

不具合と開発現場の実態に基づくテスト分析手法の提案

A Test Analysis Method Based on Failure Reports and Actual Test Processes

クラリオン株式会社 技術開発本部 コアソフト開発部 ソフト環境グループ
 Clarion Co., Ltd, Software Environment Group, Core Software Development Dept., R&D Division
 金子 昌永
 Masanori KANEKO

Abstract Failure reports are important to improve testing activities. In our company, we usually analyze failure reports especially for critical failures. However, when it comes to system testing, we often could not specify what testing activities were necessary to detect those failures. In this research, first, we analyze failure reports to extract and categorize some technical elements of testing, which are necessary to find those failures. Then, we assume that our test process have weakness on test analysis and test design processes. Second, to make sure our assumption is correct, we research the actual test processes in some software testing teams by having an interview with a test leader for each teams. Finally, we establish a test analysis method to improve our weakness, and we evaluate this method.

1. はじめに

我々の組織ではカーナビゲーション等の車載機器を主に開発している。その過程で発生した不具合のうち、再発防止すべきと判断されたものを特別なインシデント管理システムで管理し、再発防止策と共に記録している。しかし、どのようなテストを行えばその不具合を早期に発見できたのかについては十分検討されているとは言えない状況である。特に、システムテストなどの上位のテストレベルで発見すべき不具合であるほどその傾向が強く、その中には平易な手順で発生するものも含まれていることが経験的・感覚的にわかっている。ただし、定量的には定かではない。

そこで、本研究では、上述のインシデント管理システムに記録されている不具合情報から、それらの不具合を発見するためにどのようなテスト分析/設計技術が必要であったのかを分析する。また、幾つかの開発現場におけるシステムテストレベルのテスト分析/設計プロセスを調査し、その問題点を特定する。更に、それらの現状調査の結果から判明した不足しているテスト技術を補うテスト分析手法を確立する。最後に、アンケートによって提案手法が開発現場に受け入れられるかどうかを評価し、当社製品の1機能を対象とした実験によって従来手法に対する優位性を評価する。本研究のプロセスと論文構成の対応を図1に示す。

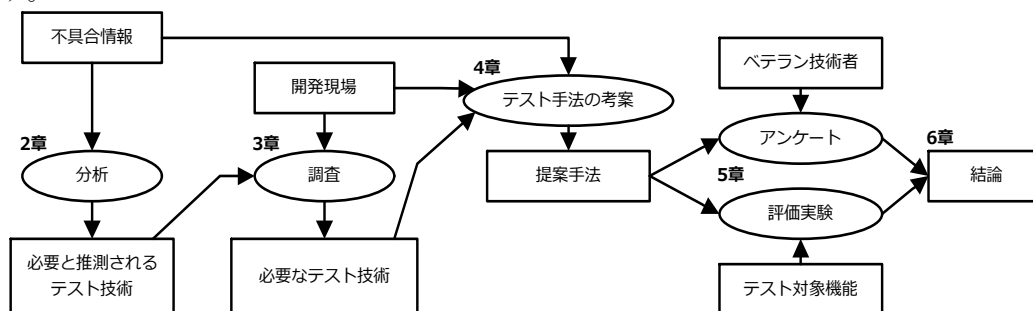


図 1：本研究のプロセスと論文構成の対応

クラリオン株式会社 技術開発本部 コアソフト開発部 ソフト環境グループ
 Clarion Co., Ltd, Software Environment Group, Core Software Development Dept., R&D Division

埼玉県さいたま市中央区新都心 7-2 Tel:048-601-3457 e-mail:masanori_kaneko@clarion.co.jp
 7-2, Shintoshin, Chuoh-ku, Saitama-city, Saitama, Japan

2. 不具合情報の分析

2.1 不具合情報から抽出した技術要素

特別なインシデント管理システムに記録された直近 50 件の不具合を対象に分析を行い、その不具合を発見するために必要なテスト技術要素を抽出・分類する。分類の種別は ISTQB シラバス[1]と先行研究[3]を参考にし、少なくとも { テストレベル, テストタイプ, テスト設計技法 } という技術要素があることがわかった。

