生成 AI を活用した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

Proposal for Reducing Test Perspective Omissions Using a Test Pattern Matrix Leveraging Generative AI

○田口真義 1）飯沼 真一 2）
○Masayoshi Taguchi　Shinichi Iinuma

It is important to extract the implicit information as test viewpoints, which are not described in the test base, such as functional specifications, to realize the software testing with higher quality and shorter period than previous it.
Any tester can extract the test viewpoint from test base.
However, it is difficult for their to extract the test viewpoint from the implicit information except test base, because of veteran testers did not formalize their knowledge and experience of extracting test viewpoint.
In this paper, we propose turning the veterans's knowledge of extracting test viewpoint to explicit information by using a report of detected bugs and generative AI.
This method is named "Veteran's Test Viewpoint extracted by using generative AI".
We confirmed that any tester can extract new test viewpoint from reports of detected bugs and using our method by experiments.

1. はじめに

23 年度 SQiP 研究会ソフトウェアテスト分科会のメンバーの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ためには，テスト設計に高い能力を持つベテランのテストエンジニアが必要であるが，そのようなテストエンジニアはテストマネージャとなりテスト設計業務から離れていく傾向がある．そこで，ベテランの持つテスト観点を新人や中堅であっても活用でき，テスト観点の抽出漏れを減らす活動が重要であると考えた．

Web アプリや組み込み開発などのプロジェクトで収集した欠陥 154 件に対し，テスト経験年数ごとの 1 名あたりの欠陥検出数およびテストベースに記載がなかったもの（以降　テストベース外）の欠陥の検出割合を調査した．その結果が表 1 である．

表 1 テスト経験と検出した欠陥の分析

テスト経験（括弧内は経験年数）	1 名あたりの欠陥検出数（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

1)リコーIT ソリューションズ株式会社　Ricoh IT Solutions Co., Ltd.
神奈川県横浜市都筑区新栄町 16-1 Tel: 050-3817-3900
16-1 Shineichou, Yokohama-City Kanagawa Japan

2)株式会社 AGEST　AGEST, Inc.
東京都文京区後楽 1 丁目 7-27 後楽鹿島ビル 4F　Tel:03-5333-1246
Kouraku-Kajima Building 4F, 1-7-27 Kouraku, Bunkyo-ku, Tokyo, 112-0004, Japan

【キーワード：】ソフトウェアテスト，テスト観点，生成 AI，暗黙知の形式知化

この結果は，欠陥検出数に経験年数による差はないが，テストベース外の欠陥の検出割合はテスト経験を積むほど高くなっていることを示している．

次に，テスト関係者 50 名の現在のテスト時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

表 1 と表 2 の結果をまとめると，次のとおりである．

a. 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外についてもテストを実施し，そこから多くの欠陥を検出する．
b. 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．

テストベース外からのテスト観点の抽出は，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，テスト経験から学んだテストのやり方やテストに関連するドメイン知識がテスト観点の抽出に必要であるが，ベテランは暗黙知として使用しており，形式知化されていないためテストベース外のテスト観点を意識して抽出することができないためである．

そこで本研究では，この課題解決のために，ベテランの暗黙知となっている情報を人の手で抽出し，生成 AI を用いて分類することで形式知化する．そして，形式知化した情報を新人や中堅が効果的かつ効率的にテスト観点の抽出ができるように「バルタンメソッド（Veteran Test Viewpoint using generative AI - method)」としてメソッド化することを提案する．

2 章では先行研究の調査結果と，そこから導き出した解決すべき課題を述べる．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，バルタンメソッド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

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，既存機能と過去の欠陥情報を関連付け，その情報から欠陥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，既存機能と過去の欠陥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．この手法により，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ックスで表現して作成する小島ら[5]や吉岡ら[6]の研究があげられる．

