

## 生成 AI によるテスト分析の実験的評価とプロンプト設計の考察

An Experimental Evaluation of Test Analysis Using Generative AI and a Study on Prompt Design

研究員：川合 泰史（富士電機株式会社）  
齋藤 美穂（株式会社ベリサーブ）  
倉田 優輝（株式会社日立ソリューションズ・クリエイト）  
主査：石川 冬樹（国立情報学研究所）  
副主査：徳本 晋（富士通株式会社）  
アドバイザー：栗田 太郎（フリー株式会社）

### 研究概要

ソフトウェアテストでは、仕様の曖昧さやシステムの複雑化により、不具合の見落としが課題となっている。そのため、要件定義書などを基にテスト対象を明確化し、リスクに基づいてテスト条件を抽出するテスト分析が重要である。

しかし、テスト分析は担当者の経験やスキルへの依存度が高く、品質のばらつきや工数増大といった問題を抱えている。これを解決するために、既存のテスト分析手法を整理し、その構造を基に作成した「テスト分析観点」を用いた生成 AI と協調しテスト分析を行う手法を提案する。また、テスト分析に関しては多様な手法が提案されているため、手法の違いによるテスト分析結果の影響を評価する。

In software testing, ambiguities in specifications and increasing system complexity have made it challenging to detect defects effectively. Therefore, test analysis—which clarifies test targets based on requirement specifications and other documents and derives test conditions based on risk—is essential. However, test analysis heavily depends on the experience and skills of individual testers, leading to issues such as variability in quality and increased effort.

To address this, we organize existing test analysis methods and propose a method for performing test analysis in collaboration with Generative AI, utilizing "test analysis perspectives" structured based on those methods. In addition, since a wide variety of test analysis methods have been proposed, this study evaluates how differences among these methods affect the outcomes of test analysis.

### 1. はじめに

近年、ソフトウェア開発の規模と複雑さが増す中で、効率的かつ高品質なテストの重要性が高まっている。特に、テスト分析はテスト設計や実行の質を左右する重要なプロセスであり、従来は専門知識や経験に依存する部分が大きかった。しかし、生成 AI 技術の進展により、自然言語処理やパターン認識を活用してテスト分析を自動化・支援する可能性が広がっている。本研究では、生成 AI を用いたテスト分析の実験的アプローチを提案し、その有効性と課題について評価する。

### 2. 背景

ソフトウェアテストは、システムの品質保証において不可欠な工程である。特にテスト分析は、要件や仕様からテスト対象を特定し、テスト設計の方向性を決定する基盤となる。しかし、テスト分析には膨大なドキュメントや仕様書の理解、複雑な業務ロジックの把握が求められるため、人的リソースや時間の負担が大きい<sup>[1]</sup>。また、人的要因による見落としや主観的な判断が品質に影響を与えるリスクも存在する。

一方で、生成 AI は、自然言語やコードのパターンを学習し、新たな内容を生成する能力を持つ。これにより、仕様書や要件定義書から自動的にテスト分析の候補を抽出したり、テスト設計のヒントを提示したりすることが可能になる。特に、大規模言語モデル（LLM）

## 第5 研究コース「人工知能とソフトウェア品質」コース（テスト分析グループ）

を活用したアプローチは、テスト分析の自動化や補助に有効であると期待されている<sup>[2]</sup>。

本研究では、生成 AI を用いてテスト分析を実施する実験を行い、そのプロセス、成果、課題について検証する。これにより、生成 AI がテスト分析に与える影響や、今後の実用化に向けた方向性を明らかにすることで、テスト分析の質を保つと同時に作業者の負担軽減が可能なテスト分析手法の確立を目的とする。

### 3. 課題

ソフトウェアテスト工程は、ソフトウェアの品質保証において不可欠な役割を担っており、不具合の早期発見や製品の信頼性向上に直結している。中でもテスト分析は、テスト設計や実行の質を左右する重要な工程である。しかし近年、ソフトウェアの高度化・大規模化や開発プロセスの変化に伴い、以下のような課題が顕著となり、効率的かつ効果的なテスト分析およびテストの実施を阻害している。

