Vol.2022-SE-210 No.33
2022/3/12

過去のテスト目的を利用した仕様レビュー手法

谷﨑浩一[1]　田上諭[1]　蛭田恭章[1]　森龍二[1]　森崎修司[2]

概要：ソフトウェア開発の上流工程で仕様のレビューを行い，修正工数が大きい欠陥や深刻な欠陥を検出し，修正することで開発効率を高められる．レビューでのチェック箇所とチェック方法を明確化し欠陥を検出しやすくするために，テストケースをレビューに活用する方法が提案されているが，テストケースは膨大な数になりすべてをレビューでチェックするのは現実的でないという課題がある．本研究では，テストケース作成の前段階の活動の成果物であり，テストケースよりも抽象度の高い情報であるテスト目的に着目し，過去に作成したテスト目的を利用した仕様レビュー手法を提案する．ケーススタディによって，テスト目的を用いて仕様レビューを行うことで実際に欠陥を検出可能であることを確認した．

キーワード：フロントローディング，ソフトウェアテスト，テスト設計，品質保証

1. はじめに

開発の上流工程で要求仕様のレビューを行い，修正工数が大きい欠陥や深刻な欠陥を検出することは重要である[1][2][3]．修正工数が大きい欠陥を上流工程での仕様レビューで検出するために，欠陥情報の分析から業務観点でのレビューで検出すべき欠陥を特定する方法[4]や，過去の欠陥の修正工数を考慮するレビュー手法[5][6]が提案されている．また，欠陥の未然防止につなげるための手法も提案されている[7][8]．しかし，これらの方法は，欠陥情報に基づいて分析を行うため，過去に検出された欠陥以外は対象にできないという課題がある．

この課題に対して，テストケースをレビューのチェック箇所とチェック方法に使う方法が考えられる．テストケースは，欠陥の有無にかかわらず，製品全体に対してテストすべき項目が集積されているため，過去に欠陥が存在していた部分以外もチェックできる．テストケースをベースにした要求仕様のレビュー手法が提案されている[9]が，レビューの前にテストケースを準備する必要がある．すべてのテストケースをレビューで用いることはレビューの実施に多くの時間を要する上に事前にテストを準備することは簡単ではない。

一般的にテストケースは膨大な件数になるため，1 件ずつすべてをレビューで確認するのは現実的ではない．そこで，本研究では，テスト設計においてテストケースの前段階の成果物として作成される，テスト目的に着目する．テスト目的はテストレベルと検出したい欠陥からなり[10]，その欠陥を見つけられるようなキーワードとして記述される．テスト目的はテストケースよりも抽象度の高い情報だが，テストで確認すべき箇所と確認すべき内容が指定されているため，レビューのチェック箇所とチェック方法として利用することが可能と考えられる．しかし，実際に試した研究はない．

本論文では，具体的なテストケースを想定できるレベルで詳細化したテスト目的を用いて，仕様レビューを行う手法を提案する．提案手法の評価のためケーススタディを行い，テスト目的を用いた仕様レビューによって欠陥の検出が可能かどうかを評価する．以降，2 で提案手法を説明し，3 で商用ソフトウェア製品の開発プロジェクトを対象としたケーススタディを示す．4 でケーススタディの結果に関する考察を述べ，5 でまとめる．

2. 提案手法

2.1 前提

テストで検出・修正された場合の修正工数の大きい欠陥を，仕様レビューで検出できるようなテスト目的を選択するために，テスト目的から作成されうるテストケースのテスト実施にかかる工数に着目する．テスト目的を具体的なテストケースを想定できるレベルで詳細化すると，テストケースの実施にかかるおおよそのテスト実施の工数を想定できることから，テスト目的に対しておおよそのテスト実施の工数の情報をあらかじめ付与することは可能である．テスト実施にかかる工数が大きい場合，そのテスト目的に関連する欠陥がテスト工程で検出されると，再テストに多くの工数がかかることから，当該欠陥は修正工数が大きい欠陥と考えられる．そのような欠陥を仕様レビューで検出するために，本論文で提案するレビュー手法では，テストケースのテスト実施にかかる工数が大きいテスト目的を選択し，レビューのチェック箇所とチェック方法として使用する．

