

ソフトウェア開発における欠陥データ分布の 統計的性質とその品質管理への応用

小 室 睦^{†1}

ソフトウェア開発での欠陥抽出数を工程別に並べた分布を考える。このような分布が（形状パラメータが一定の）ワイブル分布にしたがうと仮定すると、異なる工程間の欠陥抽出数の間に簡単な定量的関係式が成立する。この関係式を動機づけとして、上流工程での欠陥抽出数の変化が下流工程にどのような影響を及ぼすかを定量化した影響評価モデルを構築する。さらに、この影響評価モデルを変形することで上流工程での欠陥抽出数とプロジェクト情報から下流工程の残存欠陥数を予測する品質予測モデルを構築する。最後に、これらのモデルを実プロジェクトに適用した結果について報告する。

A Certain Statistical Property of Software Defect Data and Its Applications to Quality Management

MUTSUMI KOMURO^{†1}

Consider the data of the numbers of defects along development phases. Assuming that this data follows the Weibull distribution, a simple formula is obtained between the numbers of defects in different phases. Making use of this formula, we establish a model which quantitatively shows the quality improvement effect of the activities in upper development phases. Furthermore, a quality prediction model is established, which computes the number of remaining defects in the testing phases. Finally, we report our experience of applying these models to real world projects.

1. はじめに

ソフトウェア開発がうまくいかない典型的なパターンの一つとして、上流工程での品質作りこみが不十分なためにテスト工程以降で多大の工数を費す場合がある。¹⁾このような事態を防止するためには、レビューを徹底させればよいはずであるが、どの程度まで「徹底」させれば十分なのかその判断がうまくできないと、明確な根拠もないまま楽観的な見通しのもとにプロジェクトを進行させてしまい結局は失敗を繰り返すことになる。上流工程におけるピアレビュー実績データから、テスト工程での品質向上効果を示し、さらにはその後実施するテスト工程での品質を予測するようなモデルがあれば、こういった事態を未然に防止でき、実用上重要である。

2. 工程別欠陥分布と信頼性工学手法

ウォータフォール型のライフサイクルモデルを仮定し、一連の工程 p_i ($1 \leq i \leq n$) があり、特に、最後のいくつかはテストにあてられているとする。

上流工程でレビューを徹底すれば製品品質はあがるという（定性的な）主張を定量的に定式化したい。製品品質は下流工程、例えばシステムテスト工程での欠陥件数で測ることができる。上流工程での品質管理の徹底具合をどう定量化するかが問題である。

2.1 指標定義

工程 p_i での（レビューまたはテストによる）欠陥抽出数を $f(p_i)$ であらわす。さらに、工程 p_i に対して、 $F(p_i), R(p_i)$ を以下のように定め、それぞれ累積前倒し率、残存欠陥率と呼ぶことにする。

$$F(p_i) = \sum_{j \leq i} f(p_j) / \sum_{1 \leq j \leq n} f(p_j),$$
$$R(p_i) = \sum_{j \geq i} f(p_j) / \sum_{1 \leq j \leq n} f(p_j) = 1 - F(p_{i-1}).$$

これらは信頼性工学における累積確率、残存確率（信頼度）の離散的類似である。 p_j がテスト工程の時、残存欠陥率 $R(p_j)$ は p_j 以降のテスト工程で抽出された欠陥の件数の全欠陥件数に対する比率であり、製品品質を表すものと考えられる。一方、上流工程 p_i に対する累積前倒し率 $F(p_i)$ は p_i までにどれだけの欠陥を抽出できたかという割合を示しており、レビュー等、

^{†1} 富士フイルムソフトウェア株式会社

上流工程での品質管理の徹底度合いを表しているものと解釈できる。

2.2 ワイブル分布とべき乗減衰則

信頼性工学でよく用いられる分布にワイブル分布がある。これは確率密度関数が $f_{\alpha, \beta}(t) = \frac{\alpha}{\beta} (\frac{t}{\beta})^{\alpha-1} \exp[-(\frac{t}{\beta})^\alpha]$ で与えられる分布である。歴史的には、ワイブル分布は最弱リンクモデルに関連して導入され、システムの弱い部分から欠陥が発生していくと考えれば欠陥の時系列分布としてワイブル分布を期待するのは自然である。特に、ワイブル分布で $\alpha = 2$ とした分布はレイリー分布と呼ばれ、ソフトウェアの欠陥件数や工数の分布に適用する研究²⁾³⁾がある。

