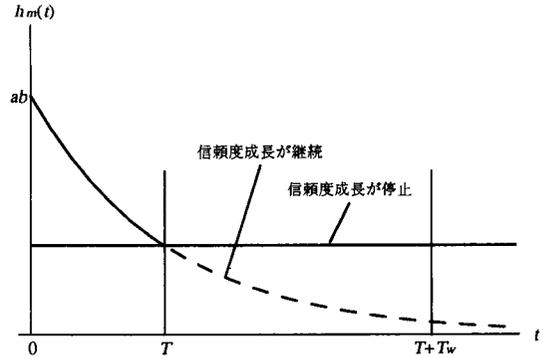


図1 保証期間の信頼度成長過程

$$T^* = \frac{1}{b} \ln \left[\frac{c_w a b^2 T_w}{c_t} \right], \quad (19)$$

である。

(2) $h_m(0) \leq c_t / (c_w b T_w)$ ならば、 $T^* = 0$ である。

[方策II-2 (ケース2)]

(1) $h_m(0) > c_t / (c_w (1 - e^{-b T_w}))$ ならば、最適リリース時刻は

$$T^* = \frac{1}{b} \ln \left[\frac{c_w a b (1 - e^{-b T_w})}{c_t} \right], \quad (20)$$

である。

(2) $h_m(0) \leq c_t / (c_w (1 - e^{-b T_w}))$ ならば、 $T^* = 0$ である。

[方策II-3 (ケース3)]

(1) $h_m(0) > c_t / (c_w \xi)$ ならば、最適リリース時刻は

$$T^* = \frac{1}{b} \ln \left[\frac{c_w a b \xi}{c_t} \right], \quad (21)$$

である。

(2) $h_m(0) \leq c_t / (c_w \xi)$ ならば、 $T^* = 0$ である。ここで、

$$\xi = 1 - \frac{e^{-\frac{\sigma^2 b^2}{2} - b\mu} \{1 - \Phi(-\frac{\mu}{\sigma} + b\sigma)\}}{1 - \Phi(-\frac{\mu}{\sigma})}, \quad (22)$$

である。

5. 数値例

ここでは、第3章および第4章で得られたソフトウェアの最適リリース問題の各最適方策にもとづく数値例を示す。まず、文献[9]で適用された累積発見フォールト数に関するデータを用いることとし、これに式(1)および式(4)で与えられる指数形ソフトウェア信頼度成長モデルを適用した。最尤法を用いて未知パラメータを推定すると、 $\hat{a} = 513.2$ および $\hat{b} = 5.365 \times 10^{-2}$ となった。以下では、この推定結果を用いる。

5.1 ライフサイクルを考慮した最適リリース問題に対する数値例

まず、第3章で議論したソフトウェアのライフサイクルを考慮した場合について、各コストパラメータを、 $c_1=1.0$ として c_2 および c_3 の値を変化させた場合の、[方策I-1]にもとづく最適リリース時刻の変化を表1に示す。これらのコストパラメータの値は相対的なコスト比と見なすことができる。ここで、式(9)で与えられるライフサイクルの確率分布のパラメータ、および割引率の値を、それぞれ $\mu=100$ 、 $\sigma=10$ 、および $\alpha=0.001$ とした。このとき、式(11)により与えられる γ の値は $\gamma=0.9167$ となった。表1から、運用段階におけるフォールト修正コストが増加するとソフトウェアのリリース時期は遅くなって十分にテストを行なう必要があり、逆に単位時間当りのテストコストが増加するとリリース時刻は早める必要があることがわかる。次に、ライフサイクルのパラメータ μ と割引率 α をそれぞれ変化させたときの最適リリース時刻の変化を表2に示す。ここで、 $c_1=1.0$ 、 $c_2=20.0$ および $c_3=10.0$ とし、ライフサイクルの分布の変動係数に相当する値を $\sigma/\mu=0.10$ とした。表2より、割引率 α の値が増加するとリリース時刻は早める必要があるといえるが、その平均値に相当する μ の増加によるリリース時刻の変動は非線形的な関係があることがわかる。

次に、信頼度要求を考慮した場合の最適リリース時刻について数値例を示す。コストパラメータの値を $c_1=1.0$ 、 $c_2=20.0$ 、および $c_3=10.0$ とし、 $\mu=100$ 、 $\sigma=10$ 、および $\alpha=0.001$ とする。このとき、コスト評価基準による最適リリース時刻は表1より $T_{c_2}=72.0$ である。そこで、テスト終了後、運用時間 $x=1.0$ (週)に対して目標信頼度を $R_0=0.9$ に設定したものとす。これを達成するのに必要な総テスト時間は式(7)より、 $T_{Re}=103.2$ であることがわかる。したがって、[方策I-2]の(1)より、最適リリース時刻は $T^*=T_{Re}=103.2$ となる。

5.2 保証期間を考慮した最適リリース問題に対する数値例

前節と同様に、第4章で得られた各最適方策にもとづく数値例を示す。保証期間中の1個当りのフォールト保守コストを $c_w=1.0$ 、テスト工程における最小限必要なコストを $c_0=1000.0$ とする。

表1 ライフサイクルを考慮した最適リリース時刻(1)

($\alpha = 0.001, \mu = 100, \sigma = 10, c_1 = 1.0$)

		c_2					
		2.0	5.0	10.0	20.0	50.0	100.0
c_3	1.0	58.4	85.9	100.9	115.0	132.7	145.8
	5.0	28.4	55.6	70.9	85.0	102.7	115.8
	10.0	15.5	42.7	58.0	72.0	89.8	102.9
	50.0	0	12.7	28.0	42.1	59.8	72.9
	100.0	0	0	15.1	29.1	46.8	60.0

表2 ライフサイクルを考慮した最適リリース時刻(2)

