5B-4

効率的な機能テスト設計のための欠陥情報の活用方法

湯本剛† 松尾谷徹‡ 津田和彦††

日本ヒューレット・パッカード株式会社† デバッグ工学研究所‡ 筑波大学††

1. はじめに

ソフトウェアの品質を確保する手段として，開発ライフサイクルのさまざまな段階でソフトウェアテスト（以降「テスト」と記す）を行う．各々のテストは，テスト対象の欠陥を発見することが目的となることが多い．

要件・設計と合致しない欠陥は，要件・設計文書の記載内容を基にしたテストケースを実行して発見することが出来る．けれども，要件・設計文書に記載がない事柄を発見するためには，何らかの方法を用いて欠落した情報を補完する必要がある．典型的な補完の方法として，過去の類似ソフトウェアの欠陥を分析した結果が挙げられる．しかし，欠陥の分析結果の多くは，開発の問題箇所特定を目的としている［1］．

その分析結果をテストケースを設計する情報として使用することは，分析の目的が違うため，それだけでは困難である．そこで本論文では，テスト対象ソフトウェアと類似するソフトウェアの欠陥情報を，テストケース設計のための補完情報として活用する手法を提案する．

2. テストケースの構成要素

テストケースは，IEEE610において，特定の目的のために開発されたテスト入力，実行条件，期待結果の３つで構成されると定義している[2]．また，機能テストは，選択した入力と実行条件のレスポンスとして生成されたアウトプットを確認する，と定義している[2]．すなわち，テストとは，テスト入力，実行条件，を入力して生成されたアウトプットが期待結果と一致するかを確認することである．このプロセスを図示すると，図1のようになる．

テスト入力，実行条件には,事前に設定されているものと，実行時点で設定するものがある．本論文では，おのおのを表1に示すよう定義する．その上で，事前入力と事前条件をまとめたものをパラメータ，イベントと操作をまとめたものをアクションと呼ぶこととする．

表1 テストの構成要素の再分類

	事前に設定（パラメータ）	実行時点で設定（アクション）
テスト入力	事前入力	イベント
実行条件	事前状態	操作

アクションはテスト対象を動作させてアウトプットを導く直接的な要因であるので1つに特定できるのに対し，パラメータは多くのバリエーションが想定できる．パラメータの選定と選定したパラメータの数，その組合せ方により，テストケースの数や内容に大きな差異がでるため，有効なパラメータを選択するための補完情報をどう得るかが重要になる．

3. テスト設計のための欠陥情報の知識化

欠陥のを分析結果をテストケースの設計に活用するためには，テスト実行の際に欠陥を識別するプロセスを理解し，そのプロセスのインプットに合わせたフォーマットで分類すると情報の活用が容易となる．

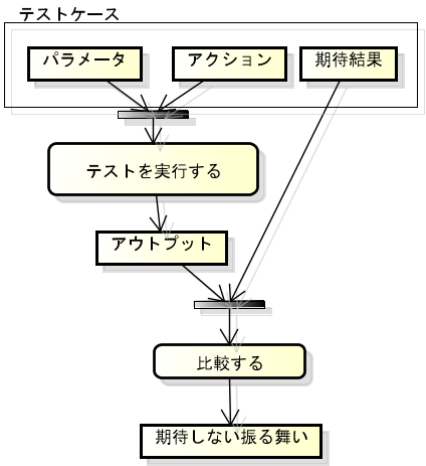

図1 テスト実行の際に欠陥を識別するプロセス

テスト実行の際は，パラメータとアクションをインプットにし，アウトプットと期待結果の比較を行い，欠陥の識別をしている．ただし，個々で識別しているものは欠陥そのものではなく，欠陥の結果として引き起こされた期待しないソフトウェアの振る舞いである．欠陥情報の知識化とは欠陥情報を図1に掲載したパラメータ，アクション，期待結果，アウトプット，期待しない振る舞いというフォーマットに分類できるよう変換することとなる．

A Practical Using Method for Efficient Design of Functional Testing.

†Tsuyotshi Yumoto, Hewlett-Packard Japan, Ltd.
‡Tohru Matsuodani, Debug engineering research laboratory.
††Kazuhiko Tsuda, University of Tsukuba.

