

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

山 田 茂[†] 木 村 光 宏[†]
寺 上 英 治[†] 尾 崎 俊 治[†]

信頼性の高いソフトウェアを効率的に生産するためには、品質、納期、および費用の観点からソフトウェア開発の各工程に対して検討を加える管理技術が必要である。特に、ソフトウェア開発の最終段階であるテスト工程では、テストの進捗状況の把握とユーザにソフトウェアを引き渡すのに最適な時期を決定することが重要な問題となる。これはソフトウェアの最適リリース問題 (optimal software release problem) と呼ばれているが、従来から、ソフトウェアのリリース時期については、納期を重視するあまりに開発管理者の経験や勘により決定されることも多かった。本論文では、ソフトウェアの信頼度成長モデルによる信頼性評価結果の開発管理への応用の一つとして、コスト評価基準によるソフトウェアの最適リリース問題を議論する。特に、ソフトウェアのライフサイクルの長さに着目してその変動を確率分布で表し、さらに総期待ソフトウェアコストの現在価値を考慮する。

Optimal Software Release Problems with Life-Cycle Distribution and Discount Rate

SHIGERU YAMADA,[†] MITSUHIRO KIMURA,[†] EIJI TERAUE[†] and SHUNJI OSAKI[†]

It is necessary for a software development manager to investigate the progress of each phase of the development process from the view points of many management factors such as quality, delivery time, cost, and so on. In particular, it is very important to determine the time when to stop the testing and deliver the software product to the user. Such a decision-making problem is called an optimal software release problem, and several researchers have treated this problem in which the optimal software release time is determined both by minimizing a total expected software cost and achieving a software reliability objective. In this paper, we discuss optimal software release problems based on cost and reliability criteria, as one of applications of software reliability growth models to controlling the testing phase of software development. Especially, we describe the variation of life-cycle duration of a software system as a probability distribution and take into account of the present value of the total expected software cost.

1. ま え が き

今日、コンピュータシステムが我々の社会生活の多くの分野において利用され、より複雑な処理を高速で行う大規模なシステムが要求されるようになってきている。このようなシステムを構成するソフトウェアも、数百万ステップを越える大規模なものとなってきた。現在、ソフトウェアの開発手法はウォーターフォール型の開発過程に従う人的作業の積み重ねからなるものが多い。したがって、設計などの上流工程で多くの人

が要求仕様に従って期待通りに動作することを確認し、ソフトウェア内に潜在している論理的な誤り、いわゆるソフトウェアエラー（以下エラーと略す）を発見し修正することを目的としたテスト工程が重要となる。このテスト工程では、実際の運用環境を反映したテスト用入力データがソフトウェアに入力され、出力との関係を吟味することによってソフトウェアのテストが行われている。

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経験などから行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エラー発見現象に確

[†] 広島大学工学部第二類（電気系）
Department of Industrial and Systems Engineering,
Faculty of Engineering, Hiroshima University

率・統計論を導入したソフトウェア信頼度成長モデル (software reliability growth model) が数多く提案されてきている^{1)~4)}。このモデルを用いることの利点は、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、および期待残存エラー数など) を予測して、ソフトウェア開発における管理上の意志決定を支援することができる点である。このことにより、テスト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の関係から、ソフトウェアの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、これまでいくつかの研究が報告されている^{5)~8)}。これらの研究の多くは、ソフトウェアが廃棄されるまでの期間、すなわちソフトウェアのライフサイクルの長さを一定として取り扱っていた。しかし、実際はユーザによりソフトウェアが廃棄される時刻は様々である。

本論文では、テスト終了時点から測ったソフトウェアのライフサイクルの長さが適当な確率分布に従うものと仮定し、テスト工程および運用段階におけるエラー発見事象をソフトウェア信頼度成長モデルにより記述する。ここで、最適リリース時刻の決定に関する評価基準として、テスト工程および実際の運用段階において発生するコスト要因を考慮した総期待ソフトウェアコストと、テストにより確認される信頼性の達成度合を示すソフトウェア信頼度の2つを採用する。これら2つの評価基準は、ソフトウェア信頼度成長モデルを使って導出される。また、最適リリース時刻の決定に関する評価基準として総期待ソフトウェアコストを採用する場合、大規模ソフトウェアを想定して、運用期間が長期に渡ることを考慮するためにコストの時間的価値を導入する。以上の仮定のもとで、総期待ソフトウェアコストの最小化、ソフトウェアの具備している信頼度、およびこれらを同時に考慮した最適リリース時刻に関する方策を導出する。さらに、以上の最適リリース時刻に関する方策の数値例も示す。