直近 50 件の発生年は 2014 年から 2008 年である。それ以前の不具合情報は詳細が記録されていないため、分析対象から除外した。また、内 48 件は当社の主力製品群において発生したものである。このため、当社におけるテストの弱点を発見するには、これら 50 件の分析で十分と考えた。

表 1: テストレベル別件数

表 1 は、各不具合がどのテストレベルで発見可能かを分析した結果である。テストレベルの名称は ISTQB シラバスに習う。不具合情報からはシステムテストと受け入れテストの判別は不可能なのでシステムテストとしてまとめる。複数のテストレベルで発見可能と考えられる不具合はそれぞれで数える。このため、総計は 50 件とならない。また、動的テストでは検出が困難と判断した不具合もここにまとめる。

テストレベル	件数
システムテスト	30
統合テスト	12
コンポーネントテスト	10
動的テストでは検出困難	6

表 2: テストタイプ別件数

表 2 は、各不具合がどのテストタイプで発見可能かを分析した結果である。従来、自組織のテスト標準にはテストタイプという概念が存在しなかったため、ISO/IEC25010 で定義されている品質特性に基づいた名称としている。また、互換性、セキュリティ、保守性、移植性に関しては、該当する不具合が無かったために記載していない。

テストタイプ	件数
機能適合性テスト	27
性能効率性テスト	10
信頼性テスト	5
使用性テスト	2

表 3: テスト設計技法別件数

表 3 は、各不具合を発見するために必要なテスト設計技法の分析結果である。この分析では「基本的な技法で検出可能か？」に着目し、5つの技法に限定している。これらの技法は主に機能適合性テストにおいて適用可とみなし、他のテストタイプについては分析を行っていない。

テスト設計技法	件数
同値分割法	9
境界値分析法	3
デシジョンテーブルテスト	2
状態遷移テスト	5
制御フローテスト	2

また、件数の数え方について補足する。例えば、境界値分析法は同値分割法の使用を前提とするが、両方ともに数えることはしない。同値分割はできていたが境界値を使用してテストできていなかった場合と、同値クラスを抽出できていなかった場合を区別する。他の技法についても同様である。

2.2 不具合誘発条件

不具合分析の結果、テストレベル、テストタイプ、テスト設計技法、のいずれにも該当しない要素が多数存在することがわかった。それらの例を以下に示す。

{ 状態遷移完了直後, 状態遷移完了直前, 前の状態に戻るイベント, 同一機能の繰り返し実行, フォーマット違反, 似ているデータ, 重複したデータ, 矛盾した複数の入力, 単一リソースの同時利用, ファイルシステムの破損, 通信路の切断, 通信相手のフリーズ, 部分的な再起動, シーケンスの逆転, シーケンスの欠落, CPU 使用率の高い機能, メモリ消費量の多い機能, ボタンの連打, 複数の入力デバイスによる同時操作, 不安定な電源, 特殊なエンドユーザー }

これらの要素は、ISTQB シラバスにおけるテスト条件や、西が提唱するテスト観点[5]に類似するが、それらよりも狭い集合に属する要素であると考えられる。むしろ、エラー推測[2][4]において着目すべき要素であり、不具合が起りやすい条件であると考えられる。よって、本研究では、これらの要素を「不具合誘発条件」と分類する。

表 4：テストタイプ別の不具合誘発条件考慮必要有無

そして、不具合発見のために不具合誘発条件の考慮が必要と考えられるものとそうでないものをテストタイプ別に分けると表 4 のようになる。このことから、機能のテストで検出可能な不具合は、不具合誘発条件の考慮が必要である不具合とそうでない不具合に分かれることが考えられる。逆に、非機能のテストで検出可能な不具合は、その多くが不具合誘発条件の考慮が必要であると考えられる。

テストタイプ	不具合誘発条件の考慮が必要	不具合誘発条件の考慮が不必要
機能適合性テスト	9	18
性能効率性テスト	9	1
信頼性テスト	4	1
使用性テスト	2	0

2.3 推測される問題

不具合の分析結果から、主にシステムテストレベルにおいて次の問題があることが考えられる。

- 非機能のテストタイプを選択できていない、または定義できていない。
- 不具合誘発条件を考えるプロセスに問題がある。またはプロセス自体が存在しない。
- 機能適合性テストにおいて、基本的なテスト設計技法を適用できていない。

