

開発における不確定性と時間変化を考慮した 一般化信頼性モデル

本田 澄^{1,a)} 鷺崎 弘宜^{1,b)} 深澤 良彰^{1,c)}

概要: 欠陥の発見におけるソフトウェア信頼性モデルについて、確率過程を用いた一般化されたモデルを提案する。開発での様々な不確定要素と開発の時間変化をモデルに組み込み、定量的に開発工程を扱えるようにする。これにより、開発の時間変化や不確定要素を考慮した開発の予測が行える。特に多くの既存モデルでは扱っていない、時間変化に関して複数のタイプを例として扱い、実際の開発との比較を行う。本発表では既存のソフトウェア信頼性モデルと提案するモデルを比較し、提案するモデルでのシミュレーション結果を示し、時間変化と不確定要素について考察を行う。

キーワード: 信頼性モデル, 欠陥予測, 確率過程

A Generalized Software Reliability Model Including Uncertainty and Dynamics in Development

KIYOSHI HONDA^{1,a)} HIRONORI WASHIZAKI^{1,b)} YOSHIAKI FUKAZAWA^{1,c)}

Abstract: Development environments have changed drastically in recent years. The development periods are shorter than ever and the number of team has increased. These changes have led to difficulties in controlling the development activities and predicting the end of developments. In order to assess recent software developments, we propose a generalized software reliability model based on a stochastic process, and simulate developments that include uncertainties and dynamics, such as unpredictable requirements changes, shortening of the development period, and decrease in the number of members. We also compare our simulation results to those of other software reliability models. Using the values of uncertainties and dynamics obtained from our model, we can evaluate the developments in a quantitative manner.

Keywords: Software Reliability Model, Prediction of bugs, Stochastic Process

1. はじめに

昨今、開発において様々な課題があげられている。ソフトウェア開発において開発の見積もりや工数の予測については、経験や勘といった属人的要素をもって判断材料とすることがほとんどである。また、比較的早期な開発や、人員の不足が蔓延しており、開発の終了をどのようにして決定していいかどうか、明確な基準がない。開発者の技術的

習熟が少ないまま開発を行うことは少なくない。技術者レベルもまた何を持って判断すべきかどうか明確な指針や論理的根拠が乏しい。

従来の研究において、こうした課題に対していくつかの提案がなされてきた。一つは、開発を始めるにあたってどれだけの規模のソフトウェアになるか見積もることである。過去の開発実績に基づき新たに開発するソフトウェアに対して、工数やLOCを見積もる、デルファイ法やCOCOMOなどが挙げられる [1]。また、開発する規模に応じてどれだけのテストを実施し、どれだけの欠陥を発見すれば開発を終了とみなせるかどうか、総欠陥数がどれだけあるのかを

¹ 早稲田大学 Waseda University

a) khonda@ruri.waseda.jp

b) washizaki@waseda.jp

c) fukazawa@waseda.jp

予測することである。この、欠陥の予測について様々な手法が提案されている。特に欠陥の発見をソフトウェアの信頼度成長過程とみなすソフトウェア信頼度成長モデルが有名である [11]。

このソフトウェア信頼度成長モデルにおいても様々な解析手法が提案されている。一つとして、実際の開発から得られた発見された欠陥数と時間の関係を統計的に解析するモデルとして、ゴンベルツ曲線モデルやロジスティック曲線モデルがあげられる [2]。この方法では実際のデータに対して近似を行うものであり、欠陥の発見に原因を考慮したものではない。本稿では、欠陥の発見に確率的原因を考慮し確率モデルを用いた、非同次ポアソン過程 (nonhomogeneous Poisson process:以下, NHPP と省略する) モデルについて説明を行い、我々が提案するモデルと比較を行う。

従来のモデルにおいては開発における時間変化や不確定性を十分に考慮することができていない。そこで我々は開発における時間変化と不確定性を考慮した新しいモデルを提案する。

我々が提案するモデルと NHPP モデルとの明確な違いが 2 点あり、一つは時間変化を反映できること、もう一つは不確定性を考慮できることである。

昨今の開発において、要求や仕様の変更といった時間変化は激しさを増している。時間変化については NHPP モデルを拡張した先行研究も、Tamura[7], Yamada[8] and Zhang[9] によって行われている。

また、我々が提案するモデルと NHPP モデルとの明確な違いについては、開発における不確定要素を考慮したモデルの構築が行えることである。開発において、開発チームの技術的なばらつきや、開発環境のばらつきといった不確定な要素が存在する。そこで、我々は信頼性モデルに対して不確定な要素を考慮したモデルを提案する。

時間変化と不確定性を考慮したモデルを用いることで、開発期間をより精度よく予測できるようになる。また、開発チームの技術的なばらつき具合を考慮することで、開発の見積もりについて、事前に精度よく見積もることが可能となる。

