

テストパターンマトリックスを用いたテスト観点漏れ低減の提案

A Proposal for Decreasing Omissions of Test Viewpoints using

Test Pattern Matrix

リーダー：飯沼 真一（株式会社 AGEST）
研究員：田口 真義（リコーIT ソリューションズ株式会社）
 会田 健太郎（株式会社日立システムズ）
 柳井 聡（株式会社日立システムズ）
 北島 優介（株式会社日立ソリューションズ・クリエイト）
 宮田 奈津子（株式会社 IHI エスキューブ）
主査：喜多 義弘（長崎県立大学）
副主査：秋山 浩一（株式会社日本ウィルテックソリューション）
アドバイザー：西田 尚弘（株式会社日新システムズ）

研究概要

ソフトウェアテストで高品質かつ短納期のテストを実現するには、テストベースの記載内容に加え、暗黙的な情報をテスト観点として抽出しテストすることが重要である。テスト経験の浅いテスト担当者であっても、テストベースを基にした観点抽出は可能である。しかし、テストベース外の暗黙的な情報からの観点抽出は、テスト経験の浅いテスト担当者には困難である。その理由は、多くのベテランが形式化せずに経験を使って観点を抽出しているためである。

そこで本研究では、ベテランの暗黙知となっているテスト観点を形式化して、活用することを考えた。具体的には、テストで発生したバグから得た知見を利用し、マトリックス化し、経験の浅いテスト担当者へ適用する VTV メソッド (Veteran Test Viewpoint method) を提案する。このメソッドを適用し、経験の浅いテスト担当者でも新たな観点抽出ができることを実験にて確認した。

Abstract It is important to extract the implicit information as the test viewpoints, other than the contents described in the test base, such as the functional specifications, to realize the software testing with higher quality and shorter delivery time than the previous it. Even if the tester is inexperienced, he/she can extract test viewpoints based on the test base. However, it is difficult to extract the test viewpoints from the implicit information which are not described in the test base.

In this paper, we aim the formalize of test viewpoint analyzed from the veteran tester's implicit knowledge. And we propose VTV method (Veteran Test Viewpoint method) which is the method to create the matrix of test viewpoints using the knowledge gained from bugs detected by software testing. We experimented with VTV method in each affiliation's field to confirm the availability of this method.

1. はじめに

各研究員の担当しているソフトウェアテストの現場では、以前に実施したテストよりも高品質かつ短納期でのテスト実施を求められることが増えている。この要求を実現するには、機能仕様書等のテストベース内外からテスト観点を抽出しバグを多く見つけ、後工程での手戻りを減らすことが重要である。

研究員が担当するプロジェクトで収集したバグ 154 件に対し、テスト経験年数ごとの 1 名当たりのバグ検出数ならびに、テストベース外のバグの検出割合を調査した結果、表 1 の結果となった。

第 39 年度 ソフトウェアテストコース

この結果は、バグ検出数に経験年数による差はないが、テストベース外のバグの検出割合はテスト経験が進むほど高くなっていることを示している。

表 1 テスト経験と検出したバグの分析

テスト経験 (括弧内は経験年数)	1名あたりの バグ検出数 (件/人)	テストベース外の割合 (%)
新人 (4年未満)	11.3	0.0
中堅 (4~10年未満)	9.2	13.0
ベテラン (10年以上)	9.2	28.2

一方、研究員がテスト関係者 50 名の現在の役割を調査した結果、表 2 の結果となった。この結果は、ベテランはテストマネージャとなり、テスト観点抽出に工数をかけることが難しいことを示している。

表 2 テスト経験年数とテスト時の役割比率

テスト経験 (括弧内は経験年数)	テストマネージャ (%)	テスト担当者 (%)
新人 (4年未満)	0.0	100.0
中堅 (4~10年未満)	12.5	87.5
ベテラン (10年以上)	37.8	62.2