3. 開発現場の実態調査

本章では、不具合分析の結果を踏まえ、自組織の 3 つの開発現場 A, B, C におけるシステムテストレベルの活動を調査し、補強すべきと考えられるテスト技術を求める。

3.1 システムテストに関する文書の調査

各開発現場について、システムテストに関する文書として何が存在するかを調査した。

表 5：システムテストに関する文書の有無（○：存在する X：存在しない）

文書名	A	B	C
要求仕様書などの明文化されたテストベース	○	○	○
テスト計画書	○	○	○
テスト分析・テスト設計に関する文書	X	X	X
テストケース	○	○	○

いずれの開発現場においても、テスト計画書には幾つかの非機能テストタイプに相当する記載があるが、それらの実施が必要な理由は記載されておらず、テスト計画書のレビュー資料にも議論された形跡が無い。また、テストベースには機能とパフォーマンス以外に関する仕様がほとんど書かれていない。そして、テストケースにはテストベースの文面に類似した記述が多く、表形式で書かれている。これらの事実から以下の内容が推測される。

- 推測 1：テストベースの内容を転記するようなテスト実装プロセスがある。
- 推測 2：品質特性に基づいた非機能のテストタイプの定義・選択プロセスに問題がある。
- 推測 3：テスト分析プロセスが存在しない。
- 推測 4：テスト設計プロセスにおいて、基本的なテスト設計技法が使用されておらず、その過程も文書として残されていない。

3.2 インタビュー

文書を調査するだけでは上記の推測が正しいかどうかを判断できないため、3 つの開発現場 A, B, C について、システムテストを指揮する立場のメンバー(テスト計画立案者)に対してインタビューを行った。内容は以下の設問 Q1～ Q4 の通りであり、3.1 の推測 1～4 にそれぞれ対応する。

- Q1：要求仕様書に書かれていない内容をテストすることの必要性を理解しているか？
- Q2：ISO/IEC25010 等の品質特性に基づいたテストの種類を定義し、行っているか？
- Q3：不具合が発生しやすい条件を考え、文書に残しているか？
- Q4：基本的なテスト設計技法を理解し、必要に応じた選択ができているか？

表 6：現場 A, B, C の回答

各現場の回答を[はい/いいえ]としてまとめたものを表 6 に示す。

	現場 A	現場 B	現場 C
Q1	いいえ	いいえ	はい
Q2	はい	いいえ	いいえ
Q3	いいえ	いいえ	いいえ
Q4	はい	いいえ	いいえ

3.3 課題

不具合分析結果、開発現場の実体調査結果を踏まえると、自組織のシステムテストレベルのテスト分析・テスト設計プロセスには次のような課題があり、改善が必要であると考えられる。

- 課題 1：品質特性に関するテストタイプの定義が不十分である。
- 課題 2：テスト分析プロセスが存在せず、不具合誘発条件を導出していない。
- 課題 3：基本的なテスト設計技法の活用が不十分である。

4 提案手法

本章では、3.3 で示した課題 1, 2, 3 を解決するテスト分析手法を提案する。

4.2 標準テストタイプの定義と選択

ISO/IEC25010 の品質特性と ISTQB のテストタイプ [1][2]を踏まえ、3 章の各開発現場における被インタビュー者と協議し、自組織における標準テストタイプを次のように定義する。

表 7：自組織における標準テストタイプ

テストタイプ名	確認事項の概要
機能テスト	機能要件に定められた通りに機能するか？
パフォーマンステスト	各機能の性能が規定範囲内であるか？
ストレステスト	ストレス環境下においてどの程度の機能と性能を求めるか？
ロングランテスト	どれだけ連続稼働できるか？
フォールトインジェクションテスト	異常発生時にできるだけ機能を提供できるか？ 異常から復帰できるか？
異常原因解析テスト	異常発生時に原因究明に必要な情報を残しているか？
互換性テスト	旧仕様や旧フォーマット等との互換性があるか？
ユーザビリティテスト	エンドユーザーから見て使いやすいか？
セキュリティテスト	秘匿情報が書き換えられたり漏洩したりしないか？