本稿の構成としては、2 章において既存研究の信頼性モデルについて、主に NHPP モデルを挙げて説明を行う。3 章では我々が提案する一般化信頼度モデルについて説明を行う。その中で時間変化と不確定性をそれぞれ 3 つのタイプに分け、それぞれの組み合わせについて 4 章においてシミュレーション結果を示す。次に、4 章では、2 章で挙げた NHPP モデルとの比較を行い、実際の開発データを用いて信頼性の制度について議論する。それとともに、3 章で分けた時間変化と不確定性のタイプについてシミュレーションを行う。最後に 5 章で、4 章で得た結果を基に考察を行い、今後の展望を記す。

2. 信頼性モデル

本章では信頼性モデルの例を挙げ、次章で我々が提案するモデルを説明する。広く用いられる NHPP モデルとの比較については第 3 章にて、実際の開発で得られたデータをもとにモデルの精度比較を行う。

信頼性モデルについては様々なモデルが提案されている。広く用いられるモデルとして、NHPP モデルがあげられる。我々が提案するモデルでは確率微分方程式を用いた信頼性モデルを提案する。

まずはじめに、実際の開発において信頼性を保証する手立てとして、欠陥の累積数を測定する方法が広く用いられている。これを用いた信頼性モデルとして発生した欠陥数を数える「フォールトカウントモデル」[3] があげられる。我々が考えるモデルもまた欠陥数を数えるモデルである。このモデルは欠陥の数とそれが発生した時間を計測することでソフトウェアの信頼性を測るものである。このモデルでは欠陥数が確率過程に従うことを仮定しており、欠陥の発生と時間の関係をポアソン過程ととらえるモデルとして NHPP モデルがあげられ、広く用いられている。実際の開発では欠陥数を数えることにより、残存欠陥数のある程度の数と予測し開発の終了を NHPP モデルを用いて妥当かどうか判断することができる。

ここでは一般的な NHPP モデルについて説明を行う。NHPP モデルでは、時間区間内に発見される欠陥の数 $N(t)$ について時間 t と発見されると期待される欠陥数 $N(t) = n$ について、その期待値を $Pr\{N(t) = n\}$ を用いて以下の式と仮定している。

$$Pr\{N(t) = n\} = \frac{\{H(t)\}^n}{n!} \exp -H(t) \quad (1)$$

ここで $H(t)$ はテスト時間 t までに発見される総期待欠陥数を表す [11]。

次に、総欠陥数が一定数 N_{max} であるとし、単位時間あたりに発見される欠陥数が、残存欠陥数に比例すると仮定して以下の式で定式化される。

$$\frac{dH(t)}{dt} = b(N_{max} - H(t)) \quad (2)$$

これを解くことで得られ平均値関数は以下となる。

$$H(t) = N_{max}(1 - \exp -bt) \quad (3)$$

これは指数形ソフトウェア信頼度成長モデルと呼ばれ、Goel と Okumoto が提案した [4]。本稿ではこのモデルを用いて我々の提案モデルと比較した。

この解を考えた場合、欠陥数は指数関数型になることが読み取れる。しかし、実際の開発では指数関数型にならないことがあり、ロジスティック曲線といったより複雑な曲線への近似が採用されることがある [11]。そこで、我々はより実際の開発に即したモデルを提案する。

3. 一般化信頼性モデル

既存研究として様々な提案されているモデルに対して、我々は一つのモデルで、様々なモデルについて適用可能な一般化されたモデルを提案する。まず、我々が仮定することを以下にあげる。

- (1) 単位時間あたりに発見できる欠陥の数は、残存欠陥数と発見できた欠陥数の積に比例する。
- (2) テストに割ける時間は一定ではなく、時間変化を持つ。
- (3) 単位時間あたりに発見できる欠陥の数は、不確定な要素を持つ。

まずはじめに、仮定 (1) については、欠陥が発見されないテスト初期の段階と、残る欠陥が少ないテスト終了の段階では、単位時間あたりに発見できる欠陥は少ないことを考慮している。これはテスト初期の段階では、欠陥を見つけずらく、ある程度欠陥が発見された段階で、発見できる欠陥が増えることを反映している。また、テスト終了の段階では残る欠陥が少ないため、発見できる欠陥が少ないことを反映している。この仮定から、一般的なロジスティック方程式を出発点とする。具体的には許容力として、発見された欠陥数が全欠陥数に近づくにつれて残存欠陥数が少なくなることを考慮する項を用いた以下の微分方程式とする。

$$\frac{dN(t)}{dt} = (a + \beta N(t))N(t) \quad (4)$$

α は単位時間あたりに発見できる欠陥の数を示す。 β は許容力を示しており、全欠陥数に近づくにつれて発見できる欠陥の量が減少するように働く。つまり $\beta N(t)$ は $N(t)$ が大きくなるにつれて、増加量を減少させるように働く。