#### ① ソフトウェア技術者の知識不足

IT 業界全体で技術者不足が深刻化する中、特に高い専門知識と経験を要するテスト設計者やテスト分析者の確保が困難となっている。これにより、テスト設計の質が均一化せず、属人化や知識の継承問題が生じやすくなっている。また、新技術や複雑化するシステムに対応できる人材を育成するための時間とコストも増加している。

#### ② 従来のテスト分析の時間と労力の多さ

設計ドキュメントや仕様書からテスト対象の抽出およびテストケースへの展開は従来手作業で行われることが多く、多大な作業時間と労力を要する<sup>[1]</sup>。さらに、アジャイル開発や DevOps の普及によりリリースサイクルが短縮されている中で、テスト分析工程の効率化が追いつかず、全体の開発スケジュールに影響を与えている。これが原因で十分なテストが実施されないリスクも増大している。

#### ③ テストの網羅率向上と効率的設計の難しさ

高品質なソフトウェアを実現するためにはテスト網羅率を高めることが不可欠であるが、単にテストケースを増やすだけではコストと時間が膨大になる。適切なリスク評価や優先順位付けを行いながら効率的に網羅性を確保することは高度な技術を要し、多くのプロジェクトで品質向上のボトルネックとなっている。特に変更頻度の高い部分に対して効果的なテストを実施するための動的なケース設計は困難である。

### 4. 仮説

生成 AI を活用したテスト分析は、課題解決の有効な手段と考えられる。生成 AI は、大量の仕様書や設計文書からテスト観点や条件を自動抽出し、経験豊富な技術者に匹敵する知見を短時間で提供できる。この技術により、技術者の負担を軽減し、少人数でも効率的かつ一定の品質を担保したテスト計画の作成が可能になる。また、生成 AI は新規参画者や未経験者の支援ツールとしても有効である。専門知識が不足しているメンバーでも、生成 AI を活用することで一定水準の分析を実施できる環境を構築できる。この環境整備はチームの生産性向上を促進し、技術者不足への対応にも役立つ。

ただし、生成 AI に分析観点を指示しない場合、期待通りの結果が得られない可能性がある。この課題を解決するには、ソフトウェアの品質要求（例：リスク重視、操作性重視など）に応じた分析観点を明確に指示することが重要である。適切な指示を与えることで、より効果的なテスト分析が可能となり、品質向上に寄与すると考える。

以上より、生成 AI を活用したテスト分析は、技術者不足による課題を克服する有効なアプローチであると考え、実験を行うことにした。また、生成 AI の効果を最大化するには、品質要求に応じた分析観点を適切に指示することが重要である点を踏まえ、実験を進める。

## 第5 研究コース「人工知能とソフトウェア品質」コース（テスト分析グループ）

### 5. 提案手法

本研究では、要件定義書を入力とし、システムテストにおけるテスト分析を、生成 AI を主体に協調して行うプロセスを提案する。

提案手法では、既存のテスト分析を参考に、AI と協調してテスト分析を行うプロセスを定義し、さらに整理した結果である「テスト分析観点」を基にテスト分析用プロンプトを構築する。

#### 5.1. テスト分析の定義

本研究では、JSTQB<sup>[3][4]</sup>にて定義される「テスト分析」の定義に準拠する。

JSTQB における「テスト分析」とは、図 1 に示すテストプロセスのライフサイクルのうち、テスト設計に先立って実施される工程であり、テストベースを分析することによって、テスト対象となる機能や特性を特定し、それぞれに対するテスト条件を定義するとともに、重要度およびリスクに基づいて優先順位を決定する活動である。また、テスト対象に関連するリスクを識別し、各リスクのレベルを評価することで、重点的に検証すべき領域を明確にする。テスト分析は、計測可能なカバレッジ基準に基づいてテスト対象の範囲を明確化し、「何をテストすべきか」を定義することを目的とする工程である。