以上のことをまとめると、次のとおりとなる。

1. テスト経験年数が増えると、テスト担当者はテストベースに記載がない点についてもテストを実施するようになる。
2. テスト経験年数が増えると、テスト担当者はテストマネージャとなる可能性が高まり、その結果としてテスト観点抽出の業務から外れるようになる。

そこで、ベテランの暗黙知を形式知化し、その情報を利用することでテスト観点の漏れを減らすことを本研究の目的として設定する。

本研究ではテストベース外のテスト観点を抽出し、マトリックスで整理してテスト実施する手法を提案する。そして、本手法の有効性を検証するために実験を行う。

2 章では、テスト実施上の問題点と課題を述べる。3 章では、ベテランの暗黙知を形式知化して新人と中堅が活用する方法を述べる。4 章では、マトリックスの有効性を検証するために行った実験および評価を述べる。5 章では、実験結果について考察する。6 章では、残課題と今後の進め方について述べる。

2. 関連研究

2.1 先行調査

経験の浅い新人、あるいはテスト対象に対し経験が不十分な中堅テスト担当者であっても、テストベース内に記載されていることを基にしたテスト仕様書の作成は可能である。しかしながら、テストベースに記載されていないことに対してテスト仕様書を作成するときには、バグの出そうな箇所を見出せず、テスト観点の漏れが発生している。

新人や中堅がテストベース外のテスト観点の抽出漏れを起こす原因は、テストベース外のテスト観点の多くがベテランの暗黙知であり、形式知化されていないためである。

上記「ベテランの暗黙知の形式知化」に関する既存研究については、以下が挙げられる。

中辻ら^[1]は、テスト仕様書を作成する際に、熟練者の業務知識を「業務」、「業務機能」、「部品」の関係性で整理し、更にインプット時にベテランのエラー推測の情報を追加し、テスト観点の考慮漏れを防ぐ方法を提案している。この手法により、テスト観点考慮漏れは防ぐことができるが、暗黙知継承の適用条件として類似のテストでの適用が条件となるため、汎用性に欠ける。

川上ら^[2]は、既存機能と過去のバグ情報を関連付け、その情報からバグ推測を行いテスト観点に追加していくテスト設計手法の方法を提案している。この手法により、既存機能と過去のバグ情報の関連付けは有効ではあるが、関連付けに時間がかかることや新機能への適用が難しいといった課題がある。

飯沼^[3]は、探索的テストでのベテランの留意点を決められた記述形式でパターン化して表現することで形式知化させ、中堅へと適用させる方法を提案している。こちらも、ベテランの技術の継承は有効に行えるが、探索的テスト実施時とシチュエーションが限られるため、汎用性に欠ける。

いずれの先行研究も、テスト全般で広く活用するためにはさらなる考慮が必要である。

なお、その他類似の研究として、品質特性を基にテスト項目をマトリクスで表現して作成する小島ら^[5] や吉岡ら^[5] の研究があげられる。

2.2 解決すべき課題

ベテランの暗黙知の形式知化は先行研究されているが、暗黙知継承の適用条件として類似のテストでの適用が条件になることや関連付けに時間がかかること、新機能への適用が難しい探索的テスト実施時の適用とシチュエーションが限られており、汎用性に欠けるといった残課題がある。

そのため本研究では、このベテランの暗黙知を形式知化し、より汎用的に新人や中堅も広く利用できるプロセスを提案する。

3. ベテランの暗黙知を形式知化し新人/中堅が活用する方法

3.1 課題解決の考え方

ソフトウェアテストにおけるベテランの暗黙知を形式知にするためには、暗黙知を適用した結果として検出されたバグを分析することが有効と考えた。

また、形式知化されたテスト観点を一定の条件でマトリクスに整理することで、テスト対象全体を俯瞰しやすくなり、新人や中堅が使いやすいものになると考えた。

3.2 テストパターンマトリクスの作成

本研究では、ベテランの暗黙知を形式知化し、ある記述書式でまとめたテスト観点を「テストパターン」とする。このテストパターンを一定の条件でマトリクスに整理したものを「テストパターンマトリクス」と名付けた。なお、テストパターンマトリクスの行と列は以下の条件で作成する。

