

## 生成 AI を活用した探索的テストの学習基盤の構築

## Building a learning platform for exploratory testing using generative AI

○飯沼 真一 1)

○Shinichi Iinuma

In order to effectively and efficiently verify the usability of products in application development, the demand for exploratory testing is increasing, and there is an urgent need to increase the number of engineers who can perform exploratory testing. However, it is difficult to train engineers who can perform exploratory testing because exploratory testing is carried out by veterans with abundant domain knowledge and experience, and the method of exploratory testing is tacit knowledge of veterans. Therefore, we extracted the exploratory testing method of veterans, defined the "exploratory testing process", and used it as a learning platform for exploratory testing. In addition, we lowered the hurdle to conducting exploratory testing by showing how to create a test charter using generative AI. In addition, the procedure for creating an exploratory test matrix as a guidepost for effective exploration is presented. If you follow the defined exploratory testing process, you will be able to understand how to do exploratory testing, and you can expect to efficiently train engineers who perform exploratory testing.

### 1. はじめに

近年、アジャイル開発の浸透やシステムの大規模化／複雑化／短納期化が進み、探索的テストにより、短期間で欠陥を検出することの重要性が増している。しかしながら、探索的テストは豊富なドメイン知識と経験を持つベテランが担っており、ベテランの数が非常に少ないのが現状である。なぜなら、これまでのテストは記述式テスト（スクリプトテスト）が主に採用されており、探索的テストのニーズが少なかった。そのため、探索的テストはできる人だけがやるものだった。それだけでなく、探索的テストのやり方がベテランの暗黙知となっていることに加えて、ベテランは比較的高齢であることから、ベテランのもつ暗黙知が後進に時間をかけて十分に伝承される機会もなく、異動や退職などによって喪失している側面がある。この側面は、探索的テストを担う技術者の養成が難しいことを意味し、探索的テストを担える技術者を増員することの妨げとなっている。

そこで、探索的テストを担える技術者を効率的に増員するために、探索的テストの活動の中で、ベテランがどのように探索を開始し進めているかを「探索的テストプロセス」として定義し、学習基盤とすることを考えた。そして、技術者が学習を継続していくために「ゴモラメソッド (Generative AI support and Master exploratory testing and Revelation from Tacit knowledge - method)」としてメソッド化することを提案する。具体的には、探索的テストのやり方を経験から抽出／整理して、探索的テストの経験の少ない技術者のために形式知にする。さらに、何をテストすべきかを判断するためには、豊富なドメイン知識が必要になる。そこで、何をテストすべきかを判断して探索する方向性を見定めるために、生成 AI からテストチャーターを生成する方法を示すことで、何をテストすべきかの判断を容易にする。2章では、探索的テスト実施上の問題点と課題を述べる。3章では、学習基盤となるゴモラメソッドで定義した探索的テストプロセスと探索的テストマトリックスについて説明後、ゴモラ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。4章では、実験結果について考察する。5章では、成果と今後の進め方を述べる。

---

1)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

## 2. 探索的テスト実施上の問題点と課題

### 2.1. 探索的テストの問題点とその原因

探索的テストについては、筆者が 2022 年に投稿した論文<sup>[1]</sup>で説明したとおりであるが、探索的テストの特徴は、テストケースを書かないことでテスト実行に早く着手できることである。そして、直前のテスト結果から得られた気づきを次のテストに活かすことで、テストを深掘りしていく。しかしながら、テストケースを書かないため、探索的テストを実施する技術者には、アプリの仕様や実際に利用されるシチュエーションについて十分に理解している必要があることが実施上の問題点である。

### 2.2. 探索的テスト実施上の課題

2.1 で述べた論文<sup>[1]</sup>では、テスト実施においてはベテランの留意点（経験と勘）に依存するところが大きいことを問題点として挙げた。そこで、非ベテランがベテランと同等に探索的テストをできるようにするために、ベテランの留意点をパターン化して非ベテランへ適用することを考えた。しかしながら、非ベテランへパターンを適用しようとしても、探索的テストのやり方がよくわからないため、適切な実施が行えない。従って、探索的テストのやり方を修得する必要があるのではないかと考えた。