表 8：テストタイプの要否判断

テスト分析プロセスにおいて、これらの標準テストタイプのうち、どれが必要かを選択し、要否の判断理由を記載する。これは課題 1 の解決に寄与する。

テストタイプ名	要否	理由
機能テスト	○	………のため
(中略)		
セキュリティテスト	×	………のため

4.3 不具合誘発条件とテストファクター

次に不具合誘発条件を導出する手法を考える。2.2 に記載した例から考えると、これらをテストベースから直接導き出すことは難しい。そこでまず、先行手法である HAYST 法[6]におけるラルフチャートを参考に、2 章で導出した不具合誘発条件にどのような種類があるのかを分類することにした。結果、入力・内部状態・外部状態に関するものに分類できることがわかった。

しかし、不具合誘発条件の中にはそれらに当てはまらないものが複数存在する。例えば 2.2 で示した例の中では、以下のものが入力・内部状態・外部状態には当てはまらない。

{同一機能の繰り返し実行, シーケンスの逆転, シーケンスの欠落, ボタンの連打, 複数の入力デバイスによる同時操作, 特殊なエンドユーザー }

特殊なエンドユーザー以外のものは入力に関するものではあるが、入力そのものではない。む

しろ、入力の仕事に関するものと考えるのが妥当である。したがって、不具合誘発条件の分類は、入力・内部状態・外部状態・入力の仕事・ユーザーの5種類であるとする。

表 9：不具合誘発条件の分類結果

実際に2章の不具合分析結果から導かれた不具合誘発条件をこれらの5種に分類にすると表9のようになり、5種の分類に当てはまらないものは無かった。

本研究では、HAYST法における因子をテストファクターと呼び、5種それぞれを入力ファクター・内部状態ファクター・外部状態ファクター・アプローチファクター(入力の仕事)、ユーザーファクターと定める。

不具合誘発条件の分類	件数
入力に関するもの	10
内部状態に関するもの	12
外部状態に関するもの	3
入力の仕事に関するもの	5
ユーザーに関するもの	2

HAYST法におけるFL表に習うと、各テストファクターにはそれを具体化したバリエーションが存在する。それらのバリエーションの中から不具合誘発条件を選出し、テスト設計書に記述することで、効率的に不具合を検出できるようになると考える。テストファクターとそれらのバリエーションの洗い出しはFL表の作成過程と同様とし、アプローチファクターとユーザーファクターも同様に行う。アプローチファクターとユーザーファクターの記述例を表10表11に示す。

表 10：アプローチファクターの例

入力ファクター	アプローチファクター	バリエーション
ボタン	操作スピード	ゆっくり、ふつう、いそいで
	同時押しボタン数	1, 2, 3 以上
ストリームデータ	時間あたりの送信データ量	少しずつ、大量に

表 11：ユーザーファクターの例

ユーザーファクター	バリエーション
ユーザー	一般ユーザー、生産ライン作業員、タクシードライバー
車に乗る目的	通勤、買い物、旅行
音楽を聞く習慣	なし、まれに、ときどき、いつも
よく使用する機能	音声認識、自宅へ戻る、Bluetooth オーディオ

4.4 テスト設計書の書式

課題2,3を解決するため、テスト分析プロセスにおける成果物の書式を考える。3章の調査結果を踏まえ、この成果物は表形式とし、従来のテストプロセスから移行しやすいことを狙う。尚、この成果物はテスト分析の結果であるが、我々の組織においてはテスト設計という用語が十分浸透していないため、テスト設計書と呼ぶことにする。

まず、従来から行っている機能テストとパフォーマンステストから考える。テストベースとのトレーサビリティをとるため、参照しているテストベースを記載する欄と、それに対する確認内容を記載する欄を設ける。基本的に、確認内容欄はテストベースに書かれた仕様単位で書くが、テストベースを読む過程において、テストベースに本来記載されているべきことが抜けていることがテスト設計書作成者の経験に基づいて発見できることが考えられる。その場合、テストベースの欄は空欄として確認内容欄にそれを記載する。

そして、その確認内容が満たされない可能性が高いと考えられる条件、すなわち不具合誘発条件を記載する欄を設ける。機能テスト・パフォーマンステストに関わるテストファクターは、ユーザーファクター以外の4種であると考え、それぞれ4列を設ける。

