ソフトウェア工学 71−10
（1990. 2. 9）

エラーのライフサイクル・モデル

松尾谷 徹
日本電気㈱ システム開発本部

本報告はソフトウェア信頼度のライフサイクル・モデルを提案する。欠陥（error)のライフサイクルを３つの過程に分け、「欠陥の生成過程」「欠陥の除去過程」そして『障害の発生過程」とした。これら３つの過程における信頼性の現象を整理した。

A Software Reliability Life Cycle Model

Toru Matsuodani

NEC Corporation
29−23 Shiba 5−Chome Minatoku,
Tokyo 108 Japan

This paper descripes a software reliability life cycle model. The model consists of three phases: error implementaion phase, error remove phase, and fault occurrence phase. and discusses the characteristics of each phase.

1．まえがき

近年、ソフトウェアに対する需要は飛躍的に増大し、それと共にソフトウェアの信頼度が大きな問題として取り上げられている。

ソフトウェア信頼度の改善をして行くためには、より秀れた開発方法論の研究と、信頼度を定量的に計測し評価する問題とがある。

計測の問題はソフトウェア・メトリクスと呼ばれ、数学モデルを構築しデータを基にして統計的な評価を行うアプローチが用いられている。(1)

テスト過程におけるテスト量と発見した累積欠陥数の関係を表す信頼度成長モデル(software reliability model)の研究もこのような技法が用いられている。(2)

本稿ではソフトウェアの信頼度に関するメトリクスをそのライフサイクルに着目し３つの現象としてとらえ、さらに分析の手法として時間軸に注目した縦方向(longitudinal)の分析と、複数のソフトウェアを対象とする横方向(horizontal)の分析の二つの面から信頼度モデルを整理した。この整理をもとにソフトウェア・ライフサイクルを通したマクロな信頼度のモデルとして、欠陥(error)の生成から消滅までを取り扱うエラーのライフサイクル・モデルのフレームを提案する。

2．エラーのライフサイクル

信頼度の問題をソフトウェア・ライフサイクル(software life cycle)から見ると、テスト過程以外に「運用と保守段階」における信頼度（運用信頼度:operational reliability)と、「設計や実現段階」においてどの位の欠陥が生成されているかの問題がある。これをライフサイクルの順に、
①欠陥の生成過程(error implementation phase)
②欠陥の除去過程(error remove phase)
③障害の発生過程(fault occurrence phase)
と呼ぶことにする。通常のソフトウェア・ライフサイクル(ANSI/IEEEの定義）との対応を図１に示す。図に示すように開発過程におけるレビューは欠陥の除去過程に含める。

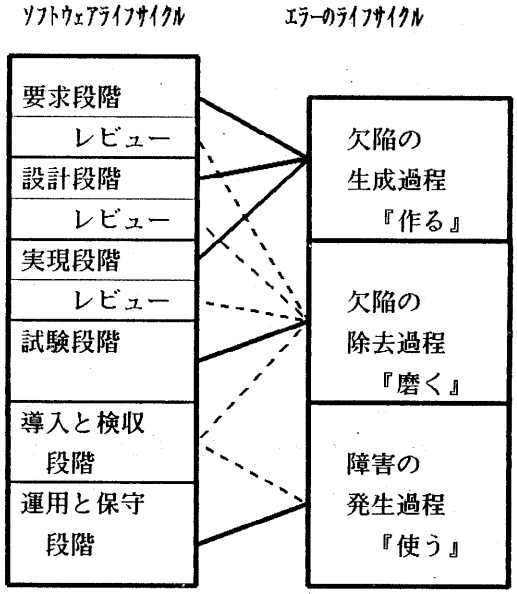


図１ソフトウェアライフサイクル と エラーのライフサイクル

2．1　欠陥の生成過程

この過程は、ソフトウェアの設計あるいは製造により欠陥が生成されるとする。対象となるソフトウェア自身の難易度や、設計を行う側の技術水準が影響すると考えられる。両者を合わせた単位規模当たりの信頼度の尺度をｑとし、ソフトウェアの規模を表す尺度をｍとする。（以降、小文字は単位規模当たりの尺度とする）