行：テスト対象に対し、どのようなことが起こったかの状況

例)仕様として定義されていなかった（または仕様が曖昧だった）

列：製品品質モデルの品質特性

例)品質特性^[4]

品質特性は ISO/IEC 25010 の製品品質モデルを選択した。選択した理由は、利用時品質のみにとどまらず幅広く確認を行えるようにするためである。

また、今回の研究では製品品質モデルは機能適合性、性能効率性、使用性、信頼

第 39 年度 ソフトウェアテストコース

性、セキュリティの 5 種に絞り設定する。

また、分類の際、煩雑になりすぎないように、副特性は用いないこととした。

なお、対象外とした保守性や移植性は仕様策定の段階で内容が確定している部分が多く、経験の違いにより観点到に差が生まれにくいこともあり、対象外とした。

これらの定義に基づき、テストパターンマトリックスは、以下の手順で作成する。

<テストパターンマトリックスの作成方法>

1. 定義した内容で、行と列を作成する。
2. バグ発見者やテスト有識者にヒアリングを行い、バグ情報を収集し以下の手順で分類する。
 - テストベース内外：収集したバグはテストベースに記載があったのか確認し、記載がないものを抽出する。
 - 品質特性：収集したバグは品質特性で分類するとどの部分のバグにあたるのか。
 - テスト観點：どのようなバグを見つけるためにどのようなテスト観點を利用したのか。
3. 2. で分類した情報を以下の記述書式に従って加工し、テストパターンとしてマトリックスへ配置する。この際、行で分類した状況に該当しないものは、「その他」で分類し、マトリックスの見直しのタイミングで行の分類の追加を検討する。

【記述書式】

目的：どのようなバグを見つけることを目的とするかを記載する

状況：バグを見つけるために行うテストのシチュエーションを記載する

3.3 アンケート調査

作成したテストパターンマトリックスのテスト観點が有用かという点について、研究員の社内メンバー 50 名（新人 5 名，中堅 8 名，ベテラン 37 名）に対しアンケート調査した。

その結果、図 1 に示すように新人と中堅の多くから、「なんらかの新たなテスト観點が思いつきそう」との回答を得た。一方で、テストパターンマトリックスの要素の表現が抽象的であり、経験の浅いテスト担当者にはどのような狙いでテストが行われているのか明確に理解できないとの回答も存在した。この結果を踏まえ、作成したテストパターンマトリックスのテスト観點は有用だが、新人や中堅に適用するためには、テストパターンマトリックスをブラッシュアップする必要があることがわかった。