この微分方程式に対して、仮定 (2)(3) を適用する。

仮定 (2) (3) として、テストに割ける時間が一定ではなく、時間変化をもつことと、単位時間あたりに発見できる欠陥の発見数が不確定な要素を持つことを挙げた。

時間に依存するとは、テスト作業やデバッグ作業が実施可能な時間によることを考慮した仮定である。例として、テストチームが他の業務に時間を割かれ、テストの実施を行えなかったことが考えられる。

また、不確定性についてはテスト作業やデバック作業にあたって作業できる数に対して、ある程度のばらつきを考慮した仮定である。例として、テストにおいて一日に発見できる欠陥数に対して、テストチームの能力により変化し、それは概ねばらつきがあるものと考えられるためである。

これを考慮し、テスト実施数について時間的変動と不確定性を含んだ以下の式を導入する。

$$a = \alpha(t) + \sigma dw(t) \quad (5)$$

$\alpha(t)$ は時間に依存した項である。 σ は不確定性に起因するばらつきである。 $dw(t)$ についてはばらつきが従う確率

分布である正規分布 (ガウシアンホワイトノイズ) である。

次にこの式 (5) を用いて式 (4) を確率微分方程式へと展開すると以下の式となる。

$$dN(t) = (\alpha(t) + \sigma dw(t))N(t)dt + \beta N^2(t)dt \quad (6)$$

この式に対して伊藤の確率積分を行うと次の式が得られる。

$$dN(t) = (\alpha(t) + \frac{\sigma^2}{2} + \sigma dw(t))N(t)dt + \beta N^2(t)dt \quad (7)$$

このモデルでは時間依存と不確定要素を含んだ項 $\alpha(t) + \frac{\sigma^2}{2} + \sigma dw(t)$ と、 β の非線形項が明確に表れる。本稿ではこの式を用いて時間依存性と不確定性について各 3 タイプについてシミュレーションを行った。

3.1 節と 3.2 節においてシミュレーションを行うそれぞれ 3 タイプについて説明を行い、表 1 の「時間変化 $\alpha(t)$ と不確定要素 $dw(t)$ による組み合わせ」において、それぞれの場合を整理する。

3.1 時間依存

時間依存項について 3 タイプを用いてシミュレーションを行う。今回扱うタイプは以下のとおりである。

$$\alpha(t) = a \quad (8)$$

$$\alpha(t) = \begin{cases} a_1, & (t \leq t_1) \\ a_2, & (t > t_1) \end{cases} \quad (9)$$

$$\alpha(t) = a_3 t \quad (10)$$

このタイプを選んだ理由として、式 (8) では時間変化がない場合の我々の提案するモデルとその他のモデルを比較するためである。

式 (9) については、単位時間あたりに発見できる欠陥数が t_1 で変化するタイプを考える。これは、時間 $t = t_1$ において、テストチームのメンバーが増加するといったことに対応する。実際の開発でも考えられるものであるため、このタイプのシミュレーションを行う。

式 (10) については、単位時間あたりに発見できる欠陥数が時間とともに増加するタイプを考える。メンバーが徐々に増えることや、メンバーのスキルが時間とともに成長することに対応する。実際の開発でも考えられるものであるため、このタイプのシミュレーションを行う。

3.2 不確定要素

式 (7) においてランダムな要素 $dw(t)$ を含む項について、我々が提案するモデルでは発見した欠陥数に比例することがわかる。つまり、欠陥を発見すればするほど不確定な要因が与える影響が大きくなる。しかし、実際の開発では開発の初期段階で不確定要素が影響を与えることが多いことも考えられる。そこで、我々は不確定要素が与える影響について 3 つのタイプを検証のため考え、モデルとして扱う。

式 (7) のランダム項 $N(t)\sigma dw(t)$ を $\gamma(t) = N(t)\sigma dw(t)$ と置き換え、以下の式を用いる。

$$dN(t) = (a(t) + \frac{\sigma^2}{2})N(t)dt + \gamma(t) + \beta N^2(t)dt \quad (11)$$

以上をまとめて、 $\gamma(t)$ を以下の 3 タイプを扱うこととする。

$$\gamma(t) = N(t)\sigma dw(t) \quad (12)$$

$$\gamma(t) = \sigma dw(t) \quad (13)$$

$$\gamma(t) = \frac{1}{N}\sigma dw(t) \quad (14)$$

このタイプを選んだ理由として、式 (12) では我々が求めたモデルを評価することと、欠陥が発見されるにつれて不確定な要素による影響が増える場合を考慮するためである。欠陥が発見されるにつれて不確定な要素が増える場合とは、テスト終了になるにつれて、欠陥が多いテストが残り、一つの欠陥の発見により、他のテストケースに対してに多くの欠陥が発見されることがあげられる。