2. ソフトウェア信頼度成長モデル

開発プロセスのテスト工程、あるいは運用段階にあるソフトウェアの信頼性を定量的に評価するために、従来より様々な数理モデルが提案されている。ソフト

ウェアが開発者による知的生産物であるため、その開発プロセスにおいて多くのエラーが潜入することは避けられない。これらのエラーによって引き起こされるソフトウェア故障現象やその原因となったエラーの発見事象は不確定現象であるため、その中からソフトウェアの信頼性特性を抽出し、計量化するためには確率・統計論を適用するのが適切である。ここで、ソフトウェア故障とはソフトウェア内に潜在するエラーにより、期待通りにソフトウェアが動作しないことと定義する。このような不確定現象を記述し、ソフトウェアの信頼性を評価する数理モデルはソフトウェア信頼性モデルと呼ばれる。特に、ソフトウェア開発のテスト工程では、所定の手続きに従ってソフトウェア内に潜在するエラーを発見し、それを修正する作業が行われる。ここで、エラーの修正時に新たなエラーは作り込まれないものと仮定すると、テスト時間の経過とともにソフトウェア内に潜在しているエラー数は減少していくので、ソフトウェア故障の起こらない確率、すなわちソフトウェアの信頼度は増加していくことになる。このような現象を記述するソフトウェア信頼性モデルは、ソフトウェア信頼度成長モデルと呼ばれる。

本論文では、テスト工程あるいは運用段階において発見されるエラー数の累積値を確率変数として捉える非同次ポアソン過程 (nonhomogeneous Poisson process, 以下 NHPP と略す) モデルを用いることにする。いま、テスト時刻 t において発見されたエラー数を $N(t)$ とすると、NHPP によりテスト工程あるいは運用段階におけるエラー発見事象を次式で記述できる。

$$\Pr[N(t)=n] = \frac{H(t)^n}{n!} \exp[-H(t)] \quad (n=0, 1, 2, \dots) \quad (1)$$

$$H(t) = \int_0^t h(x) dx. \quad (2)$$

ここで、 $H(t)$ は時間区間 $[0, t]$ における発見エラー数の期待値を表す NHPP の平均値関数 (mean value function) であり、 $h(t)$ は時刻 t におけるエラー発見率を表す NHPP の強度関数 (intensity function) と呼ばれる。

実際に NHPP モデルを用いて信頼性評価を行うには、平均値関数に具体的な物理的意味を持つ関数形を指定しなければならない。本論文では、テスト工程において発見される総エラー数の成長曲線が指数 (関数) 形を示す場合に適切であり、以下に述べる2つの平均値関数をもつ NHPP モデルについて議論する。

(1) 指数形ソフトウェア信頼度成長モデル

(exponential software reliability growth model)⁹⁾

指数形ソフトウェア信頼度成長モデルは、ソフトウェア故障発生現象を記述するのに適切であり、NHPPの平均値関数および強度関数は次式により与えられる。

$$H(t) \equiv H_e(t) = a(1 - e^{-bt}),$$

$$h(t) \equiv h_e(t) = abe^{-bt} \quad (a > 0, b > 0). \quad (3)$$

ここで、 a はテスト開始前にソフトウェア内に潜在する初期内蔵エラー数であり、 b は1個当りのエラー発見率である。

(2) 修正指数形ソフトウェア信頼度成長モデル

(modified exponential software reliability growth model)¹⁰⁾

修正指数形ソフトウェア信頼度成長モデルは、テスト初期において発見されやすいエラーと、中期以降に発見されやすいエラーとを本質的に性質が異なるものとしてとらえ、前者をタイプ1エラー、後者をタイプ2エラーとしてエラーの発見難易度を区別することにより、テスト工程におけるエラー発見過程をより詳細に記述しようとするものである。NHPPの平均値関数および強度関数は、次式により与えられる。

$$H(t) \equiv H_p(t) = a \sum_{i=1}^2 p_i (1 - e^{-b_i t}),$$

$$h(t) \equiv h_p(t) = a \sum_{i=1}^2 p_i b_i e^{-b_i t}$$

$$\left(a > 0, 0 < b_2 < b_1 < 1, \sum_{i=1}^2 p_i = 1, 0 < p_i < 1 \right).$$

$$(4)$$

ここで、 a はテスト開始前にソフトウェア内に潜在する初期内蔵エラー数であり、 b_1 および b_2 はそれぞれタイプ1エラーおよびタイプ2エラー1個当りの発見率である。また p_i はソフトウェア内におけるタイプ*i*エラーの含有率を意味する ($i=1, 2$)。

3. 最適リリース問題

ソフトウェア信頼度成長モデルに基づいてテスト工程で観測された発見エラー数に関するデータを解析した後の興味ある問題として、テスト工程にあるソフトウェアを実際の運用段階に移行するのに最適な時期を決定するという問題が挙げられる。これは適当な評価基準を導入し、テスト時間とテストにより確認されるソフトウェアの信頼性の関係を反映するように定式化され、ソフトウェアの最適リリース問題と呼ばれ