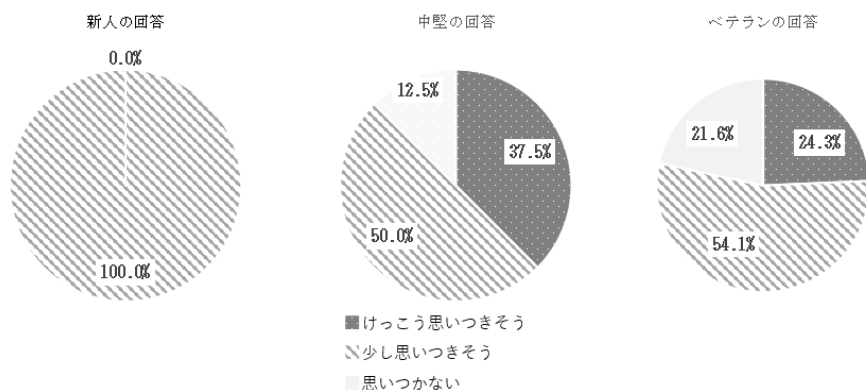


図 1 アンケート結果

ブラッシュアップを行い、実際に作成したテストパターンマトリックスを図 2 に示す。

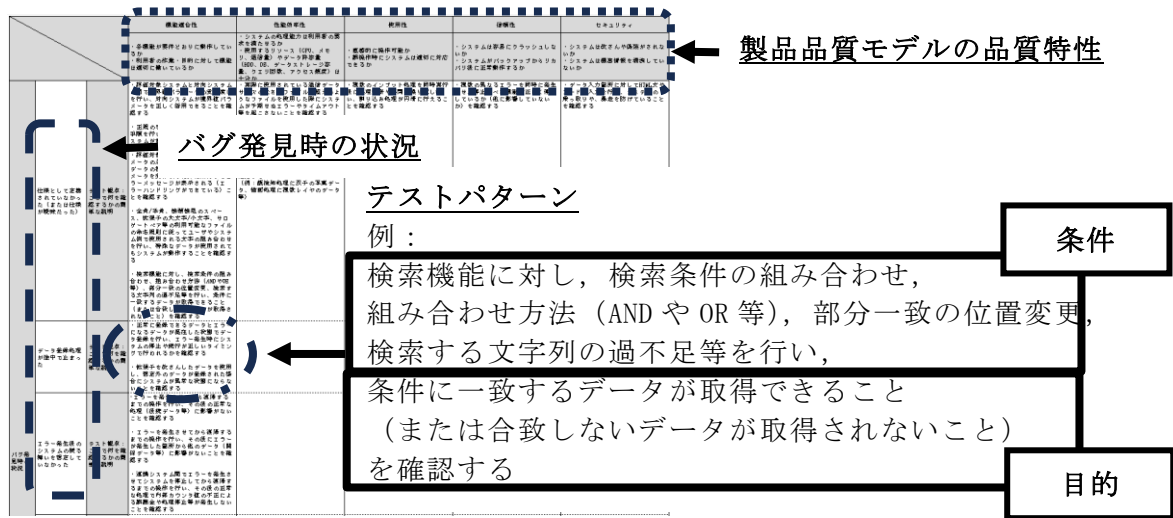


図 2 テストパターンマトリックス

3.4 テストパターンマトリックスの活用メソッド (VTV メソッド)

本節では、3.2 節で作成したテストパターンマトリックスを活用してテスト観点漏れを軽減する手法である VTV メソッド (Veteran Test Viewpoint method) を提案する。

VTV メソッドの「活用手法」, 「利用時の注意」は下記のとおりである。

[活用手法]

1. テスト設計者はテスト分析を実施するときに、テストベースの内外を意識して「テスト観点」を抽出する。
2. 抽出したテスト観点に対し抜け漏れが発生していないか、更に追加できるテスト観点が存在しないかを、テスト戦略に基づきテストパターンマトリックスの行と列の交点となる要素を参照する。
3. 2 で確認した要素内のテスト観点のうち漏れや追加が必要となるものをテスト仕様書へ追加する。
4. テスト活動終了後、テストで見つかったバグ情報を収集する。収集したバグの情報を分析し、テストベース外から出たバグについての扱いや、テストで使用しなかった要素の扱いを関係者間で合意を得て、テストパターンマトリックスをブラッシュアップする。

ブラッシュアップする際、新規に追加する項目はバグが製品やユーザに対する影響度の大きかったものをピックアップし、類似の観点がなければ追加する。削減する場合は、テスト対象および求められているテスト対象とは関係のないものから削除を行う。

[利用時の注意]

- テストパターンマトリックスの要素中の観点を変更なしでテスト観点として活用することも可能である。変更なしでの利用は問題ないが、ベテランのバグ検出時の考え方や検出方法を理解し習得することが重要である。
- テストパターンマトリックスは実際の事例から導き出しているため、同じテストを実施しても常に同じ交点に配置できるわけではない。
- テストパターンマトリックスはすべての項目を取り込むのを目的にするのではなくテスト戦略に従いテスト対象物、テスト内容に応じテストマネージャやテスト設計者が情報の取捨を行い活用することで効率的に活用できる。

