

## テスト結果の履歴情報を用いたテスト項目間の関連抽出手法の提案

佐々木 愛美<sup>†</sup> 河村 透<sup>†</sup> 小笠原 秀人<sup>†</sup>

株式会社東芝 ソフトウェア技術センター<sup>†</sup>

### 1. はじめに

本稿では、ソフトウェアの不具合を修正した後に実施する再テストにおいて、限られた時間で、より早期にデグレードを検出することを目的として、テスト結果の履歴情報を用いて優先的に実施すべきテスト項目を選び出す方法を提案する。

### 2. 本稿の課題

ソフトウェア開発におけるテスト工程ではテスト実施と不具合修正を繰り返し行う。不具合修正の後に実施する再テストでは、不具合を修正した変更個所の動作確認だけでなく、その変更によりそれまで正しく動作していた機能に新たな不具合が生じる“デグレード”についても確認する必要がある[1]。

近年のソフトウェアの大規模・複雑化に伴い、テスト項目の規模は数千～数万件に増加している。不具合修正を行うたびに、全てのテスト項目を実施することは非常にコストがかかる。限られた時間の中でデグレードを検出するためには、ソースコードの変更による影響を確認できるテスト項目を選び出し、優先的に実施しなければならない。

デグレードを検出する可能性があるテスト項目を選択するには、テスト対象のドメイン知識やソフトウェアの全体構成・挙動、テスト項目が対象とするソースコードの範囲などを把握し、テストすべき範囲を見定める必要があり、高いスキルが必要とされる。

そこで本稿では、テスト技術者のスキルに左右されずに、デグレードを検出する可能性があるテスト項目を選び出す方法を提案する。これまでに実施したテストの結果をもとに、不具合を修正した変更個所の動作確認のために実施するテスト項目と以前同時にテスト結果が変化したことがあるテスト項目を、デグレードを検出する可能性があるテスト項目として選び出す。

### 3. テスト項目間の関連

過去に実施したテストのテスト結果には、同じ欠陥に起因して不具合が発生した状況やその欠陥を除去して不具合が修正された状況などが、“テスト結果の変化”という形で残されている。これには次の2つのパターンがある。

1つ目のパターンは、2つのテスト項目のテスト結果が同時に不合格から合格または、合格から不合格へ変化する場合である(図1)。あるモジュールに潜在する欠陥の影響で、そのモジュールを利用する機能A, Bに不具合が発現し、機能A, Bのテスト項目が不合格となる。ソースコードを修正し欠陥を取り除くと、機能A, Bの不具合は修正され、テスト項目はともに合格に変化する。

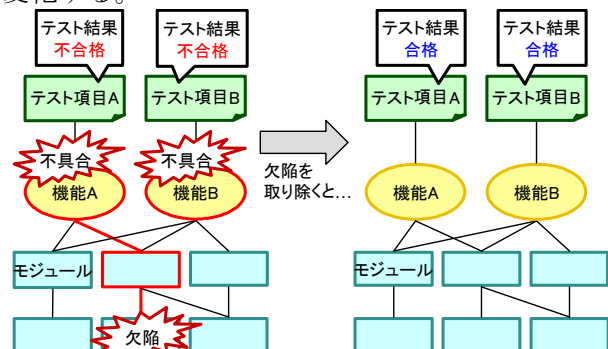


図1 テスト結果の変化(パターン1)

2つ目のパターンは、あるテスト項目のテスト結果が不合格から合格へ変化すると同時に、別のテスト項目のテスト結果が合格から不合格へ変化する場合である(図2)。あるモジュールに潜在する欠陥の影響で、そのモジュールを利用する機能Aには不具合が発現し、テスト項目Aは不合格となるが、機能Bには不具合が発現せず、テスト項目Bは合格となる。ソースコードを修正し欠陥を取り除くと、機能Aの不具合は修正されテスト項目Aは合格に変化するが、機能Bはソースコードの変更による悪影響を受けテスト項目Bが不合格に変化する。

このように、同時にテスト結果が変化するテスト項目は、同じソースコードの変更による影響を受けてテスト結果が変化した可能性があると考えられる。このテスト項目間の関連をテス

A method to relate between the test cases with a history of test results

Manami SASAKI<sup>†</sup>, Toru KAWAMURA<sup>†</sup>,

Hidetoshi OGASAWARA<sup>†</sup>

<sup>†</sup> Corporate Software Engineering Center, Toshiba Corporation

ト結果の合格/不合格の履歴から抽出することで、変更箇所の動作確認のテスト項目と関連があると思われるテスト項目を、ソースコードの変更箇所の影響を受けてテスト結果が変化する可能性があるテスト項目として推定できると考える。