る^{5)~8)}。

ソフトウェアの性能や信頼性は、適用されるテスト技法やテストツールの良悪とともにテストに費やされる時間に依存している。テスト時間が長ければ、ソフトウェア内に残存するエラー数は数多く発見・修正され、運用段階に移行した後の信頼性は向上する。しかし、テスト実施に伴うコストは増加し、運用開始時刻(ソフトウェアの納期)は遅れることになる。一方、テスト時間が短かければ、テストに要するコストは減少するが、ソフトウェア内に残存している未発見のエラーが多いために、運用段階へ移行した後に多くのソフトウェア故障が発生するであろう。このため、運用段階におけるソフトウェアの保守コストが増大することになる。一般に、運用段階で発見されるエラーの修正コストは、テスト工程におけるそれよりも大きい。したがって、ソフトウェアの最適リリース問題として、テスト工程と運用段階において発生するエラーの修正コストを考慮して、総期待ソフトウェアコストを最小化する最適リリース時刻を求めることが考えられる。

本論文では、評価基準として代表的なコスト要因を考慮した、総期待ソフトウェアコストを採用し、これを最小化するような最適リリース時刻を決定する方策について議論する。ここで、より現実的な運用期間を考えるため、ソフトウェアのライフサイクル、すなわちテストを終了して運用段階に移行し、そのソフトウェアが廃棄されるまでの時間が確率分布により記述されるものとする。さらに、大規模なソフトウェアシステムを対象とした場合に特に有効となる。コストの時間的価値を導入する。

最適リリース問題を定式化するにあたり、以下の諸量を定義する。

c_1 : テスト工程における1個当りのエラー修正コスト。

c_2 : 運用段階における1個当りのエラー修正(あるいは保守)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 : テスト期間の長さ ($T \geq 0$)。

$C(T)$: 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。

3.1 基本問題

テスト終了時点以降のソフトウェアのライフサイクルが分布関数 $G(t)$ に従うものとする。いま、テスト工程および運用段階におけるエラー発見事象が、式(1)および式(2)の NHPP モデルにより記述されるものと仮定する。このとき、テスト工程および運用段階において発見されるエラー修正コストと、テスト実施に伴うコストを考慮して、通常の NHPP モデルに基づく最適リリース問題 (例えば Okumoto and Goel⁶⁾, Koch and Kubat⁵⁾, Yamada and Osaki⁸⁾参照) と同様にして、総期待ソフトウェアコストは、

$$C(T) = c_1 H(T) + c_3 T + c_2 \int_0^{\infty} \{H(t+T) - H(T)\} dG(t), \quad (5)$$

により与えられる。したがって、式(5)を最小にする総テスト時刻 $T = T^*$ が最適リリース時刻となる。式(5)において、分布関数 $G(t)$ を一定分布

$$G(t) = U(t - t_0) \quad (t_0 > 0) \quad (6)$$

とすれば、従来の最適リリース問題 (例えば、文献6)および8)に帰着することが示される。ここで、 t_0 はテスト終了時点から測定したライフサイクルの長さであり、 $U(t)$ はステップ関数を表し、 $U(t) = 1$ ($t \geq 0$), $U(t) = 0$ ($t < 0$) である。これは式(5)の定式化が、ライフサイクルの時間的変動を確率分布を用いて表すことにより、従来の最適リリース問題を特別な場合として含むような、より一般性のあることを意味している。

3.2 コストの時間的価値を考慮する場合

式(5)により表される総期待ソフトウェアコストにおけるコストパラメータ c_1 , c_2 , および c_3 はそれぞれ経過時間に関して一定であった。しかし、ソフトウェア開発におけるテストを開始してからそのソフトウェアが廃棄されるまでの時間が長期に渡るような大規模ソフトウェアを想定する場合、各コストパラメータの時間的価値は低下していくことになる。そこで、コストの時間的価値を考慮して (例えば尾崎ら¹¹⁾参照) 割引率 α を導入することにより、式(5)は

$$C(T; \alpha) = \int_0^T [c_1 h(t) + c_3] e^{-\alpha t} dt + c_2 \int_0^{\infty} [H(t+T) - H(T)] \cdot e^{-\alpha(t+T)} dG(t), \quad (7)$$

と書き換えられる。式(7)において $\alpha = 0$ とすれば式(5)に帰着できるので、以後は式(7)を本論文で考察する総期待ソフトウェアコストの評価基準式として用

いることにする。

3.3 信頼度要求を考慮する場合

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。これは、コスト評価基準のみを考える場合に比べて、より現実的な要因を考慮しており、またテストコストの信頼度成長に対する影響度合を確認することにおいても有意義である。式(1)および式(2)の NHPP に基づくソフトウェア信頼度成長モデルでは、テストを T 時間行った後の時間区間 $(T, T+x]$ に対するソフトウェアの信頼度、すなわちソフトウェア故障の起こらない確率は

$$R(x|T) = \exp[-\{H(T+x) - H(T)\}] \quad (x \geq 0), \quad (8)$$

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 ($0 < R_0 \leq 1$) として、式(8)が R_0 を満足しながら式(7)を最小化するような最適リリース時刻 $T = T^*$ を求めればよい。すなわち、最適リリース問題は x の所定の値に対して

$$\text{minimize } C(T; \alpha) \text{ subject to } R(x|T) \geq R_0, \quad (9)$$

と定式化される。

4. 最適方策

4.1 コストの時間的価値を考慮した基本問題に対する解析

前章で定式化された各最適リリース問題に対して、ライフサイクルの期間の長さが以下に示す密度関数をもつ切断正規分布 (truncated normal distribution)¹²⁾ に従うものと仮定する。

$$\frac{dG(t)}{dt} = \frac{1}{A\sqrt{2\pi}\sigma} \exp\left[-\frac{(t-\mu)^2}{2\sigma^2}\right], \\ A = \frac{1}{\sqrt{2\pi}\sigma} \int_0^{\infty} \exp\left[-\frac{(t-\mu)^2}{2\sigma^2}\right] dt \quad (t \geq 0). \quad (10)$$