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が活用できる手法の確立を目指す．

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化し新人や中堅がその情報を活用するには，テスト活動において暗黙知を適用した結果として検出された欠陥を分析することが有効と考えた．最初に暗黙知を活用した欠陥の情報を人の手で収集する．そして，収集した情報を，生成 AI を活用して一定の条件で効率的に分類する．これをテストパターンマトリックスとして表現することで形式知化する．このように，収集した情報をマトリックスとして表現することで，新人や中堅は形式知化された情報を俯瞰しやすくなる．

さらに，このマトリックスを用いたテスト活動を新人，中堅が効果的に活用できるようにバルタンメソッドとしてメソッド化することを考えた．この考え方を図 1 に示す．

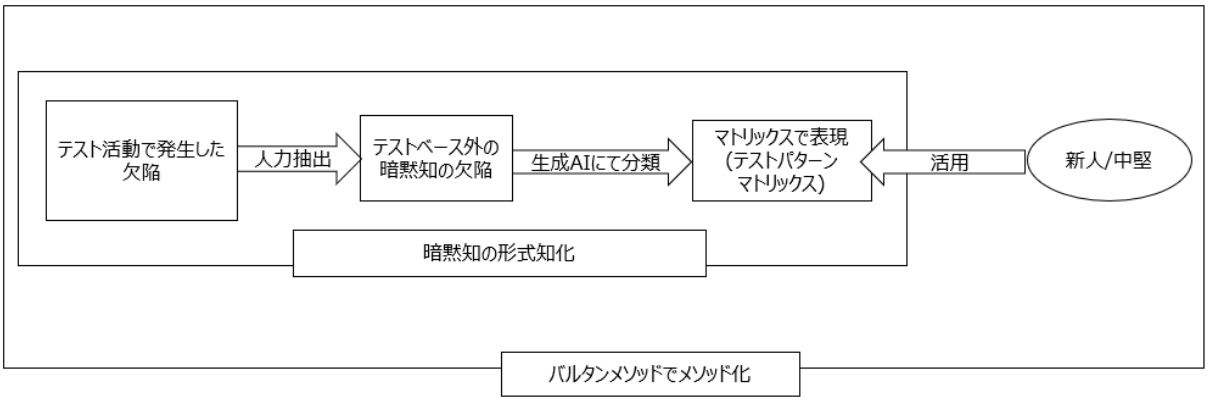

図 1 課題解決の考え方

3.2 バルタンメソッドの提案

本節では，ベテランの暗黙知を形式知化し，テスト経験の浅いテスト担当者のためにメソッド化して活用するバルタンメソッドについて説明する．

なお，手順 1～5 はテスト関係者が行い，手順 6～9 はテスト担当者が行うものとする．

[バルタンメソッドの手順]

1. テスト活動により検出された，過去の欠陥を収集し，「テストベースの内外どちらの条件で発生したか」と「欠陥検出時に利用したテスト観点」の情報を付与する．
2. テストベース外の条件により発生した欠陥を「テスト目的の観点」と「テスト条件の観点」の 2 つの要素からなる「テストパターン」の書式で記述し直す．
3. 2 で作成したテストパターンを生成 AI に読み込ませることで「システム開発の運用側面」のカテゴリを生成する．
4. 「システム開発の運用側面」を行，「品質特性」を列とした「テストパターンマトリックス」を作成する．

5. 2 で作成した「テストパターン」をテストパターンマトリックスの要素として埋め込み，テストパターンマトリックス（ベテランのテスト観点が詰まった一覧表）を完成する．
6. はじめにテストパターンマトリックスを利用せずにテスト観点を抽出する．
   なお，テストパターンマトリックスをはじめから利用しないのは，テスト担当者がテスト観点抽出を行う際に自身のテスト観点抽出能力を把握すること，ならびにテストパターンマトリックスに頼り切ることでテスト観点の発想力を狭めてしまわないためである．
7. テストパターンマトリックスを参照し，ベテランの暗黙知となっていたテスト観点の情報を利用してテスト観点の抜け漏れを補完する．
8. テストを実施し，品質の保証および欠陥の検出を行う．
9. 検出された欠陥をもとにステークホルダー間で協議し，テストパターンマトリックスをさらにブラッシュアップする．

