仕様に関する観点を加味した回帰テストケース選択手法の提案
The proposal of regression test case selection method that contains the viewpoints for the specification

主査　　：奥村　有紀子（有限会社デバッグ工学研究所）
副主査　：秋山　浩一　（富士ゼロックス株式会社）
　　堀田　文明　（有限会社デバッグ工学研究所）
研究員　：畑中　義和　（GE ヘルスケア・ジャパン株式会社）
　　吉田　善之　（株式会社インテック）
　　杉野　敦司　（株式会社インテック）
　　渋谷　英里子（株式会社エフネット）
　　平山　智美　（株式会社エフネット）

研究概要
ソフトウェアに変更を加えた際，デグレードという現象が生じることがある．「ソフトウェアへの変更」によって生じる「デグレード」の有無を調べるテストを回帰テストと呼ぶ．もし，デグレードがソフトウェアのどこに発生するのか予測できない場合は，ソフトウェアの品質を保証するためには実施済みのテストをすべてやり直さなければならない．しかし，回帰テストを実施する時間をそれまでのテスト時間と同程度に確保できることは通常ない．そこで，過去の経験，変更範囲および変更した機能との関連性等からデグレードが発生すると思われる範囲を推測し，回帰テストを行うことになる．しかしながら，影響を受ける範囲および影響の受ける程度をどのように割り出したのかを明示することは難しい．そこで，仕様に関する観点についてあてはまるかチェックをし，回帰テストを実施する際の優先順位を算出することで，より確証が得られるテストケース選択手法を提案する．

Abstract　The side-effect bugs occurs at the timing of software changes. The regression test checks for the presence of software change and side-effect bugs. If the occurrence location of the side-effect bugs inside the software cannot be predicted, it is necessary to perform all of software tests from the top/beginning for quality evidence. However, it is difficult to secure the same amount of time for regression test as the full test. Therefore, the regression test scope is conjectured on the basis of experience, but there is difficulty delivering satisfactory explanation to the stake holders. So, an idea of regression test case is proposed in which orders the test priority by checking specification point of view. As a result, more of the conclusive evidence is achieved to choose the best test scope for finding side-effects bugs.

1.　はじめに
不具合の修正や機能の拡張などによってソフトウェアに変更を加えた際，以前のテストでは不具合が検出されずに合格したテストケースに不具合が発生することがある．これをデグレードと呼ぶ．このデグレードの有無を調べるのが回帰テストである．もし，デグレードがソフトウェアのどこに発生するのか予測が困難な場合，ソフトウェアの品質を保証するためには実施済みのテストをすべて最初からやり直さなければならない．しかし，研究員の開発現場では短期間かつ低コストが求められており回帰テストに割り当てられる時間および工数が限られている．さらにマイナーリリースであってもソフトウェア全体をテストするため，メジャーリリースと同じケース数を実施している．したがって，回帰テストについて，すべてのテストケースを再度実行することは困難である．そのため実際には，影響範囲に関係すると思われるテストケースだけに絞ってテストされる．影響範囲はソフトウェアの変更に関する情報と有識者およびテスト担当者の過去の経験により決定するが，その方法が妥当であるか十分な確証が得られていない．そのため，回帰テストの十分性の判断に困るケースや，回帰テストを行ったにもかかわらず，市場で不具合が見つかるケースがあるという現状が明らかになった．なおテストの自動化を検討しているが，実現

するためにはツールの選定やコード作成をはじめ準備期間が必要であるため，短期開発が続く現状では導入に至っていない．このような背景から本グループでは回帰テストケースの定量的選択基準を研究テーマとした．

本論文の構成は以下のとおりである．2 章では研究課題を述べる．3 章では先行研究とその適用結果について説明する．4 章では本論文の手法と適用結果を述べ．5 章では考察を述べる．

