ソフトウェア品質シンポジウム 2022

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

Patterning of points of attention for effective exploratory testing

株式会社 AGEST
AGEST, Inc.
○飯沼 真一[1]
○Shinichi Iinuma

Abstract

To be effective in exploratory testing, knowledge of the usage environment and knowledge of the application and experience to test are necessaries, so it is important to be familiar with the target area. However, the current situation is that there is a shortage of such skilled test designers. Therefore, it is necessary to keep in mind that mid-level test designers can effectively execute exploratory tests without failing. For them, I extracted the essence from successful and unsuccessful cases in the experience of conducting exploratory tests. It was explicit knowledge as "exploratory test pattern". By utilizing this pattern for mid-level test designers, we tried to solve the shortage of exploratory testers.

1. はじめに

ソフトウェア開発の現場では，アプリ品質が問題となることが多いため，テストケースベースドテストに加え，探索的テストが導入されている．これにより，運用にかかわるような重大な不具合の検出やアプリ品質向上など，探索的テストの導入効果が得られている．近年，熟練テスト設計者が不足し，探索的テストを担うことが難しくなった．そこで，熟練テスト設計者のもとで育成もかねて中堅テスト設計者にやってもらった結果，うまくいかなかった．中堅テスト設計者がうまくいかなかった理由は，探索的テストのやり方の特徴である，「動的にテストを設計し，問題を探す」という行為には，経験と勘（以下，留意点）が必要なためであった．つまり，探索的テストを効果的に行うには，中堅テスト設計者に熟練テスト設計者の探索的テストの留意点を移転可能な状態にしなければならない．

本論文では熟練テスト設計者の探索的テストの留意点を経験から抽出し，グッドパターンとバッドパターンに分け，整理して中堅テスト設計者が探索的テストで活用できるように形式知化した．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，パターンを用いた探索的テストの留意点を移転可能な状態とする方法について説明し，4 章でパターンの検証と結果を述べる．5 章では，パターンを中堅テスト設計者に活用してもらうことで，探索的テストの実施者不足の解消に繋がるかをまとめる．

[1] 株式会社 AGEST
AGEST, Inc.

東京都新宿区西新宿三丁目 20 番 2 号　東京オペラシティビル 41 階
Tokyo Opera City Building 41st floor,3-20-2,Nishi-Shinjuku,Shinjuku,Tokyo Japan

Tel:03-5333-1246
e-mail: shinichi.iinuma@agest.co.jp

【キーワード：】探索的テスト，ソフトウェア工学全般，人材育成，パターン，記述書式

2. 探索的テスト実施上の問題点と課題

2.1 探索的テストとは

探索的テストは形式的ではないテストで，テスト実施時に動的にテストケースの作成とテスト実行とテスト結果の記録及び評価をする．そして，テスト結果を使用し，システムについての理解を深め，さらにテストすべき領域を見出し，テストケースを作成する．また，探索的テストを効率的に実施するために，セッションベースドテストを使用する場合がある．セッションベースドテストとは，探索的テストをあらかじめ決められた時間枠内で実施するもので，テスト担当者はテストチャーター（テスト目的を定義したもの）に従いテスト実施する．

テストのやり方の特徴は，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．また，テストケースを書かないため，テストはテスト担当者に一任される．したがって，テスト担当者は，アプリの仕様や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ことが求められる．一方，テストケースベースドテストでは，テスト実行前にテストケースを書くため，書き出したテストケースをレビューすることでテストの網羅性を高めることができる．

探索的テストには次のような特徴がある．

① 仕様書に基づいてテスト設計するわけではないため，仕様書が最新でなくてもテスト実施できる（ただし，出力結果が正しいかどうかの判断を誤る可能性がある）．

② テスト工数を十分に取れない場合や派生開発の場合など，広く粗いテストができる（ただし，テストの十分性は担保できない）．

③ 熟練テスト設計者が行うと，短時間で意外な不具合を発見できることがある（一方，非熟練テスト設計者が行うと，テスト内容の重複やありきたりのシナリオでのテストとなり，無駄が多い）．

つまり，探索的テストとは仕様が不十分な場合やテストのスケジュールに余裕がない場合や他の形式的なテスト技法を補完する場合に効果が大きいものであり [1]，テストプロセス全体から見ればヒューリスティクスで早い着手と低コストがメリットである．

2.2 探索的テスト実施上の問題点と課題

2.1 で述べたように，探索的テストはテスト実施時に動的にテストケースの作成とテスト実行とテスト結果の記録及び評価を繰り返し行っていくものであるため，次のような問題がある．

