

## 多属性効用理論に基づいた最適リリース時刻および チェンジポイントの推定に関する研究

南野 友香 (Yuka Minamino)<sup>†</sup> 井上 真二 (Shinji Inoue)<sup>†</sup> 山田 茂 (Shigeru Yamada)<sup>†</sup>

<sup>†</sup> 鳥取大学大学院・工学研究科 (Graduate School of Engineering, Tottori University)

### 1 はじめに

ソフトウェア製品は、通常、品質や信頼性の計測を行うテスト工程を経て出荷される。テスト工程では、発見されたフォールトの修正・除去が行われているが、ソフトウェア内のすべてのプログラムパスに潜在するフォールトをすべて修正することは、テスト資源の制約や、納期といった観点から困難である。一方で、テスト工程で完全に修正・除去することが出来なかったフォールトは、運用段階で発見され、修正・除去されるが、運用段階における保守コストは、かなり膨大なものであるといわれている。

また、コスト面に加えて、ソフトウェア開発においては、品質（信頼性）と納期（リリース時刻）の制約も、同時に満たしていなければならない。例えば、テスト期間を長くした場合、信頼性は向上する。これにより、運用段階でのフォールト発見数は減少し、保守コストを抑えることが出来るが、長期間のテストを行うことでテストコストが増加してしまう。一方で、テスト期間を短くした場合、テストコストを抑えることは出来るが、信頼性が低下してしまうため、保守コストが増加してしまう。したがって、最適なリリース時刻を決定する問題が生ずる。このような問題は、ソフトウェア最適リリース問題と呼ばれ、ソフトウェア開発管理面における、興味深い問題の1つである。

本研究では、ソフトウェア最適リリース問題を議論するにあたり、複数の属性を考慮することが可能な多属性効用理論 [7,8] を用いる。特に、コストと信頼度を属性とし、2種類のチェンジポイントモデルを適用する。これは、最適リリース時刻に加え、最適なチェンジポイントの発生時刻を推定することを目的としている。ここで、チェンジポイントとは、意図的なテストの増員やフォールトターゲットの変更などのテスト環境の変化によって、ソフトウェア故障発生時間（間隔）の確率・統計的性質が変化する時刻のことをいう [3-5]。また、チェンジポイントは、従来のソフトウェア信頼度成長モデル (SRGM: software reliability growth model) [1,2] を用いた信頼性評価技術に影響を与えている。そこで、従来の指数形および遅延 S 字形 SRGM にチェンジポイントの概念を導入することにより、チェンジポイントモデルへの拡張を行う。

以上より、チェンジポイントモデルを適用しながら、一定の管理戦略に基づいて、各属性における単一属性効用関数を構築する。最終的に、多属性効用関数の効用最大化を行い、最適リリース時刻、最適なチェンジポイント発生時刻および効用、総期待ソフトウェアコストを導出する。

### 2 SRGM の基本的な構築枠組み

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従来の SRGM は、以下の仮定に基づいて構築することができる [6]。

- (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および除去される。
- (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、各ソフトウェア故障発生時刻は、それぞれ、独立かつ同一の確率分布関数  $F(t) \equiv \Pr\{T \leq t\} = \int_0^t f(x)dx$  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非負の確率変数とする。

ここで、確率過程  $\{N(t), t \geq 0\}$  をテスト時刻  $t$  までに発見された総フォールト数と定義する。  $N_0$  が平均  $\omega$  ( $\omega > 0$ ) のポアソン分布に従うと仮定すると、

$$\begin{aligned} \Pr\{N(t) = m\} &= \sum_{n=0}^{\infty} \binom{n}{m} \{F(t)\}^m \{1-F(t)\}^{n-m} \Pr\{N_0 = n\} \\ &= \frac{\{\omega F(t)\}^m}{m!} \exp[-\omega F(t)] \quad (m = 0, 1, 2, \dots), \end{aligned} \quad (1)$$

となり、平均値関数  $E[N(t)] = \omega F(t)$  をもつポアソン過程 (NHPP: nonhomogeneous Poisson process) と本質的に等価となる。このとき、ある具体的な NHPP モデルを構築するためには、ソフトウェア故障発生時間分布  $F(t)$  を具体的に与える必要がある。