($c_1 = 1.0, c_2 = 20.0, c_3 = 10.0, \sigma/\mu = 0.10$)

		α				
		0	0.0001	0.001	0.005	0.01
μ	50	72.3	72.3	71.7	69.1	65.7
	60	72.9	72.8	72.1	68.7	64.3
	70	73.3	73.2	72.2	68.1	62.6
	80	73.5	73.3	72.3	67.3	60.8
	90	73.6	73.4	72.2	66.4	58.8
	100	73.7	73.5	72.0	65.5	56.8
	150	73.8	73.5	71.2	60.4	45.8
	200	73.8	73.4	70.2	55.2	33.1
	300	73.8	73.2	68.2	43.9	0
	400	73.8	73.0	66.2	30.8	0
	500	73.8	72.8	64.1	13.3	0

まず、[ケース1]において、単位時間当りのテストコスト c_t および保証期間の長さ T_w を変化させて、[方策II-1]にもとづく最適リリース時刻を求めたのが表3である。表3より、単位時間当りのテストコストが高くなるにつれて最適リリース時刻は早くなることがわかる。また、保証期間が長くなるにつれて最適リリース時刻は遅くなり、テストを十分に行なう必要があることがわかる。

次に、[ケース2]に対する[方策II-2]の最適リリース時刻を求めたのが表4である。表3および表4を比較すると、同じ条件の下では[ケース1]に比べて[ケース2]の方が最適リリース時刻は早くなることがわかる。これは[ケース2]の場合、テスト終了後もソフトウェアの信頼度成長が継続するため、運用段階では時間の経過に伴って発見されるフォールト数が[ケース1]の場合に比べて少なくなるからである。

6. むすび

本稿では、ソフトウェアの開発管理上重要な問題として、適切なリリース時刻を総期待ソフトウェアコストにもとづいて定量的に決定する最適リリース問題について議論した。特に、コストの時間的価値とライフサイクルの不確実性を考慮した場合と、ユーザに対する保証期間を考慮した場合について最適方策を導出した。問題の定式化に際して、NHPPにもとづく指数形ソフトウェア信頼度成長モデルを用いて、テスト工程および運用段階におけるフォールト発見事象を記述

表3 保証期間を考慮した最適リリース時刻 [方策II-1]

($c_0 = 1000.0, c_w = 1.0$)

		c_t					
		0.5	1.0	10.0	30.0	50.0	100.0
T_w	1.0	20.2	7.3	0	0	0	0
	2.0	33.1	20.2	0	0	0	0
	5.0	50.2	37.3	0	0	0	0
	10.0	63.1	50.2	7.3	0	0	0
	20.0	76.0	63.1	20.2	0	0	0
	50.0	93.1	80.2	37.3	16.8	7.3	0
	100.0	106.0	93.1	50.2	29.7	20.2	7.3

表4 保証期間を考慮した最適リリース時刻 [方策II-2]

($c_0 = 1000.0, c_w = 1.0$)

		c_t					
		0.5	1.0	10.0	30.0	50.0	100.0
T_w	1.0	19.7	6.8	0	0	0	0
	2.0	32.1	19.2	0	0	0	0
	5.0	47.7	34.8	0	0	0	0
	10.0	58.3	45.4	2.5	0	0	0
	20.0	66.9	54.0	11.1	0	0	0
	50.0	73.4	60.5	17.6	0	0	0
	100.0	74.6	61.7	18.8	0	0	0

した。

今後の課題として、本稿で得られた最適リリース方策の実際のソフトウェアプロジェクトに対する妥当性の検証や、コストパラメータの実際の与え方、あるいは見積り方法などについて検討する必要がある。なお、本稿では、指数形ソフトウェア信頼度成長モデルのみを取り扱ったが、他のNHPPモデルを適用して最適リリース方策を同様に議論することができる。

参考文献

- [1] Malaiya, Y.K. and Srimani, P. K. (eds.) : "Software Reliability Models : Theoretical Developments, Evaluation and Application", IEEE Computer Society Press, Los Alamitos, CA (1991).
- [2] Musa, J. D. : "Software Reliability : Measurement, Prediction, Application", McGraw - Hill, New York (1987).
- [3] 山田茂 : "ソフトウェア信頼性評価技術", HBJ 出版局, 東京 (1989).
- [4] 山田茂, 大寺浩志 : "ソフトウェアの信頼性 ~理論と実践的応用~", ソフト・リサーチ・センター, 東京 (1990).
- [5] 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).
- [6] Koch, H. S. and Kubat, P. : "Optimum release time for computer software", IEEE Trans. Software Eng., Vol. SE -9, No. 3, pp. 323 -327 (1983).
- [7] Okumoto, K. and Goel, A. L. : "Optimum release time for software systems based on reliability and cost criteria", J. Syst. & Software, 1, pp. 315-318 (1980).
- [8] Yamada, S. and Osaki, S. : "Optimal software release policies with simultaneous cost and reliability requirements", European J. Operational Research, Vol. 31, No. 1, pp. 46-51 (1987).
- [9] Goel, A.L. and Okumoto, K. : "Time - dependent error - detection rate model for software reliability and other performance measures", IEEE Trans. Reliability, Vol. R -28, No. 3, pp. 206-211 (1979).
- [10] 山田茂, 木村光宏, 寺上英治, 尾崎俊治 : "ライフサイクルの確率分布とコストの時間的価値を考慮したソフトウェアの最適リリース問題", 情報処理学会論文誌, Vol. 34, No. 5, pp. 1188-1197 (1993).
- [11] 山田茂 : "ソフトウェア保守コストモデルに基づく保証期間を考慮した最適リリース問題", 情報処理学会論文誌, Vol. 35, No. 10, pp. 2197-2202 (1994).