2. 研究員の現状と課題

2.1. 前提条件

本研究を進めるにあたり以下を前提条件とした．

・システムテストレベルで検出した不具合改修時の回帰テストを対象とする．
・デグレードの発見を対象とし，改修部分で発生した新規不具合の発見は対象外とする．
・回帰テストは，すべてのテストケースを実施後，バグが改修済みのソフトウェアに対して実施する．
・本手法を利用する人物は，プログラムコードを読むことがない，テスト専門の担当者とする．

2.2. 用語の定義

本論文で使用する用語について表 1 に定義する．

表 1 用語定義

用語	本論文での定義
改修	不具合の修正や機能の拡張にともなうプログラムの変更および追加
デグレード	不具合を修正したことで改修のなかった部分で発生する不具合（side effect bug）
確認テスト	不具合を見つけて改修したら，その不具合を見つけたテストケースを再度実行して，不具合が修正されていることを確認するテスト
回帰テスト	不具合の修正や機能の拡張などによって，従来問題のなかった機能の動作に影響がないことを検証するテスト．回帰テストは，すべてのテストを終了後，バグが改修済みのソフトウェアに対して実施するときなどがある．

2.3. 現状と課題

研究員が回帰テストの現場での実態や，改善したい事項の現状調査を行なった結果，以下のような意見が挙がった．またそれぞれの意見に対しての課題を Q:Quality，C:Cost，D:Delivery の観点で分類し関連性を表 2 に示した．

2.3.1. 現状

(1) ステークホルダに対して，回帰テストケースの選択範囲について，根拠および確証があり，かつ定量的な説明ができないため，できる限り多くのテストケースを実施せざるを得ない．
(2) テストケースに優先度を付けて，不具合が発生する可能性が高い箇所からテストを行いたいが，有識者でないと優先度を判断することができず，テストケースを今までのテスト実施順番と同じ順で実施している．
(3) 回帰テストに割り当てられる時間が十分でない場合，回帰テストの途中であっても，タイムオーバーでテストが打ち切りになり，テストの質が低下する．
(4) 全テストケースを回帰テストとして行う場合，機能およびテストケースが追加される度に回帰テストもテストケースが増えていくため，回帰テストを実行する時間が増えていく．
(5) 過去の経験，変更範囲および変更した機能との関連性等からデグレードが発生すると思われる範囲を推測し，回帰テストを行うのであるが，テスト終了後に回帰テストを省いたところから不具合が見つかることが稀にある．
(6) ソフトウェアのマイナーリリースの場合でも，メジャーリリースと同時間の回帰テスト時間が必要となっている．（付録 1.参照）

2.3.2. 課題

(1) ステークホルダに対して，回帰テストケースの選択範囲を定量的に説明ができるよう，根拠となるデータを提示できること．
(2) 有識者に頼ることなく，回帰テストとして行うべきテストケースを選択できること．それにより，不具合発生率が高いケースや潜在的に不具合検出が高いケースからテスト可能になること．
(3) 回帰テストに割り当てられる時間により，優先順位の高いテストケースからテストを実行できること．
(4) 回帰テストを実施する時間をそれまでのテスト時間と同程度に確保できなくても，限られた時間の中で，より効率的にデグレードを検出できるようにするできること．
(5) テストの選択範囲の推測の根拠および成否を振り返ることができること．

表 2　現状と課題

分類	現状	課題
Q:Quality（品質根拠）	(1)，(3)，(5)，(6)	(1)，(2)，(5)
C:Cost（工数）	(1)，(2)，(4)，(6)	(1)，(2)，(3)，(4)
D:Delivery(時間)	(1)，(2)，(4)，(6)	(1)，(2)，(3)，(4)

2.4. 本研究の目的

2.3 の課題を解決するための，回帰テストケース選択手法を研究した．また本研究の目的を以下のとおり設定した．

「回帰テストのテストケースを選択する時に，優先すべきテストケースの選択範囲を的確に抽出するための，“テストケース選択メトリクス”を作成する」

3.　先行研究の調査とその効果検証結果

3.1. 先行研究の調査

回帰テストのテスト範囲の選択方法として，いくつかの報告がされている[1][2][3]．それらを下記ポイントに基づいて比較検討した．

・2.3.2 で述べた課題が解決できるか
・2.1 の前提条件で述べたテスト担当者だけで定量的なテスト範囲選択が判断できる方法を提案しているか．

検討した結果，「数理計画モデルを利用したテストケースの選択手法」[1]を参照することにした．そこで，研究員の実プロジェクトの過去データ（不具合記録，テスト実施記録）を元に，先行研究で回帰テストのテスト範囲選択が適切に行えるかについて検証を行った．