### 3 チェンジポイントモデル

チェンジポイントの発生前後において、 $i$  番目 ( $i = 1, 2, \dots$ ) のソフトウェア故障が観測される時刻をそれぞれ  $X_i$ 、および  $Y_i$  とし、ソフトウェア故障発生時間間隔をそれぞれ、 $S_i = X_i - X_{i-1}$ 、および  $T_i = Y_i - Y_{i-1}$ 、その時間間隔に対する確率分布をそれぞれ、 $\Pr\{S_i \leq t\} = F_i(t)$ 、および  $\Pr\{T_i \leq t\} = J_i(t)$  とする。ここで、チェンジポイント前後のソフトウェア故障発生現象に関する確率量の関係を以下のように仮定する。

$$\begin{cases} Y_i = \alpha(X_i), \\ T_i = \alpha(S_i), \\ K_i(t) = J_i(\alpha^{-1}(t)). \end{cases} \quad (2)$$

また、テスト環境関数は、以下のように仮定する。

$$\alpha(t) = \alpha t \quad (\alpha > 0), \quad (3)$$

チェンジポイント前後におけるソフトウェア故障発生現象の関係性を考慮したとき、チェンジポイントを考慮した NHPP の平均値関数  $\Lambda(t)$  は、以下のように求められる。

$$\Lambda(t) = \begin{cases} \Lambda_1(t) = M_B(t) & (0 \leq t \leq \tau), \\ \Lambda_2(t) = M_B(\tau) + M_A(t) \\ \quad = M_B(\tau + \frac{t-\tau}{\alpha}) & (t > \tau). \end{cases} \quad (4)$$

ここで、 $\tau$  はチェンジポイント発生時刻を表し、 $M_B(t)$  および  $M_A(t)$  は、それぞれ、チェンジポイント前およびチェンジポイント後の平均値関数である。したがって、チェンジポイント前における平均値関数が指数形および遅延 S 字形 SRGM に従うとき、チェンジポイントを考慮した NHPP の平均値関数は、それぞれ、以下のようになる。ここで、 $b$  はハザードレートを表す。

$$\Lambda(t) = \begin{cases} \Lambda_1(t) = \omega\{1 - \exp[-bt]\} & (0 \leq t \leq \tau), \\ \Lambda_2(t) = \omega\{1 - \exp[-b(\tau + \frac{t-\tau}{\alpha})]\} & (t > \tau). \end{cases} \quad (5)$$

$$\Lambda(t) = \begin{cases} \Lambda_1(t) = \omega\{1 - (1 + bt) \times \exp[-bt]\} & (0 \leq t \leq \tau), \\ \Lambda_2(t) = \omega\{1 - (1 + b(\tau + \frac{t-\tau}{\alpha})) \times \exp[-b(\tau + \frac{t-\tau}{\alpha})]\} & (t > \tau). \end{cases} \quad (6)$$

## 4 ソフトウェア最適リリース問題

### 4.1 属性の選択

まず、コスト属性を考慮するために、式 (5) および式 (6) で表された 2 種類のチェンジポイントモデルに基づきながら総期待ソフトウェアコストを定式化する。ここで、コストパラメータは次のように定義する。

$c_1$  : チェンジポイント発生前のテスト工程において発見されるフォールト 1 個当たりの修正コスト ( $> 0$ )

$c_2$  : チェンジポイント発生後のテスト工程において発見されるフォールト 1 個当たりの修正コスト ( $c_2 > c_1$ )

$c_3$  : 運用段階において発見されるフォールト 1 個あたりの修正コスト ( $c_1 < c_3, c_2 < c_3$ )

$c_4$  : テストに要する単位時間当たりのテストコスト ( $0 < c_4$ )

テスト終了時刻を  $T$ 、チェンジポイント発生後からテスト終了時刻  $T$  までのテスト期間を  $s$  としたときの総期待ソフトウェアコスト  $C(T, s)$  は次のように定式化できる：

$$C(T, s) = c_1 \Lambda_1(T - s) + c_2 \{\Lambda_2(T) - \Lambda_1(T - s)\} + c_3 \{\omega - \Lambda_2(T)\} + c_4 T. \quad (7)$$

コスト属性は予算を消費する割合とし、次式のように表すとする。

$$\min: C = \frac{C(T, s)}{C_B}. \quad (8)$$

ここで、 $C_B$  は予算を表す。次に、2 つ目の属性として、コスト属性と同様に、チェンジポイントモデルを適用した信頼度属性を考慮する。ここで、信頼度属性は、フォールト発見数の割合とし、次式のように表すとする。

$$\max: R = \frac{\Lambda(T)}{\text{初期潜在フォールト数}}. \quad (9)$$

### 4.2 単一属性効用関数の構築

本研究では、一例として、以下の管理戦略に基づいて、単一属性効用関数を構築する。

1. 総期待ソフトウェアコストは、予算の 50% 以上を消費する。
2. 少なくともソフトウェアフォールトの 50% は発見されなければならない。
3. 経営陣は各属性に対してリスク中立の立場をとる。