更に、テストベースの記載・確認内容・不具合誘発条件のセットに対して適用すべきテスト設計技法を選択する欄を設ける。テスト設計技法の欄は{状態遷移テスト、デシジョンテーブルテスト、制御フローテスト}の3択とし、課題3の解決に寄与する。テストベースに状態遷移モデル・デシジョンテーブル・制御フローモデルの図が記載されている場合にこれらの技法を適用する。記載されていない場合であっても、テストベースの文面からこれらのモデルが読み取れる場合はこれらの技法を適用する。同値分割法と境界値分析法についてはFL表の作成と同様に、テ

5 評価と考察

本章では、4章で提案したテスト分析手法を評価し、その結果を考察する。

5.1 アンケートによる評価

提案手法が受け入れられるかどうかを調査目的とし、開発現場 A, B, C の被インタビュー者 3 名を含むソフトウェア開発経験 9 年～23 年のベテラン技術者 8 名に対してアンケートを行った。設問は、八木らが提案するマフィアオファー手法[7]における”抵抗の 6 階層”を参考に作成した。以下 7 つの設問を設け、回答は「1. 非常にそう思う」「2. そう思う」「3. そう思わない」「4. 全くそう思わない」の 4 択とした。

- Q1：簡単な手順であっても発生する重大な不具合が見逃されていると思いますか？
- Q2：重大な不具合を見逃さないためにテスト分析/設計手法が必要だと思いませんか？
- Q3：テスト設計書を作成することで、見逃していた不具合を発見できると思いませんか？
- Q4：あなたの開発組織には、テスト設計書の作成能力があると思いませんか？
- Q5：テストファクターから不具合誘発条件を導出しやすくなると思いませんか？
- Q6：あなたの開発組織では、テスト設計書を作成するための工数があると思いませんか？
- Q7：テストケースよりもテスト設計書をレビューする方が、テストの抜け・漏れの発見において効率的であると思いませんか？

表 15：アンケート結果

アンケート結果(表 15)からは、提案手法が解決しようとする問題の存在、解決方法の方向性について理解が得られており、問題解決の期待もされていることが読み取れる。しかし、実践のための能力と工数が不足していると考えられている。これらのことから、抵抗の 6 階層のうち 3 階層目まではクリアできていると考える。

	a	b	c	d	e	f	g	h
Q1	2	3	2	2	2	2	3	1
Q2	2	2	1	1	1	2	1	1
Q3	2	2	2	2	2	2	1	2
Q4	3	4	2	3	3	2	3	3
Q5	2	2	2	2	2	2	2	2
Q6	4	4	3	3	4	3	3	4
Q7	2	2	2	2	2	2	2	2

5.2 カーナビゲーションシステムの 1 機能に対する評価実験

システムテストが完了しているカーナビゲーションシステムの音楽再生機能に対して提案手法を適用した。この機能は音楽ファイルを再生する機能であり、タッチパネルやリモコン等のデバイスを使用して再生・停止・曲送りなどの操作することができるという特徴がある。この機能に関するソフトウェア要求仕様書(A4 用紙換算 82 ページ)をテストベースとしてテスト設計書を作成し、テストケースの作成、テストの実施を行った。また、既存のテストケースとテスト設計書の確認内容を比較した。本評価の実施者は 1 名であり、組込みソフトウェア開発 7 年、組込みシステムテスト 1 年の経験があり、JSTQB Foundation Level を保有している。以下に結果と考察を示す。※注：我々の組織では、実機を用いたブラックボックステストをシステムテストと呼ぶ。

(1) 結果

作成したテスト設計書の確認内容項目数(表 12 における行数)76 件のうち、既存のテストケース群に同等に存在するものは 54 件であった。残りの 22 件についてテストケースが不足している理由の内訳を表 16 に示す。

表 16：テストケースの不足理由

また、従来は開発組織内で発見できなかった 8 件の機能的不具合のうち、5 件を発見することができた。5 件のうち 2 件は 0 スイッチカバレッジの状態遷移テスト、1 件は 0 スイッチカバレッジの状態遷移テストにアプローチファクターを加えたテストケース、1 件は境界値分析、1 件はアプローチファクターにより検出ができた。他 3 件を検出