式(10)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について解析を行う。

式(7)の総期待ソフトウェアコストは

$$C(T; \alpha) = \int_0^T [c_1 h_e(t) + c_3] e^{-\alpha t} dt + c_2 \int_0^{\infty} [H_e(t+T) - H_e(T)] \cdot e^{-\alpha(t+T)} dG(t), \quad (11)$$

となる。式(11)の $C(T; \alpha)$ を最小にする $T=T^*$ を求めるために、式(11)を時間 T について微分して 0 とおくことにより

$$h_e(T) = \frac{c_3}{c_2\gamma - c_1}, \quad (12)$$

なる関係式を得る。ここで、

$$\gamma = \frac{\alpha + b}{b} \left\{ \exp \left[-\alpha\mu + \frac{\alpha^2\sigma^2}{2} \right] \frac{1 - \Phi \left(\frac{-\mu - \alpha\sigma^2}{\sigma} \right)}{1 - \Phi \left(-\frac{\mu}{\sigma} \right)} - \exp \left[-(\alpha + b)\mu + \frac{(\alpha + b)^2\sigma^2}{2} \right] \frac{1 - \Phi \left(\frac{-\mu - (\alpha + b)\sigma^2}{\sigma} \right)}{1 - \Phi \left(-\frac{\mu}{\sigma} \right)} \right\}, \quad (13)$$

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-y^2/2} dy, \quad (14)$$

である。

式(12)における強度関数 $h_e(T)$ が $h_e(0) = ab$ とする T に関する単調減少関数であることから、 $h_e(0) > c_3/(c_2\gamma - c_1)$ の関係が満足されるとき式(12)は有限かつ唯一の解をもち、その解を T_{Ce} とすると

$$T_{Ce} = \frac{1}{b} \ln \left[\frac{ab(c_2\gamma - c_1)}{c_3} \right], \quad (15)$$

を得る。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 1]

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

つぎに、修正指数形ソフトウェア信頼度成長モデルを用いた場合について同様に解析を行う。このとき、式(7)の総期待ソフトウェアコストは

$$C(T; \alpha) = \int_0^T [c_1 h_p(t) + c_3] e^{-\alpha t} dt + c_2 \int_0^\infty [H_p(t+T) - H_p(T)] \cdot e^{-\alpha(t+T)} dG(t), \quad (16)$$

となる。式(16)の $C(T; \alpha)$ を最小にする $T=T^*$ を求めるために、式(16)を時間 T について微分することにより

$$\begin{aligned} \frac{dC(T; \alpha)}{dT} &\equiv C'(T; \alpha) \\ &= c_3 - \alpha \sum_{i=1}^2 (c_2\gamma_i - c_1) p_i b_i e^{-b_i T} = 0, \end{aligned} \quad (17)$$

なる関係を得る。ここで、

$$\gamma_i = \frac{\alpha + b_i}{b_i} \left\{ \exp \left[-\alpha\mu + \frac{\alpha^2\sigma^2}{2} \right] \frac{1 - \Phi \left(\frac{-\mu - \alpha\sigma^2}{\sigma} \right)}{1 - \Phi \left(-\frac{\mu}{\sigma} \right)} - \exp \left[-(\alpha + b_i)\mu + \frac{(\alpha + b_i)^2\sigma^2}{2} \right] \frac{1 - \Phi \left(\frac{-\mu - (\alpha + b_i)\sigma^2}{\sigma} \right)}{1 - \Phi \left(-\frac{\mu}{\sigma} \right)} \right\} \quad (i=1, 2), \quad (18)$$

である。

式(17)は $C'(0; \alpha) < 0$ ならば有限解が存在し、その解を T_{Cp} とすると次の方策を得る。

[方策 2]

- (1) $C'(0) < 0$ ならば、最適リリース時刻は $T^* = T_{Cp}$ である。
- (2) $C'(0) \geq 0$ ならば、 $T^* = 0$ である。