テスト目的を記述する方法としては，テスト目的を漏れなく洗い出すことができるように，カテゴリ分けして階層構造で記述する方法がある[11][12]．この方法では，上位の階層には抽象度の高いテスト目的を記述し，下位の階層にはより具体的なテスト目的を記述する．末端階層のテスト目的は具体的なテストケースを想定できるレベルまで詳細

1 (株)ベリサーブ
VeriServe Corporation
2 名古屋大学大学院情報学研究科
Nagoya University Graduate School of Informatics

化して記述する．たとえば，ウォータフォール開発における統合テストフェーズでのテスト目的として，最上位の階層に機能テスト，性能テスト，互換性テストといったテストの種類を挙げ，機能テストの下位の階層には「入力データの入力範囲の確認」や「出力データの初期化タイミングの確認」，性能テストの下位の階層には「ソフトウェアの処理時間」，互換性テストの下位の階層には「データ種別（画像や動画）ごとの確認」や「表示先種別（PC や携帯電話）ごとの確認」といった具体的なテストの内容を挙げることができる．

2.2 手順

図 1 に提案手法による仕様レビューのプロセスを示す．提案手法による仕様レビューでは，図 1 の手順 1 および手順 2 においてレビューで使うテスト目的の選択を行い，手順 3 においてテスト目的を使ってレビューを行うことにより欠陥を検出する．テスト目的は過去の開発プロジェクトで作成済みのものを流用する．リポジトリ T には過去のプロジェクト r で作成したテスト目的の集合 T_r が蓄積されているものとする．リポジトリ T には m 件のテスト目的の集合が含まれる．つまり，$T=\{T_1, T_2, \cdots, T_m\}$である．テスト目的の集合 T_m には，n 件のテスト目的 t が含まれる．つまり $T_m=\{t_{m1}, t_{m2}, \cdots, t_{mn}\}$である．

本論文で提案するレビュー手法の具体的な手順は以下の通りである．

手順 1: テスト目的の集合 T_i の選択

レビューアはリポジトリ T から，今回レビューする対象の製品 i と類似する製品のテスト目的 T_i を選択する．

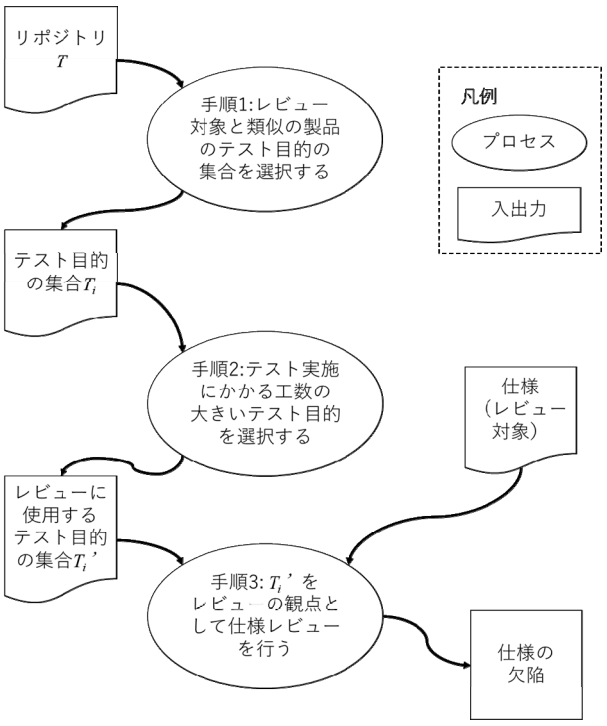

図 1　提案手法による仕様レビューのプロセス

表 1　テスト目的とテスト実施にかかる工数の例

テスト目的	テスト実施工数（person hours）
t_{31}	3
t_{32}	6
t_{33}	8
t_{34}	2
t_{35}	5

手順 2: テスト工数が大きなテスト目的の選択

レビューアは T_i に含まれるテスト目的のなかから，テスト実施にかかる工数が大きくなる t_{ij} を選択する．このとき，テスト目的 t_{ij} はテスト実施にかかる工数 e_{ij} の情報を持ち，それぞれ(t_{i1}, e_{i1}), (t_{i2}, e_{i2}), ・・・とすると，レビューアが t_{ij} を選択する際は閾値 e_{ip} を設定して e_{iv}, e_{iw}, e_{ix}, e_{iy}, ・・・の順に t_{ij} を選択する．ただし、$e_{iv} => e_{iw} => e_{ix} => e_{iy} => \cdots => e_{ip}$ である．選択した t_{ij} の集合を T_i ’とする．T_i ’$\subseteq T_i$ である．