テストケースの不足理由	件数
入力ファクターの不足(同値分割・境界値分析の不足)	11
アプローチファクターの不足	5
状態遷移テストの設計不足	3
デシジョンテーブルテストの設計不足	1
内部状態ファクターの不足	1
テストベースに記載のある項目が漏れている	1

できなかった理由をそれぞれ次に示す。

- 発生に関連するパラメータがテストベースからは読み取れなかったため
- 1 スイッチカバレッジの状態遷移テストを行っていないなかったため
- 発生的前提条件と手順がきわめて特殊であったため

更に、2 件のソフトウェア要求仕様書とは異なる動作を発見できた。いずれもテストベースでは明示されていない無効同値クラスの入力から導かれたテストケースによる発見である。

(2) 考察

0 スイッチカバレッジの状態遷移テストで発見できた 2 件については、提案手法において、どの確認内容に対して状態遷移テストを適用するかを検討したことの成果と考える。境界値分析で発見できた 1 件、ソフトウェア要求仕様書と異なる動作 2 件については、提案手法における入力ファクターの FL 表作成過程において、無効同値クラスと境界値を発見したこと、および、そのパラメータを不具合誘発条件としてどの確認内容で用いるかを検討した成果と考える。

一方、アプローチファクターによって発見できた不具合 2 件については、評価実験実施者の経験に基づいてテストケースを導いたため、提案手法による成果とは言えない。しかし、それらのアプローチファクターが不具合発見に寄与したことは事実であるため、テスト設計書の作成プロセス自体は有用であったと考える。

6 おわりに

6.1 まとめ

本研究では、不具合の分析と開発現場の実態調査を行うことにより、テスト分析/設計プロセスの強化が課題であることを見出した。また、その課題を解決する手法を提案した。そして、ベテラン技術者へのアンケートにより、提案手法が開発現場の期待に沿うものであることを確かめた。最後に、提案手法を製品の 1 機能に適用し、その有効性を確かめた。

6.2 今後の課題と展望

(1) 提案手法の自組織への普及と効果の更なる立証について

まず、アンケートでネガティブな回答のあった能力と工数について理解を得ることが最優先の課題であり、テスト設計書の作成工数の調査、テスト設計に関するレビューの効率化に関する調査、演習形式の教育などを行う必要があると考える。また、より多くのシステムや機能を対象とした評価実験や、機能テスト以外のテストタイプに関する評価実験も必要と考える。

(2) 提案手法の発展について

提案手法はテスト設計技法ではない。このため、不具合誘発条件を活用したテスト設計技法を確立することが重要な課題と考える。また、本研究で思考過程と書式を確立することのできなかったユーザビリティテスト・セキュリティテスト・互換性テストについて提案手法に取り入れることも課題である。更に、本研究で行った不具合の分析と開発現場の実態調査を継続的に行うことは、提案手法の洗練とテスト分析/設計プロセスの改善のために重要であると考え。ほか、ガイドワードを用いた方法[8]で不具合誘発条件を導出することも今後の展望として考えられる。

参考文献

- [1] ISTQB テスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2011.J02
- [2] 大西健児 他, ソフトウェアテスト教科書 JSTQB Foundation 第 3 版, 翔泳社, 2011
- [3] 清水剛史 他, バグの流出防止を考える, ソフトウェア品質管理研究会 第 5 分科会, 2012
- [4] 情報処理推進機構, 高信頼化ソフトウェアのための開発手法ガイドブック, 2012, pp.147-171
<http://www.ipa.go.jp/sec/publish/tn10-005.html>
- [5] 西康晴, テスト観点に基づくテスト開発方法論 VSTeP の概要, 2013, pp.12
<http://qualab.jp/materials/VSTeP.130510.color.pdf>
- [6] 秋山浩一, 事例とツールで学ぶ HAYST 法, 日科技連出版社, 2014
- [7] 八木将計 他, XDDP のマフィアオファー, SWEST15, 2013
- [8] 河野哲也, ソフトウェア要求仕様における HAZOP を応用したリスク項目設計法, ソフトウェアテストシンポジウム 2012 Tokyo