

ソフトウェアの簡易品質評価の実験

2H-2

謝花 喜宏* 平山 昭弘* 若木 守**

* NTTソフトウェア株式会社

** NTT情報通信処理研究所

1. はじめに

ソフトウェアの品質を向上させるには、開発技法、開発環境、及び開発体制の強化を図り、開発段階で確実に品質を作り込むことが最も重要であることは言うまでもないが、ソフトウェアバグの利用者に与える迷惑度を考えると、利用者の立場にたった開発部門とは別の部門による検査・確認も大切である。

しかし、マシンによる製品検査は、検査環境等も大きくなり一件当たりのバグ修理費用が膨大となる。⁽¹⁾ また、ソフトウェア開発工程の最後に位置するため、検査不合格時の品質向上期間が確保出来ないという課題がある。

この為、製品検査の実施に先立って、品質の作り込み状況を探針・評価することが重要となる。

品質評価手法には、試験の進行に伴うバグの成長傾向から評価する縦方向予測法や、品質決定要因値を経験式に当てはめて評価する横方向予測法が知られているが、⁽²⁾ 試験進捗が均一でないとバグ成長の軌跡が不正確になる、経験式を求めるのは困難等の問題がある。

本論文では、バグの検出率や収束率等により簡易に品質を評価する手法を提案する。また、本手法を大規模オンラインシステムのオペレーションシステムを構成するプログラムの品質評価に適用させたところ、品質が悪いプログラムの指摘に有効であったことを示す。なお、ここで言う品質とは、信頼性のことを示す。

2. 製品開発モデル

以下のwater fallモデルに沿って開発している。接続試験工程の後に製品検査工程がある。

表1 開発モデル

開発工程	作業内容
機能設計	システム構成に必要な全ての機能を抽出し外部仕様を決定
詳細設計	構造 上記機能をプログラムにマッピングする。また、当該プログラムをサブプログラムに分割、サブプログラム間インタフェースを決定する
	論理 サブプログラムの詳細手順を設計する
製造	コーディング・コンパイル
単体試験	サブプログラムの単体試験
接続試験	サブプログラム間接続試験

以下サブプログラムのことをプログラムと呼称する。

3. 品質評価運用モデル

製品開発部門と検査部門との関係は、以下の通り。先取り品質評価とは、製品開発部門が製品を開発中又は開発完了後直ちに行うものである。

先取り品質評価の為のデータは、製品開発部門から提示されるものであるが、先取り品質評価の後に、マシンによる製品検査があるので、虚偽のデータの申告が出来ない体制となっている。

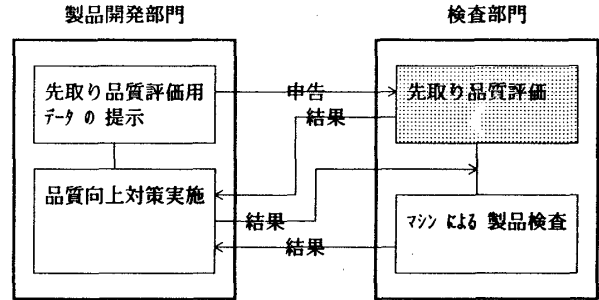


図1 品質評価運用モデル

提示される時期・データは次の通り。

- 単体試験工程着手時 : 予測バグ数の提示
- 単体試験・接続試験中・完了直後: バグ検出実績値の提示

4. 先取り品質評価法

先取り品質評価とは、要注意プログラムの抽出が目的であり、本当に品質が良いか悪いかは、先取り品質評価結果に基づいて、開発の進め方を詳細に調査・分析して判断する。

(1) 残留バグ数による評価

改造規模当たりの残留バグ数の推定値(申告値)の大ききで評価する。

残留バグ数/KS > 標準値 の条件を満足するモジュールは要注意。

原則 その数値が大きい程、注意度が高いが、改造規模が小さいモジュールは、数字のトリックにより、非常に大きな値を示すので、改造規模が小さいモジュールの数値は、参考程度にする。

