

カバレッジテストの品質向上効果に関する分析

1 R-4

高橋健司 下村隆夫 磯田定宏

NTT ソフトウェア研究所

1. はじめに

カバレッジテストは、テストの網羅性を定量的に把握できるという利点を持つ。単体テストにカバレッジテストを適用することは、モジュール単体を対象とするため容易である。実際、カバー率100%を目標にカバレッジテストを実施しているところもある。しかし、結合テストへの適用に関しては、具体的な適用法やテスト工数および効果が明らかでないという問題点がある。

そこで、実際のテスト作業の分析結果に基づきカバー率とテスト工数やバグ検出率との関係を明らかにした。この結果に基づき、機能テストにカバレッジツールを援用した効果的なテスト手法を提案する。

2. テスト作業

(1) テスト対象プログラム

デバugg(C言語記述 102KS)の一部(データ表示部 14KS)で、開発部隊の結合テスト完了後のプログラムをテスト対象とした。

(2) テスト手法

機能テストおよび構造テストの2つのテスト手法をデータ表示部に対して独立に適用し、工数、カバー率、検出バグ数を収集した。

[機能テスト]

機能仕様書を基に機能要因を洗い出し、それらを組み合わせることにより機能テスト項目を作成。

[構造テスト]

未通過ブランチを通過させるための、テスト項目を作成。

また、機能テストでは、ある時点でカバー率が向上しなくなったため、構造テストに切り替えた。

3. テスト作業結果の分析

(1) 機能テストの有効性

各テスト手法における工数・KS当りのカバー率、検出バグ数の変化を図1,2に、テスト効率を表1に示す。

表1. テスト効率

指標	機能テスト	構造テスト
(a)カバー率向上工数	5.2 人時/KS・%	13.9 人時/KS・%
(b)カバー率向上によるバグ検出効果	0.37 件/KS・%	0.27 件/KS・%
(c)バグ検出効率	0.071 件/人時	0.019 件/人時

以上より、機能テストの方が以下の点で有効であることが明らかになった。

- ① 機能テストの方がカバー率を容易に向上させることができる。
- ② カバー率が同じだけ向上しても、機能テストの方がバグ検出効果が大きい。

(2) 機能テストの限界

機能テストでは、あるところまでくるとカバー率がまったく上がらなくなるが、ここで構造テストに切り替えると、さらにカバー率を上げることが可能であった(なお、切り替え後の指標は、それぞれ(a)6.9人時/KS・%、(b)0.10件/KS・%、(c)0.015件/人時であった)。この構造テストのテスト項目を調べてみると、機能テストにおける機能要因が漏れていたことがわかった。すなわち、外部仕様からでは機能要因を十分洗い出せないことが、機能テストにおいてカバー率が上がらなくなる原因である。

(3) カバレッジ援用機能テスト手法の提案

機能テストで限界に達した場合には、漏れていた機能要因を未通過のブランチの通過条件を調べることにより発見し、これを補充して機能テストを強化する方法(カバレッジ援用機能テスト手法)が効果的であると考えられる。

[カバレッジ援用機能テスト手法]

① まず、機能仕様書を基に、各機能を一通り漏れなく検査する機能テスト項目を作成し、テストする。

② つぎに、未通過ブランチの通過条件を調べることによりテストされていない機能要因を洗い出し、それらの機能要因を補充して機能テスト項目を作成し、テストする。ただし、ここでは未通過ブランチから容易にわかる機能要因のみを洗い出すこととし、ソースコードを深追いすることはしない。

③ 以上のテストでも残った未通過ブランチについては、ソースコードを追いながら通過条件を調べ、テストされていない機能要因を洗い出し、それらの機能要因を補充して機能テスト項目を作成し、テストする。

4. おわりに

今後は、カバレッジ援用機能テスト手法についてデータ収集、分析を行っていく予定である。

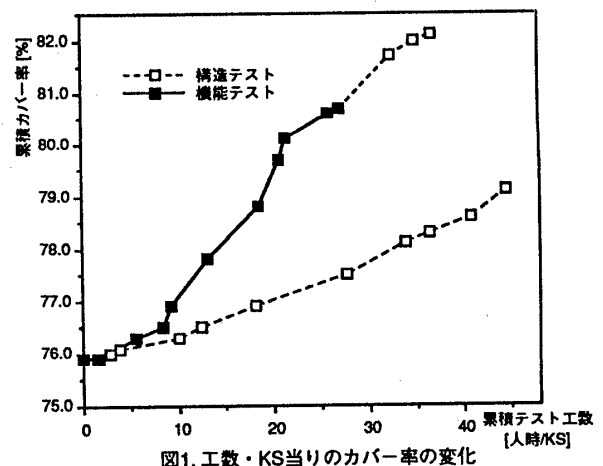


図1. 工数・KS当りのカバー率の変化

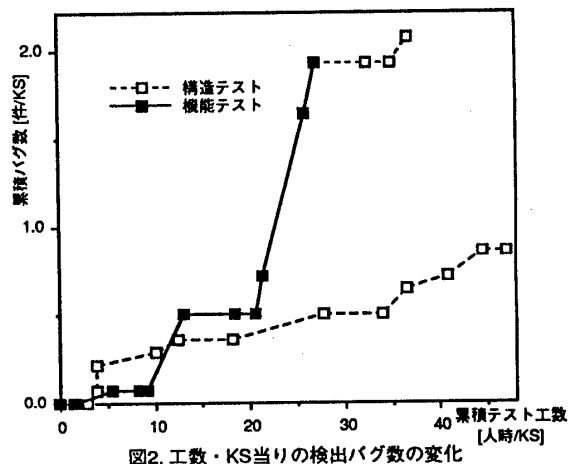


図2. 工数・KS当りの検出バグ数の変化

An analysis of program quality improvement through coverage test

Kenji Takahashi, Takao Shimomura, Sadahiro Isoda
NTT Software Laboratories