手順 3: 選択したテスト目的を利用したレビュー

レビューアは T_i ’に含まれるテスト目的 t_{ij} をレビューのチェック箇所とチェック方法として，レビュー対象の仕様を読み，欠陥を検出する．

2.3 例

例として，製品の種別やバージョンごとに作成したテスト目的の集合がリポジトリ T に蓄積されているケースを考える．製品 A のバージョン 1 のテスト目的の集合 T_1，製品 A のバージョン 2 のテスト目的の集合 T_2, 製品 B のバージョン 1 のテスト目的の集合 T_3 がリポジトリ T に蓄積されている場合，$T=\{T_1, T_2, T_3\}$である．製品 B と類似の製品 C の開発プロジェクトでテスト目的を仕様レビューに流用する場合は, 製品 B のバージョン 1 のテスト目的の集合 T_3 を選択する．T_3 に 5 個のテスト目的が含まれる場合 $T_3=\{t_{31}, t_{32}, t_{33}, t_{34}, t_{35}\}$であり，レビューアは t_{31}～t_{35} の中からテスト実施にかかる工数が大きいものを選択する．このとき，表 1 に示すようにテスト実施にかかる工数の情報がテスト目的に付与されている場合, 閾値 e_{3p} を 4 person hours とすると，テスト実施にかかる工数が 4 person hours 以上で大きい順に，t_{33}, t_{32}, t_{35} をレビューに使用するテスト目的として選択できる．つまり T_3’$=\{t_{33}, t_{32}, t_{35}\}$となる．閾値は対象ソフトウェアや必要であったテスト工数に応じて適切に設定する．

3. ケーススタディ

3.1 目的と方法

提案手法によってテスト目的を用いてレビューすることで，仕様の欠陥を検出できるかをケーススタディによって確認する．

予備調査として，テスト目的を利用した仕様のレビューによって，検出済みの欠陥を仕様レビューの段階で未然に検出できたかどうかを，筆者らが確認する．

実際のレビューの実施による提案手法の評価として，ソ

フトウェアテストの実務経験があり，要求仕様のレビューの実務にも携わっているメンバーに提案手法を利用してもらい，インタビューにより(i) レビューでチェックする項目としてテスト目的を選ぶことと(ii) 選んだテスト目的からレビューすることの二つにわけて意見を得る．

3.2 対象

予備調査では PC アプリケーションの開発プロジェクトにおいて過去に検出済みの欠陥を対象とした．対象のプロジェクトで検出された欠陥のうち，受け入れテスト以降で検出された欠陥を対象とした．

テスト目的は，過去のプロジェクトの知見として作成されたテスト目的の集合が複数蓄積されたリポジトリから，ケーススタディの対象プロジェクトで利用可能なテスト目的の集合を選択して利用した．なお，当該リポジトリに蓄積されたテスト目的には，提案手法の手順で解説したテスト実施にかかる工数 e_{ij} の情報は付与されていなかった．

修正工数の大きい欠陥を提案手法による仕様レビューで検出できるか確認するため，対象プロジェクトの既知の欠陥から合計作業時間が 8h を超える欠陥を抽出し，欠陥の埋め込み要因の工程が実装よりも前であるものを対象とした．ここでは，レビュー実施工数を 4 person hours と想定し，手戻り工数がその 2 倍となる 8 person hours を閾値として選んだ．

対象とした欠陥の，埋め込み要因の工程ごとの件数を表 2 に示す．実装より前の工程で混入した欠陥は，要件定義・設計・性能設計の各工程で埋め込まれたもので，計 21 件だった．これら 21 件の欠陥を提案手法で検出可能かどうか確認した．

レビュー実施による評価では，機能要求が記載された仕様書の一部を対象とした．テスト目的は予備調査と同様に，テスト目的の集合が複数蓄積されたリポジトリから，対象プロジェクトで開発している製品と類似する製品のテスト目的の集合を選択して利用した．ソフトウェアテストの実務経験があり，対象の仕様書のレビューを担当しているエンジニア 3 名に評価に協力していただいた．