3.2. 先行研究の検証結果

先行研究で紹介されている手法[1]図 2 では，過去の不具合記録，過去のテスト実施記録を最大 5 世代前まで記入し“不具合検出の期待値”，“未実施期間” を求めることができる．この手法を表計算ソフトと研究員の実プロジェクトの過去データ（不具合記録，テスト実施記録）を用いて検証を行った．その結果，論文の通りにテスト範囲を選択できることが確認できた．ただし，検証で算出された結果対して，研究員のプロジェクトでの実状との間に相違点があることが分かった．詳細は 3.3 に述べる．

3.3. 先行研究検証結果と目的との相違点に関する問題点の把握

(1) 不具合が検出されなかったテストケース
先行研究では過去のテスト結果で不具合が検出されたテストケースに対して，今後も不具合が検出されるであろうテストケースとして“不具合検出の期待値”が高くなる．これは「不

具合の 80%は全体の 20%の箇所に集中するという経験則が知られており，過去に不具合が頻発した箇所は，今後も不具合を引き起こす可能性が高い．」という根拠に基づいている．したがって，不具合が検出されていないテストケースにも不具合が潜んでいる可能性があり，これについては対策が取られていない．その結果，テスト範囲の選択対象外となる点についてステークホルダへ十分に説明ができないという問題が判明した．図 1 にイメージを示す．

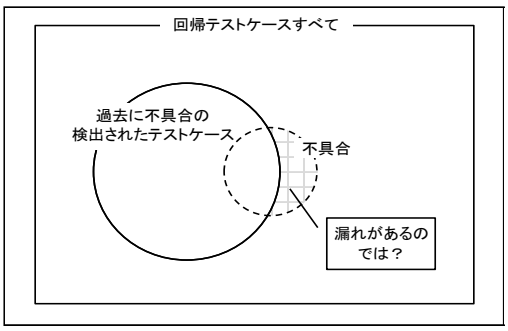

図 1 先行研究おける不具合検出イメージ

(2) 未実施期間の長さ

先行研究では未実施期間の長いテストケースに対して，デグレードを検出する期待が高いと考え，未実施期間の長さを“不具合検出の期待値”算出に利用している．ところが，研究員のプロジェクトには回帰テストのテスト範囲選択が定量的に行えないことが原因で，テスト範囲選択をせずにすべての回帰テストを実施しているプロジェクトが存在する．そのため，未実施期間自体が存在せず，未実施期間の長さが“不具合検出の期待値”算出として有効に利用できないという問題が判明した．

3.4. 問題点解決のための解決策の検討

3.3 で述べた問題を解決するためには，これまで不具合が検出されていないテストケース，および未実施期間のないテストケースもテスト範囲として選択の対象となるよう，先行研究の手法に新たな指標を加える対策が必要との結論に至った．次の 4 章に考案した手法について述べる．

4. 本論文の手法と適用結果

4.1. 解決策

4.1.1. 解決策のポイント

テストケースの選択基準や優先順位を研究員のプロジェクトの有識者にヒアリングした結果，そのテストケースが評価対象としている仕様からみた指標に基づき回帰テストのテストケースを選択していることわかった．さらに各プロジェクトで見つかったデグレードを分析した結果，原因の多くが傾向を持つことがわかった．この 2 つの結果をまとめたものを指標とし，先行研究に加えることで 3.3 で挙げた問題を解決できると考えた．図 2 にイメージを示す．