そこで今回は、テストのやり方に焦点を当てて検討を行った。検討の結果、探索的テストは特定のプロセスを決めず、経験に基づいて探索することが特徴であるが、経験が少ない人がベテラン並みにできるようになるまでには、試行錯誤して成功体験を積む必要があるため、時間とコストがかかってしまう。せめて、テストすべき領域を特定して深掘り始める部分をベテラン程度に引き上げることができれば、これまでよりも早くベテランと同等の探索的テストが可能になると考えた。

そこで、本論文の課題を「探索的テストのやり方の大まかなプロセスを定義し、学習基盤を構築すること」とした。これにより、効果的かつ効率的に探索的テストの経験を積むことが可能となる。

### 2.3. 先行調査結果

探索的テストを実施する技術者にベテランのやり方を移転可能とするためには、それらを何らかの方法で学習しなければならない。そこで、探索的テストの学習方法の観点で先行事例を調査した。

Cem Kaner ら<sup>[2]</sup>は、さまざまな探索的テストの体験からの教訓が示されており、探索的テストについての理解をさらに深めていくには有効である。しかし、探索的テストの経験の少ない技術者には、教訓を活用するには至らないという問題があった。

飯泉ら<sup>[3]</sup>は、探索的テストの進め方や留意点が示されているが、一般的な解説にとどまっている。

高橋<sup>[4]</sup>は、探索的テストの留意点が示されており、探索的テストの知見と考え方をさらに深めていくには有効である。しかし、具体的なやり方が分からないといった点で解決には不十分である。

先行事例を調査した結果、探索的テストの経験の少ない技術者にベテランのやり方を移転可能とし、繰り返し伝授できる状態にする方法は見つからなかった。それだけでなく、探索的テストの実施者は、テスト関連の学習を行いながら、継続的にテストスキルを洗練させる必要があることがわかった。

## 3. 探索的テストの学習基盤の構築

探索的テストを担える技術者を効果的かつ効率的に養成するためには、ベテランの暗黙知となっている探索的テストのやり方を真似るのが効率的である。そこで、筆者の探索的テストの実施経験からエッセンスを抽出して整理を行い、探索的テストプロセスとして形式知化し、経験の少ない技術者のドメイン知識を補う目的で生成 AI からテストチャーターを生成する学習基盤を構築することを考えた。そして、探索的テストの成功を達成するためのリスク回避として、探索的テストを実施する前に考案した探索的テストプロセスを確認し、テスト活動の中で活用してもらう。さらに、探索的テストにおいては、「テスト済の機能」や「セッション回数（テスト回数）と検出したインシデント」が重要な道しるべとなる。そこで、探索的テストマトリックスの作成手順を形式知化し、併用することとする。

### 3.1. ゴモラメソッドの提案

本節では、技術者が形式化された探索的テストプロセスを使い、探索的テストの学習を継続していくゴモラメソッドについて説明する。

### 3.1.1. 探索的テストプロセス

探索的テストプロセスとは国際規格「ISO/IEC/IEEE 24774：プロセス記述のための仕様」を参考にして、ベテランのやり方がある記述書式でまとめたもので、以下の記述書式を定義した。

#### 1) 探索的テストプロセスの記述方法

<記述書式（項目：記載方法）>

##### 第一段落

- ・プロセス名称：プロセスにわかりやすい名前を付けたもの
- ・プロセス目的：プロセス目的を具体的に示したもの
- ・プロセス活動：プロセス活動の概要をわかりやすく示したもの

##### 第二段落

- ・プロセス活動（Activity）：プロセス活動を具体的に示したもの
- ・プロセス活動の説明：プロセス活動の主要なタスクを連想できるもの
- ・プロセス活動への入力：プロセス活動ごとの作業成果物に取り込むデータ
- ・タスク：プロセス活動に必要なタスクを具体的に示したもの
- ・プロセス活動の出力：プロセス活動ごとの作業成果物
- ・注記：タスクやプロセス活動のやり方を注意事項として定義したもの