4.2 信頼度要求を考慮した問題に対する解析

指数形ソフトウェア信頼度成長モデルの時刻 T におけるソフトウェア信頼度は、式(8)より

$$R(x|T) = \exp[-\{H_e(T+x) - H_e(T)\}], \quad (19)$$

となる。式(19)は総テスト時間 T に関して $R(x|0) = \exp[-H_e(x)]$ および $R(x|\infty) = 1$ の単調増加関数であるので、目標信頼度が R_0 のとき、これを満足するテスト時刻 T_{Re} が存在して、

$$T_{Re} = \frac{1}{b} \ln \left[-\frac{H_e(x)}{\ln R_0} \right], \quad (20)$$

となる。したがって、式(9)の最適リリース問題に対して、以下の方策を得る。

[方策 3]

- (1) $h_e(0) > c_3/(c_2\gamma - c_1)$ 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) $h_e(0) > c_3/(c_2\gamma - c_1)$ かつ $R(x|0) \geq R_0$ ならば、 $T^* = T_{Ce}$ である。
- (3) $h_e(0) \leq c_3/(c_2\gamma - c_1)$ かつ $R(x|0) < R_0$ ならば、 $T^* = T_{Re}$ である。
- (4) $h_e(0) \leq c_3/(c_2\gamma - c_1)$ かつ $R(x|0) \geq R_0$ ならば、 $T^* = 0$ である。

修正指数形ソフトウェア信頼度成長モデルの場合、目標信頼度 R_0 を達成するテスト時刻すなわち

$$H_p(T+x) - H_p(T) = \ln \left(\frac{1}{R_0} \right), \quad (21)$$

を満足するテスト時刻を $T = T_{Rp}$ とすると、[方策 3]

と同様にして以下の方策を得ることができる。

【方策4】

- (1) $C'(0) < 0$ かつ $R(x|0) < R_0$ ならば、最適リリース時刻は $T^* = \max\{T_{c_p}, T_{R_p}\}$ である。
- (2) $C'(0) < 0$ かつ $R(x|0) \geq R_0$ ならば、 $T^* = T_{c_p}$ である。
- (3) $C'(0) \geq 0$ かつ $R(x|0) < R_0$ ならば、 $T^* = T_{R_p}$ である。
- (4) $C'(0) \geq 0$ かつ $R(x|0) \geq R_0$ ならば、 $T^* = 0$ である。

なお、上記の4つの方策は、式(13)および式(18)における σ を $\sigma \rightarrow 0$ とすることにより、ライフサイクルを確定値として取り扱った従来のモデルによる方策に一致することを付記しておく。

5. 数 値 例

本章では、実際のソフトウェア開発のテスト工程で得られたエラー発見数に関するデータ（以下、DS-1と呼ぶ）を用いて、前章で得られた各最適方策に基づくソフトウェアの最適リリース問題の数値例を示す。DS-1はOhba¹³⁾が引用したもので $\{t_k, y_k\}$ ($k=1, 2, \dots, 19$) の形で与えられている。ここに t_k はテスト時刻(週)であり、 y_k はテスト時刻 t_k までに発見された累積エラー数である。またデータはPL/I言語によるアプリケーションソフトウェアから採取されたものである。

5.1 指数形ソフトウェア信頼度成長モデルの場合

まず、指数形ソフトウェア信頼度成長モデルに基づく場合について考察する。式(3)におけるモデルパラメータ a および b の値を最尤法^{2),3)}により推定すると、それぞれ $\hat{a}=513.2$ および $\hat{b}=0.5365 \times 10^{-1}$ となった。そこで、 $c_1=1$ として c_2 および c_3 の値を変化させた場合の、【方策1】に基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(10)で与えられるライフサイクルの確率分布の近似的な平均値と標準偏差、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(13)により与えられる γ の値は $\gamma=0.9167$ とな

った。表1から、運用段階におけるエラー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。つぎに、ライフサイクルの平均値 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、 $c_1=1$, $c_2=20$, および $c_3=10$ とし、ライフサイクルの変動係数の逆数に相当する値を $\mu/\sigma=10$ 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、平均値 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。

つぎに、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

表1 指数形ソフトウェア信頼度成長モデルに基づく最適リリース時刻 (1)
($\alpha=0.001$, $\mu=100$, $\sigma=10$, $c_1=1.0$)

Table 1 The optimum release time based on exponential software reliability growth model (1)
($\alpha=0.001$, $\mu=100$, $\sigma=10$, $c_1=1.0$).

C_2	2	5	10	20	50	100
C_3						
1	58.40	85.58	100.93	114.96	132.67	145.80
5	28.40	55.58	70.94	84.96	102.68	115.80
10	15.48	42.66	58.02	72.04	89.76	102.88
50	0	12.67	28.02	42.05	59.76	72.89
100	0	0	15.10	29.12	46.84	59.97