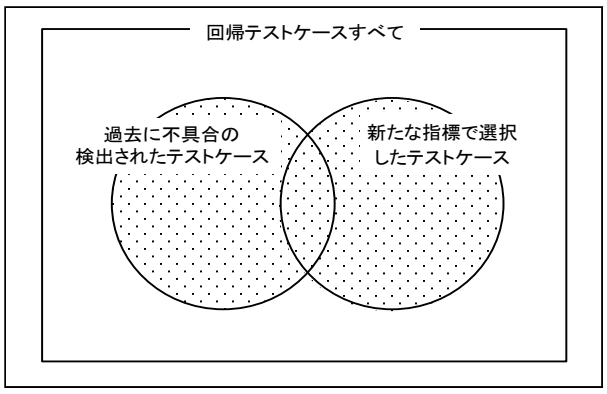

図 2 RTCS 法でのテスト選択ケースイメージ

指標を検討する際の指針として，以下のポイントを考慮する．

・テスト経験が浅い人でも仕様書を読んで，判断できること
・リリース後にデグレードが見つかった場合に，ソフトウェア利用者に与える不利益の重要度が大きいテスト機能が選択されること
・回帰テスト以前にデグレードを見落としやすいと懸念される機能が選択されること
（例：回帰テスト以前のテスト時にテストパターンやケース数が多い機能）
・効率的なテストケース選択とするために，指標となる項目は多くなりすぎないこと

4.1.2. 解決策

研究員で 4.1.1 の解決策のポイントをもとに議論し，5 つの指標を選定した（表 3）．先行研究の結果に，5 つの指標を加味し， 表計算ソフトで 4.1.3 で述べる算出式を設定しておくことで適切テストケースが選択できる「RTCS（Regression Test Case Selection）法」を考案した（図 3）．この「RTCS 法」を使うことで，仕様や機能の特徴をテスト範囲選択の観点として捉えることができる．そうすることで過去の不具合検出の結果だけでなく，不具合が起こる可能性があるテストケースを優先的にテストケースとして選択できるようになると考えた．

表 3 仕様観点でチェックする指標

No	指標	実例
1	入力項目が多い	10 個以上の入力項目
2	操作の自由度が高い	必須項目ではないものが 5 か所以上
		必須項目が 5 か所以上
		操作順があるもの
3	１つの要求に複数のステークホルダ	他システムと連携している
4	１つの要求に複数の機能	１操作で複数機能が動作するもの
5	操作が複雑	１操作で入力内容が変化するもの
		１操作で入力項目が変化するもの

未実施期間の閾値 2　　今回の回帰テスト時間 20 分

テストケースNo	世代ごとの実施結果 1	2	3	4	5	未実施期間	テスト必要度T	仕様観点の指標 1	2	3	4	5	テスト必要度K	テストの必要度	テスト時間(分)	選択結果
ケース1	×	○	○	○	−	1	0.3	○	○	○			0.6	0.9	5	選択
ケース2	×	○	×	○	○	0	0.4	○					0.1	0.5	5	未選択
ケース3	○	○	−	−	−	3	1		○			○	0.2	1.2	5	選択
ケース4	○	○	−	○	○	0	0			○	○	○	0.8	0.8	10	選択

○：合格　X：不合格　－：未実施　　○：該当

図 3 RTCS 法

4.1.3. 算出方法について

先行研究は過去のテスト結果から「不具合検出の期待値（以降テスト必要度 T）」が大きい値（最大 1）のテストケースから順番に選択する手法である．そのため本研究で選定した 5 つの指標にも「仕様観点のテスト必要度（以降テスト必要度 K）」を最大 1 となるように決めた．両者を足し合わせた値を「テストの必要度（テストの必要度　＝　テスト必要度 T　＋　テスト必要度 K）とする．この「テストの必要度」が大きい値（最大 2）のテストケースから順に回帰テストケースとして選択することで，より不具合発生率が高いケースや潜在的に不具合検出が高いケースからテスト可能になると考えた．

さらに研究員のプロジェクトで見つかったすべてのデグレードを本研究で選定した 5 つの指標に分類した（付録 2.参照）．その結果，不具合の傾向から各指標にテストの必要度を重みづけすることで,より効果的な結果が得られると考えた．