この過程によってソフトウェアに作り込まれる欠陥数をｅ 個とし、その関係を関数Ｆとして表す。

$$e = F(q) \qquad (1)$$

$$E = F(Q) = F(q, m) \qquad (1')$$

2．2　欠陥の除去過程

テストやレビューにより欠陥を除去する過程である。この過程は除去の程度を表すテスト量としてｒ、その結果除去した欠陥を作り込まれた欠陥ｅと区別するためバグと呼びｂで表す。観測されるバグ件数ｂはｒとｅによって決まる、その関係をＧとする。

$$b = G(e, r) \qquad (2)$$

２．３　障害の発生過程

開発過程を終了し、運用段階において発見される欠陥を障害(fault) と呼び区別する。障害件数 f はそのソフトウェアが運用段階においてどの程度使われるかを示す「使用の頻度」(frequency of use) により左右されると考え、その尺度をuとする。障害f は「使用の頻度」u と出荷時点に含まれる欠陥の数epにより決まると考え、その関係をHとする。

$$f = H(ep, u) \quad (3)$$

$$ep = e - b \quad (4)$$

２．４　縦方向と横方向

統計的な方法、つまり回帰分析によりモデルの係数を求める方法について考えると、縦方向の分析と横方向の分析の二つの方法がある。[3]

除去過程における信頼度成長モデルを例に考えると、対象とするソフトウェアとそのテスト行為についてテストの時間軸に沿って分析を行っている。すなわちテスト量r の時間的な変化r(t)により発見されるバグがどの様に増加するかを示している。これが縦方向の分析に相当する。

$$b = G(e, r(t)) \quad (5)$$

横方向の分析とはもう少し広い範囲を対象とするもので、開発コストの分析でよく知られているBoehm のコストモデル(COCOMO)[4] のような分析方法のことである。[5] 信頼度に影響すると考えられる要因と、その係数を観測値から求める方法である。

除去過程について考えると一つのソフトウェアを構成するモジュールなどの構成要素をi で表しそれぞれのテスト量riとそれぞれのバグbiについて分析することを横方向の分析と呼ぶ。

$$bi = G(ei, ri) \quad (6)$$

このような分類をライフサイクルに適用すると表１に示すようなモデルのフレームになる。

表の中の変数は分析の対象となるものを示している。それぞれの項目に対する研究の程度を大まかに記号◎○△で示した。

表１　ｴﾗｰﾗｲﾌｻｲｸﾙﾓﾃﾞﾙ のフレーム

	欠陥生成過程	欠陥除去過程	障害発生過程
横方向	(品質計)　△	r(t)　◎	u(t)　△
縦方向	qi　△	bi,ri　○	fi,ui　△

３．欠陥除去過程におけるモデル

３．１　縦方向のモデル

欠陥除去過程におけるモデルは古く1972年に発表されたZ.JelinskiとP.B.Morandのモデル[6] 以来いろいろなモデルが提案され、多くのモデルが実際のデータを基に検証を行っている。[2]

これらのモデルは欠陥除去過程の縦方向モデルとして分類する。

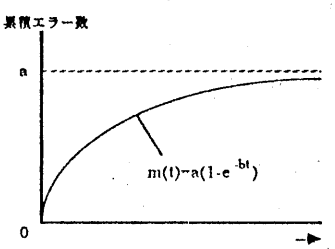
$$b = G(e, r(t)) \quad (7)$$

多くのモデルがあるが、欠陥除去の終了状態についてマクロに見ると共通点がある。それはr(t)を大きくした時に次の式が成立するとしている。図２に指数型とＳ字型の例を示す。

$$b = e(1-Exp(-\alpha r(t))) \quad (8)$$

$$b = e \quad r(t) \rightarrow 大の時 \quad (8')$$

αは係数、以降α、β、γ、κ 等は係数を表す。



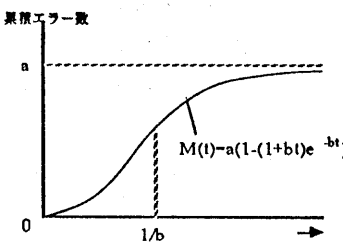


図2-1 指数型信頼度成長曲線　　図2-2 字型信頼度成長曲線

３．２横方向のモデル

異なるソフトウェアや異なる除去過程について分析を行うモデルである。縦方向のモデルに分類した信頼度成長モデルも成長の型が共通であれば、異なるソフトウェア間で係数を求めることにより一般化が考えられる。しかしまだその様な報告は見当たらない。