式 (13) については、欠陥の発見数によることがないタイプを考える。これは、欠陥の発見数にかかわらず、不確定な要素による影響が起因する場合を想定する。実際の開発でも考えられるものであるため、このタイプのシミュレーションを行う。

式 (14) については、欠陥の発見が少ない場合に不確定な要素による影響が増えるタイプを考える。欠陥の発見が少ない初期段階での不確定な要素による影響が起因する場合等を想定する。実際の開発でも考えられるものであるため、このタイプのシミュレーションを行う。

4. 結果と考察

本章では、初めに既存の信頼性モデルである NHPP モデルとの比較について記す。ここでは実際の開発で得られたデータを用いて、そのデータとのフィッティングを最小二乗法を用いて比較する。

また、3 章で得た我々の提案するモデルについて 9 つのタイプについてシミュレーションを行い、結果を示す。9 つのタイプについては、3 つのタイプの時間変化と 3 つのタイプの不確定要素について、それぞれの組み合わせでシミュレーションを行い、それぞれを比較する。

4.1 NHPP モデルとの比較

昨今、様々な信頼度モデルが提案されており、その中で最も多く使われている NHPP モデルと提案するモデルとの比較を行う。比較の内容は実際の開発データを基にモデルへのあてはめを行った際の誤差にフォーカスを当てて比較する。この信頼度モデルと我々が提案するモデルについて実際の開発データを用いて比較を行う。ここでは実際のデータとの比較結果を示す。

表 2 モデル比較

ソフトウェア 故障番号 k	実測データ (日)	NHPP モデルの 推定値	提案モデル の推定値
1	9	1.462347524	1.726368822
2	21	2.303096785	3.892613973
3	32	3.430465855	5.750365354
4	36	3.943042134	6.397108696
5	43	4.986493676	7.493485721
6	45	5.320022062	7.798660783
7	50	6.223053204	8.546315041
8	58	7.864465848	9.698440515
9	63	8.998248803	10.39188463
10	70	10.68862626	11.32959057
11	71	10.93697871	11.46047297
12	77	12.44393764	12.23005117
13	78	12.69596244	12.35573497
14	87	14.9254177	13.45473451
15	91	15.87041267	13.92509506
16	92	16.10022927	14.04099291
17	95	16.77195078	14.38468724
18	98	17.41455368	14.72246314
19	104	18.60247452	15.3806656
20	105	18.78710724	15.48816005
21	116	20.55754092	16.63041784
22	149	23.43497089	19.65142657
23	156	23.72420965	20.2213468
24	247	24.61792945	25.86455432
25	249	24.61973136	25.95821941
26	250	24.62057858	26.00464679

解析対象としたデータについては、Goel と Okumoto[4] から引用したものである。このデータは、NTDS(Naval Tactical Data System) の中核をなすシステムの開発プロセスで観測されたものである。開発データとモデル比較を行った数値を表 2 に示す。また、図 1 に比較結果をプロットしたグラフを示す。

我々の提案するモデルについて、開発データとのフィッティングしたパラメータは”R”を用いて求めた。また NHPP モデルについては、文献 [11] より引用した。それぞれの具体的な数値を用いた式を以下に示す。