効用関数を構築するため、 $u(C^W) = 0$ 、 $u(C^B) = 1$ 、 $u(R^W) = 0$ 、 $u(R^B) = 1$  より、以下の式を解く。

$$u(C^W) = y_1 + y_2 \times 0.5, \quad (10)$$

$$u(C^B) = y_1 + y_2 \times 1.0, \quad (11)$$

$$u(R^W) = y_1 + y_2 \times 0.5, \quad (12)$$

$$u(R^B) = y_1 + y_2 \times 1.0. \quad (13)$$

したがって、コストおよび信頼度属性の単一属性効用関数は、以下のようになる。

$$u(C) = 2C - 1, \quad (14)$$

$$u(R) = 2R - 1. \quad (15)$$

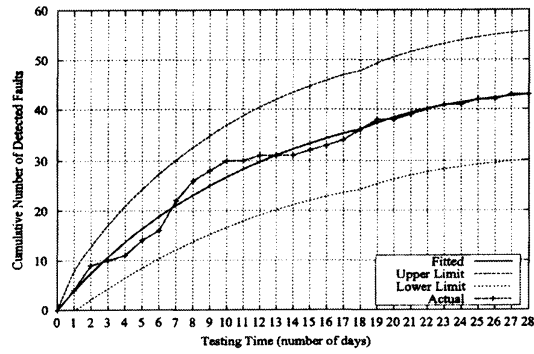


図 1： チェンジポイント前における平均値関数が指数形 SRGM に従う場合の推定された平均値関数とその 95% 信頼限界 (DS5,  $\hat{\omega} = 44.8226$ ,  $\hat{b} = 0.0908$ ,  $\hat{\alpha} = 0.5789$ ,  $\tau = 18$ ,  $MSE=2.2943$ ) .

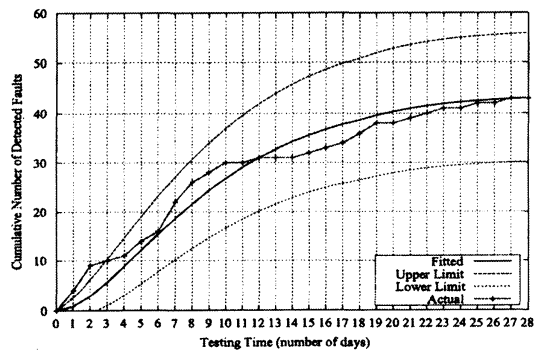


図 2： チェンジポイント前における平均値関数が遅延 S 字形 SRGM に従う場合の推定された平均値関数とその 95% 信頼限界 (DS5,  $\hat{\omega} = 43.5009$ ,  $\hat{b} = 0.2084$ ,  $\hat{\alpha} = 0.765$ ,  $\tau = 18$ ,  $MSE=7.4995$ ) .

### 4.3 多属性効用関数の構築

本研究では、コスト属性および信頼度属性の重みは、それぞれ、 $w_C$  および  $w_R$  とし、これら重みの和は 1 であるとする。また、今回は加法型多属性効用関数を適用するため、最終的に以下のような最適化問題となる。

$$\begin{aligned} \max: \quad u(C, R) &= w_R \times u(R) - w_C \times u(C) \\ &= w_R \times (2R - 1) - w_C \times (2C - 1) \\ \text{subject to: } \quad w_R + w_C &= 1, \frac{C(T, s)}{C_B} \leq 1. \end{aligned} \quad (16)$$

## 5 数値例

本研究では、テスト工程において観測された、以下のフォールト発見数データを用いて適用例を示す。

$$\text{DS1 : } (t_k, y_k) (k = 1, 2, \dots, 26; t_{26} = 26, y_{26} = 40, \tau = 18)$$

$$\text{DS2 : } (t_k, y_k) (k = 1, 2, \dots, 29; t_{29} = 29, y_{29} = 73, \tau = 24)$$

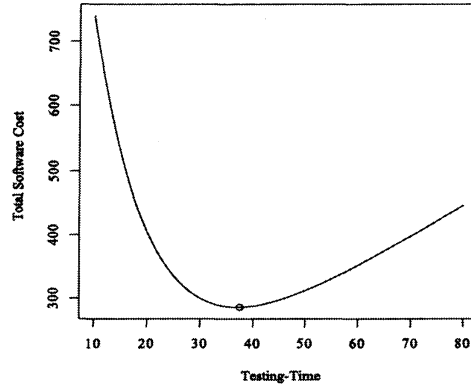


図 3： 推定された総期待ソフトウェアコストの挙動 (DS5, EXP) .