除去過程における横方向のモデルとしては、1984年に真野、塩川と共に発表した「ソフトウェア検査モデル」がある。[7]

このモデルは図３に示すような３つのサブモデルから構成されている。

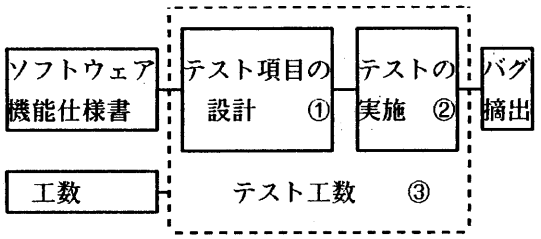


図３　検査モデルの構成

①検査項目抽出モデルは検査対象となるソフトウェアに対しどの位のテスト量が必要かを求める。主な要因は機能仕様書の規模mであり、その他の要因αjとして対象ソフトウェアの特性と検査の技術水準等がある。

②バグ摘出モデルは実施した検査項目数Riを主な要因とし、その他の要因αkとして対象となるソフトウェアの信頼度に関する特性（qi：改造の有無、新規性、汎用性など）と検査の特性（限界値検査やタイミングに関連する検査の有無など）から構成される。

③検査工数モデルは検査の生産性を表すものであり本稿では省略する。

それぞれのモデルは次式で表される。

$Ri = \Pi \alpha j \cdot m^{\beta}$　検査項目抽出モデル　(9-1)

$Bi = \Pi \alpha k \cdot Ri^{\gamma}$　バグ摘出モデル　(9-2)

色々な部門で運用した結果βはほぼ１に近く、γは0.5 に近いと見なせる。

式 (9-1)をm で割り (9-2)に代入すると、

$$bi = \Pi \alpha k \cdot (\Pi \alpha j)^{\gamma} \qquad (9\text{-}3)$$

この式（9-3 ）が意味することは、検査によって摘出されるバグの数は係数$\Pi \alpha k$ と、平均的な規模当りの検査量からの増減$\Pi \alpha j$ のγ乗（経験的にはγはほぼ0.5)に比例する。

この検査モデルが実際に観測されるデータとどの程度適合するかを示したのが図４である。図の横軸は観測値を示し、縦軸はモデルによる予測値を示している。両軸とも対数(log) で表示してある。

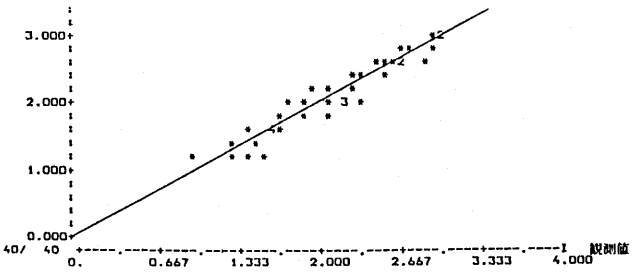


図４　検査モデルの適合性

４．障害発生過程におけるモデル

実際の運用段階において信頼度がどの様なふるまいをするかに関するモデルである。この分野における研究は極度に少ない。

４．１　縦方向のモデル

運用段階において、時間軸から見たとき信頼度がどの様に成長して行くか、あるいは障害件数がどの様に成長して行くかのモデルである。

モデルは運用段階における二つの区別が考えられる、利用者の数が一つか又は非常に少ない場合と、もう一つは利用者の数が多く、製品のコピーが多く出荷される場合である。

前者の例として1989年に報告した「フィールドにおける信頼度成長」がある。[8]