これらの分析を体系的かつ客観的に実施し、テスト条件を識別するために、テスト分析ではテスト技法が用いられる。

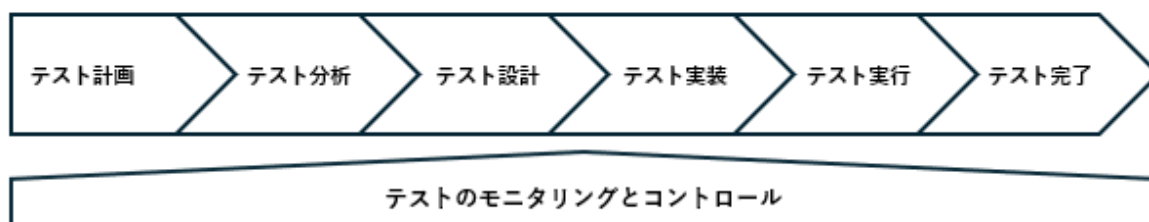


図 1. テストプロセスのライフサイクル

テスト分析の一般的な手順は以下の通りである。

- ①テストベース（要求仕様書、設計書、ユースストーリーなど）の収集と整理
- ②テスト対象の機能や品質特性の把握
- ③テストすべき条件（テスト条件）の抽出
- ④テスト条件の分類・体系化
- ⑤リスク評価の実施とリスクレベルに基づくテスト条件の優先順位付け
- ⑥テスト範囲の網羅性の確認と不足条件の検出
- ⑦テスト条件のレビューと関係者との合意形成
- ⑧テスト設計およびテストケース作成のための基盤情報として文書化

#### 5.2. テスト分析観点

本研究にて、JSTQB が定義するテスト分析の内容を考慮した上で、自分たちの業務で実際に行うテスト工程に必要な情報を検討した結果、テスト分析はテストベース（要件定義書／要求仕様書）を入力として表 1 に示す「テスト分析観点」を出力することが、ソフトウェアテストの品質向上および効率的な実施に不可欠であると考えた。

表 1. テスト分析観点一覧

No	項目	説明
1	テスト対象	テストの範囲を明確化するために必要である。明確な対象設定がなければ、重要機能の漏れや不要な部分への過剰なテストが発生し、テストの効率性と網羅性が損なわれる。

## 第5 研究コース「人工知能とソフトウェア品質」コース（テスト分析グループ）

2	テスト目的	テスト活動の方向性を定める役割を持つ。機能検証、性能評価、セキュリティ確認など多様な目的が混在する現代のソフトウェアにおいて、目的の共有と明確化は適切なテストケース設計の基盤となる。
3	テスト条件	具体的な入力値、環境設定および前提条件を示し、テストの再現性と結果の信頼性を担保する。条件を体系的に整理することで、異なる状況下での挙動検証が可能となり、品質評価の精度向上につながる。
4	優先度	潜在的な不具合発生箇所や影響度を評価し、資源配分の優先順位付けに寄与する。限られたテストリソースを効率的に配分し、重要度の高い領域を重点的に検証するために不可欠なプロセスである。
5	トレーサビリティ・カバレッジ	テスト項目と要求仕様の対応関係を明示することで、仕様の抜け漏れを防止し、テスト網羅性を保証する。さらに、不具合発生時の原因追跡や変更影響範囲の特定を効率化し、品質管理全体の透明性向上に寄与する。

### 5.3. 提案手法：AI と協調するテスト分析プロセス

5.2 にて作成した「テスト分析観点」から、テスト分析を行う際に出力すべき内容・観点を出力する「テスト分析プロンプトテンプレート」を作成した。このプロンプトでは実際にテスト分析を行う部分（以降、「方針」）に、分析対象の特性や適用するテスト分析手法に合わせたテスト分析観点として追記するものとする。

また、AI により生成される成果物には、AI が事実と異なる情報やもつともらしい嘘であるハルシネーションが発生する可能性がある。これを防止するために、AI による自己採点と人間による分析結果・採点結果のチェックをプロセスとして含め、一定の基準を満たすまでテスト分析と採点を繰り返すことで、テスト分析結果の十分性を担保する。なお AI による自己採点には、5.2 にて作成した「テスト分析観点」を元に作成した「採点プロンプト」を用いる。