表2 指数形ソフトウェア信頼度成長モデルに基づく最適リリース時刻 (2)

$$(c_1=1.0, c_2=20.0, c_3=10.0, \frac{\mu}{\sigma}=10)$$

Table 2 The optimum release time based on exponential software reliability growth model (2)

$$(c_1=1.0, c_2=20.0, c_3=10.0, \frac{\mu}{\sigma}=10).$$

μ	α	0.0	0.0001	0.001	0.005	0.01
50		72.31	72.25	71.69	69.11	65.71
60		72.91	72.83	72.09	68.72	64.30
70		73.26	73.16	72.24	68.07	62.60
80		73.46	73.34	72.25	67.28	60.76
90		73.58	73.43	72.17	66.39	58.80
100		73.65	73.49	72.04	65.45	56.76
150		73.75	73.49	71.16	60.43	45.77
200		73.76	73.40	70.17	55.15	33.12
300		73.76	73.20	68.18	43.88	0
400		73.76	73.01	66.17	30.84	0
500		73.76	72.81	64.14	13.25	0

$c_1=1$, $c_2=20$, および $c_3=10$ 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 $T_{Ce}=72.04$ 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(19)より, $T_{Re}=103.24$ であることがわかる. したがって, [方策3]の(1)より, 最適リリース時刻は $T^*=T_{Re}=103.24$ (週) となる. この結果を図1に示す.

5.2 修正指数形ソフトウェア信頼度成長モデルの場合

前節と同様にして, 修正指数形ソフトウェア信頼度成長モデルを DS-1 に適用した場合についての数値例を示す. まず, 式(4)に含まれている未知パラメータを最尤法により推定すると $\hat{a}=538.3$, $\hat{b}_1=0.5503 \times 10^{-1}$, および $\hat{b}_2=0.1564 \times 10^{-1}$ となった. ここで, タイプ1エラーおよびタイプ2エラーの含有率は, あらかじめ開発管理者の経験などから決定されているものとして, $p_1=0.90$, および $p_2=0.10$ とした. また $\mu=100$, $\sigma=10$, および $\alpha=0.001$ とすると式(18)により与えられる γ_1 および γ_2 の値はそれぞれ, $\gamma_1=0.9168$ および $\gamma_2=0.7583$ となった. 以上のようにパラメータの値を設定し, $c_1=1.0$ としてコストパラメータ c_2 および c_3 の値を変化させた場合の最適リリース時刻を表3に示す. 表3においても表1と同様な傾向がうかがえる. しかし, 表1と表3を比較すると, c_2 の値が小さいかまたは c_3 の値が大きいときには両者の最適リリース時刻にはあまり差がないが, そうでない場合には両者の値はかなり異なっていることがわかる. これは, DS-1 に対して指数形および修正指数形ソフトウェア信頼度成長モデルを適用することにより, 推定された総期待発見エラー数 $\hat{a}$ がそれぞれ513.2および538.3となり, これらの差がコスト比 c_2/c_1 の影響を受けるため結果として最適リリース時刻の差として現れるものと考えられる. これらのことから, テスト工程における

エラー修正コストに対する運用段階でのエラー修正コストの比が大きいときには, テスト工程から採取されたデータにどのようなソフトウェア信頼度成長モデルを当てはめるかによって最適リリース時刻は大きく影響されることがわかる.

つぎに表2で用いた各コストパラメータの値を用い

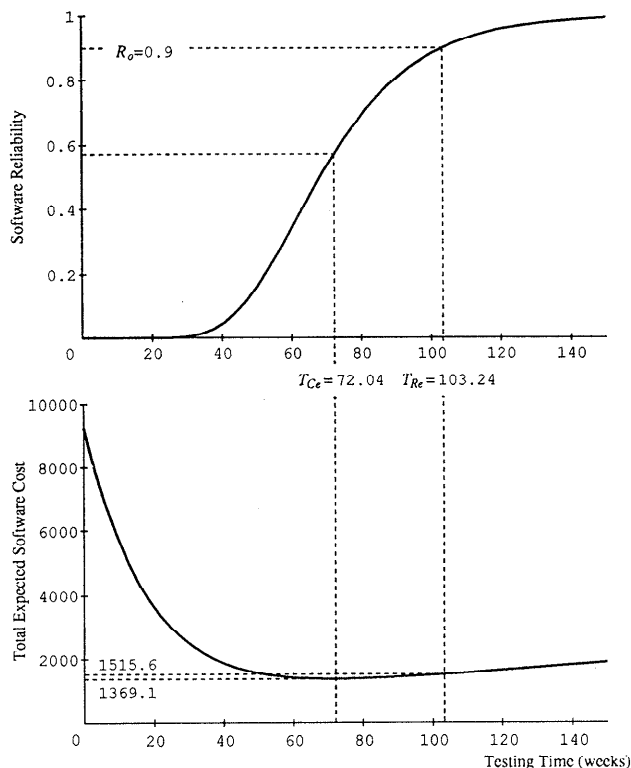


図1 信頼度およびコストの両評価基準を考慮した最適リリース時刻 (指数形ソフトウェア信頼度成長モデル)

Fig. 1 The optimum software release time with cost and reliability requirements (exponential software reliability growth model).

表3 修正指数形ソフトウェア信頼度成長モデルに基づく最適リリース時刻 (1) ($\alpha=0.001$, $\mu=100$, $\sigma=10$, $c_1=1.0$)

Table 3 The optimum release time based on modified exponential software reliability growth model (1) ($\alpha=0.001$, $\mu=100$, $\sigma=10$, $c_1=1.0$).