$$N(t) = \frac{24.64}{1 + 22.90 \exp(-0.04093t)} \quad (15)$$

$$N_{NHPP}(t) = 34.00(1 - \exp(-0.000579t)) \quad (16)$$

これらの式と実際のデータを用いて、提案モデルと NHPP モデルとで実際の計測値と違いを残差平方和を用いて比較した。比較した値は以下である。

- NHPP モデル：残差平方和 129.5874329
- 我々の提案するモデル：残差平方和 11.2030334

この値を比較すると、我々の提案するモデルの方が精度がよいことがわかる。特に、残差平方和ではおよそ 10 分の 1 の精度で我々のモデルが優位である。

表 1 時間変化 $\alpha(t)$ と不確定要素 $dw(t)$ による組み合わせ

	$\gamma(t) = N(t)\sigma dw(t)$	$\gamma(t) = \sigma dw(t)$	$\gamma(t) = 1/N(t)\sigma dw(t)$
$\alpha_1(t) = a_1(\text{const.})$	単位時間あたりに発見できる欠陥数が一定であり、発見された欠陥の累積数が増えれば不確定な影響が大きくなる. (Model 1-1)	単位時間あたりに発見できる欠陥数が一定であり、不確定な影響が常に一定. (Model 1-2)	単位時間あたりに発見できる欠陥数が一定であり、発見された欠陥の累積数が増えれば不確定な影響が小さくなる. (e.g. 開発チームが成熟する) (Model 1-3)
$\alpha_2(t) = a_2(t < t_1)$ $\alpha_2(t) = a_3(t \geq t_1)$	単位時間あたりに発見できる欠陥数が t_1 で変化し、発見された欠陥の累積数が増えれば不確定な影響が大きくなる. (e.g. 新しいメンバーが t_1 で加わる) (Model 2-1)	単位時間あたりに発見できる欠陥数が t_1 で変化し、不確定な影響が常に一定. (Model 2-2)	単位時間あたりに発見できる欠陥数が t_1 で変化し、発見された欠陥の累積数が増えれば不確定な影響が小さくなる. (Model 2-3)
$\alpha_3(t) \propto t$	単位時間あたりに発見できる欠陥数と、不確定性の影響が発見された欠陥の累積数が増えれば大きくなる. (e.g. 時間とともに開発メンバーが増える) (Model 3-1)	単位時間あたりに発見できる欠陥数が時間とともに増加し、不確定な影響が常に一定. (Model 3-2)	単位時間あたりに発見できる欠陥数が時間とともに増加し、発見された欠陥の累積数が増えれば不確定な影響が小さくなる. (Model 3-3)

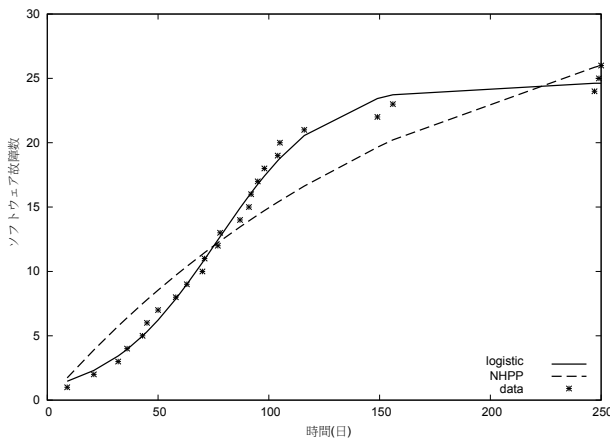


図 1 NHPP モデルとの比較

我々が提案するモデルと NHPP モデルとの比較を行った。縦軸がソフトウェア故障の数を示し、横軸が故障が発生した時間を示す。実線が我々が提案するモデルの値を示し、点線が NHPP モデルの値を示す。ドットが適用対象としたデータを示す。

なぜこのような結果が得られたかについては、我々の提案するモデルにおいて、NHPP モデルには含まれない非線形項によるものと考えられる。この非線形項が含まれることで、より複雑なデータに対して、精度がよいフィッティングが行えた。

4.2 時間変化と不確定要素のシミュレーション

表 1 に示したタイプについてシミュレーションを行い、図 2、図 3、図 4 に結果を示す。それぞれのタイプにおいて、主なパラメータの違いは $\alpha(t)$ である。 $\alpha_1(t) = a_1$ を基にして、 $\alpha_2(t)$ では $a_2 = a_1$ 、 $a_3 = 2a_1$ 、 $t_1 = t_{max}/2$ とした。 $\alpha_3(t)$ では $\alpha_3(t) = a_1 t$ とした。 $\alpha_2(t)$ のタイプでは、 $t = t_1$ に開発チームのメンバーが 2 倍になったことに対応する。 $\alpha_3(t)$ のケースでは、チームのスキルが時間とともに成熟し、 $t = t_{max}$ で 2 倍になることに対応する。シミュレーションの目的は我々の手法が開発の時間変化と不確定性のもとで表れるソフトウェア信頼性について正しいかを確認することである。また、他のモデルと比較して正しいかどうかを確認することである。

4.2.1 Model 1-1

単位時間あたりに発見できる欠陥の数は一定であり、不確定要素の影響は発見した欠陥の数が増えると大きくなる。このシミュレーションの結果はロジスティック曲線によくフィットすることを予測した。図 2 より、ロジスティック曲線へのフィッティングがよいことがわかる。この結果は非線形項を含まない他の確率モデルを用いては得ることは簡単ではない。また、発見された欠陥数が大きいほど、不連続な値となっていることがわかる。

