

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

研究員：飯沼 真一（ソーバル株式会社）
主査：飯泉 紀子（株式会社日立ハイテク）
副主査：足立 久美（株式会社デンソー）

研究概要

探索的テストで効果を出すには、テスト対象の利用環境とアプリケーションに関する知識、テスト経験が必要なため、熟練テスト設計者が担うことが多い。探索的テストはその特徴から採用されることが多くなってきているが、それを担う熟練テスト設計者は不足しているのが現状である。そこで、中堅テスト設計者が探索的テストを失敗することなく効果的に実行するための留意点が必要と考え、筆者の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、それを「探索的テストパターン」として形式知化した。このパターンを中堅テスト設計者が活用することにより、探索的テストの実施者不足の解消の目処を付けることができた。また、このパターンは探索的テストに限らず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。

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。そのような場合には、これまで品質向上施策として探索的テストを導入し、運用にかかわるような重大な不具合の検出やテスト対象アプリケーションの弱点強化に一定の効果を示すことができた。近年、多重請負構造で熟練テスト設計者が抱える開発案件が増加したため、多くの管理工数が必要となったことで探索的テストにかけられるテスト工数が減少した結果、方針／目的の検討や指定が不十分となり、十分な探索的テストを担うことが難しくなった。そこで、中堅テスト設計者にやってもらったが、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グッドパターン／バッドパターンに分けて洗い出し、中堅テスト設計者の探索的テストのために形式知化した。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。

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、テスト実行時に動的に設計、実行、ログ記録、および評価をする。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、さらにテストを行わなければならない領域のテストケースを作成する。また、活動を体系的にするために、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）に従いテスト実行していく。

テストのやり方の特徴は、テスト実行時に動的にテストケースを作成すること。また、テスト結果は記録しないことである。そのため、テストはテスト担当者に一任される。

テスト担当者の特徴は、テスト対象アプリケーション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である。

一般的に使われているテストケースベースドテストとのテストプロセスの違いは、図 1 に示したように、探索的テストではテスト実行前にテストケースを書かないため、テスト実行に早く着手できることである。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設計するため、第三者レビューによりテスト設計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くと、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くと、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。

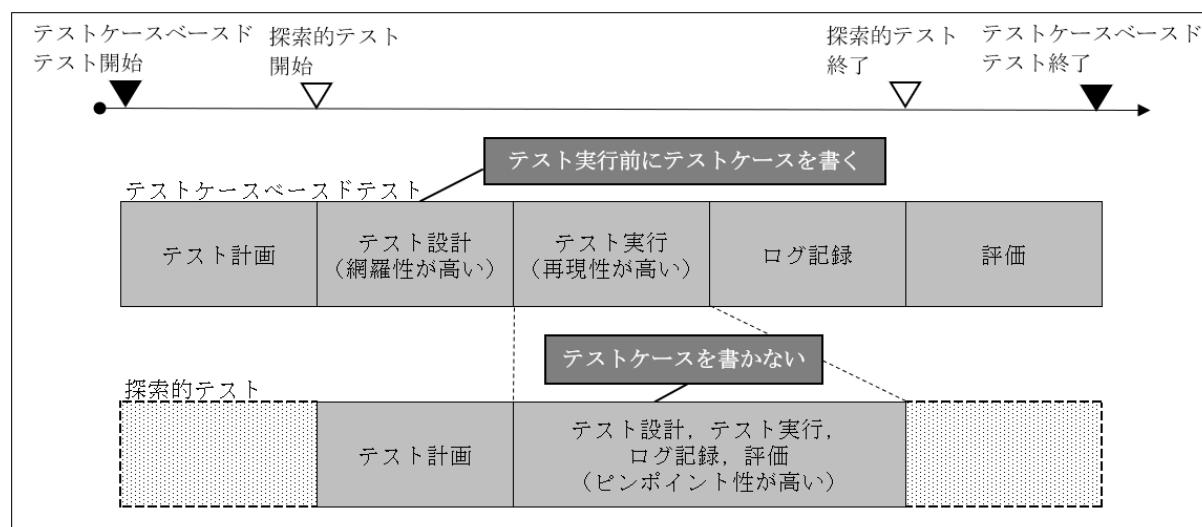


図 1 テストケースベースドテストと探索的テストのテストプロセス比較

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行時に思考しながら不具合がありそうな手順や入力を試行していく。このため、次のような問題がある。