以上をまとめた提案手法を、図 2 に示す。

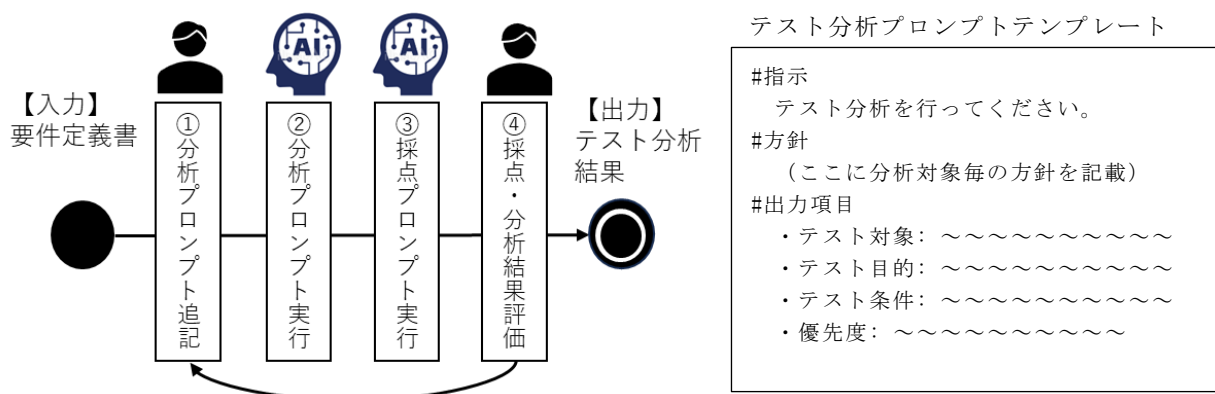


図 2. 提案手法：AI と協調するテスト分析プロセス

## 6. 実験

### 6.1. 実験の目的

本実験では、本研究の目的である「テスト分析の質を保つと同時に、作業者の負担軽減が可能なテスト分析手法の確立」を実現できるかどうかを、提案手法でのテスト分析結果、人手でのテスト分析結果、AI で方針を指定しなかった場合のテスト分析結果をそれぞれ評価・比較することで、十分性及び有効性を評価する。

## 第5 研究コース「人工知能とソフトウェア品質」コース（テスト分析グループ）

### 6.2. 実験方法

実験として、同じドキュメントに対して人手・AIにて提案手法でテスト分析し、その採点結果と実行時間を評価し比較する。実験方法の全体概要を図3に示す。

テスト分析対象は「話題沸騰ポット 要求仕様書 第7版」<sup>[5]</sup>を使用する。比較対象は、手動でテスト分析を行った結果（以降、「人力結果」）と、「テスト分析プロンプトテンプレート」を使用せずテスト分析をAIに任せたとした結果（以降、「AI方針なし結果」）と、「テスト分析プロンプトテンプレート」に既存のテスト分析手法を「方針」として記載し分析させた結果（以降、「AI方針結果」）の3種類を用いる。これを表2に示す。採点では、マークダウン形式で出力したテスト分析結果を、表1. テスト分析観点一覧の各項目に対して、1点を「非常に悪い（基準をほとんど満たしていない）」、5点を「非常に良い（基準を完全に満たしている）」とした5段階評価で十分性を評価した平均値を、各観点の採点結果とする。実行時間は、図3の「②テスト分析実施」の開始から「④分析結果・採点結果の妥当性評価」終了までの合計時間とする。