このような記述書式とした理由は、ベテランのやり方を、予め探索的テストを実施する技術者が十分に認識／理解しておかなければ、探索的テストの習得は難しいと考えたためである。

探索的テストプロセスは、探索的テストを実施する技術者間で、探索的テストの実施状況の共有を可能にするものである。そして、探索的テストの「道しるべ」とし、そこから得られた「製品」「市場」「リスク」「欠陥」「テストで失敗した方法」について学んだ重要なナレッジを蓄積／共有するという基本プロセスの位置付けである。

#### 2) 探索的テストプロセスの定義

実際に考案した探索的テストプロセスの第一段落および第二段落を以下の表 1 と表 2 にそれぞれ示す。また、本プロセスはシステムテストにもとからテスト計画があり、それを基に記述式テストを行っている前提である。

表 1 第一段落

名称	探索的テストプロセス (ET : exploratoy Testing)
目的	1. 探索的テストを担える技術者の養成をすること。 2. 利用時品質の検証と評価を行うこと。
活動	ET1：テスト計画の策定 ET2：テスト計画の検証と承認 ET3：テストの実施 ET4：テスト結果の検証と報告

表 2 第二段落

活動	ET1：テスト計画の策定
説明	探索的テストの実施計画を作成する。
入力	・ソフトウェアシステムテスト計画書 ・テスト対象のソフトウェア要求仕様書 ・システムテストのテスト仕様書（記述式テスト用）
タスク	a. 探索的テスト目的の設定 目的を決める上では、利用者が具体的に何に困っていて、どうなれば解決するかを考え、誰がどのように使うかを想定するために、利用者の目的やニーズを把握して利用者像を特定する必要がある。そこで、ユーザー要求分析によって定義された要求事項に対し、

	<p>システムがどのように具現化されているかの確認を行うことが重要である。また、ユーザビリティについて、生成 AI を利用して確認し、探索的テストの目的を設定することで探索する方向性を決定する。 注 1</p> <p>b. 探索的テスト範囲の設定 テスト対象の探索的テスト範囲を決定する。 注 2 注 3</p> <p>c. 探索的テストマトリックスの作成 決定した探索的テスト目的と探索的テスト範囲をマトリックス状に並べ変えることで、探索エリアを決定する。なお、探索的テスト範囲（機能と特別確認項目）に対する探索的テスト目的をチャーターと呼んでいる。 注 4</p> <p>d. 探索的テスト計画の作成 探索的テスト計画を立てる。</p>
出力	<ul style="list-style-type: none"> <li>・探索的テストマトリックス</li> <li>・探索的テスト計画書</li> </ul>
注記	<p>1. ユーザビリティについては SQuaRE 規格の ISO/IEC25010 で定義されている「利用時の品質モデル」を参考にしている。また、生成 AI に確認した実際の AI プロンプト情報を以下に記載する。</p> <p>AI プロンプト 1：一般的な SNS の利用時品質におけるテスト目的を次の 5 つの品質特性ごとに詳しく教えてください。「有効性」、「効率性」、「満足性」、「リスク回避性」、「利用状況網羅性」</p> <p>なお、このとき生成 AI は同じ質問を行っても異なる回答をする場合があるため、AI プロンプト 1 の質問に続けて、AI プロンプト 2 で具体的な機能概要からテスト目的の詳細を確認してテスト目的の抜け漏れが発生しないようにする。</p> <p>AI プロンプト 2：&lt;機能概要&gt;（機能 1）（機能 2）（機能 3）・・・（機能 N）</p> <p>AI プロンプト 3：一般的な SNS で過去に発生したバグを教えてください。</p> <p>2. システムがサービス拡大により、複数存在して連携するような場合は、他システムに類似機能が存在するかもしれない。</p> <p>3. 探索的テスト範囲については、特別確認したいような項目があるかもしれない（以下 特別確認項目）。</p> <p>4. 探索的テストマトリックスを作るためには、高度なドメイン知識とテスト設計スキルが必要になる。そのため、実際には有識者や経験者とよく協議することで、探索的テストの目的を設定しやすくなる。</p>
活動	<b>ET2：テスト計画の検証と承認</b>
説明	探索的テストの実施計画を、PM/PL によりレビューして問題があれば解消する。また、探索的テストの実施計画に、問題がなければ PM/PL により承認する。
入力	<ul style="list-style-type: none"> <li>・探索的テスト計画書</li> <li>・テスト対象の設計書一式</li> </ul>
タスク	<p>a. 探索的テスト計画書のレビュー PM/PL により、探索的テスト計画書のレビューを実施する。 注 1</p> <p>b. レビュー議事録の作成 探索的テストの技術者が指摘事項を記録する。また、指摘事項の是正を行い、PM/PL による是正後の承認を得る。 注 2</p>
出力	<ul style="list-style-type: none"> <li>・レビュー議事録</li> </ul>
注記	<p>1. テスト目的の妥当性について問われることが多い。 &lt;レビュー実施前に確認すること&gt;</p> <ul style="list-style-type: none"> <li>・見積もりの具体的な内容</li> <li>・テスト目的の内容はプロジェクト状況を踏まえているか</li> </ul> <p>2. 同じ指摘を受けないように、レビュー議事録から確認すべきポイントを水平展開する。</p>