4.2.2 Model 2-1

単位時間あたりに発見できる欠陥数が t_1 で変化し、発見された欠陥の累積数が増えれば不確定な影響が大きくなる。

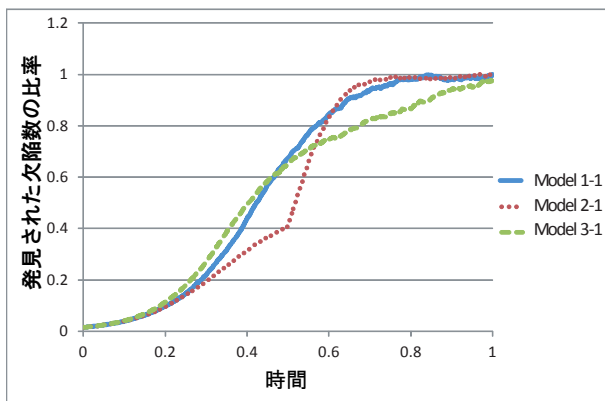


図 2 Model 1-1, Model 2-1, Model 3-1 のグラフ

縦軸に発見された欠陥数を総欠陥数で割った比率を示し、横軸に時間を示す。単位として $t_{max}=1$ and $t_1 = 0.5$ をもちいた。また、この図では $\gamma(t) = Ndw(t)$ をプロットした。Model 1-1 では、単位時間あたりに発見できる欠陥数が一定である。Model 2-1 では、単位時間あたりに発見できる欠陥数が $t = t_1$ で変化する。Model 3-1 では、単位時間あたりに発見できる欠陥数が時間とともに増加する。

る。我々の予測としては、 t_1 で発見した欠陥数が急激に増加することを予測した。図 2 より、 t_1 で急激に増加していることが見て取れる。これは、 t_1 で開発メンバーを増員し、単位時間あたりに発見できる欠陥数を増加したためである。実際の開発において人員の増加により必ずしも欠陥の発見ができるとは限らないため、この点に関して今後の課題としたい。また、発見された欠陥数の増加が t_1 で急なため、不確定要素の影響が少なく、不連続な値となっていることが図 2 に示す他の 2 モデルよりすくない。

4.2.3 Model 3-1

単位時間あたりに発見できる欠陥数が時間とともに増加し、発見された欠陥の累積数が増えれば不確定要素の影響が大きくなる。我々は結果として、Model 1-1 より発見された欠陥の累積数の傾きが大きくなると予測したが、図 2 より、途中から Model 1-1 が Model 1-1 より大きくなっていることが見てわかる。この結果から考察すると単位時間あたりに発見できる欠陥の数が時間とともに増加することにより、非線形項の影響が大きくなり、発見された欠陥の累積数が小さくなったと考えられる。また、発見された欠陥数の増加が Model 1-1 より速く、不確定要素の影響が図 2 に示す他の 2 モデルより顕著に表れており、不連続な値となっていることがわかる。

4.2.4 Model 1-2

単位時間あたりに発見できる欠陥の数は一定であり、不確定要素の影響は発見した欠陥の数に影響しない。このシミュレーションの結果はロジスティック曲線によくフィットすることと、不確実性が発見した欠陥の数によらず一定であることを予測した。図 3 より、ロジスティック曲線へのフィッティングがよいことがわかる。この結果は非線形

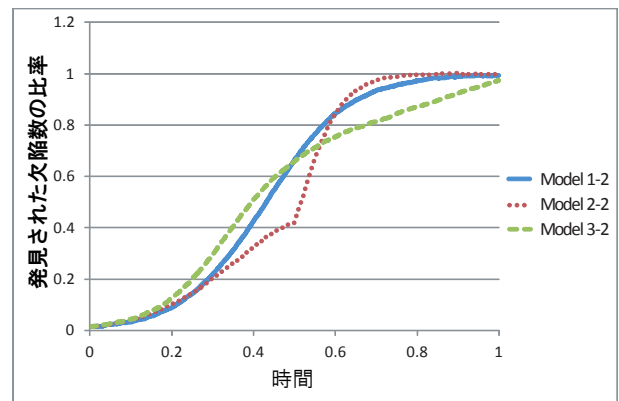


図 3 Model 1-2, Model 2-2, Model 3-2 のグラフ

縦軸に発見された欠陥数を総欠陥数で割った比率を示し、横軸に時間を示す。単位として $t_{max}=1$ and $t_1 = 0.5$ をもちいた。また、この図では $\gamma(t) = dw(t)$ をプロットした。Model 1-2 では、単位時間あたりに発見できる欠陥数が一定である。Model 2-2 では、単位時間あたりに発見できる欠陥数が $t = t_1$ で変化する。Model 3-2 では、単位時間あたりに発見できる欠陥数が時間とともに増加する。