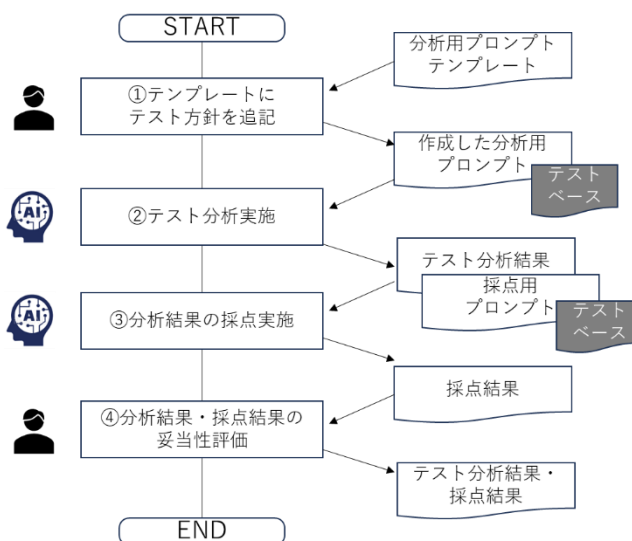


図3. 実験方法

表1. テスト分析観点一覧の各項目に対して、1点を「非常に悪い（基準をほとんど満たしていない）」、5点を「非常に良い（基準を完全に満たしている）」とした5段階評価で十分性を評価した平均値を、各観点の採点結果とする。実行時間は、図3の「②テスト分析実施」の開始から「④分析結果・採点結果の妥当性評価」終了までの合計時間とする。

表2. 実験で評価するテスト分析の特徴

No	テスト分析方針	テスト分析方針の説明
1	指定なし(人力)	AIを使用せず人力(経験30年のQA)によるテスト分析する
2	指定なし(AI)	分析方針を指定せずに「テスト分析を行うこと」のみを記載する
3	ユーザーストーリー	対象製品を使用するユーザーとそのストーリーを入力ドキュメントから想定し、その想定ストーリーに対して実行・確認すべきテストを分析する
4	リスクベース	入力ドキュメントから想定できるリスクとその優先度を設定し、優先度毎にテストすべき内容を定義する
5	リスクドリブン	潜在的な障害が発生した場合の影響度や発生確率といったリスクを基準に、テストの優先順位や範囲を決定する手法によるテスト分析を行う
6	体系的手法強調	ソフトウェアテストに関する国際標準規格群（ISO/IEC/IEEE 29119）に準拠したテスト分析を行う
7	抽象・戦略	「テストをどのように進めるか」という全体的な方針や指針を、初期段階で決定するテスト戦略立案の観点からテスト分析を行う
8	開発時品質 (主特性・副特性)	ISO/IEC 25010 <sup>[6][7]</sup> に定義される品質特性のうち開発時品質の主特性および副特性の観点から、外部品質を対象としたテスト分析を行う
9	開発時品質 (副特性)	#8 開発時品質(主特性・副特性)から副特性に特に注目し、ISO/IEC 25010の副特性の定義に基づいてテスト分析を行う
10	HAYST法	HAYST法 <sup>[8]</sup> を用いて要求、設計、実装・環境の3階層から体系的にテスト条件を抽出し、抜け漏れを防ぎ網羅性を確保する

## 第5 研究コース「人工知能とソフトウェア品質」コース（テスト分析グループ）

### 6.3. テスト分析に使用した AI

今回実験にて用いた AI モデルを以下に示す。

表 3. 実験で使用した AI モデル

モデル	Gemini 2.0 Flash Experimental(思考モード、高速モード、Pro モード)
	Gemini 3.0 (思考モード、Pro モード)

### 7. 実験結果と考察

本実験の結果を以下に示す。

表 4. 実験結果一覧

No	テスト分析方針	テスト対象	テスト目的	テスト条件	優先度	トレーサビリティ・カバレッジ	平均	実行時間(分)
1	指定なし(人力)	3.67	4.67	4.50	1.33	4.50	3.89	360
2	指定なし(AI)	4.33	3.33	4.33	1.00	3.50	3.47	20
3	ユーザーストーリー	5.00	4.67	4.50	5.00	3.25	4.42	23
4	リスクベース	5.00	5.00	4.67	5.00	4.50	4.79	26
5	リスクドリブン	5.00	5.00	4.50	5.00	4.75	4.79	20
6	体系的手法強調	4.33	4.67	3.83	4.33	4.00	4.16	20
7	抽象・戦略	4.67	4.67	4.33	5.00	4.50	4.58	20
8	開発時品質(主特性・副特性)	5.00	4.33	3.67	5.00	4.75	4.42	45
9	開発時品質(副特性)	5.00	4.33	4.67	5.00	5.00	4.79	10
10	HAYST 法	5.00	4.33	3.83	5.00	4.75	4.47	45