(a) テストケースを書かずにテストを進めるため，不具合を発見した際に再現できない可能性がある[2]．

(b) アプリに対する知識が必須であり，テスト実施においては熟練テスト設計者の留意点に依存するところが大きい．

(a) については録画用のツールを使い，テスト実行した手順を録画しておく方法や，テストした内容を把握できるだけの情報をテスト結果に記録しておく方法で，不具合を発見した際に再現できない可能性がある問題の緩和に繋げていくことができる．

(b) については熟練テスト設計者の数は潤沢ではないため，例えば表 1 に示す中堅テスト設計者に熟練テスト設計者の留意点を伝授することが解決の糸口となる．

そこで，本論文の課題を「中堅テスト設計者に熟練テスト設計者の留意点を移転可能とする」とした．なぜなら，中堅テスト設計者に熟練テスト設計者の留意点を繰り返し伝授できる状態にすることで，留意点の一貫性を保ちながら効率的に育成することが可能となり，探索的テストの品質を確保できるからだ．

表 1 テスト設計者の役割

役割 （担当者）	テスト実行 （若手テスト設計者）	テストケース作成 （中堅テスト設計者）	全体チェック （熟練テスト設計者）
テスト技法	テストケースベースドテスト		探索的テスト

2.3 先行事例調査結果

留意点を移転可能とする方法について，先行事例を調査した．

事例「デザインパターンへのソフトウェア工学的取り組み[3]」では，自由度の高いソフトウェアの設計において，偶然ではなく繰り返される問題や解決をデザインパターンとして識別し，再利用している．デザインパターンとは，ソフトウェアの設計における特定の文脈上で頻出する問題，その解決や考慮すべき事柄をまとめた記述である．

先行事例を調査した結果，中堅テスト設計者へ熟練テスト設計者の留意点を移転可能な状態とするには，留意点を形式知化してまとめる必要があることがわかった．

3. パターンを用いた探索的テストの留意点の形式知化

中堅テスト設計者が熟練テスト設計者と同等に探索的テストをできるようにするためには，中堅テスト設計者に熟練テスト設計者の留意点を移転可能な状態にする必要がある．そこで，探索的テストの実施経験における成功事例と失敗事例からエッセンスを抽出し，3 回以上実績がある事例[3]をパターンとして整理して移転可能な状態にすることを考えた．そして，熟練テスト設計者のもとで育成もかねて中堅テスト設計者へパターンを適用し，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パターンを確認して備えてもらう．

パターンとは留意点等を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．

＜記述書式（項目：記載方法）＞

- タイトル：パターンの名称（覚えやすいワンワードで表現する）
- 説明：パターンの目的（具体的な事柄で表現する）
- 状況：パターンの状況（わかりやすい事例で表現する）
- 問題点：具体的な問題点（状況から捉えた問題点で表現する）
- 解決方法：パターンの解説（概要と詳細で表現する）
- メリットまたはデメリット：パターンの効果（具体的な効果で表現する）
- 利用シーン：パターンの利用シーン（いつだれが利用するかを具体的に表現する）
- 使用上の留意点：パターンの留意点（わかりやすい事例で表現する）

このような記述書式とした理由は，中堅テスト設計者が抱えている問題を，予め自身で十分に認識しておかなければ，探索的テストの実施は難しいと考えたためである．

実際に考案したパターンは，良い行動を取りやすくするグッドパターンと悪い行動を取りにくくするバッドパターンに区別し，それぞれ表 2 と表 3 に示している．

※「セッションベース」と「ペアテスト」に関しては既存のテスト技法を流用した．

表 2 考案したグッドパターン

タイトル	No. 1 ： 水先案内人
説明	対象ドメインに明るい熟練テスト設計者の協力を得て水先案内人になってもらい，探索的テストを効果的かつ効率的に行う．
状況	ドメイン知識に乏しい状況で探索的テストを実施しても効果が薄い．
問題点	どこからテストすれば効果的なのかわからない．
解決方法	熟練テスト設計者を協力者として探索箇所の推定と怪しい箇所を指し示してもらう．また，探索不要な箇所があれば示してもらう．複数のテスト担当者が存在する場合は探索範囲が重複しないように機能と機能の間に関連性があるもの同士を 1 つの区画として並び替えを行い（以下，区画整理），各区画にテスト担当者をアサインする．
メリット	・案内人と共に戦略を立てることで探索範囲を約 1 時間という短時間で決める事も可能で，費用対効果が高い． ・セッションベースドテストにテストの重複が同時に起きることを避ける点と偏在する不具合をピンポイントで狙い撃つ点を考慮できる．