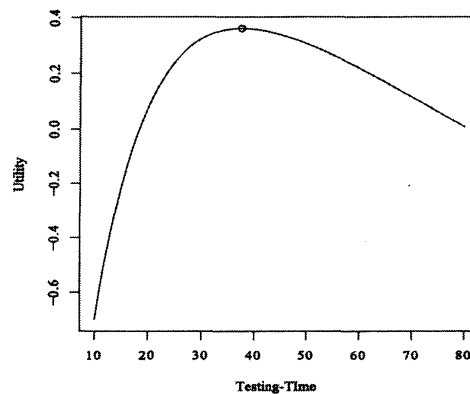


図 4： 推定された多属性効用関数の挙動 (DS5, EXP) .

DS3 :  $(t_k, y_k)(k = 1, 2, \dots, 26; t_{26} = 26, y_{26} = 34, \tau = 17)$

DS4 :  $(t_k, y_k)(k = 1, 2, \dots, 29; t_{29} = 29, y_{29} = 25, \tau = 18)$

DS5 :  $(t_k, y_k)(k = 1, 2, \dots, 28; t_{28} = 28, y_{28} = 43, \tau = 18)$

ここで、 $t_k$  はテスト時刻、 $y_k$  は時刻  $t_k$  までに発見された総フォールト数、および  $\tau$  はチェンジポイント発生時刻である。また、DS1, DS2, および DS3 は、S 字形信頼度成長曲線を示すデータ、DS4 および DS5 は、指数形信頼度成長曲線を示すデータである。まず、2つのチェンジポイントモデル内に含まれるパラメータ  $\omega$ ,  $b$ , および環境係数  $\alpha$  に対する推定値  $\hat{\omega}$ ,  $\hat{b}$ , および  $\hat{\alpha}$  をそれぞれ最尤法に基づいて推定する。ただし、チェンジポイントは所与のものとして与え、上記のパラメータを推定する。ここで、一例として、図 1 および図 2 に、DS5 において、それぞれ推定された平均値関数と 95% 信頼限界を示す。平均偏差平方和 (MSE: mean squared errors) [2] を用いた適合性比較により、DS5 においては、遅延 S 字形よりも指数形のチェンジポイントモデルの方が適合性が高いことがわかった。これは、DS5 が示す信頼度成長曲線の時間的挙動が指数形であることから、指数形のチェンジポイントモデルがよく適合していると考えられる。

次に、提案モデルのパラメータ推定の結果を用いながら、構築した多属性効用関数の効用最大化を行い、最適解を導出する。図 3 および図 5 は、コスト関数の挙動を表し、図 4 および図 6 は、多属

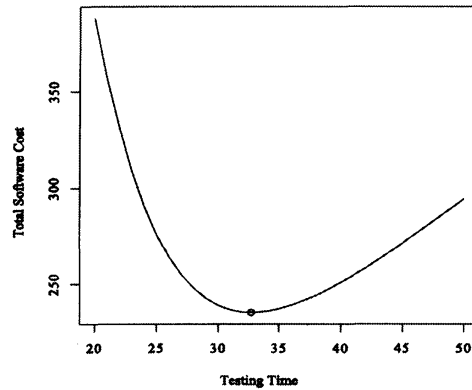


図 5： 推定された総期待ソフトウェアコストの挙動 (DS5, DSS)。

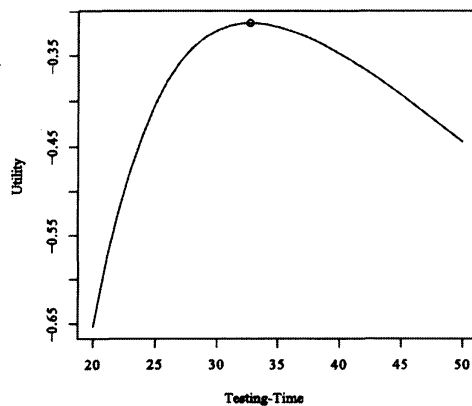


図 6： 推定された多属性効用関数の挙動 (DS5, DSS)。

性効用関数の挙動を表している。ここで、予算は 450 と仮定し、各属性の重みは 0.5 ずつであるとした。DS5 における結果として、指数形のチェンジポイントモデルの場合、ソフトウェア最適リリース時刻  $T$  の最適値  $T^*$  は、 $T^* = 37.51$ 、チェンジポイント発生後からテスト終了時刻  $T$  までのテスト期間  $s$  の最適値  $s^*$  は  $s^* = 30.25$ 、および効用の値は  $u(C, R) = 0.36$  となった。このとき、総期待ソフトウェアコストは  $C(T, s) = 285.40$  となった。一方、遅延 S 字形のチェンジポイントモデルの場合、 $T^* = 32.72$ 、 $s^* = 17.34$ 、および、 $u(C, R) = -0.31$  となった。このとき、総期待ソフトウェアコストは  $C(T, s) = 235.22$  となった。