活動	ET3：テストの実施
説明	探索的テストの実施前に、関係者全員で作戦会議「ブリーフィング」を行う。また、探索的テストを実施して、その結果を記録する。
入力	<ul style="list-style-type: none"> <li>探索的テスト計画書</li> <li>テスト対象の設計書一式</li> </ul>
タスク	<p>a. ブリーフィング 探索的テストの関係者との間で、ブリーフィングを行う。 注1</p> <p>b. 探索的テストの実施 設定したセッション単位で、探索的テストを実施する。 注2</p> <p>c. 検出したインシデントの処理 探索的テスト実施中に検出した不明な動作や事象を記録し、必要に応じて欠陥票の起票、および修正された欠陥の是正確認をする。</p> <p>d. 探索的テストの実施結果の記録 セッション単位で、実施した探索的テストの実施結果を記録する。</p>
出力	<ul style="list-style-type: none"> <li>探索的テスト結果</li> <li>インシデント一覧</li> <li>探索的シナリオ一覧</li> </ul>
注記	<p>1. ブリーフィングは、探索的テストの価値を最大化するために、テスト後半にも実施して、アイデアが出尽くしているかどうかを確認し、探索的テストの実施計画の見直しも行うとよい。</p> <p>2. 偶発的に思い付いたようなことは、すぐにメモを取るとよい。また、探索するやり方の例を以下に列挙した。</p> <p>&lt;探索のやり方&gt;</p> <ul style="list-style-type: none"> <li>製品概要から深掘りしていくように進める</li> <li>市場トラブルを基にテストで拾えそうなことやテスト設計書に記載していないようなことを分析してそこを探す</li> <li>類似製品の過去欠陥を基に同じような問題が潜んでいないかを分析してそこを探す</li> <li>システムに起きたら嫌なことや類似製品の過去にテストで失敗したようなやり方がなかったかなどを考えてそこを探す</li> </ul>
活動	ET4：テスト結果の検証と報告
説明	探索的テスト結果をレビューして、問題があれば解消する。また、探索的テスト結果を発注元に報告する。
入力	<ul style="list-style-type: none"> <li>探索的テスト結果</li> <li>探索的テスト計画書</li> <li>インシデント一覧</li> </ul>
タスク	<p>a. 実施結果のレビュー 探索的テストを実施した技術者以外のベテランが、実施結果のレビューを行う。</p> <p>b. レビュー後のフィードバック 実施結果の不明点や不足の有無について、探索的テストを実施した技術者へ即日のうちにフィードバックする。</p> <p>c. 探索的テスト結果報告書の作成 目安として、週次に発注元へ報告する。</p>
出力	<ul style="list-style-type: none"> <li>レビュー議事録</li> <li>探索的テストレポート</li> </ul>
注記	なし