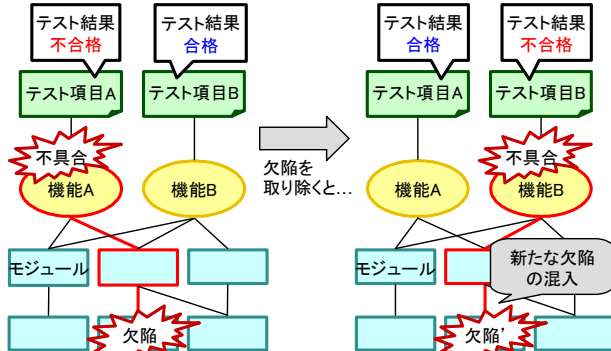


図 2 テスト結果の変化 (パターン 2)

4. テスト結果の履歴情報を用いたテスト項目間の関連抽出手法

テスト結果の合格/不合格の履歴から、3章で示したテスト項目間の関連を抽出するには、テストした回数ごとにテスト結果を並べ、それをテスト項目間で比較する。その方法として、①同じ回に実施したテストの結果が一致しているか比較する方法と、②同じ回でテスト結果が変化しているか比較する方法の2つを検討した。

方法① (図 3) では、テスト結果の履歴(a)を使って2つのテスト項目を比較し、同じ回のテスト結果が一致する数を合計する。一致した数が多い順にテスト項目間の関連が強いとする。

	1回目	2回目	3回目	4回目	5回目
テスト項目 1	○	○	×	×	○
" 2	×	×	○	○	○
" 3	○	○	×	×	○
" 4	○	×	×	×	×
" 5	○	○	○	○	×

(a) テスト結果の履歴

テスト項目	2	3	4	5
テスト項目1と テスト結果が一致する数	1	5	3	2

テスト項目1と関連が強いテスト項目は、3、4、5、2の順

図 3 テスト項目間の抽出方法①の例

ただし、テスト結果の特性として、テスト工程終盤では多くのテスト項目のテスト結果が合格である状況が続くため、同じ回のテスト結果が一致する数が増え、多数のテスト項目間の関連が強くなるという問題点がある。

一方、方法②(図 4)では、テスト結果の履歴(a)から作成したテスト結果の変化の履歴(b)を使って2つのテスト項目を比較し、同時にテ

スト結果が変化した回数を集計する。回数が多い順にテスト項目間の関連が強いとする。この方法では、テスト工程終盤で合格が続いてもテスト結果は変化しないため、同時にテスト結果が変化した回数が増えず、必要以上にテスト項目間の関連が強くなることはない。

	1回目	2回目	3回目	4回目	5回目
テスト項目 1	○	○	×	×	○
" 2	×	×	○	○	○
" 3	○	○	×	×	○
" 4	○	×	×	×	×
" 5	○	○	○	○	×

(a) テスト結果の履歴

変化がある: "1"  
変化がない: "0"

	1~2	2~3	3~4	4~5
テスト項目 1	0	1	0	1
" 2	0	1	0	0
" 3	0	1	0	1
" 4	1	0	0	0
" 5	0	0	0	1

(b) テスト結果の変化の履歴

テスト項目	2	3	4	5
テスト項目1と同時に テスト結果が変化した回数	1	2	0	1

テスト項目1と関連が強いテスト項目は、3、2、5の順

図 4 テスト項目間の抽出方法②の例

以上より、テスト結果の履歴情報からテスト項目間の関連を抽出するには、方法①よりも方法②の方が良いと考えられる。

5. おわりに

不具合修正を確認する時に優先的に実施すべきテスト項目として、テスト結果の履歴情報から、ソースコードの変更箇所の動作確認のテスト項目と過去に同時にテスト結果が変化したことがあるテスト項目を選び出す方法を提案した。この提案方法は、現在開発中のテスト管理システム[2]に、一機能として実現している。

提案方法を実際のソフトウェア開発に適用する際は、毎回すべてのテスト項目に対してテスト結果が残されているとは限らないことを考慮する必要がある。今後はテスト結果が多少欠落しても、必要なテスト項目が導き出せるように提案方法の改良を進める。

参考文献

- [1]Rex Black/テスト技術者交流会監訳/トップスタジオ訳: 基本から学ぶテストプロセス管理, 日経 BP 社(2004)
- [2]河村透, 小笠原秀人: ソフトウェアのテスト管理システム, 東芝レビューVol.66 No.1(2011)