(a) 思考しながらテストを進めるため、たどった手順を記録することはなく、不具合を発見した際に再現できない可能性がある^[2]。

(b) テスト対象アプリケーション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多い。

(a) については録画用のツールを使い、たどった手順を録画しておく方法や、定期的な「何をテストしたか」と「他に何をテストすべきか」についてチーム内で議論し、必要最低限の情報を記録しておく方法でテスト内容を把握することができるため、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、中堅テスト設計者が実施できるようにする必要がある。ここでいう中堅テスト設計者とは、図 2 に示したような担当

するテスト技法が「探索的テスト」の部分を担当できるスキルと経験を有する者である。ある程度のスキルと経験を有する者であれば、テスト対象アプリケーションに対する知識の獲得方法は既に熟知していると考え、熟練テスト設計者の留意点に依存するところをどうやって中堅テスト設計者が獲得していくかが解決の糸口となる。

そこで、本論文の課題を「中堅テスト設計者が探索的テストを失敗することなく効果的に実行できるようにすること」とした。なぜなら、中堅テスト設計者はプロジェクトの状況に応じて探索的テストの方針／目的の検討や指定をしていく必要があるためである。

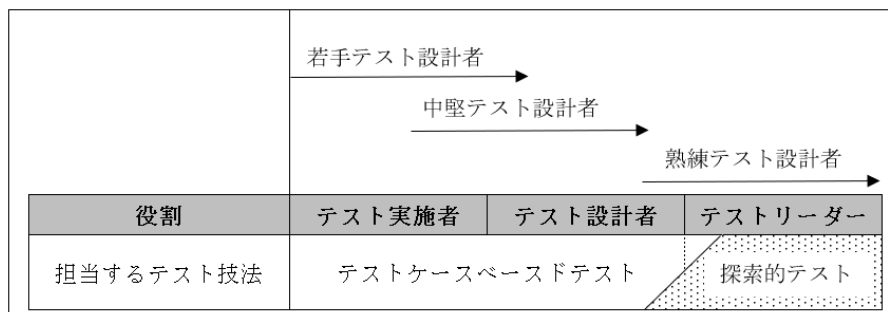


図 2 テスト設計者の役割

2.3. 先行事例調査結果

短納期開発において探索的テストが採用されることが多いことからこの観点で先行事例を調査した。

- ・マインドマップによるテスト分析を活用した探索的テストを適用した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) [3]」

上記事例から、短納期開発では開発プロセスの並行によって、仕様変更の多発を招くことが多い。そこで、仕様変更に伴うテストの効率化のため、探索的テスト実施前に業務分析やシステム分析を行い、網羅性を高めている。問題と考えているところのテスト対象アプリケーションに対する知識の獲得については解決できるが、熟練テスト設計者の留意点に依存する問題は解決できない。

次に、開発期間が長い場合の事例として、テストケースベースドテストと探索的テストを併用する工夫を述べる事例を調査した。

- ・事例「チームで継続的に探索的テストの効果を出す工夫[4]」

上記事例から、開発期間が長い場合には、テストケースベースドテストと探索的テストを行っているが、リリース後に欠陥が検出されることがある。そこで、探索的テストの問題の洗い出しとそれらの施策を検討し、探索的テストの効果を高めている。問題と考えているところの熟練テスト設計者の留意点に対して、テスト実施ログや検出した不具合の情報を基に共有していることから、検出した不具合のテスト観点を獲得することはできるが、テスト観点を新たに導き出せるようになるまでは難しいと考える。

先行事例を調査した結果、効果的に不具合を検出するためには、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し、熟練テスト設計者並みのスキルを付けることが必要で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、何らかの知見や留意点を移転可能な状態にする必要がある。そこで、筆者の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、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、探索的テストを実施する前に、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、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のある記述書式で書いたもので、今回は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）
 - ・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（有効性）またはデメリット（無効性）：パターンの効果（具体的な効果で表現する）
 - ・使用上の留意点「使用シーン」：パターンの留意点（わかりやすい事例で表現する）
- ※テスト計画時・テスト設計時・テスト実行時・ログ記録時・評価時を記載する