ワイブル分布の残存確率関数は任意の異なる時刻 $t_1 < t_2$ に対して次のような等式を満たす。これをべき乗減衰則と呼ぶことにする。

$$R(t_2) = R(t_1)^c$$

ただし、 $c = c(t_1, t_2)$ は、 β に依存しない定数。

3. 2つのモデル

べき乗減衰則（の離散版）を仮定すると、異なる工程間の $F(p_i) = 1 - R(p_{i+1})$ と $R(p_j)$ の間の定量的関係を得ることができる。さらに、総欠陥数 $\sum_{1 \leq j \leq n} f(p_j)$ が開発規模 $size$ でほぼ決定されるという経験的事実を用いると次のような2つの仮説（モデル）を立てることができる。

工程記号 ST でシステムテストを表す。 p_i はコーディング以前の工程を表すものとする。

効果説明モデル

$$\log(ST) = -k_1 F(p_i) + A_1 \log(size) + (other\ factors) + \epsilon$$

品質予測モデル

$$\log(ST) = -k_2 \sum_{j \leq i} p_j / size + A_2 \log(size) + (other\ factors) + \epsilon$$

ただし、 $k_1, k_2 > 0, A_1, A_2$ は定数、(other factors) は難易度、開発経験などの要因の寄与、 ϵ は誤差項を表す。

効果説明モデルは p_i 以前の工程が ST 工程の欠陥件数にどう影響するかという品質向上効果を表すモデルであるが、説明変数に $F(p_i)$ を含むため、予測能力は持たない。品質予測モデルは、 $F(p_i)$ を $size$ を用いた項で置き換える近似操作により、予測（工程途中で入手可能なデータからの計算）を可能としたモデルである。近似のため精度は効果説明モデルより悪くなる。

4. 検証

前節で述べた2つの仮説（モデル）を実データによ

り検証した。使用したデータは47件のプロジェクトデータで、ポアソン分布を仮定した重回帰分析を適用した。 $F(p_i), d(p_i) = \sum_{j \leq i} p_j / size$ の係数、p値、回帰式の疑似決定係数の値を表に示す。

表1 効果説明モデルの検証結果

工程 p_i	疑似決定係数	$F(p_i)$ の係数	p 値
機能設計	86.1%	-3.48	2.2 %
詳細設計	87.8%	-2.96	0.2 %
コーディング	85.9%	-2.46	3.0 %
単体テスト	91.9%	-5.57	0.1 % 未満

表2 品質予測モデルの検証結果

工程 p_i	疑似決定係数	$F(p_i)$ の係数	p 値
詳細設計	85.4%	-0.069	0.9 %
コーディング	83.7%	-0.039	5.0 %
単体テスト	83.9%	-0.035	0.1 % 未満

5. 応用

効果説明モデルは、品質計画を立案する際のシミュレーションや組織レベルでの改善効果の算定に用いている。品質予測モデルは開発途中で品質見通しを立てるのに用いている。2つのモデルを適用したプロジェクトでは、組合せテスト以降の欠陥密度が従来実績と比較して半減するなどの成果があがっている。

6. 結論

ソフトウェア開発における工程別欠陥件数データの分布についてワイブル分布と同様の性質（べき乗減衰則）を仮定することで、上流工程の品質向上効果を示すモデルおよび工程途中で品質予測を実現するモデルを構築できることを示した。これらのモデルは実プロジェクトに適用され、テスト工程の欠陥密度を半減させるなどの成果をあげている。

参考文献

- 1) Humphrey, W. S.: *Winning with Software*, Pearson Education, Inc. (2002).
- 2) Putnam, L. H.: A General Empirical Solution to the Macro Software Sizing and Estimation Problem, *IEEE Transaction on Software*, Vol.4, No.4, pp.345-361 (1978).
- 3) John E. Gaffney, J.: On Predicting Software Related Performance of Large-Scale Systems, *Int. CMG Conference*, pp.20-23 (1984).