このモデルは発生する障害件数を指数型の式で表現したものである。

$$F = \alpha\ (1 - Exp\ (-\beta \cdot U(t))) \qquad (10)$$

そのグラフを図５に示す。

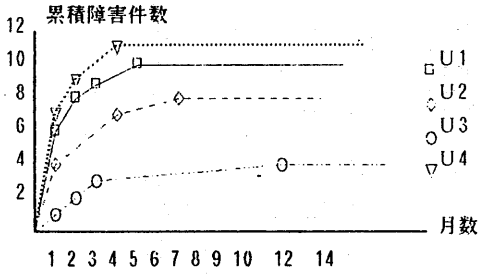


図５　単一利用者における信頼度成長

複数の利用者に出荷された場合の信頼度成長については1983年に報告した「のべ利用者数による信頼度成長モデル」がある。[9]　このモデルは製品の使われ方u(t)をのべ利用者月数で表し、モデルとして次式を使っている。

$$F = \alpha u(t)^{\beta} \qquad (11)$$

この式は製品出荷の初期段階で良く適合しており、中期以降になると観測値の方が少なくなる傾向がある。つまり式(11)は無限に障害が発生するようなモデルであるが、モデルの適合範囲に制限がある。図６に示すように適合性の高いモデルである。図の両軸は対数で示してある。

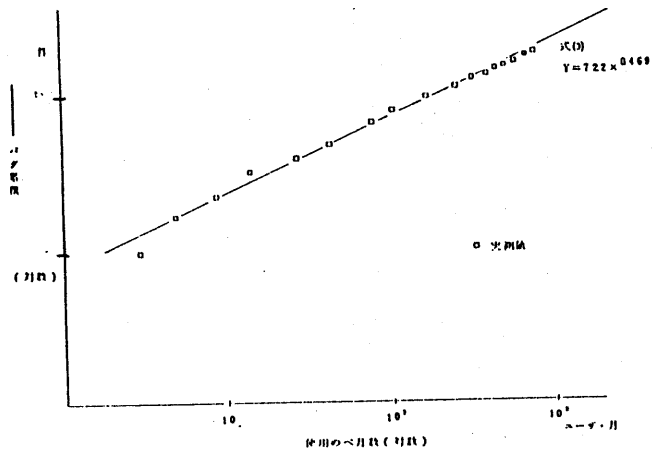


図６　のべ利用者数によるモデルと実測値

４．２　横方向のモデル

障害の発生過程における横方向の分析には、「利用の頻度」uiと製品の出荷時における欠陥数Epi の二つの要因が考えられる。

図５に示した「フィールドにおける信頼度成長」でもUiに差が有ることを示している。図５の４つのグラフは同じ製品を異なる利用者で運用した時のデータである。

多くの利用者が同じ製品を運用した時のUiについては同じ文献(8) の中に「ポアソン分布モデル」として示してある。

Epi とUiが共に観測されている例としては文献(9)で報告した「規模と使用頻度による障害モデル」が有る。この報告例はosのリリースのように一度に複数の製品をまとめて出荷する時の障害件数を分析したものである。図７が個々の製品規模と障害件数をプロットしたもので、図８がモデルにより「使用の頻度」と「規模」による補正をした後のものである。