最後に、表 1 および表 2 に、指数形 (EXP) および遅延 S 字形 (DSS) のチェンジポイントモデルにおける感度分析の結果を示す。表 1 より、ソフトウェア最適リリース時刻  $T^*$  および、チェンジポイント発生後からテスト終了時刻までの最適なテスト期間  $s^*$  が増加すると、チェンジポイントの発生時刻は減少傾向となり、総期待ソフトウェアコストおよび効用の値が増加していることが確認出来る。一方、表 2 では、感度が非常に小さく、主に各属性の重みによる変化によって、効用の値が変動していることがわかる。

表 1：感度分析の結果 (EXP, DS5) .

$w_R$	$w_C$	$T^*$	$s^*$	Change-Point	Total Cost	Utility
0.1	0.9	37.14034	29.87466	7.26568	285.3361	-0.142298
0.3	0.7	37.27178	30.01017	7.26161	285.348	0.1094483
0.5	0.5	37.50686	30.25121	7.25565	285.4025	0.3612723
0.7	0.3	38.02501	30.76971	7.2553	285.6632	0.6133076
0.9	0.1	40.1202	32.8696	7.2506	288.4347	0.8664326

表 2：感度分析の結果 (DSS, DS5) .

$w_R$	$w_C$	$T^*$	$s^*$	Change-Point	Total Cost	Utility
0.1	0.9	32.72308	17.33735	15.38573	235.2164	-0.898841
0.3	0.7	32.72308	17.33735	15.38573	235.2164	-0.60571
0.5	0.5	32.72308	17.33735	15.38573	235.2164	-0.312579
0.7	0.3	32.72308	17.33735	15.38573	235.2164	-0.019448
0.9	0.1	32.72308	17.33735	15.38573	235.2164	0.273683

## 6 おわりに

本研究では、指数形と遅延 S 字形 SRGM に基づいたチェンジポイントモデルを適用し、多属性効用理論に基づいたソフトウェア最適リリース問題について議論した。今回の数値例では、DS5 が指数形信頼度成長データであったことから、遅延 S 字形よりも指数形のチェンジポイントモデルの方が精度良くパラメータの推定が出来たと考えられる。したがって、適用するモデルの選択によって、最適リリース時刻やチェンジポイント発生時刻などの推定結果が大きく変化することも考慮し、適切なモデル選択を行うことが必要である。また、本手法にて導出される効用の値は、感度分析などにより相対的な評価のみが可能である。そのため、今後は、効用の値に対する具体的な評価尺度を考慮していく必要がある。

## 参考文献

- [1] S. Yamada, *Software Reliability Modeling — Fundamentals and Applications*, Springer, Tokyo, 2013.
- [2] H. Pham, *Software Reliability*. Springer-Verlag, Singapore, 2000.
- [3] C.Y. Huang, “Cost-reliability-optimal release policy for software reliability models incorporating improvements in test efficiency,” *J. Syst. Softw.*, Vol. 77, No. 2, pp. 139–155, 2005.
- [4] M. Zhao, “Change-point problems in software and hardware reliability,” *Communication Statistics — Theory Methods*, Vol. 22, No. 3, pp. 757–768, 1993.
- [5] S. Inoue and S. Yamada, “Software reliability measurement with change-point,” *Proc. Fifth Int. Conf. Quality and Reliability*, Chiang-Mai, Thailand (2007), pp. 170–175.
- [6] N. Langberg and N.D. Singpurwalla, “A unification of some software reliability models,” *SIAM J. Sci. Comput.*, Vol. 6, No. 3, pp. 781–790, 1985.
- [7] P.K. Kapur, V.B. Singh, O. Singh and J.N.P. Singh, “Software release time based on different multi-attribute utility function,” *Int. J. Reliab. Qual. Saf. Eng.*, Vol. 20, No. 4, 1350012-1–1350012-12, 2013.
- [8] P.K. Kapur, K.K. Sunil, T. Anshul, and S. Omar, “Release time determination depending on number of test runs using multi attribute utility theory,” *Int. J. Syst. Assur. Eng. Manag.*, pp. 186–194, 2014.