項を含まない他の確率モデルを用いては得ることは簡単ではない。また、不確定要素の影響により、発見した欠陥の数に影響せず、不連続な値となっていることがわかる。

4.2.5 Model 2-2

単位時間あたりに発見できる欠陥数が t_1 で変化する、不確定要素の影響は発見した欠陥の数に影響しない。我々の予測としては、 t_1 で発見した欠陥数が急激に増加することを予測した。不確実性が発見した欠陥の数によらず一定であることを予測した。図 3 より、 t_1 で急激に増加していることが見て取れる。これは、 t_1 で開発メンバーを増員し、単位時間あたりに発見できる欠陥数を増加したためである。実際の開発において人員の増加により必ずしも欠陥の発見ができるとは限らないため、この点に関して今後の課題としたい。また、不確定要素の影響により、発見した欠陥の数に影響せず、不連続な値となっていることがわかる。

4.2.6 Model 3-2

単位時間あたりに発見できる欠陥数が時間とともに増加し、不確定要素の影響は発見した欠陥の数に影響しない。我々は結果として、Model 1-2 より発見された欠陥の累積数の傾きが大きくなると予測したが、今回のシミュレーション結果を見ると、途中から Model 1-2 が Model 1-2 より大きくなっていることが見てわかる。この結果から考察すると単位時間あたりに発見できる欠陥の数が時間とともに増加することにより、非線形項の影響が大きくなり、発見された欠陥の累積数が小さくなったと考えられる。

4.2.7 Model 1-3

単位時間あたりに発見できる欠陥の数は一定であり、不確定要素の影響は発見した欠陥の数が増えると小さくなる。このシミュレーションの結果はロジスティック曲線によく

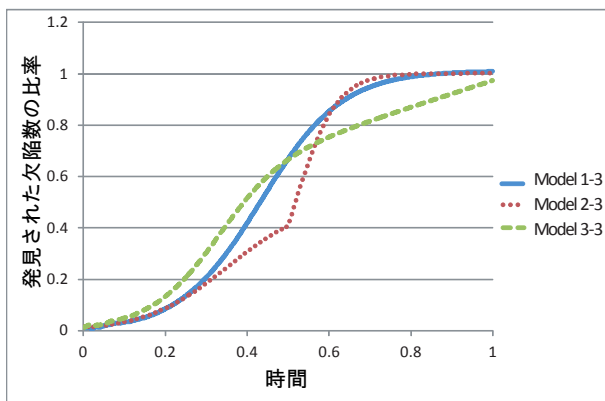


図 4 Model 1-3, Model 2-3, Model 3-3 のグラフ

縦軸に発見された欠陥数を総欠陥数で割った比率を示し、横軸に時間を示す。単位として $t_{max}=1$ and $t_1 = 0.5$ をもちいた。また、この図では $\gamma(t) = \frac{1}{N}dw(t)$ をプロットした。Model 1-3 では、単位時間あたりに発見できる欠陥数が一定である。Model 2-3 では、単位時間あたりに発見できる欠陥数が $t = t_1$ で変化する。Model 3-3 では、単位時間あたりに発見できる欠陥数が時間とともに増加する。

フィットすることを予測した。図 4 より、ロジスティック曲線へのフィッティングがよいことがわかる。この結果は非線形項を含まない他の確率モデルを用いては得ることは簡単ではない。また、不確定要素の影響は初期の段階で影響を及ぼしているが、終了に向かうにつれて影響が見取れない。

4.2.8 Model 2-3

単位時間あたりに発見できる欠陥数が t_1 で変化する、不確定要素の影響は発見した欠陥の数が増えると小さくなる。我々の予測としては、 t_1 で発見した欠陥数が急激に増加することを予測した。図 2 より、 t_1 で急激に増加していることが見て取れる。これは、 t_1 で開発メンバーを増員し、単位時間あたりに発見できる欠陥数を増加したためである。実際の開発において人員の増加により必ずしも欠陥の発見ができるとは限らないため、この点に関して今後の課題としたい。また、不確定要素の影響は初期の段階で影響を及ぼしているが、終了に向かうにつれて影響が見取れない。

4.2.9 Model 3-3

単位時間あたりに発見できる欠陥数が時間とともに増加し、不確定要素の影響は発見した欠陥の数が増えると小さくなる我々は結果として、Model 1-3 より発見された欠陥の累積数の傾きが大きくなると予測したが、今回のシミュレーション結果を見ると、途中から Model 1-3 が Model 3-3 より大きくなっていることが見てわかる。この結果から考察すると単位時間あたりに発見できる欠陥の数が時間とともに増加することにより、非線形項の影響が大きくなり、発見された欠陥の累積数が小さくなったと考えられる。また、不確定要素の影響は初期の段階で影響を及ぼしているが、終了に向かうにつれて影響が見取れない。