ゴモラメソッドの流れをDFDとして図1に示す。

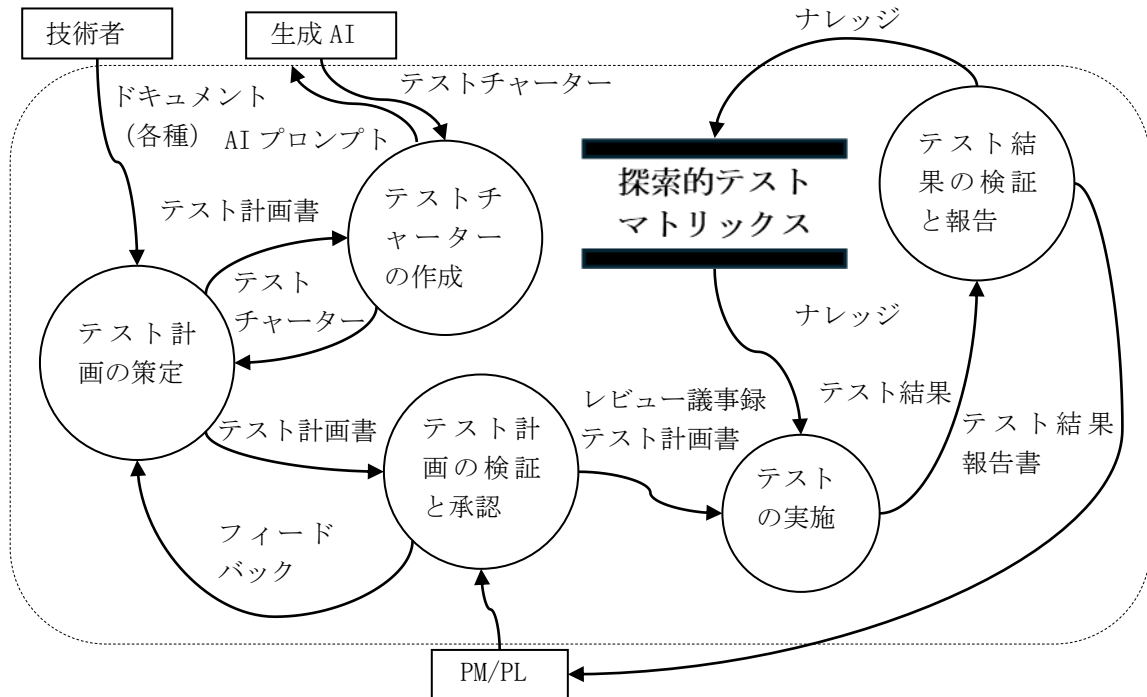


図1 ゴモラメソッド

### 3.1.2. 探索的テストマトリックスについて

探索的テストマトリックス<sup>[5]</sup>（以下 マトリックス）とは、開発スケジュールの変更に応じて、テスト優先度の高い箇所を特定して関係者同士で共有することを目的としたものである。筆者が作成するマトリックスは、テスト対象の機能および特別確認項目として利用時品質を検証することを主目的としており、探索エリアを正確に特定するためのものである。また、そのための入力情報として、記述式テストの結合／システムテスト仕様書のテストケースの内容から組み合わせパターンを確認する。そして、組み合わせで網羅していないようなところを探索エリアとしている。なぜなら、特定した探索エリアを探索することで、記述式テストでは何を確認していないかをよく知ることに繋がるため、記述式テストのテスト設計の抜け漏れがわかることでテストの価値が向上するからだ。

マトリックスは、実施したセッション回数と検出したインシデントの数によって、探索エリアの怪しさを表現できる。例えば、インシデントの数は欠陥の多さを示すものではなく、探索している技術者の理解の深さに依存しているものと考えられ、仕様が不明瞭な点が多いことが想定できる。そのような場合には、実施したセッション回数が例え十分だとしても、別の技術者を割り当てることがテストの抜け漏れの観点で特に重要となる。また、検出されたインシデントの情報は、次に同じ探索エリアを探索する人の参考情報としても有用である。マトリックスの作成手順を以下に示す。