バルタンメソッドの流れを DFD として図 2 に示す．

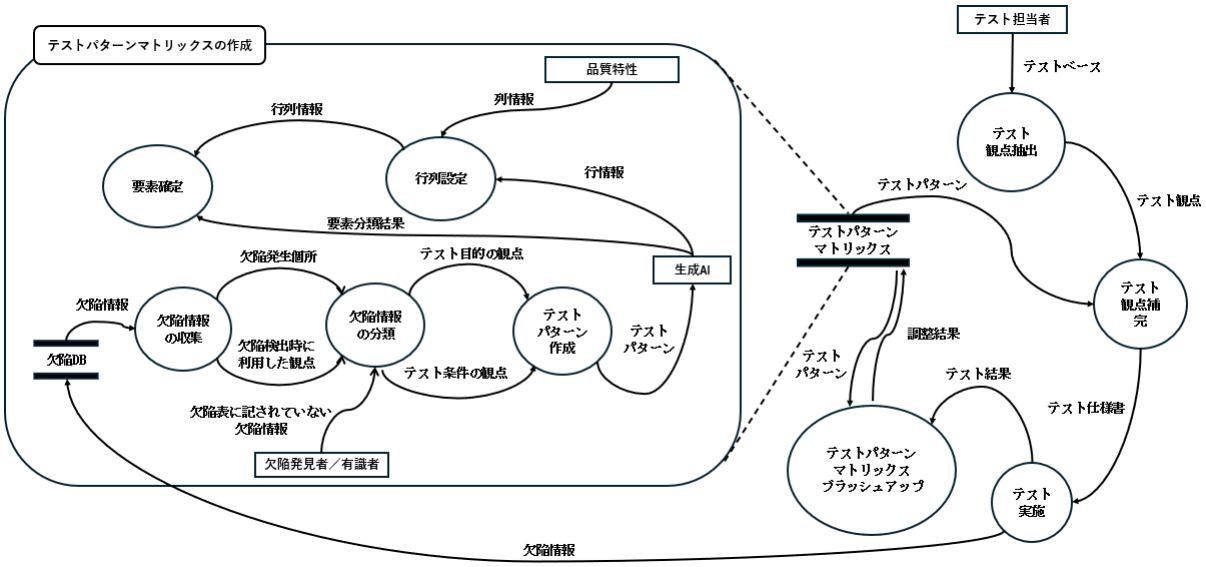

図 2　バルタンメソッド

次に，バルタンメソッドの手順 1～5 で表現したテストパターンマトリックスの生成方法について，詳細を記述する．なお，テストパターンをマトリックスで表現する理由は，テストベース外のテスト観点全体を俯瞰できるため，手順 6 で抽出したテスト観点との重複を避けながらテストで気にすべきポイントが探しやすいことと，テスト戦略に応じて必要なテスト観点にたどり着くことが容易になるためである．

［テストパターンマトリックスの生成］

1. テスト活動により発生した全ての欠陥に対し，欠陥発見者やテスト有識者からヒアリングを行い，欠陥発生個所および使用したテスト観点の情報を付与する．

   ＜欠陥に付与する情報＞
   欠陥発生個所：テストベース内とテストベース外のどちらで欠陥が発生しているのか．
   テスト観点:どのような欠陥を見つけるために，どのようなテスト観点を利用したのか．

2. 1 で付与した欠陥発生個所のうち，ベテランの暗黙知として表現されるテストベース外の欠陥に対し，「テスト目的の観点」と「テスト条件の観点」の 2 つの要素を追加し，「テストパターン」として表現し直す．
   なお，テストパターンはテスト経験の浅い担当者が，欠陥検出の考え方や着眼点を理解しや