4.2.10 シミュレーションに関するまとめ

我々が提案するモデルにおいては、時間変化と不確定要素の影響について様々なモデルが考えられる。代表的と考えられる 3 つのタイプの時間変化と 3 つのタイプの不確定要素の影響を組み合わせた合計 9 つのモデルについてシミュレーションを行った。

9 つそれぞれに異なった結果を得ることができ、様々な状況へ適応することができる可能性が高いことがわかる。しかし、それぞれのタイプを実際の開発と乖離することなく選択することは困難な可能性がある。

5. まとめと今後の展望

本稿では、時間変化と不確定性を考慮したソフトウェア信頼性モデルを提案した。提案したモデルでは、従来モデルより精度のよいモデルであることを示した。また、従来モデルでは扱うことが困難であった開発における時間変化と不確定性を考慮したモデルを用いて複数の開発モデルについてシミュレーションを行った。以下にまとめと今後の展望を記す。

従来から使われている NHPP モデルと我々の提案するモデルの比較を行い、我々の提案するモデルが実際の開発データを用いて精度の違いを確認することができた。我々の提案するモデルの用が NHPP モデルよりおよそ 10 倍精度がよい結果が得られた。この結果は、今回適応したデータに対してだけではあるが、我々の提案モデルがより良いモデルであることを示している。しかし、現状では NHPP モデルよりも自由度が大きいため、パラメータの設定において多少複雑である。今後の展望として、この点を改善すべく、より簡単なパラメータを用いたモデルへ拡張を行いたい。

また、開発における時間変化と不確定要素を考慮したシミュレーションを行い、9 つのタイプでのモデルの比較が行えた。このシミュレーションでは非線形項の影響を考慮した、ソフトウェア信頼度曲線を得ることができた。また、時間変化と不確定要素を考慮したシミュレーションを行うことで、想定する複数のタイプの開発に対して、時間変化と不確定要素が影響を及ぼすか、考察を行うことができた。今後の展望として、以下があげられる。

(1) 実際の開発で得られる値を用いた開発終了予測への適用

(2) 不確定要素の根本原因の解明

上にあげた 3 つについて詳細を以下に記す。

(1) については、現在提案されている欠陥数の予測や開発工数の予測に対して、我々が提案するモデルを用いてより精度のよい予測が行える手法を提案する。この予測を行うことで、ある開発にどれだけのリスクがあるのか、遅くともいつまでに終わるのかといったコストや納期についても予測が行え、リスクが回避できるからである。

(2)について、開発において不確定な要素としてあげられる開発チームのスキルのばらつきや開発環境の違いによるばらつきといった根本原因を解明し、定量的な尺度をもって開発全体を評価する方法を提案する。原因を解明を定量的な尺度を用いて原因を解明することでばらつきの少ない開発チームの選別や、開発環境の選定が行え、それによりより効率的な開発が行えるためである。

参考文献

- [1] ライルド, M, リンダ・ブレナン, M, キャロル, 中野誠・鷺崎弘宜 訳: ”演習で学ぶソフトウェアメトリクスの基礎【ソフトウェアの測定と見積もりの正しい作法】”, 日経BP社 (2009)
- [2] Yamada, S., et al. : S-Shaped Reliability Growth Modeling for Software Error Detection. Reliability, IEEE Transactions on R-32, Issue: 5 475 - 484 (1983)
- [3] GOEL, A. :Software Reliability Models: Assumptions, Limitations, and Applicability. IEEE TOSE, VOL. SE-11, NO. 12 (1985)
- [4] GOEL, A. and Okumoto, K. :”Time-dependent error-detection rate model for software reliability and other performance measures”, IEEE Trans. Reliability, Vol.R-28, No3, pp.206-211 (1979)
- [5] J. D. Musa and K. Okumoto ”A Logarithmic Poisson Execution Time Model for Software Reliability Measurement” Proceedings of the 7th international conference on Software engineering, p.230-238, March 26-29, (1984)
- [6] Z. Jelinski, P. B. Moranda ”Software reliability research”, ”Statistical Computer Performance Evaluation” W. Freiberger, Ed., New York: Academic, pp. 465-484, (1972)
- [7] Tamura, Y., et al. :A flexible stochastic differential equation model in distributed development environment. EJOR 168, 143—152,(2006)
- [8] Yamada, S., et al. :Software Reliability Measurement and Assessment with Stochastic Differential Equations. IEICE TFECCS E77-A(1), 109-116, (1994)
- [9] Zhang, N., et al. :A Stochastic Software Reliability Growth Model with Learning and Change-point.CDC’ 2010
- [10] A.L. Goel : ”Software error detection model with applications”, J. Systems and Software, vol. 1, pp.243 -249 (1980)
- [11] 山田 茂: ”ソフトウェア信頼性の基礎 - モデリングアプローチ”, 共立出版株式会社 (2011)