まず指標 4 は，件数が多いため，他の指標よりもテストの必要度が高いと考えた．さらに改修を行わないとリリースできないような致命的デグレードであるかどうかを判断した．その結果,指標 3 は，他の指標と比べると明らかに件数が高くなっている．そのため指標 3 は最も重要な観点であると考えたためテストの必要度を一番高くした．

本論文では各指標の優先順位を「指標 3>指標 4>指標 1= 指標 2=指標 5」と定義付け，合計が 1 となるように各指標のテストの必要度の重みを決めた．

4.2. RTCS 法のテストケース選択手法の手順

RTCS 法とは，先行研究の手順に 4.1.2 で検討した 5 つの指標を加えることで最適なテストケースを選択することを目指した手法である．なお本研究では，表計算ソフトを用いることで，テストケースを選択するためのテスト必要度を自動算出し，適用できるよう考慮した．

テストケースを選択する手順は次の 4 ステップで構成される（付録 3.参照）．

ステップ 1　過去に実施したテスト結果を記入（毎回入力）
ステップ 2　5 つの指標の判定結果を記入（初回と変更時のみ入力）
ステップ 3　テストケースごとのテスト時間を記入（初回と変更時のみ入力）
ステップ 4　今回の回帰テスト時間(テストに使える時間)を記入（毎回入力）
ステップ 5　テストケースの選択（毎回入力）

1，2，5 ステップの詳細については，以下に述べる．

4.2.1. 過去に実施したテスト結果（ステップ 1）

表計算ソフトでは過去最大 5 世代まで記入することができ，回帰テストの実施結果が合格であれば○，不合格であれば×，未実施であればーをリストから選択する．すると，“不具合検出の期待値”，“未実施期間”，“テスト必要度 T”が自動算出される．先行研究にて未実施期間があることで不具合に気付かないリスクが高くなることが述べられているため，未実施期間に対して必ずテストを実施したい回数を決める．その回数を閾値としてテスト必要度 T を算出している．それぞれの算出方法については表 4 を参照．算出例について付録 4.参照.

表 4 過去に実施したテスト結果からの算出方法

項目名	算出方法
不具合検出の期待値	過去に実施したテストのうち，不合格となったものの割合
未実施期間	最後にテストを実施してから未実施だった回数
閾値	未実施期間に対して必ずテストを実施したい任意の回数
テスト必要度 T	未実施期間が閾値以上の場合は 1(最大値) それ以外は不具合の期待値

4.2.2. 5 つの指標の判定結果（ステップ 2)

テストケースごとに仕様観点の指標に該当するかどうかの判定結果を記入する．有識者に頼らず，仕様書を読んで該当すれば○，該当しない場合はブランクのままとする．一度判定結果を記入すると，仕様変更や機能追加が行われない限り，変更する必要はない．入力した判定結果から，“テスト必要度 K”を算出する．それぞれの算出方法については表 5 を参照．算出例について付録 5.参照.

表 5 仕様観点からの期待値合計の算出方法

項目名	算出方法
テスト必要度 K	該当する指標に対するテストの必要度の重みを足し上げる.

4.2.3. テストケースの選択（ステップ 5)

4.2.1 で算出した“テスト必要度 T”と 4.2.2 で算出した“テスト必要度 K”を合計し，“テストの必要度（T+K）”を算出している．算出例について付録 6.参照.

”テストの必要度（T+K）”が大きいほど，優先的に選択する．なお，テストケースごとのテスト時間の合計が，ステップ 4 で毎回入力する今回の回帰テスト時間を超えないようにテストケースを手動で選択する．テストケース選択例について付録 7.参照.

4.3. 検証結果

RTCS 法を研究員の 2 プロジェクトに適用して検証する事により，以下の相違点を確認することができた.
各プロジェクトの条件と結果については表 6，表 7 に記載する.