$$Fi = Ui * m^{\kappa} \qquad (12)$$

Uiの尺度としては４段階の使用頻度の層別を行っている。意外なのは規模と使用頻度以外に製品の出荷品質の差Epi が統計分析で有意にならない点である。

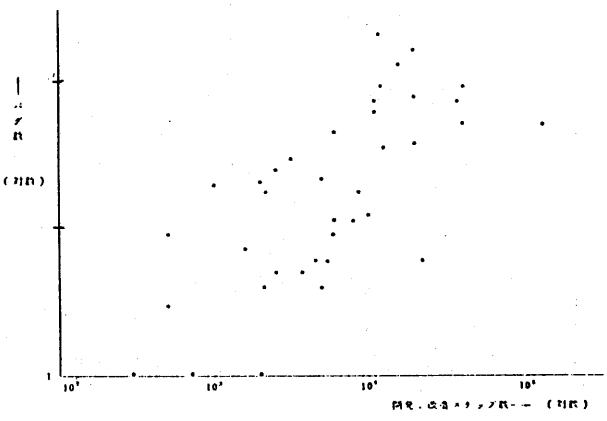


図７　製品の規模と障害件数

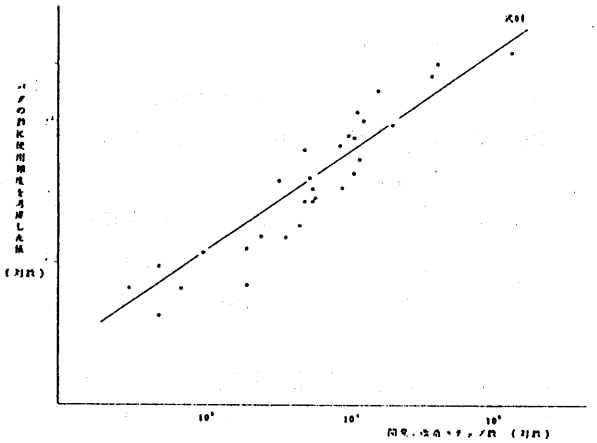


図８　製品の規模と使用頻度と障害件数

５．欠陥の生成過程におけるモデル

どの位の欠陥を作り込んだかと言う問題は、ソフトウェア開発技法の評価にとって重要な問題である。しかしながら作り込んだ欠陥の数eiを知るためには欠陥の除去過程でみつかったバグの数biと障害として発生した数fiから推測することになる。ところがbi,fi 共にモデルとして未完成である。そのためこの過程におけるモデルをいっそう難しくしている。

５．１　縦方向のモデル

生成過程の中だけで縦方向のモデルを作るのは難しいが、レビューやテストで除去したバグの数を工程毎にその種類と共に記録する方法がある。我々はその記録を品質会計と呼んでいるが、表２に示すようなものである。

表は欠陥の種類を発生した工程毎に分類し、それをレビューやテストで発見する毎に累計を取る。表の事例では要求段階での欠陥が要求段階のレビューにより32件、次の設計段階のレビューで7 件除去したと読み取る、そして最後の合計45件が要求段階で生成された欠陥数E に相当する。

実際の現場では広く行われているモデルである。永続的に記録して行くと作り込む欠陥数が明らかになる。

表２　品質会計の例

	要求段階欠陥	設計段階欠陥	実現段階欠陥	合計
要求段階	32			32
設計段階	7	42		49
実現段階	1	5	130	136
試験段階	3	5	49	57
保守段階	2	3	11	16
合計	45	55	190	290

５．２　横方向のモデル

作り込まれる欠陥数を表すモデルは、何等かの方法でその値を知る手段が必要となる。次に示すデータは、パッチと呼ばれるソフトウェアの一部分のみ修正する行為を行った時の誤り件数に関するものである。[9] パッチを作成する事は小さなソフトウェアを作ることに対応する。作るロジックが小さい事（93% は20行以内）と、確実に検証されるため発見される誤り数は作り込んだ誤り数にほぼ一致すると考えられる。図９に示すように行ったパッチの件数とその誤り件数との間には定常的な関係が成立していた。

同じ仕事ならqiやeiは大きな変動が無い　(13)