3.3 手順

予備調査では，まずテスト目的の集合が蓄積されたリポジトリから，ケーススタディの対象プロジェクトで利用可能なテスト目的の集合を選択した．当該テスト目的の集合に含まれるテスト目的から作成されうるテストケースのテスト実施の工数を考慮して，テスト実施にかかる工数が大きいテスト目的を選択した．既知の欠陥に対して，選択したテスト目的をレビューのチェック箇所とチェック方法として使っていたら，仕様レビューの段階でその欠陥を検出可能だったか確認した．

表 2　対象とした欠陥の混入工程ごとの件数

工程	件数
要件定義	2
設計	15
性能設計	4
実装	17
マニュアル作成	1

レビュー実施による評価では，対象のエンジニア 3 名に対して筆者らが提案手法の手順を説明し，実際のプロジェクトでレビュー済みの仕様書に対して追加で，提案手法でのレビューを試行してもらった．レビューの試行後に，当該エンジニア 3 名と筆者らでオンラインミーティングを実施し，テスト目的を選択することやテスト目的を用いてレビュー実施することが可能だったか，テスト目的を用いてレビューすることが有用と感じたか，といった手法の有効範囲を確かめるためのインタビューを行った．

3.4 結果

予備調査では，テスト目的をレビューのチェック箇所とチェック内容として活用することで，既知の欠陥の一部を検出可能であることを確認できた．

図 1 の手順 1 では，リポジトリから PC アプリケーションに関連するテスト目的の集合を検索するために，「アプリ」という単語でリポジトリ内を検索したところ，図 2 に示す汎用的なアプリケーションに対するテスト目的の集合が見つかった．

図 1 の手順 2 では，このテスト目的の集合の中から，テスト実施にかかる工数の大きいテスト目的を 11 個選択し，レビューに使用するテスト目的とした．実際に選択したテスト目的は，図 2 にチェックマークのついたものである．

各テスト目的を選択した理由は以下のとおりである．

- 例外処理に関するテストは，例外処理のパターンを考慮するとテストケースの件数が多くなる可能性が高い．テストデータの準備に時間がかかり，テスト実行に時間がかかる可能性が高い．
- 入力データの範囲外やデータ管理に関するテストは，範囲外となるデータのパターンや様々なデータのパターンの CRUD を考慮すると，テストケースの件数が多くなる可能性が高い．テストデータの準備に時間がかかり，テスト実行に時間がかかる可能性が高い．
- 出力データのデータ管理に関するテストは，様々なデータのパターンの CRUD を考慮すると，テストケースの件数が多くなる可能性が高い．テストデータの準備に時間がかかり，テスト実行に時間がかかる可能性が高い．
- 競合や排他に関するテストは，複数の機能や処理の組み合わせを考慮すると，テストケースの件数が多くなる可能性が高い．競合や排他を発生させるための手順が複雑になり，テスト実行に時間がかかる可能性が高い．
- 相互運用性に関するテストは，機能と機能の組み合わ