(1) RTCS 法では，先行研究に比べて，プロジェクト A，プロジェクト B ともにテスト必要度の値にメリハリが付くようになった.

(2) RTCS 法では，先行研究に比べて，プロジェクト A，プロジェクト B ともにテストケースの選択数がプロジェクト A で，3 件から 15 件に，プロジェクト B で 27 件から 50 件に増加した.

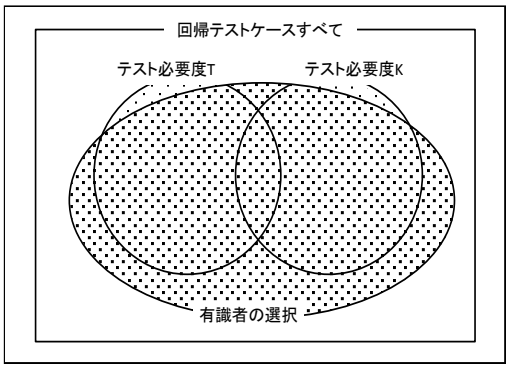

図 4 RTCS 法でのテストケース選択結果と有識者のテストケース選択イメージ

(3) プロジェクト B で有識者 A の 58 件，有識者 B の 72 件，有識者 C の 76 件に対して，RTCS 法は 50 件とすべての有識者よりテストケースが少ない結果となった．イメージを図 4 に示す.

表 6 プロジェクト A の結果

テスト総件数	75
テスト総工数(分)	375
回帰テスト工数(分)	60

先行研究		
	テストケース数	
テスト必要度T	選択数	未選択数
0		72
0.3	3	
合計	3	72

(1)

(2)

RTCS法		
	テストケース数	
テスト必要度K	選択数	未選択数
0		32
0.1		22
0.2		6
0.3	8	
0.4	4	
0.5	1	
0.6	1	
0.8	1	
合計	15	60

(1)

(2)

表 7 プロジェクト B の結果

テスト総件数	222
テスト総工数(分)	1926
回帰テスト工数(分)	480

	テストケース数
有識者A	58件
有識者B	72件
有識者C	76件

(3)

先行研究		
	テストケース数	
テスト必要度T	選択数	未選択数
0		195
0.3	6	
0.5	17	
1	4	
合計	27	195

(1)

(2)

RTCS法		
	テストケース数	
テスト必要度K	選択数	未選択数
0		127
0.1		15
0.2		10
0.3	20	20
0.4	6	
0.5	8	
0.6	4	
0.7	4	
0.8	3	
0.9	1	
1	3	
1.1	1	
合計	50	172

(1)

(2)

5. 考察

5.1. RTCS 法適用に対する本研究の達成度

先行研究と本研究の結果を比較し，2.3.2 で本研究課題として挙げた(1)～(5)の各項目の達成度について考察する．

(1) ステークホルダに対して，回帰テストケースの選択範囲を定量的に説明ができるよう，根拠となるデータを提示できることについて

先行研究では，過去のテストでデグレードが検出されたテストケースだけが回帰テストケースの選択対象となる．これに対し，本研究の「テスト必要度 K」を加味することで，デグレードが検出されていないテストケースであってもテストケースの選択対象となる．

このことから，ステークホルダに対して，「テストの必要度」の指標をテスト選択の根拠とし，定量的に示すことができる．さらに RTCS 法で算出した表計算ソフトの結果をデータとして提示するできる．

(2) 有識者に頼ることなく，回帰テストとして行うべきテストケースを選択できること．それにより，不具合発生率が高いケースや潜在的に不具合検出が高いケースからテスト可能になることについて

本研究では，普段有識者が回帰テストケースの選択を行う際に考慮していることも指標に含めている．前項 4.3.2 の結果から，本研究のケース選択範囲は，有識者に頼らなくても，先行研究以上に達成できたといえる．

ただし「本研究で選択したテスト範囲」と，「有識者が選択したテスト範囲」の差についての課題は 5.2 で述べる．

(3) 回帰テストに割り当てられる時間により，優先順位の高いテストケースからテストを実行できることについて

先行研究で算出した「テスト必要度 T」に本研究の指標「テスト必要度 K」を加え「テストの必要度」の検証結果を比較すると，テストの優先順位が変わることは確認できた．

ただしこの優先順位の差についての課題は 5.2 で述べる.