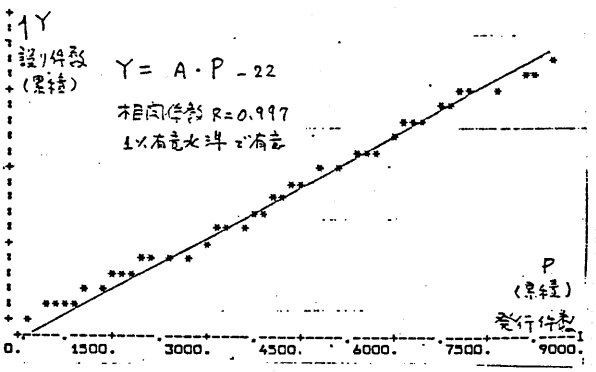


図９　パッチ件数とその誤り件数（累積）

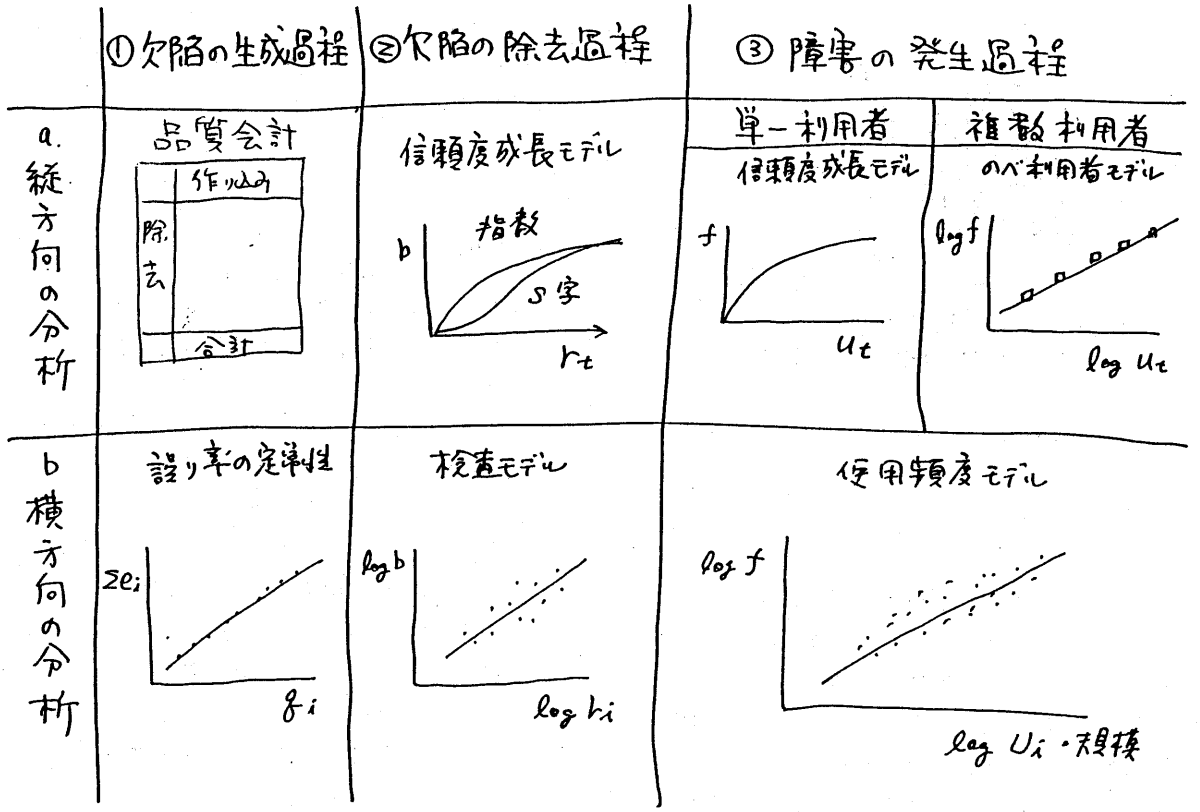


図１０　エラーのライフサイクル相関図

qiに明らかな差がある例としては、文献(10)で行った新規開発と改造の比較において２倍程度の差が報告されている。それ以外、難易度のような要因で製品間に大きな信頼度の差を見出だす事例に遭遇したことがない。（プログラマーの個人的な差は別）

６．エラーのライフサイクル・モデル

以上、欠陥のライフサイクルについて今まで知られていることを整理した。これを簡単に表したのが図10の相関図である。

またライフサイクルを連結して表したのが図11である。障害の発生過程における３つのモデル｛式（10）～(12)｝はいずれも障害件数を決める要因としてEpよりUiすなわち「使用の頻度」が遥かに大きな係数を持っていることを示している。

この事は運用が終わった段階でもなお障害Erが存在する事をも意味する。

テストや運用段階においてみつかる欠陥の数が飽和する現象は、含有する欠陥数が無くなる事ではなく、図12に示すように探す範囲に限りがあるため、その範囲内において飽和現象が観測されるのである。