#### 7.1. 考察

「人力結果」と AI によるテスト分析結果を比較したところ、全ての項目について概ね同等であり、そのうち「優先度」においては優位な結果を得られた。また、「AI 方針なし結果」と「AI 方針結果」を比較したところ、「テスト目的」と「優先度」については優位な結果を得られた。一方で、どちらの場合でも、「テスト条件」および「トレーサビリティ・カバレッジ」についてはテスト方針を指定した一部のプロンプトでは下回る結果となった。

##### 7.1.1. 「テスト対象」について

「人力結果」および「AI 方針なし結果」と比較して、「AI 方針結果」は同等および優位な点数となった。「人力結果」で採点が低くなった原因は、テスト対象外の項目や除外条件といったテストの要否を判断する項目が記載されていなかったためである。

##### 7.1.2. 「テスト目的」について

「人力結果」および「AI 方針なし結果」と比較して、「AI 方針結果」は同等および優位な点数となった。「AI 方針なし結果」では他の結果と比較して、テストの目的が機能的な確認の充分性に限られており、また具体性も欠けた結果となっていたことが、採点で差が見られた原因と考えられる。

##### 7.1.3. 「テスト条件」について

「テスト条件」については、実行結果毎に採点結果が異なった。「人力結果」と「AI 方針なし結果」と比較して、概ね同程度の点数となったが、一部の「AI 方針結果」では採点が下回った。採点が下回ったものについて分析すると、以下の2点の傾向が見られた。

- (1) テスト条件として考慮すべき「テストの制約条件」の考慮が不足していた事に起因して点数が低くなっている傾向が見られた。プロンプトにて指定した「テストの制約

## 第5 研究コース「人工知能とソフトウェア品質」コース（テスト分析グループ）

条件」について、人間が想定したテスト分析で出力すべき「テストの制約条件」とAIの「テストの制約条件」で認識の差異があると考えられるため、今後の課題として、プロンプトの改修を検討する必要がある。

- (2) テスト条件として考慮すべき「テストの開始条件」または「終了(目的)条件」の記載が不十分なものが見られた。こちらについては一部プロンプトでのみ発生しているため、今後の課題として該当「AI 方針結果」のプロンプトの改修を検討する必要がある。

### 7.1.4. 「優先度」について

「人力結果」および「AI 方針なし結果」と比較して、「AI 方針結果」は著しく点数が高い結果となった。「人力結果」及び「AI 方針なし結果」で採点が低くなった原因は、それぞれのテスト目的に対して優先度付けが全く行われておらず、考慮できていないためである。

一方、「AI 方針結果」では、テスト分析方針をプロンプトとして明示したことで、各テスト項目に対する優先度付けが一貫した基準に基づいて行われたと考えられる。

### 7.1.5. 「トレーサビリティ・カバレッジ」について

「人力結果」および「AI 方針なし結果」と比較して、「AI 方針結果」は概ね同程度となったが、一部、「AI 方針なし結果」と「ユーザーストーリー」は採点が低くなった。

今回提案手法ではチャットで出力されるテスト分析結果をマークダウン記法の中間ファイルに出力し、それを採点している。この中間ファイルの出力において、チャット上の参照形式がテキストファイルへ上手く反映できないことが稀に発生していることが、一部提案手法で採点が低くなった原因と考えられる。この中間ファイル出力方法について、今後の課題として検討する必要がある。

### 7.1.6. その他

- (1) 「AI 方針結果」の出力内容について

各「AI 方針結果」にてプロンプトに記載したテスト方針に合わせたテスト分析結果を、一定の品質で出力が可能であった。ただし、人間にて想定した単語・「テスト分析」における単語と、AIにて解釈される単語間で認識の差異が見られたため、プロンプトにて単語を定義するなどの検討が必要である。

- (2) 実行時間について

「人力結果」と、「AI 方針結果」の実行から評価までの時間（以降、「実行時間」）を比較したところ、「人力結果」約6時間、「AI 方針結果」では約25分となり、「AI 方針結果」が大きく優位な結果となった。

## 7.2. 結論

提案手法、既存のテスト分析を「方針」として与えたAIによるテスト分析とその採点によるテスト分析の品質確認により、人力でのテスト分析と大きな相違がなく、また実行者のスキル・経験によって検討漏れ・検討不足となり得る項目を一定の品質で抽出した上で、大幅に実行時間の短縮が可能である。