すくするために，汎用的かつ分かりやすい表現で記載する．

3. 作成した全てのテストパターンを生成 AI に読み込ませる．そして，行として活用するカテゴリを確定させる．
   なお，列は品質特性を利用するため，品質特性とは別の視点で生成 AI に分類させてカテゴリを定義する必要がある．
   今回は以下のプロンプトにおいて「システム開発の運用側面」のカテゴリを抽出した．

   【プロンプト入力値】
   ＜全テストパターン＞を入力．ここに N 個のテスト観点が存在します．
   これらを ISO/IEC 25010 の製品品質モデルとは別の方法で，M 種くらいに分類し，
   どのような条件で分類したかを明確に表現してください．

   なお，利用時にテストパターンの内容が推測しやすいように分類は 7 種±2 くらいで絞るとよい．
4. 3. で抽出した「システム開発の運用側面」を行に設定し，列には品質特性[4]の ISO/IEC 25010 の製品品質モデルの中から機能適合性，性能効率性，使用性，信頼性，セキュリティを設定し，マトリックスを準備する．なお，列に製品品質モデルを 5 種のみ使用した理由は，使用しなかった 3 種（保守性，移植性，互換性）は仕様策定の段階で内容が確定している部分が多いことに加え，経験の違いによりテスト観点の差が生まれにくいと考えたためである．また，情報が煩雑になることを避けるため，今回品質副特性は活用していない．
5. 2. の各テストパターンが，行（システム開発の運用側面）と列（品質特性）のどのカテゴリにそれぞれ属しているかについて，生成 AI を利用して確認し，「テストパターンマトリックス」を完成させる．
   なお，このとき生成 AI は複数の要素を回答する場合があるため，回答を導きだす際，最も重要な構成要素は何であるかを生成 AI から聞き出すこと．その際，誤った分類をしていないかを確かめるためにテストパターンを生成 AI に再確認し，適切な回答を人力で選ぶ必要がある．
   実際に使用したプロンプト情報を以下に記載する．

   【プロンプト入力値】
   プロンプト：ここに＜列）5 種の品質特性＞＜行）M 種のカテゴリ分類＞が存在します．
   ＜個々のテストパターン＞は例で挙げている 5 種の品質特性および M 種のカテゴリ分類のどれに最も当てはまりますか？