ソフトウェアが持っている全ての欠陥の集合を考えその集合をEとする。

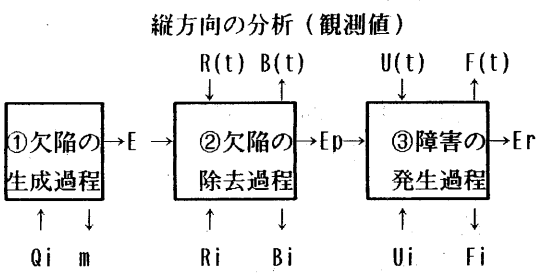


図11　ライフサイクル・モデル

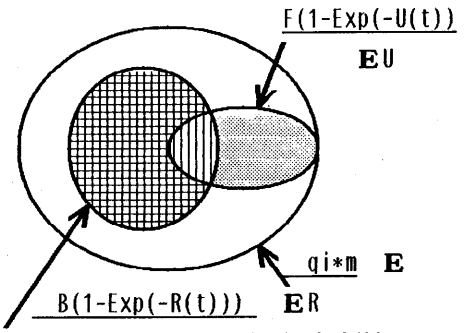


図12　欠陥の空間と探索空間

欠陥の生成過程はこの集合**E**を作ることに相当する。

$$\mathbf{E} = F\ (Qi, m) \qquad (14)$$

欠陥の除去過程は、集合**E**の中の部分集合**ER**の範囲内について除去を行う。従来の信頼度成長モデルは**E**＝**ER** を前提にしていたが、現状のテスト技術では**E**⊃**ER** と思われる。結果的に**ER**に含まれる欠陥数がB に相当する。

障害の発生過程は、集合**E**の中の部分集合**Eu**の範囲内について探索を行う。その結果**Eu** に含まれる欠陥数が障害F として発生する。**Eu** は利用者によって異なるため４．障害発生過程において説明した諸現象が発生する。

欠陥除去との関係は、**E**＝**ER** か**ER** ⊃**EU** が成立すれば障害は発生しない。しかしどちらも難しいようである。特に除去した欠陥の数であるBの量より**ER** ⊃**EU** の関係の方が重要である。

観測される結果としては、次式で**ER EU** が表される。

$$\mathbf{ER} = B(1-Exp(-R(t))) \qquad (15)$$

$$\mathbf{EU} = F(1-Exp(-U(t))) \qquad (16)$$

７．むすび

以上、信頼度のモデルについてライフサイクルの観点と横方向／縦方向の観点から分析を行った。

モデルとして未完成の部分が多く、これから観測値をもとに分析しなければならない課題が多い。

この分野の研究があまりにも信頼度成長曲線に集中していることに危機感をいだいおり、少しでも問題の整理に役立てば幸である。

参考文献

(1) 寺本・上村・松尾谷：”ソフトウェア・メトリクスにより生産性と品質を向上”，日経エレクトロニクス，No.344，1984年

(2) A A.Abdel-Ghaly ：” Evaluation of Competing Software Reliability Predictions" , IEEE VOL.SE-12, 1986-No9

(3) 菅野文友：”ソフトウェア・エンジニアリング”，日科技連出版社，p143，1973年 3月

(4) Boehm,B.W ："Software Engineering Economics", Prentice-Hall , 1981年

(5) 松尾谷：”開発のコストのメトリクスの現状と課題”，昭和63年電気・情報関連学会連合大会　1988年9 月

(6) Z.Jelinski他：” Software Reliability Research",(Statistical Computer Performance Ev-aluatin) , Press,Inc.NY, 1972年

(7) 松尾谷・真野他：”ソフトウェア検査モデル１～3"，情処学会29回全国大会，1984年後期

(8) 松尾谷：”ﾌｨｰﾙﾄﾞにおけるｿﾌﾄｳｪｱ信頼度に関する現象とそのモデル”，情処学会，ソフトウェア工学研究会64-1，1989年 2月

(9) 吉川・松尾谷：”ｿﾌﾄｳｪｱ誤りの統計的解析”信学会R研究会R83-11，1983年5 月

(10)松尾谷　徹、他：”ソフトウェア改造における生産性と品質”，日科技連、第3 回ｿﾌﾄｳｪｱｼﾝﾎﾟｼﾞｳﾑ，1983年9 月

(11)藤野・松尾谷：”デバグ工学における精度配分問題”，平成元年電気・情報関連学会連合大会，1989年9 月

(12)松尾谷・真野他：”ソフトウェア検査の外部発注における管理技術”，情報ｻｰﾋﾞｽ産業協会ｿﾌﾄｳｪｱ･ｼﾝﾎﾟｼﾞｪｳﾑ84，1984年6 月