4. 実験と評価

4.1 実験

VTV メソッド有効性評価として、アンケート調査にて経験の浅いテスト担当者がテストパターンマトリックスのみを用いて試験を実施することは難しいとの回答を得た。そのため、各要素の表現が汎用的な表現であったものをより具体的な表現に変更し、テストの狙いを明確にしたテストパターンマトリックスを用いて実験を行った。また、テストパターンの数も研究員内で再度精査し、重複やまとめられるところがあったため、46 個から 39 個へと整理したものをを用いて行った。

なお、3.2 節で論じたテストパターンマトリックスの作成方法は本変更を取り込んだものとなる。実験で確認したいことは、作成したテストパターンマトリックスを利用することで、ベテランが形式知化していたテスト観点を新人や中堅が導きだせるようになることである。

本実験は、新人 2 名中堅 4 名の計 6 名に対し、以下の手順および環境で実施した。

<実験手順>

1. 実験用の Web ページ，テストベースとして機能仕様書を被験者へ提示する。
2. 被験者は 1 で提示された Web ページに対し，機能仕様書外のテスト観点を抽出する。
3. 被験者に対しテストパターンマトリックスを提示する。
4. テストパターンマトリックスを参照して追加の機能仕様書外のテスト観点を抽出する。
(テストパターンマトリックスを使用する実験手順 3, 4 は時間測定を行う)

<実験環境>

- ・実験には機能仕様書に記載されていないようなテスト観点を抽出するため，画面間の相互作用のある Web ページを選定した。
- ・実験には詳細設計書でよく使われている種類を選定した。

(選定した Web ページの概要)

-顧客情報の入力およびその検索

(選定した機能仕様書の種類：35 ページ)

-処理概要

-入出力概要

-画面レイアウト：画面数 2

-入力項目条件

-画面出力条件

-イベント機能定義

4.2 実験結果

実験用の Web ページに対して，テストパターンマトリックスを参照して抽出した機能仕様書外のテスト観点の平均値を表 3 に示す。

表 3 実験結果 キャリアごとの 1 名あたりテスト観点抽出件数

	参照なし	参照あり
新人	6.0	25.5
中堅	14.8	28.4
全体(新人+中堅)	11.8	27.3

表 3 から読み取れることは以下である。

- ・新人はテストパターンマトリックスを参照することで、テストパターンマトリックスを参照する前に対して平均 4.25 倍のテスト観点数となった。
- ・中堅もテストパターンマトリックスを参照する前に対して平均 1.92 倍のテスト観点数となった。

図 3 は表 3 の全体の分布を箱ひげ図で表したものである。図 3 から読み取れることは以下である。

- ・テスト観点抽出数が 0 の人がいなくなった。
- ・中央値が 7.5 から 30.0 に変わったことから、テスト観点抽出数が全体で向上した。

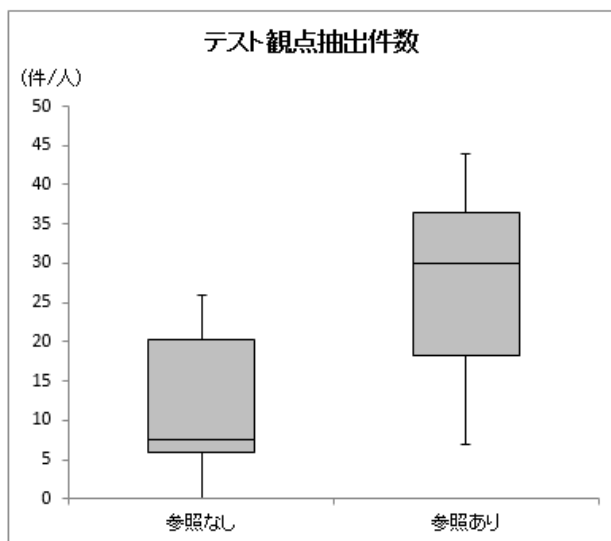


図 3 全体の分布

なお、今回の実験における本メソッドに費やした作業時間は以下の通りであった。

表 4 実験結果 本メソッドに費やした作業時間

作業内容	作業時間
テストパターンマトリックス作成	20 時間
テスト担当者への使用方法説明	15 分
テストパターンマトリックスを使用したテスト観点抽出	1 時間