<利用時品質を検証するためのマトリックスの作成手順>

1. 探索的テスト計画書のテスト目的から探索的テスト対象とする利用時の品質特性を確認し、マトリックスのテスト方針欄の同じ利用時の品質特性に○を付ける。
2. ソフトウェアシステムテスト計画書からテスト範囲を特定する。
3. テスト範囲で指定された機能をシステムテストのテスト仕様書からすべて抜き出して、マトリックスのテスト範囲の機能欄に転記する。
4. テスト範囲で指定された機能以外の特別な確認項目をソフトウェアシステムテスト計画書から抜き出して、マトリックスのテスト範囲の特別確認項目欄に転記する。
5. テスト範囲に記入された各項目（機能／特別確認項目）に対して、手順1の探索的テスト対象の品質特性のみ、利用時品質のどこに該当するかを判定し、該当するところに○を入れる。また、対象外の品質特性の欄には、N/A（適用外）を入れる。
6. 探索的テスト方針で決定した利用時品質のところに該当する機能および特別確認項目を

探索的テスト対象とする。

- 探索的テスト対象にチャーターを設定する。チャーターには探索的テスト対象に対して何を確認するかの簡単な説明を記載する。実施中は必要に応じてチャーターを更新し、セッション回数/インシデント数/欠陥数を記録する。

### 3.1.3. 探索的テストプロセスおよびマトリックスの効果

表 1 と表 2 に示した探索的テストプロセスにより、探索的テストを実施する技術者に基本的な手順を支援することができる。そして、探索的テストを実施する技術者が、ET1 と ET2 を経ることで、生成 AI のサポートおよび有識者の知見を得て利用者像の特定方法とドメイン知識を習得し、マトリックスを自らの力で作れるようにしていくことができる。ただし、考えられる課題として、有識者の知見を得る際に、だれがどのような知見を有しているかが不明瞭のため、プロセス活動に慣れるまでには実施経験のある程度積んでいく必要がある。

マトリックスの役割は、探索的テストの「道しるべ」であり、「製品」「市場」「リスク」「欠陥」「テストで失敗した方法」について学んだ重要なナレッジを蓄積/共有することである。具体的には、チャーターに設定/入力した情報がナレッジとなり、紐付けられた探索的テスト結果に蓄積/共有することができる。また、生み出される効果については、探索的テストを実施する技術者の学習基盤となることだけでなく、複雑なシステムテストにおいて、合理的にテストのバラツキを抑制することで、テストの価値を最大化することにある。

## 3.2 実験と評価

### 3.2.1 実験

ゴモラメソッドの有効性評価として、ゴモラメソッドを使用する 10 名（ベテラン 4 名、非ベテラン 6 名）と使用しない 5 名（非ベテラン 5 名）の社内有志で募った計 15 名に対し実験を行い、欠陥検出に違いがあるかを検証する。なお、今回の実験で利用する手順および実験環境は以下のとおりとする。

<実験手順>

- ゴモラメソッドあり（以降 チーム A）とゴモラメソッドなし（以降 チーム B）に分ける。
- チーム A, B に実験用のテストベースとして Web ページと機能仕様書を被験者へ提示する。
- チーム A にはテスト観点一覧（蓄積したナレッジ）を提示し、約 1 時間テスト観点抽出を行う。
- チーム A, B は約 1 時間探索的テストを実施する。
- Web ページに存在する欠陥をどの程度検出できるかを確認する。

<実験環境>

- 実験のテストベースはまだ仕様が固まっていない SNS の Web ページを選定した。
- 機能仕様書の構成は画面遷移図、画面レイアウト、画面項目、入出力条件、イベント定義とした。
- 欠陥は事前にテストベースの検証を行い、最低でも 17 個が存在している状態とした。

### 3.2.2 実験結果

実験の結果、1 名あたりのテスト項目数および検出した欠陥数の平均値を表 3 に示す。

表 3 探索的テストで実施したテスト項目数と検出した欠陥数の平均値（件数）

テスト経験	チーム A		チーム B	
	テスト項目数	欠陥検出数	テスト項目数	欠陥検出数
非ベテラン（10 年未満）	14.33	17.00	19.00	14.20
ベテラン（10 年以上）	21.75	16.75	-	-