(4) 回帰テストを実施する時間をそれまでのテスト時間と同程度に確保できなくても，限られた時間の中で，より効率的にデグレードを検出できるようにするできることについて
先行研究と本研究の検証結果を比較すると，過去のテストケースで不具合が検出されていないケースも，「テスト必要度 K」を加味して，テスト実施すべき優先度の判断を行うことができたといえる.

(5) 回帰テストの選択の理由を履歴として残すことで，推測の根拠および成否を振り返ることができることについて
RTCS 法で算出した表計算ソフトの結果を参照することで，テスト範囲の選択理由を振り返ることが可能である.

5.2. 今後の課題

本研究は，過去に行った回帰テストの結果から確認を行い，研究員が課題としていた項目が 5.1 で述べたように達成できたことを確認した．しかしながら RTCS 法を用いて新規の回帰テストを検証するには至らなかった．そのため研究員が所属するプロジェクトで実際に RTCS 法を適用して効果を検証する必要がある．検証した結果，図 5 となることを理想とするが，RTCS 法で選択されなかったテストケースでデグレードが発生した場合は以下の 2 点を検討したい.

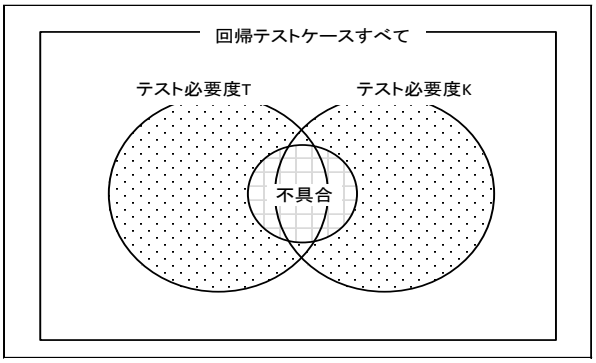

図 5 理想となる回帰テストケース

(1) 開発者視点の導入
5.1 で述べた，有識者と本研究で選択したケース選択数の差の理由の 1 つが，開発者視点でのアプローチとなっていると予想される．そのため，プログラムレベルの変更の影響範囲など，開発者視点の指標を加えることでより適切な範囲のケース選択ができる可能性があると考えている．この仮説の検証は今後の課題である.

(2) 優先順位の確認と指標のカスタマイズ
本研究では 4.1.2 で述べた通り，我々研究員の中でテスト範囲選択における加味すべき仕様観点を選定したが，本研究を適用するプロジェクトによって，各指標のテストの必要度の重みと指標のカスタマイズが必要と考えている．たとえば，指標の中に「入力項目が多い」という指標があるが，入力項目が存在しないシステムの場合，この指標はテスト範囲選択の指標として機能しない．また，我々が選定した指標以外にもテスト範囲選択として追加したい指標がある場合にも，指標のカスタマイズが必要となると考えている．それにともない優先順位が最適となるよう実プロジェクトで確認をしながら各指標のテストの必要度の重みを変えていけばよいと考える.

6. 参考文献

[1]佐々木　愛美　他，「数理計画モデルを利用したテストケースの選択手法」，東芝レビュー，69 巻 6 号，pp.40-43，2014
(http://www.toshiba.co.jp/tech/review/2014/06/69_06pdf/f03.pdf)
[2]第 25 年度ソフトウェア品質管理研究会　第 5 分科会（テスト設計グループ），「派生開発における影響個所の把握改善によるテスト範囲の特定方法の提案」，(財)日本科学技術連盟，2009
(http://juse-sqip.jp/workshop/seika/2009/attachs/5_1_report.pdf)
[3]町田欣史，「回帰テストの適用と実施」，ソフトウェア・テスト PRESS，Vol.6，pp.108-113，2008