$c_3 \backslash c_2$	2	5	10	20	50	100
1	59.76	96.25	125.36	162.48	220.16	265.00
5	28.16	57.47	76.06	96.02	130.03	165.53
10	7.62	43.32	60.39	77.61	104.06	130.60
50	0	12.49	28.08	68.11	62.52	78.80
100	0	0	14.96	29.24	47.97	62.77

表 4 修正指数形ソフトウェア信頼度成長モデルに基づく最適リリース時刻 (2)

$$(c_1=1.0, c_2=20.0, c_3=10.0, \frac{\mu}{\sigma}=10)$$

Table 4 The optimum release time based on modified exponential software reliability growth model (2)

$$(c_1=1.0, c_2=20.0, c_3=10.0, \frac{\mu}{\sigma}=10).$$

μ	α	0.0	0.0001	0.001	0.005	0.01
50		75.34	75.30	74.86	72.64	69.36
60		76.61	76.54	75.89	72.72	68.19
70		77.55	77.45	76.59	72.43	66.61
80		78.27	78.15	77.07	71.89	64.75
90		78.85	78.71	77.40	71.17	62.70
100		79.32	79.15	77.61	70.32	60.51
150		80.76	80.46	77.68	65.03	48.38
200		81.41	80.96	76.94	52.63	34.62
300		81.85	81.12	74.63	46.08	0
400		81.95	80.94	72.04	31.90	0
500		81.97	80.69	69.43	13.65	0

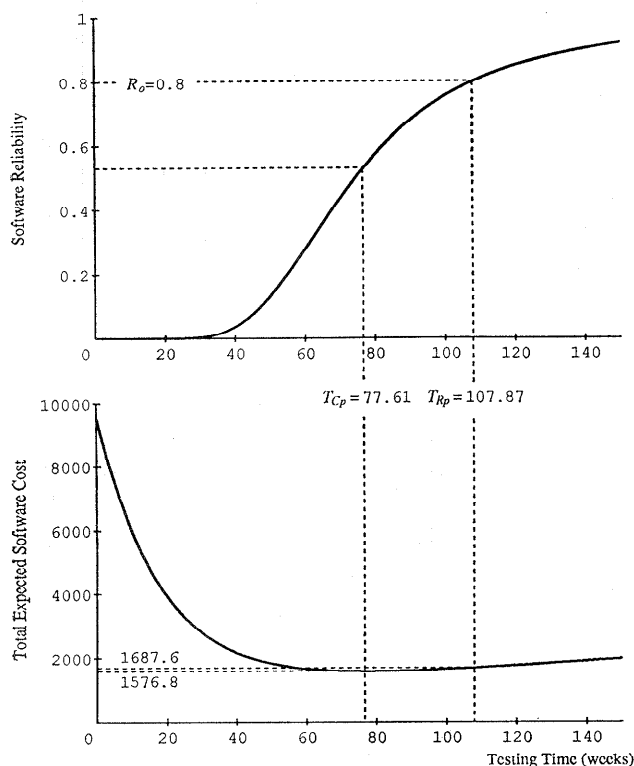


図 2 信頼度およびコストの両評価基準を考慮した最適リリース時刻 (修正指数形ソフトウェア信頼度成長モデル)

Fig. 2 The optimum software release time with cost and reliability requirements (modified exponential software reliability growth model).

て、ライフサイクルの長さの平均値 μ および割引率 α の値をそれぞれ変化させた場合の最適リリース時刻を表 4 に示す。ここで、 μ/σ の値は 10 に固定した。表 4 は表 2 とほぼ同様な傾向を示している。

さらに信頼度要求を考慮した場合についても同様に示す。コスト、ライフサイクルの長さおよび割引率の各パラメータを前節の場合と同一に設定する。また、目標信頼度を $R_o=0.8$ とする。式 (8) に式 (4) を用いることにより、式 (21) を満たす総テスト時間 $T=T_{Rp}$ を数値的に求めると $T_{Rp}=107.87$ となる。したがって、表 3 より $T_{Cp}=77.61$ から、[方策 4] の (1) より、最適リリース時刻は $T^*=T_{Rp}=107.87$ (週) であることがわかる。この結果を図 2 に示す。

6. む す び

本論文では、ソフトウェア開発管理者にとって重要であり、また興味ある問題として NHPP モデルに基づくソフトウェアの最適リリース問題について考察を行った。特に、ソフトウェアの運用開始から廃棄されるまでのソフトウェアのライフサイクルの長さを確率分布を用いて記述し、さらに長期に渡るソフトウェア開発にも本手法を適用することを踏まえてコストの時間的価値を導入した。最適リリース時刻を決定する評価基準としては、テスト工程および運用段階で発生するコスト要因を考慮した総期待ソフトウェアコスト、およびソフトウェア信頼度を用いた。これらより、前者の評価基準の最小化と後者の評価基準の目標値を考へて、NHPP モデルとして指数形および修正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時刻に関する 4 つの方策を導出した。さらに、実際のソフトウェア開発のテスト工程で観測された発見エラー数に関するデータを用いて、最適リリース時刻に関する 4 つの方策の数値例を示した。