5. 考察

表 3 の実験結果において、新人はテストパターンマトリックスを参照しない場合、テストベース外の観点抽出は 6.0 件で、14.8 件抽出した中堅と比較して約 40%程度であったが、テストパターンマトリックスを参照した場合には、新人は 25.5 件、中堅は 28.4 件と約 90%程度の差にまで縮まることがわかった。

要因として考えられることは、ベテランのテスト観点やそれに伴うバグのパターンを十分に目にすることが少なかったためである。一方、中堅は参照しない場合でも新人の約 2 倍の観点抽出ができていたため、参照した場合は参照しない場合と比較して約 2 倍増と新人の増加率よりも低い結果となったが、中堅も本テストパターンマトリックスを活用し、さらなるテスト観点を追加できることがわかった。

また、本テストパターンマトリックスを使用することで、観点やテスト知識/技術の習得にもつながり、新たなテスト観点の追加も可能になることが分かった。

以上のことから、今回提案した VTV メソッドは新人や中堅の新たなテスト観点の創出に効果的であることが確認できた。

さらに本メソッドの実用性についての考察として、導入工数について述べる。実験での時間測定の結果から、テストパターンマトリックスの作成には時間は要するが、テストパターンマトリックスを使用した観点抽出は 1 名当たり 1 時間 15 分で済むことがわかった。作成したテストパターンマトリックスは複数名のテスト担当者が同じものを利用できることから、本手法は多数のテスト担当者が行う実際のテストでも適用可能と考える。

6. まとめ

本研究の課題は、ベテランの暗黙知となっている情報を形式知化し、新人や中堅も利用できることである。それに対しテストパターンマトリックスを用いた、VTV メソッドを適用する実験を行い効果の確認をした。

また、実験結果から、VTV メソッドを利用することで、ベテランの暗黙知を形式知化し、新たなテスト観点の抽出を新人や中堅も実施できることが分かった。しかしながら、今回の実験や検証結果は少ない人数や決められた特定の Web ページを用いての実験であったため、VTV メソッドがテストの規模が大きくなったときに有効に働くかについては、母数の増加やテスト対象を変更して引き続き検証する必要がある。

さらに、今回はテスト観点抽出に絞って検証を行ったため、VTV メソッドを使用した場合のソフトウェア品質への影響についての検証が今後の課題である。

なお、今回提案したテストパターンマトリックスは開発における汎用的な利用を前提として、テストパターンマトリックスに用いたテスト観点についても汎用的なものに限定して検証を行った。

今回、本メソッドの有効性を検証できたため、今後、専門性の高い開発についても本メソッドを利用することで、効果が得られることを検証していく。

参考文献

- [1] 中辻はるひ, 松井紗彩, 土山真由美, “業務パターン整理の技能継承,” 開発工学論文誌, Vol.42, No. 1, pp.15-18, 2022.
- [2] 川上祐司, 大島祥吾, 榎本和也, 岩尾隆弘, “ソフトウェアテストにおけるバグ推測によるテスト設計手法の提案,” ソフトウェア品質管理研究会, 第 5 分科会 B, pp. 1-8, 2015.
- [3] 飯沼真一, “探索的テストを効果的に行うための留意点のパターン化,” ソフトウェア品質シンポジウム 2022, A2-1, pp.1-7, 2022.
- [4] 飯泉紀子, 鷲崎弘宜, 菅田直美監修, SQuBOK 策定部会編, ソフトウェア品質知識体系ガイド (SQuBOK Guide V3 1. 1. 2 S-KA ソフトウェア品質モデル), pp. 1-381, 2020.
- [5] 小島嘉津江, et al. “ソフトウェア品質技術が品質特性に与える効果の見える化とその検証.” SEC journal/情報処理推進機構技術本部ソフトウェア・エンジニアリング・センター 編 14.1 (2018): 50-57.
- [6] 吉岡克浩, 水野昇幸, and 西康晴. “見通しのよいテストの段階的詳細化の手法.” (2013).