実際に考案したパターンを表 1 と表 2 に示している。これらの考案したパターンは、こうするとうまくいくという習慣であるグッドパターンと、こうするとうまくいかないという習慣であるバッドパターンとで区別した。

※「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 1 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（どこからテストすれば効果的なのかわからない）状況でいたずらに探索的テストを実施しても効果が薄い。
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（有効性）	・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。
使用上の留意点「テスト計画時」	・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。また、このパターンはセッションベースドテストを前提に考えており、対象ドメインをあらかじめ複数のセッションに区切る必要がある。 ・探索的テストには明確な方針／目的の設定が必要である。 ・機能一覧のような全体を俯瞰できる仕様書が無い場合に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、「どのような経路を通過して掃除をしたのか」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	探索的テストの進め方がよく分からない。
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、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。

第 37 年度 実践コース 品質技術の実践

メリット (有効性)	<ul style="list-style-type: none"> ・検出した不具合を再現できる。 ・記録した情報を基に別のテスト担当者へ引き継ぐことができる。
使用上の留意点 「テスト計画時とログ記録時と評価時」	<ul style="list-style-type: none"> ・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品質が実態より悪く見えてしまうことがある。そこで、どのような情報を残したらよいかに着目し、直感的でわかりやすいパターン名で表したものである。 また、このパターンはセッションベースドテストを前提に考えているが、探索的テストでセッション時間を気にしたくないような場合にも適用できるため、汎用性が高い。 ・不具合の仕分けは品質分析する際に行う。 ・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、顧客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。
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 (有効性)	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。
使用上の留意点 「評価時」	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。
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	探索的テストは、どこで何をテストしたかの情報が残らないため、テスト担当者任せになる。
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 (有効性)	<ul style="list-style-type: none"> ・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。
使用上の留意点 「テスト計画時とテスト実行時とログ記録時と評価時」	<ul style="list-style-type: none"> ・セッションで設定した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。 ・セッション単位で探索的テストの方針／目的を設定する。 ・品質分析する際に不具合の仕分けをする。また、アプリケーション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。
タイトル	No. 5 : ペアテスト
説明	<ul style="list-style-type: none"> ・①教育目的を兼ねて指導を行う者と指導を受ける者がペアになり、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になり、探索的テストを行う。
状況	<ul style="list-style-type: none"> ・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。
解決方法	<ul style="list-style-type: none"> ・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 (有効性)	<ul style="list-style-type: none"> ・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。

使用上の留意点 「①テスト計画時 ②評価時」	<ul style="list-style-type: none"> ・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。 ・①ペアテストでヒューリスティックスが身に付けられるとは限らない。
------------------------------	--

表 2 考案したバッドパターン

タイトル	No. 1 : 放置プレー
説明	<ul style="list-style-type: none"> ・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	<ul style="list-style-type: none"> ・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。
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ケーションの仕様や利用されるシチュエーションを丁寧に図解する。
デメリット (無効性)	<ul style="list-style-type: none"> ・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。
使用上の留意点 「①テスト設計時 ②テスト計画時」	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。
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。
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 (無効性)	<ul style="list-style-type: none"> ・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。
使用上の留意点 「テスト実行時」	<ul style="list-style-type: none"> ・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。
タイトル	No. 3 : 必要性追求
説明	探索的テストは、不具合が多発していないようなアプリ品質がよい状況で導入したい場合、プロジェクト内外に向けて導入理由を根拠立てし、説明する必要がある。
状況	探索的テストは、具体的にどのようなテストをするつもりかについて問われる。
解決方法	テスト対象アプリケーションの弱点分析を行い、データで弱点を示す。 また、弱点分析は開発者の認識齟齬に起因するような不具合が検出されていないかとアプリケーションの内部構造でブラックボックス化しているような箇所がないかについて分析する。弱点分析の導入理由は探索的テストを導入する客観的な根拠とし、探索的テストを効果的かつ効率的に実施して費用対効果を高めるためである。
デメリット (無効性)	アプリ品質がよいという理由だけで探索的テストを導入する根拠が薄いように見られてしまう。
使用上の留意点 「テスト計画時」	<ul style="list-style-type: none"> ・弱点分析はプロジェクト内外に過剰な期待を持たせてしまうことがある。 ・弱点分析において開発担当者のスキル不足に起因するデータを含める場合はブラックリストに見られてしまう可能性があるため、システムの複雑性や担当機能数をデータに付けるような配慮も必要である (スキル不足は幅が広く、優秀な開発者になるほど担当する機能数が増え、複雑な機能を担うことが多いため定量的なデータには依存しない)。