4. 欠陥情報知識化/活用のフロー

欠陥情報知識化までの流れを図2に示す.

1) 初期情報（欠陥の現象と発生原因）の整理
2) テストケース要素を基にした分類
3) パラメータ汎化情報の付与
4) テスト設計（欠陥知識化ＤＢの活用）

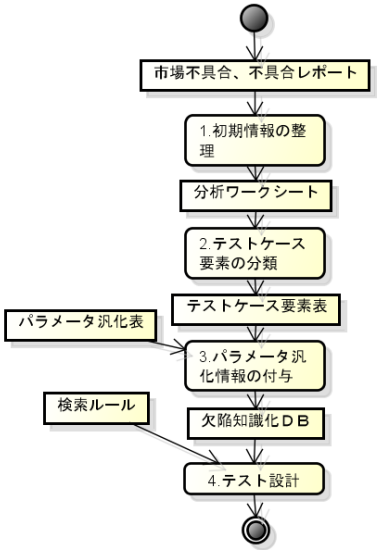

図2 欠陥情報知識化までの流れ

4.1 初期情報の整理

開発時に検出した不具合や，市場にて発覚した不具合のレポートの文言から以降のテストケース要素の分類に有用なキーワードを抽出する．また，記述内容をISTQBの定義に基づき，ソフトウェアの期待しない振る舞いである故障，ソフトウエアの間違いである欠陥，欠陥を引き起こした原因であるエラーに分類する[3]．これらの結果は分析ワークシートに一旦蓄積する．

4.2 テストケース要素を基にした分類

4.1で抽出したキーワードを，パラメータ，アクション，期待結果，というフォーマットに整理してテストケース要素表を作成する．例として，条件によりメール自動受信が出来なくなるという不具合をテストケース要素に整理したものを表2に示す．表2では1つの現象に対してパラメータが2種類あると分析している．

表2 テストケース要素表の一事例

パラメータ	値	アクション	期待結果
メモリ残量	10ＭＢ	メール自動受信	応答成功
バックグランド処理	削除優先順位1位	メール自動受信	応答成功

また，表2に整理したテストケースを実行することにより，発生したアウトプットを故障と欠陥に整理してテストケース要素表の同列に記載する．

表3 テストケース要素表の一事例（続き）

故障	欠陥
メールの自動受信が出来ない	memory kill仕様の対応漏れ
メールの自動受信が出来ない	memory kill仕様の対応漏れ

表3の例では，1つの現象を2行記載しているが，それは表2にてパラメータが2種類あると分析しているためである．

4.3 パラメータ汎化情報の付与

テストケース要素表の1行毎にパラメータを汎化した情報を付与する．他の類似ソフトウェアに対するテストケースの設計時にどのようなパラメータを考慮すべきかを展開できるようするためである.表1のパラメータの例で言えば，メモリ残量の汎化は「容量」，バックグランド処理の汎化は「順番」といったものが考えられる．これらは，パラメータ汎化表としてまとめていく．属性を付与が出来たデータを蓄積し，欠陥知識化DBとして以後のテスト設計に活用する．

4.4 テスト設計

テスト設計する際に，パラメータの過不足を認識するために欠陥知識化DBを活用する．そのための検索ルールとしては，例えば，表2で例に挙げた自動受信する，というアクションと類似のアクションがある機能ではどのようなパラメータを考慮したほうが良いかを欠陥知識化DBを使い検討する，といった活用方法があげられる．

5. おわりに

本提案では，欠陥情報をテスト設計に活用するための手法として，フォーマットとフローの定義を提案した．フォーマット定義にて,テストケースの構成要素であるテスト入力と実行条件を,パラメータとアクションに再分類したことでテスト設計への活用が容易になった． 今後，パラメータの汎化キーワードの作成ルール，知識化DBからテスト設計へ活用する際の検索ルールを明確にすることで，より有用な手法としていきたい．

参考文献

[1] B.バイザー；“ソフトウェアテスト技法”, 日経BP,1994
[2] “Standard glossary of software engineering terminology”, IEEE610.12.1990
[3] “ISTQBテスト技術者資格制度Foundation Levelシラバス日本語版”, Version2011.J02