実際に作成したテストパターンおよびマトリックスを図 3 に示す．

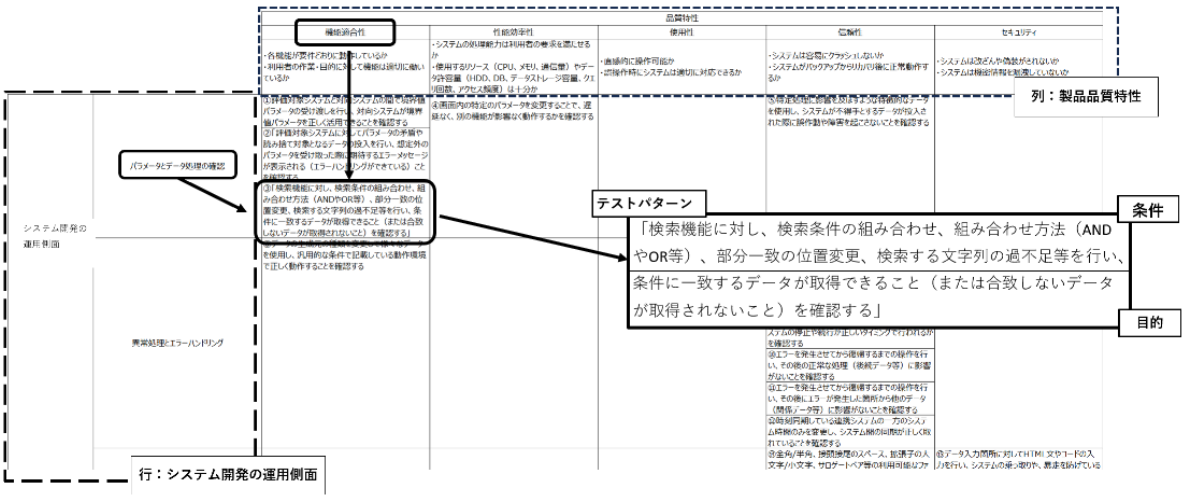

図 3　テストパターンマトリックス

4 実験と評価

4.1 実験

バルタンメソッドの有効性評価として，新人 3 名・中堅 8 名・ベテラン 13 名の計 24 名に対して実験を行い，ベテランが暗黙知化していたテスト観点を新人や中堅といったテスト経験の浅い担当者であっても抽出できるかどうかを確認する．また，抽出したテスト観点で実際の欠陥を検出できるかどうかを確認する．

なお，今回の実験で利用するテストパターンマトリックスに収録されているテストパターンは 40 個とし，実験手順および実験環境は以下のとおりとする．

＜実験手順＞

1. テストパターンマトリックスの作成時間を記録する．
2. 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．
3. 被験者は 1 で提示された Web ページに対し，テストベース外のテスト観点を抽出する．
4. 被験者に対しテストパターンマトリックスを提示し，その情報を活用し追加のテストベース外のテスト観点を抽出する．
5. 抽出したテスト観点で Web ページに仕込んだ欠陥を検出できるかどうかを確認する．

＜実験環境＞

・実験は画面間の相互作用のある SNS の Web ページを選定した．
・実験は被験者 24 名すべてが，過去バージョンのテストも行っていない新規テスト対象で実施した．
・実験用の Web ページはテストベース外の欠陥 12 個を準備した．
・機能仕様書は 16 頁で構成され，以下に示す内容が記述されている．
（機能仕様書記述内容）
画面遷移図/データフロー図/画面レイアウト/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，テスト経験ごとにテストパターンマトリックスの使用有無で抽出可能となる 1 名当たりのテスト観点の数およびベテランとの比率を表 3 に示す．

表 3 実験結果 テスト経験年数ごとの 1 名あたりテスト観点抽出件数とベテランとの比率

	参照無し		参照あり	
テスト経験	件数	ベテランとの比率（%）	件数	ベテランとの比率（%）
新人	18.33	80.78	35.33	113.67
中堅	17.75	78.23	30.13	96.94
ベテラン	22.69	-	31.08	-
全体（新人＋中堅＋ベテラン）	20.50	-	31.29	-

表 3 より，テストパターンマトリックスを参照した場合，テストの経験年数を問わず，テスト観点の抽出件数が増えた．また，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検出したテスト経験ごとの欠陥の平均値を表 4 に示す．

表 4 テスト経験と実験において検出した欠陥数の平均

テスト経験	参照なし(件)	参照あり(件)
新人	2.67	6.00
中堅	3.38	4.38
ベテラン	2.62	3.46

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および 23 年度 SQiP 研究会 研究コース 3 においてテストパターンマトリックスをすべて人力で行った際の結果を表 5 に示す．

表 5　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（h）	
		23年度SQiP研究会 実験結果 （作成はすべて人力）	24年度SQiPシンポジウム 実験結果 （作成はテストパターン分類は生成AI）
作成	テストパターン抽出	10	10
	テストパターン分類（マトリックス表へのプロット）	10	2
運用 （1名当たり）	テスト担当者への使用方法説明	0.25	0.25
	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	1