また、既存のテスト分析手法を方針として与えたAIによるテスト分析では、それぞれのテスト分析の方針に応じた結果が出力可能である。これにより、プロジェクトの特徴・状況に応じて適切な「方針」を指定することで、実施者の要求に応じたテスト分析が可能であると考えられる。

一方で、現在のテスト分析および採点プロンプトには考察に記載した通り、「テスト条件」「トレーサビリティ・カバレッジ」などを含めたプロンプトの記載において課題があるため、今後の研究ではこれらのプロンプトの改修を検討する。

## 8. 総括・今後の課題

本提案手法、既存のテスト分析手法を「方針」として与えたAIによるテスト分析と採点による品質確認により、品質・実行時間・網羅性を確保することができる。

また本手法を実際にテスト工程で活用する際の発展として、以下も考えられる。

- ・ 本提案手法の出力と人間が作成したテスト工程ドキュメントとの比較レビュー
- ・ 本提案手法の出力を元にした、高品質なテスト項目の自動作成
- ・ 本提案手法のテスト方針を組み合わせたテスト分析プロンプトの作成

## 第5 研究コース「人工知能とソフトウェア品質」コース（テスト分析グループ）

ただし、本研究では以下が課題である。

- (1) プロンプトのブラッシュアップ・AI のさらなる活用  
結論などに示した通り、プロンプトの内容にはまだ改修の余地があり、テスト分析で求められる内容と AI の認識差異を埋められるような改修が必要である。  
また、テスト分析データを集計した RAG の作成や他モデルでの実験、モデルのバージョンアップ等による出力内容の確認なども必要である。
- (2) 実験結果に対する再評価  
今回使用した要求仕様書は単一のものであるため、その性質に大きく依存している可能性がある。例えば、今回の要求仕様書はポットというハードウェア製品が対象となっているため、アプリケーションやソフトウェア開発を主体とした要求仕様書では、全く異なる出力・実験結果となる可能性がある。そのため、他の要求仕様書を利用したプロンプトの改修やテスト分析手法ごとの再評価が必要である。今回実験・採点で用いた平均値や採点方法などの実験方法に対しても同様である。
- (3) 提案手法の人間による評価  
本提案手法では研究員と AI による評価を行った。これから発展させ、人力によるさらなる評価をアンケートなどで実施し、プロンプトの改修や提案手法のブラッシュアップが必要である。

### 参考文献

- [1] SQiP 研究会 研究コース 4「ソフトウェアテスト」, “短納期でかつ小規模な開発プロジェクトにおけるテスト分析・設計導入による有効性の検証”, SQiP 研究会, pp. 7-8, 2017
- [2] Katja Karhu, Jussi Kasurinen, Kari Smolander, Expectations vs Reality -- A Secondary Study on AI Adoption in Software Testing, arXiv preprint arXiv:2504.04921, 2025.
- [3] International Software Testing Qualifications Board, テスト技術者資格制度 Foundation Level シラバス Version 2023V4.0.J02, 2023.
- [4] 湯本剛, 大西建児, 勝亦匡秀, 佐々木方規, 鈴木三紀夫, 町田欣史, 吉澤智美, 中野直樹, ソフトウェアテスト教科書 JSTQB Foundation 第3版, 翔泳社, 2011.
- [5] 組込みソフトウェア管理者・技術者育成研究会 (SESSAME), 話題沸騰ポット (GOMA-1015 型) 要求仕様書 第7版, [https://www.sesame.jp/workinggroup/WorkingGroup2/POT\\_Specification.htm](https://www.sesame.jp/workinggroup/WorkingGroup2/POT_Specification.htm) .
- [6] International Organization for Standardization, ISO/IEC 25010:2023 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuARE) – Product quality model, 2023.
- [7] 日本規格協会, JIS X 25010:2025 システム及びソフトウェア技術 – システム及びソフトウェア製品の品質要求及び評価 (SQuARE) – 品質モデルの概観及び利用法, 2025.
- [8] 秋山浩一, ソフトウェアテスト技法ドリル – テスト設計の考え方と実際 (第2版), 日科技連出版社, 2022.