利用シーン	いつ「探索的テストを計画するとき」，だれ「テスト設計者」
使用上の留意点	・対象ドメインの熟練テスト設計者が本当に水先案内人として頼れる人なのかを見抜く眼力は必要かもしれない． ・水先案内人は実際の利用状況に詳しいため説明してもらうとよい． ・まとまった仕様書等が無いような場合は新たに作成する必要がある．
タイトル	**No．2 ： お掃除ロボット**
説明	探索的テストは，ある程度の再現性が必要とされており，お掃除ロボットのように「どのような経路で掃除したのか」と「どこにゴミが多かったのか」の情報を記録する．そして，ゴミが多かったところはテストするためのアイデアが出なくなるまで行う．
状況	テスト結果をどのように記録すればよいかわからずに探索的テストを進めている．
問題点	探索的テストの進め方がよくわからない．
解決方法	検出した不具合が再現できるように探索経路を記録する．また，記録にはキャプチャツールを使ってアクションの直前を記録する．不具合が多くありそうな場所を特定しやすくするために検出した不具合の箇所と数を記録する．
メリット	・検出した不具合を再現しやすい． ・記録した情報をもとに別のテスト担当者へ業務説明や引き継ぎを行うことができる．
利用シーン	いつ「探索的テストを実施するとき」，だれ「テスト担当者」
使用上の留意点	・様々な不具合が検出されるため，多少スピード感がそがれても，必要最低限の不具合情報は残さないと後で分析できずに品質が実態より悪く見えてしまうことがある． ・不具合の仕分けは品質分析する際に行い，テストを優先する． ・効率的にテストが進んでいない場合には記録した情報をレビューし，方針／目的に沿っているかを確認する必要がある．
タイトル	**No．3 ： ローテーション**
説明	探索的テストは，利用者からの要求と文書化された仕様との間に何らかの乖離が発生した場合やテストで不具合が多く検出されていたようなシステムの弱いセクションで導入すると不具合の検出に効果が期待できるが，想定よりも不具合の検出数が少ないような場合にはテスト担当者を交代する．そして，次の担当者がすぐ探索的テストに取り掛かるため，これまで何をテストしていたのかを記録した情報に基づいて端的に伝える．
状況	想定よりも不具合の検出数が少ない．
問題点	十分にテストが行われた結果，不具合の検出数が少なかったのかがよくわからない．
解決方法	探索的テストのテスト担当者を交代する．また，引継ぎには探索的テストで記録した情報の補足説明をしてもらう．
メリット	経験の異なるテスト担当者に交代することで，新たな不具合の検出に期待できる．
利用シーン	いつ「品質分析時」，だれ「テスト設計者またはチームリーダー」
使用上の留意点	・テスト実施経験が長いほど異なる留意点を持っていることが期待できるため，交代するテスト担当者を選定する際はスキルよりもテスト実施経験を優先する． ・優先度の観点で複数アサインすることも視野に入れて考えたほうがよい．
タイトル	**No．4 ： セッションベース**
説明	探索箇所をセッション単位に区分して不具合数を見える化し，何をテストしたかの情報を必要最低限残すことで，さらに探索すべき箇所を特定しやすくする．また，テスト担当者同士で探索的テストが重複しないように方針／目的などから作業分担する．
状況	テストケースを書かないため，テストがテスト担当者任せになる．
問題点	テスト実行手順や結果等を第三者が確認できないため，品質分析が難しい．
解決方法	探索的テストをセッション単位で行うセッションベースドテストを導入する．また，1セッションあたりの時間設定は集中力の持続性を考慮して2時間に設定し，探索しきれなかった場合は次セッションに持ち越す．
メリット	・セッション単位で検出された不具合数とテスト結果（何をテストしたかの情報）を残すことができる．つまり，テスト担当者任せにならない． ・セッション単位で品質分析することができる．
利用シーン	・いつ「テスト計画時」，だれ「テスト設計者」 ・いつ「探索的テストを実施する時」，だれ「テスト担当者」