(2) 収束率による評価

バグ成長曲線の簡易版であり、単体試験でのバグ検出数と接続試験でのバグ検出数とにより、以下の様に評価する。

$$\text{収束率} = \frac{\text{接続試験でのバグ検出数}}{\text{単体+接続試験でのバグ検出数}}$$

$$\text{収束率} < \text{標準値} - 5\% \quad \text{---} \quad \textcircled{1}$$

$$\text{収束率} > \text{標準値} + 5\% \quad \text{---} \quad \textcircled{2}$$

を満足するプログラムは、要注意。

①式は、単体試験でのバグ検出数が多いこと、換言すれば、バグの成長が飽和状態にあることを示すもので、成長曲線の点からは、安定しているといえる。ここでは、インタフェース絡みのバグが検出されないのは逆に品質が不安と判断するものである。

②式は、接続試験でのバグ検出数が多いことを示す。これは、バグ成長曲線でもバグの成長が未だ飽和状態になっていないことを示すもので、ここでは、単体試験工程が不十分のまま、接続試験に突入したと判断する。

(3) 検出率による評価

予測バグ数と実発見バグ数との乖離状況により、以下の様に評価。

$$\text{検出率} = \frac{\text{実発見バグ数}}{\text{予測バグ数}}$$

$$\text{検出率} > 105\%$$

$$\text{検出率} < 95\%$$

を満足するプログラムは、要注意。

上記の3つの評価を併用して、要注意プログラムを指摘する。

- ①注意度大: 残留バグ数、収束率、検出率 全て悪し
- ②注意度中: 残留バグ数、収束率、検出率の内 2つ悪し
- ③注意度小: 残留バグ数、収束率、検出率の内 1つ悪し

5. 実験

前述の先取り品質評価手法を大規模オンラインシステムのオペレーティングシステムを構成する14個のプログラムの品質評価に適用させた。これらプログラムの特徴は、改造が主体でその規模は約5ks程度である。

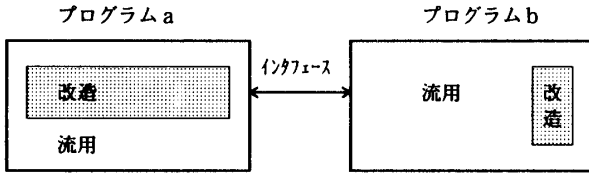


図2 試験対象プログラムの特徴

6. 先取り品質評価結果

(1) 残留バグ数による評価

改造規模と残留バグ数の関係を図3に示す。要注意プログラムとして、PRO-C、PRO-N及び改造規模が小さいものでPRO-P、PRO-Iを抽出した。

(2) 収束率による評価

改造規模と収束率との関係を図4に示す。要注意プログラムとして、PRO-C、PRO-H、PRO-M、PRO-O、PRO-D及びPRO-Tを抽出した。

なお、改造規模に比例してプログラム間インタフェースに対する影響度も変化すると想定し、改造規模が小さいものは、インタフェースバグが検出されなくとも妥当と判断している。すなわち、収束率 < 標準値-5% は適用外とする。