表 5 よりテストパターンの分類は生成 AI を活用することで 8 時間短縮した．

5　考察

表 3 より，新人や中堅がテストパターンマトリックスを参照した場合は，参照しない場合と比較してベテランとのテストベース外のテスト観点抽出件数の差が約 3%まで小さくなり，ベテランに近いテスト観点数を抽出できることがわかった．この要因は，テスト経験の浅いテスト担当者はベテランの持つテスト観点やそれに関連する欠陥のパターンを充分に目にすることがまだ少ないため，テストパターンマトリックスからそれらの情報を多く着目できたことだと考えられる．このことから，テストパターンマトリックスは先行研究で課題となっていた探索的テスト以外のシチュエーションにおいても活用できることがわかった．また，全体でテスト観点抽出件数が増加し，ベテランもテストパターンマトリックスを活用することで，まだ知見のない暗黙知の情報を吸収し，追加のテスト観点として活用できることが分かった．さらに，本実験はすべての被験者が新しいテスト対象に対してテスト観点抽出を行っており，先行研究で課題となっていた類似のテスト対象でなくてもテストパターンマトリックスを活用できることがわかった．これらのことから，テストパターンマトリックスを使用することは，経験年数を問わず，テスト観点やテスト知識/技術の習得へとつながり，新たなテスト観点の追加が可能となることが確認できた．

表 4 より，テスト経験年数を問わず，テストパターンマトリックスを使用した場合，欠陥検出数が増加することが分かった．欠陥検出数は新人や中堅がベテランよりも多くの欠陥を検出した．この要因はテストに仕込んだ 12 の欠陥が比較的単純な操作で見つかるものが多かったため，単純な操作をベテランはテストベース内のテストで見つけられると判断し，テストベース外のテスト観点としてあえて抽出しなかったことが考えられる．

表 5 より，テストパターンマトリックスの作成および運用工数について述べる．
今回生成 AI を活用し，マトリックス表へのプロットを行ったが，すべて人力で行った場合と比較し，約 80%の工数が削減できることが分かった．また，テストパターンマトリックス作成時，生成 AI を用いて分類を行ったが，生成 AI に聞き直すことはほとんど発生しなかった．この要因は，プロンプトの条件として最も重視する項目を行と列で共に指定したことが考えられる．これらのことから，テストパターンマトリックスは生成 AI を用いることで分類を短時間に的確に行うことが出来ることが証明できた．また，実運用を行う場合，テストパターンマトリックスの作成に時間がかかるが，作成後は複数のテスト担当者が同じものを使用し，従来よりも多くのテストベース外のテスト観点の抽出が行えるようになるため，実用的であることが考えられる．

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新人や中堅が利用できないことで，テスト観点の抽出漏れに繋がっていることである．それに対し，テストパターンマトリックスを用いたバルタンメソッドを適用する実験を行い，効果を確認した．

実験結果より，ベテランの暗黙知となっている情報をもとにしたバルタンメソッドを適用することで，新人や中堅であってもベテランに近い数のテスト観点を抽出可能なことがわかった．

しかしながら，今回の実験や検証結果は，24 名のテスト担当者が決められた特定の Web ページを用いての実験であったため，テスト規模やテスト対象が変更となった際や専門性の高いアプリ開発においてもバルタンメソッドが有効に機能するかどうかは引き続き検証する必要がある．同様に，ソフトウェア品質への影響についても，テスト対象やテスト目的を変更して引き続き検証する必要がある．

このほかの課題として，生成 AI は毎回同じ回答を返さない特徴があるため，その部分の品質をどのように担保するか，追加で検証する必要がある．

7 謝辞

本研究は 2023 年度 SQiP 研究会 研究コース 3 において行った研究を発展させたものとなります．本研究を行うにあたり，多くの方々にお世話になりました．主査の喜多義弘氏，副主査の秋山浩一氏，アドバイザーの西田尚弘氏には本研究を遂行する上で様々なご指導や助言をいただきました．深く感謝申し上げます．また，会田健太郎さん，柳井聡さん，北島優介さん，宮田奈津子さんには SQiP 研究会で研究員として共同研究していただき，本稿の発表に関しても多大なご協力をいただきました．感謝申し上げます．

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，誉田直美監修，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).