使用上の留意点	・設定時間が余ってしまうような場合には 2 時間を待たずに探索を終了する． ・残す情報量は探索するスピードとトレードオフになる．また，必要最低限の情報は，パターン No.2「お掃除ロボット」を使用するとよい． ・品質分析する際に不具合の仕分けをする．また，アプリの弱点分析をするためには全セッションで少なくとも 1 回はテストする必要がある． ・セッション単位で探索的テストの方針／目的を設定する． ＜方針／目的の設定例＞ A. 過去に発生した再現性の低い不具合が簡単に発生しない状態になっていること． B. UI 操作と内部仕様の間で乖離がない状態になっていること． C. 基盤の環境設定を様々に変更し，起因する不具合がない状態になっていること． D. 全体を満遍なくテストし，大きな不具合が発生しない状態になっていること．
タイトル	**No. 5 ： ペアテスト**
説明	・①教育目的をかねて指導を行う者と指導を受ける者がペアで探索的テストを行う． ・②複雑なシステムに対し，熟練テスト設計者同士がペアで探索的テストを行う．
状況	・①若手テスト設計者や中堅テスト設計者がアサインされる． ・②複雑なシステムで不具合がなかなか収束しない．
問題点	・①探索的テストのやり方や進め方がわからない． ・②不具合が取り除ききれていない．
解決方法	・①熟練テスト設計者が指導者となり，若手テスト設計者や中堅テスト設計者と一緒にテストする．また，テスト中は指導者が留意点を若手テスト設計者や中堅テスト設計者に伝える． ・②熟練テスト設計者同士がペアになり，協力して探索する．また，テスト中はお互いに検出した不具合に関する情報を共有し，声をかけあいながら進める．
メリット	・①②異質の経験を持つ人の観点や結果から若手テスト設計者や中堅テスト設計者だけでなく，熟練テスト設計者のスキル向上も見込むことができる． ・②不具合を出し切ることで収束させることができる．
利用シーン	・①いつ「テスト計画時」，だれ「テスト設計者またはチームリーダー」 ・②いつ「品質分析時」，だれ「テスト設計者またはチームリーダー」
使用上の留意点	・①②探索的テストはテスト担当者独自の留意点に基づいて行われるため，熟練テスト設計者が複数いる場合は適宜変更する． ・①②テスト重複は避けられない．
タイトル	**No. 6 ： 利用者視点**
説明	業務知識や経験が浅い場合，「利用者の視点で当たり前のことはなにか」と「システムが細かいところまでできているかどうか」を考えることが難しいため，予め利用者の要求を確認し，利用者のケース検討のベースを作成することでテストケースを考えやすくする．
状況	利用者の視点で不足しているようなことがあるかどうかがわからない．
問題点	利用者の視点で考えることができない．
解決方法	利用者のケース検討のベースを作成し，テストケースを考えやすくする． 【手順】 1. 利用者の要求を確認するために必要な確認観点を考える． 2. 考えた確認観点をまとめた確認観点一覧を作成する． 3. 確認観点一覧に基づいて可能な限りの調査を行う． ＜確認観点の例＞ A. システムがどのような振る舞いをすれば利用者は納得するか． （いつ，だれが，どのようなときに，何を，なぜ，どのように利用するか．また，よく使われるような当たり前のことは何か．） B. 実際のやりとりを考えて作られているか． （システムを複数の利用者が使う場合，システムを同時並列的に使えるか．また，ステータスやエラーの種類は状況に応じて備わっているか．また，エラーの種類に応じて利用者に正しく伝わっているか．また，利用者はエラーが表示された際，回避または対応できるか．）
メリット	起点となるテスト観点からテストを深掘りしやすくなる．
利用シーン	いつ「探索的テストを実施するとき」，だれ「テスト担当者」