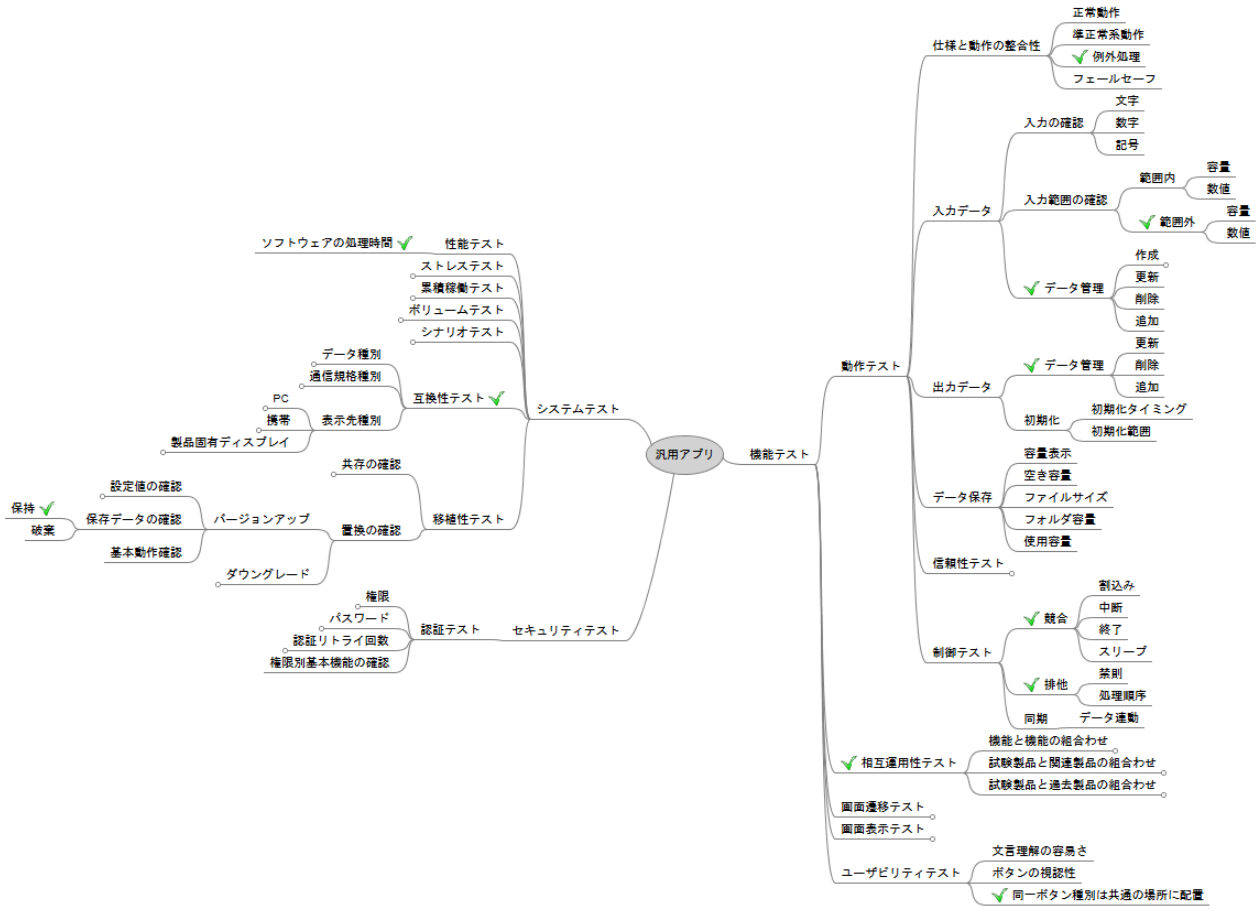

図 2　レビューに使用したテスト目的

せや，テスト対象製品と関連する製品や過去バージョンとの組み合わせを考慮すると，テストケースの件数が多くなる可能性が高い．テスト環境として，テスト対象製品以外の用意が必要となり，テスト実行に時間がかかる可能性が高い．

- 同一ボタン種別の配置に関するテストは，製品の画面などの GUI を全体的に確認する必要があり，テストケースの件数が多くなる可能性が高い．確認する部分が多く，テスト実行に時間がかかる可能性が高い．
- ソフトウェアの処理時間に関するテストは，様々な入出力データのパターンやテスト環境で処理時間を計測する必要があり，テストケースの件数が多くなる可能性が高い．処理時間の計測は複数回実施する場合もあり，テスト実行に時間がかかる可能性が高い．
- 互換性テストは，様々な入出力データのパターンやテスト環境でテストを実施する必要があり，テストケースの件数が多くなる可能性が高い．一通りの機能が動作することを確認するなど，テストの手順が多い場合もあり，テスト実行に時間がかかる可能性が高い．
- バージョンアップ時の保存データの保持に関するテストは，様々なパターンのデータを考慮すると，テストケースの件数が多くなる可能性が高い．テスト環境の用意やバージョンアップの処理自体に時間がかかる場合もあり，テスト実行に時間がかかる可能性が高い．

図 1 の手順 3 の代替として，選択した 11 個のテスト目的を仕様のレビューに用いていた場合に，対象の欠陥をレビューで検出できたかどうか確認した結果を，表 3 に示す．検出の可能性を○・△・×の 3 段階に分類し集計するとともに，該当する欠陥の実際の修正にかかった工数の平均値を算出した．○はテスト目的そのもので欠陥を検出可能だったもの，△はテスト目的を具体化するなどレビューアがテスト目的を応用すれば欠陥を検出可能だったもの，×はテスト目的を用いても検出できなかったものを表す．