(3) 検出率による評価

予測バグ数と実発見バグ数との関係を図5に示す。要注意プログラムとして、PRO-C及びPRO-Nを抽出した。

したがって、要注意プログラムの注意度を整理すると以下の様になる。

表3 先取り評価実験結果

注意度	要注意モジュール
大	PRO-C
中	PRO-N
小	PRO-P、PRO-I、PRO-H、PRO-M、PRO-O、PRO-T

7. 検査工程・サービス工程でのバグ検出状況

先取り品質評価の妥当性を検証するため、その後の検査や商用サービスでの半年間のバグ検出状況を追跡した。

PRO-C、PRO-T、PRO-N、PRO-Hの順に改造部分にバグが検出された。また、PRO-C、PRO-T、PRO-Iで流用部分との関係でバグが検出された。

8. 実験結果の傾向・分析

- (1) 改造部分にバグが検出されたプログラムは、全て先取り品質評価で指摘したプログラムである。
- (2) 注意度小のPRO-Tのバグ検出数が、注意度中のPRO-Nより多いのは、改造規模が大きいためバグ遭遇確率が高い為と推定する。
- (3) 流用部分との関連バグは、PRO-Cを除いて改造規模が小さいプログラムで発生している。これは、改造規模の小さいプログラムは、残留バグ数による評価において、製品開発部門に流用部分との関係を含めて残留バグを推定しているかを再確認することによりある程度検出可能と考える。

9. 先取り品質評価の有効性評価

残留バグ、収束率、及び検出率評価の各評価で、ORで抽出されたプログラムは確実に品質に問題ありと言える。

10. おわりに

本先取り品質評価手法は、製品開発担当と製品開発管理者間にも適用できるものである。製品開発管理者は、本手法により品質要注意プログラムを指摘し、当該プログラムの開発の進め方を詳細に調査・分析することにより高品質のプログラムの開発が可能となる。

予測残バグ数 (件/ks)

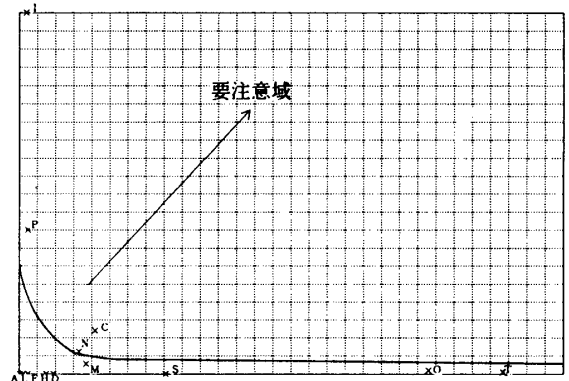


図3 残留バグ数による評価結果

収束率 (%)

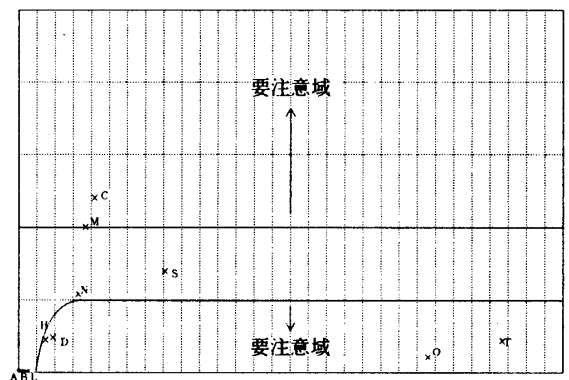


図4 収束率による評価結果

実発見バグ数 (件)

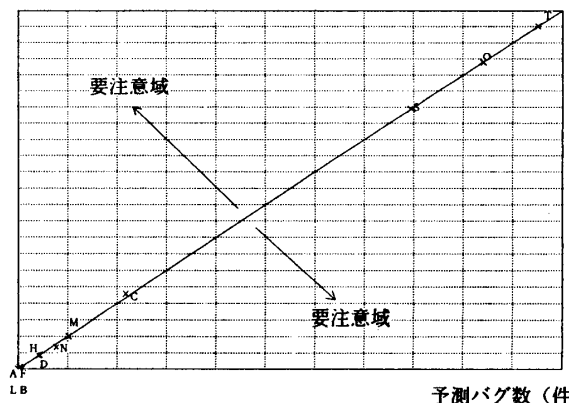


図5 検出率による評価結果

(参考文献)

- (1) 菅野文友: ソフトウェア・デザインレビュー、日科技連出版社、1982、pp. 65
- (2) 石井康夫: ソフトウェア検査と品質保証、日科技連出版社、1987、pp. 89-94