使用上の留意点	・利用開始から利用終了までの利用者の関わりを図等にすると繰り返し考えやすい. ・実際のやりとりを深掘することでステータスやエラーの意図が理解しやすくなる. ・グッドパターン「設計忘れ」との併用を検討するとよい. ＜併用時の例＞ A. 利用者視点「どのようなときに，どのように利用するか」を考える際，設計忘れ「超高速キー操作」を行うような状況を想定し，カチカチと連打したくなるところからテストする. B. 利用者視点「複数の利用者が使う場合，システムを同時並列的に使えるか」を考える際，設計忘れ「同じ操作」を行うような状況を想定し，環境設定や有効／無効などの共有機能からテストする.
タイトル	**No. 7 ： 設計忘れ**
説明	長年運用を続けているシステムを改修する場合や詳細設計書等で第三者が十分にレビューを行っていない場合，予めソフトウェア設計でレビューやテストが十分ではない観点一覧を作成し，テストケースを考えやすくする.
状況	不具合が出そうな部分を想定してテストする.
問題点	不具合が出そうなところが思いつかない.
解決方法	開発者の立場に立ち，ソフトウェア設計時に忘れていることがないか確認する. 【手順】 1. どのように開発し，レビューとテストを行っているかを調査する. 2. 調査結果を分析してレビューやテストが十分ではない観点一覧を作成する. ＜忘れやすい観点一覧の例（A.テスト不足／B.レビュー不足）＞ A. 改修前のシステムと比較して性能劣化や機能していないものがないか. （描画性能｛超高速キー操作・画面スクロール・複数ダイアログ同時起動｝に違いが出ていないか．また，カレンダー｛うるう年・年／月／日跨ぎ｝の動作に影響がないか.） B. 詳細設計書の記載漏れ等からプログラムに影響していないか. （画面表示や印刷物などの出力結果に罫線／文字切れや不明なところがないか．また，同じ操作や操作の取り消しややり直しは繰り返しできるか．また，入力欄に入力される値や種類／型｛数値型：境界値・桁あふれ，文字列型：全角／半角，その他：空白・未入力等｝は考慮されているか.）
メリット	設計書に記載されていなくても，設計忘れによる不具合流出を防止できる.
利用シーン	いつ「探索的テストを実施するとき」，だれ「テスト担当者」
使用上の留意点	・なぜ忘れやすいかの深堀した結果から仮説を立て，探索していくとよい. ・想定利用者像（ペルソナ）を作成してもよいかもしれません. ＜想定利用者像（ペルソナ）の例＞ A. PC 操作巧み「超高速キー操作」「ショートカットキー」「ブラウザバック」など B. PC 操作苦手「セッション切れ」「操作途中でブラウザ閉じる」「入力誤り」など ・グッドパターン「利用者視点」との併用を検討するとよい. ＜併用時の例＞ A. 設計忘れ「複数ダイアログ同時起動」した状態から利用者視点「どのように利用するか」を考えた場合，複数ダイアログ同時起動したときのさまざまなバリエーションが思いつくのでまだ試していないようなことからテストする. B. 設計忘れ「印刷物に不明なところがないか」を確認した状態から利用者視点「よく使われるような当たり前のことは何か」を考えた場合，利用者にとって当たり前の印刷物が思いつくのでまだ試していない印刷物からテストする.

表 3　考案したバッドパターン

タイトル	No. 1 ： 放置プレー
説明	・①ドメイン知識に乏しいことで自然に仕様理解を優先してしまい，探索することができない. ・②若手テスト設計者や中堅テスト設計者がアサインされるような状況において具体的な作業指示をしない.
状況	・①探索的テストで文書化された仕様と現物の突合せをする. ・②探索的テストに必要な留意点を伝えない.
問題点	①②探索ができていない.
解決方法	テスト対象を使った探索的テストの事例を作成する．また，ドメイン知識を補完するために，アプリの仕様や利用状況を丁寧に図解する.
デメリット	・①②探索方針／目的から外れやすい. ・①②効果的かつ効率的に不具合を検出することができない. ・①探索的テストではなく UI テストになってしまう.
利用シーン	・①いつ「探索的テストを実施する時」，だれ「テスト担当者」 ・②いつ「テスト計画時」，だれ「テスト設計者またはチームリーダー」
使用上の留意点	・①②作成した探索的テストの事例はテスト対象の機能が追加／変更された際に見直しが必要かもしれない. ・①思いつきでテストを進めているような状況において，方針／目的を是正させることは難しいため，方針／目的に沿うことは基本であるが，テストを途中で止めない方がよいかもしれない.
タイトル	No. 2 ： セッション破り
説明	セッションベースドテストにおいて 1 セッション単位に設定した時間を守らず，集中力低下とともにテスト工数が増加する．また，設定した時間の超過とともに管理工数も増加する.
状況	セッションの設定時間を守らない.
問題点	テスト工数／管理工数を有効活用できていない.
解決方法	ウォッチドックする人を立てる. また，設定した時間が経過した際には作業をすぐ止めさせるのではなく，切りのよいところで作業を止めてもらう.
デメリット	・テスト工数／管理工数が増加する. ・探索する方針／目的に沿ってやっているが，時間を有効に使うことができない.
利用シーン	いつ「探索的テストを実施する時」，だれ「テスト担当者」
使用上の留意点	・探索しきれなかったセッションは次セッションに持ち越されることが多いため，ウォッチドックする人を立てただけでは，テスト工数が抑えられないかもしれない. ・セッション内で優先順位の高いところからやる場合もあるため，設定した時間が経過していても何をテストしているかに配慮した上で止めるかどうかを判断する必要があるかもしれない.