第 5 章の表 1～表 4 から読み取れるように、最適リリース時刻を大きく左右する要因となっているのは主に各コストパラメータの値および総期待エラー数 (モデルパラ

メータ α)の推定値であると推測される。したがって、これらのパラメータをテスト工程のできるだけ早い時点に見積ることが必要であり、そのための手法の開発が今後の課題として挙げられる。さらに、本論文では指数形ソフトウェア信頼度成長モデルの適用を中心として最適方策を議論したが、NHPPに基づく代表的なS字形ソフトウェア信頼度成長モデル(例えば遅延S字形ソフトウェア信頼度成長モデル^{2),3)})に関する最適方策についても考察する予定である。

謝辞 本論文の作成にあたりご協力頂いた広島大学大学院生得能貢一君に感謝します。また、本論文の著者のうち山田は、本研究の一部に対して文部省科学研究費(一般研究(C), 課題番号 04650316)の補助を受けたことを付記する。

参 考 文 献

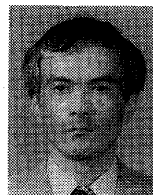
- 1) Musa, J.D., Iannino, A. and Okumoto, K.: *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, New York (1987).
- 2) 山田 茂: ソフトウェア信頼性評価技術, HBJ 出版局, 東京 (1989).
- 3) 山田 茂, 大寺浩志: ソフトウェアの信頼性—理論と実践の応用—, ソフト・リサーチ・センター, 東京 (1990).
- 4) Yamada, S. and Osaki, S.: Software Reliability Growth Modeling: Models and Applications, *IEEE Trans. Software Eng.*, Vol. SE-11, No. 12, pp. 1431-1437 (1985).
- 5) 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).
- 6) Okumoto, K. and Goel, A.L.: Optimum Release Time for Software Systems Based on Reliability and Cost Criteria, *J. Syst. & Software*, Vol. 1, pp. 315-318 (1980).
- 7) 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).
- 8) Yamada, S. and Osaki, S.: Cost-Reliability Optimal Release Policies for a Software System, *IEEE Trans. Reliability*, Vol. R-34, No. 5, pp. 422-424 (1985).
- 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) Yamada, S., Osaki, S. and Narihisa, H.: A

Software Reliable Growth Model with Two Types of Errors, *R.A.I.R.O.-Operations Research*, Vol. 19, No. 1, pp. 87-104 (1985).

- 11) 尾崎俊治, 海生直人, 一森哲男: OR による経営システム科学, 朝倉書店, 東京 (1989).
- 12) Barlow, R. E., Proschan, F. and Hunter, L. C.: *Mathematical Theory of Reliability*, John Wiley & Sons, New York (1965).
- 13) Ohba, M.: Software Reliability Analysis Models, *IBM J. Res. & Dev.*, Vol. 28, No. 4, pp. 428-443 (1984).

(平成4年8月10日受付)

(平成5年1月18日採録)



山田 茂 (正会員)

昭和27年生。昭和50年広島大学工学部経営工学科卒業。昭和52年同大学院修士課程修了。昭和58~63年岡山理科大学勤務。昭和63年広島大学工学部第二類(電気系)助教授。現在に至る。工学博士。ソフトウェア信頼性モデル, ソフトウェアマネジメントモデル, 信頼性工学, 品質管理などの研究に従事。平成3年度情報処理学会 Best Author 賞, 第8回テレコムシステム技術賞受賞。著書「ソフトウェア信頼性評価技術」(HBJ 出版局), 「ソフトウェアの信頼性~理論と実践の応用~」(ソフト・リサーチ・センター), 「ソフトウェアマネジメントモデル入門」(共立出版)など。電子情報通信学会, 日本 OR 学会, 日本経営工学会, IEEE 各会員。



木村 光宏 (正会員)

平成元年広島大学工学部第二類卒業。平成3年同大学院博士課程前期修了。現在同大学院博士課程後期在学中。ソフトウェアの信頼性に関する研究に従事。日本 OR 学会, 電子情報通信学会各会員。



寺上 英治

平成2年広島大学工学部第二類卒業。現在富士写真フイルム(株)勤務。在学中, ソフトウェアの信頼性に関する研究に従事。



尾崎 俊治 (正会員)

昭和 45 年京都大学大学院 博士課程修了。同年広島大学工学部講師，同年助教授，昭和 61 年教授。工学博士。信頼性工学およびシステム工学の研究に従事。昭和 45～47 年南カルフォルニア大学研究員。昭和 51～52 年マンチェスター大学研究員。日本 OR 学会第 4 回大西文献賞受賞。著書「応用確率論」，「Stochastic System Reliability Modeling」など。編「Stochastic Models in Reliability Theory」，「Reliability Theory and Applications」。電子情報通信学会，日本 OR 学会各会員。