4. パターンの検証

4.1. 検証方法

表 1 と表 2 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記載を求める）

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）

4.2. 検証結果

検証 1 と検証 2 の結果、表 3 に示したようにすべてのパターンで賛同が得られた。

<検証 1 の結果>

表 3 熟練テスト設計者のアンケート結果

パターン No	パターン名	賛同	非賛同
グッドパターン 1	水先案内人	4	0
グッドパターン 2	お掃除ロボット	4	0
グッドパターン 3	ローテーション	4	0
グッドパターン 4	セッションベース	4	0
グッドパターン 5	ペアテスト	3	1
バッドパターン 1	放置プレー	4	0
バッドパターン 2	セッション破り	4	0
バッドパターン 3	必要性追求	4	0

<検証 2 の結果>

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。その結果、すべてのパターンで賛同する声を得られた。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とっていいものの、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、「ローテーション」は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というアドバイスが得られた。また、「放置プレー」はテスト中（考え中）に方針／目的で縛らないほうがよいという意見があった。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自身の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／システム構成に対する理解不足と、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。

4.3. 考察

検証 1 のアンケート結果から、今回提案するすべてのパターンに対して「探索的テストを効果的に行うための留意点になり得る」との賛同が得られた。特に、パターン名を覚えやすく忘れにくいワンワードで表現した点がよかった。

賛同が得られなかったパターンでは、過去にうまくいかなかったことが挙げられた。

検証 2 のヒアリング結果から、以下の①～③に示したように、パターンをよりよくするためのヒントが得られた。

①「水先案内人」の使用上の留意点に記載したように、探索的テストを行う上では、探

索する方針／目的に沿うことが基本であるが、「放置プレー」のように若手テスト設計者や中堅テスト設計者の思いつきでテストを進めているような状態において、方針／目的を是正させることは難しい。そこで、テストを途中で止めないほうがよいと考える。今後、「放置プレー」の留意点に盛り込んでいく。

②「ローテーション」は、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る。今後、「ローテーション」の留意点に盛り込んでいく。

③「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、グッドパターン「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。

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「放置プレー」の4つに関しては、探索的テストに限らず、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られる。

熟練テスト設計者 D が抱えている問題は、探索的テストの経験が浅い中堅テスト設計者と共通の問題であると考えられる。そこで、問題解決に向けて不具合が出そうな部分を想定するためのパターンとして、過去に検出した不具合のパターンやバリエーションを洗い出すパターンをグッドパターンに加える。

提案するパターンについては、熟練テスト設計者 C と D の賛同が得られたことから、探索的テストに対する知識や経験を有する中堅テスト設計者に対しても有効と考える。

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、筆者の経験を基に探索的テストの留意点をパターン化した。また、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。

この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を実現する目処を付けることができた。

今後は中堅テスト設計者を含めてアンケートのサンプル数を増やし、本研究中にできなかった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、今回定義した記述書式のブラッシュアップと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. 謝辞

研究コース 3 喜多義弘主査，上田和樹副主査には，探索的テストについての貴重なアドバイスを賜りました。厚く御礼申し上げます。

7. 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
- [2] 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
- [3] 上田和樹, 丹場純次, 工藤修悟, ” 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”, ソフトウェア品質シンポジウム 2018, pp.1-8, 2018.
- [4] 中嶋信, 柴田威, 高木進也, ” チームで継続的に探索的テストの効果を出す工夫”, ソフトウェア品質シンポジウム 2017, pp.1-8, 2017.