4.　パターンの検証

4.1　検証方法

表 2 と表 3 に示したパターンについて，下記の検証を熟練テスト設計者 4 名と中堅テスト設計者 1 名に対して行う.

＜検証内容＞

検証 1：アンケートによる検証（賛同の可否と事例の提供を求める）

検証 2：ヒアリングによる検証（パターンの記載内容に対してヒアリングを行う）

※検証した熟練テスト設計者と中堅テスト設計者のスペック

A：探索的テストの経験豊富（テストエンジニア歴約 18 年）

B：探索的テストの経験豊富（テストエンジニア歴約 14 年）

C：探索的テストは未経験だが知識はある（テストエンジニア歴約 10 年）

D：探索的テストの経験浅い（テストエンジニア歴約 19 年）

E：中堅テスト設計者（テストエンジニア歴約 2 年）

4.2 検証結果

ヒアリングはオンライン会議ツールを介し，顔合わせの型式で 30 分程度実施した．また，アンケートはメールやチャットツールを活用した．その結果，すべてのパターンで賛同する声が得られた．ただし,「ペアテスト」は 1 名が非賛同だった．理由は，ペアテストでうまくいった経験がないことであった．例えば，熟練テスト設計者同士で探索的テストを行った際，コミュニケーションはお互いに取っていたものの，検出した不具合の重複が多発したことが挙げられた.また,熟練テスト設計者 D からは探索的テストがうまくいかない理由を教えてもらった．理由は，探索的テストのテスト技術やテスト対象の仕様／システム構成に対しての理解不足と，自身が持っている不具合のパターンの量やバリエーションに乏しいことであった．中堅テスト設計者 E からは探索的テストができない理由を教えてもらった．理由は,「どのような順番で何をすればよいのか」と「なぜその操作をしようと思ったのか」がわからないことであった．

4.3 考察

検証 2 のヒアリング結果から，賛同が得られなかった「ペアテスト」を熟練テスト設計者同士で行う場合，検出した不具合の重複が多発するかもしれない．そこで,「水先案内人」の解決方法で示した区画整理が有効と考える．今後,「ペアテスト」の留意点に盛り込んでいく．また，熟練テスト設計者 D と中堅テスト設計者 E の抱えている問題に対しては,「利用者視点」と「設計忘れ」の解決方法や使用上の留意点で示した手順や例示が有効と考える．

今回提案するパターンのうち「水先案内人」と「ペアテスト」と「ローテーション」と「利用者視点」と「設計忘れ」の 5 つに関しては，パターンの状況や問題点で示した内容が探索的テストに限らないため，他の種類のテスト実施における道標としても効果的であると考える．

5. 結論

探索的テストの実施者不足の解消を実現するために，経験をもとに探索的テストの留意点をパターン化した．今回提案するパターンは，経験豊富な熟練テスト設計者と経験の浅い中堅テスト設計者へアンケートとヒアリングし，色々なアドバイスをもらってブラッシュアップした．そして，このパターンを熟練テスト設計者のもとで育成もかねて中堅テスト設計者に活用してもらい，探索的テストの実施者不足の解消を実現する目処を付けることができた．

今後はアンケートのサンプル数を増やし，中堅テスト設計者へパターンを適用して実際の探索的テストで「報告された不具合の内容は方針／目的に沿っているか」と「設定した時間枠内で不具合が検出されているか」と「具体的にどのようなテストがされているか」の 3 つを検討することにより，パターンの有効性を確認して信頼性を高め，体系立てを行っていく．また，パターンの組み合わせや併用を通じてより効果を発揮するものがないかと他のパターンについても検討し，パターンの追加／工夫を繰り返していくことで中堅テスト設計者が探索的テストを早く担うことができるようにしていく．

6. 参考文献

[1] JSTQB,“テスト技術者資格制度 Foundation Level シラバス”, 4.4.2 探索的テスト, p.60, 2021.
[2] 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 GuideV3 - 第 3 班”, p.205, 2020.
[3] 鷲崎弘宜, 坂本一憲, 大杉直樹, 権藤克彦, 服部哲, 久保淳人, 小林隆志, 大月美佳, 丸山勝久, 榊原彰, ”デザインパターンへのソフトウェア工学的取り組み”, コンピュータソフトウェア, 岩波書店, Vol.29, No.1, pp.1_130-1_146, 2012.