表 3 より、チーム A は、チーム B と比較して非ベテランはテスト項目数が平均 4.67 件（約 25%）少なく、欠陥検出数は平均 2.8 件（約 20%）多かった。また、非ベテランはテスト項目数がベテランと比較して平均 7.42 件少なかった。また、欠陥検出数の差は平均 0.25 件だった。

また、実験後に生成 AI が作成したテストチャーターに関して以下に示す意見が得られた。テストチャーターに関する意見：「文字の羅列で読み難い」「過去欠陥は参考になった」「機能概要や一

一般的な SNS については応用性がない（なくてもよい）」「テスト対象に適していない情報と感ずることが多くテスト目的をもっと限定すべきだと感じた（AI プロンプトにもっと力を入れたほうが良い）」

#### 4. 考察

表 3 より、ゴモラメソッドを使用するとベテランと非ベテランは同じ程度の欠陥数を検出できることがわかった。このことから、先行研究で課題となっていた探索的テストの体験からの教訓を活用できることがわかった。しかしながら、非ベテランのテスト項目数については、ベテランと比較して約 65.89% だったことから、テストの網羅性に違いがあることが考えられる。この要因は、ベテランはテスト結果を踏まえて次にテストすべき点を経験からすぐに導き出していたためだと考えられる。また、チーム A とチーム B の非ベテランを比較した場合、テスト項目数についてはチーム B の方が多いが、欠陥検出数は少ないことがわかった。この要因は、チーム B の被験者へ提示した情報とテスト対象に関わった時間が少なかったため、テストすべき点を考える範囲が狭くなっていたことが考えられる。これらのことから、先行研究で課題となっていた探索的テストの具体的なやり方がわからない点についてはゴモラメソッドを活用することが効果的であることと、探索的テストの経験を積む必要があることがわかった。

実験後に得られた生成 AI に関する意見より、生成 AI が出力したテスト目的をそのまま探索的テストに使用することが難しかったことから、AI プロンプトを改善する必要があることがわかった。

#### 5. 結論

##### 5.1. 成果

探索的テストの実施者不足の解消を実現するために、探索的テストの実施経験を基に探索的テストプロセスを定義し、生成 AI を活用して探索的テストを実施するハードルを下げ、継続的に学習していく学習基盤としてゴモラメソッドを提案した。定義したテストプロセスにならって探索することにより、ベテランがどのように探索を開始し進めているかを学ぶことができるようになる。また、マトリックスを利用することで、効果的な探索的テストの実施が期待できる。

##### 5.2. 今後の進め方

今後、考案したゴモラメソッドの有効性を検証するためにサンプル数を増やし、メソッドを改善していく。また、ゴモラメソッドのタスクに探索的テストパターン<sup>[1]</sup>リスト（ベテランから収集したパターンを定義したもの）を用いて、探索的テストの留意点を確認するようなタスクを追加するなどブラッシュアップしていく。

今後の課題として、AI プロンプトを改善するプロセスの追加を検討していく。また、今回提案したマトリックスを工夫し、ユーザビリティ以外への適用を検討していく。そして、今回提案したゴモラメソッドは、アプリ開発における記述式テストと探索的テストの併用を前提としているが、他の種類のやり方も検討していく。

#### 6. 参考文献

- [1] 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”, 株式会社 AGEST, ソフトウェア品質シンポジウム, 2022.
- [2] Cem Kaner・James Bach・Bret Pettichord [著], “Lessons Learned in SOFTWARE TESTING”, Published by John Wiley & Sons, Inc., New York., 2002.
- [3] 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 GuideV3 - 第3班”, p.205, 2020.
- [4] 高橋寿一[著], “ソフトウェア品質を高める開発者テスト - アジャイル時代の実践的・効率的なテストのやり方 - 初版”, 翔泳社, 第11章 探索的テスト, 2021.
- [5] 上田和樹・丹場順次・工藤修悟[著], 日本ナレッジ株式会社, “短納期型開発プロジェクトのためのテスト手法「FaRSeT (Flexible and Rapid Software Test) の適用と効果」”, 探索的テストマトリックスの作成, 2018.