対象とした欠陥 21 件のうち 17 件は○と△に該当し，テスト目的を用いたレビューで検出できた可能性があることが分かった．○と△に該当する欠陥について，テスト目的ごとの検出可能な欠陥の件数および平均修正工数を図 3 に示す．○に該当する欠陥を検出できたテスト目的の中でも，

表 3　提案手法により検出可能な欠陥の件数

検出の可能性	件数	平均修正工数（h）
○	7	21.9
△	10	11.2
×	4	10.9

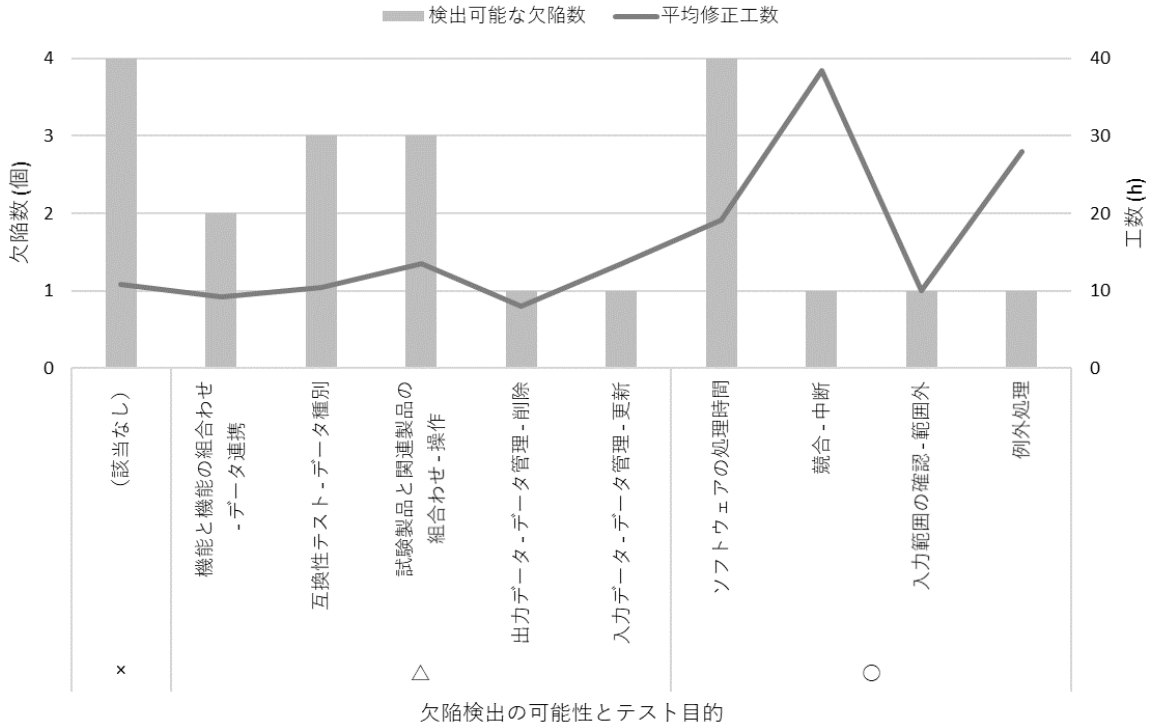

図 3　テスト目的ごとの検出可能な欠陥の件数および平均修正工数

「ソフトウェアの処理時間」「競合-中断」「例外処理」に関する欠陥は平均修正工数が 20h 以上であり，修正にかかるコストが特に大きいものだった．

レビュー実施による評価では，仕様書レビューの業務に携わるエンジニアへのインタビューの結果，以下の意見を得た．

(i) レビューでチェックする項目としてテスト目的を選ぶことに関する意見

- テスト目的を選ぶことについては戸惑わずに実施できた

(ii) 選んだテスト目的を用いてレビューすることに関する意見

- 通常行っているアドホックな仕様書レビューに追加して，提案手法によるレビューをチェックリストのような形で活用できる．
- 通常行っているレビューでは何をチェックしたかはレビューアの頭の中にしかなかったが，提案手法ではチェックする内容が明確になり，重大な欠陥がないことをチェックできることがメリットと感じた．
- 通常行っているレビューでは 5 時間程度かける仕様書に対して，提案手法によるレビューでは 3 時間程度の時間がかかった．追加でその程度の時間がかかるとしても，前述のメリットを考えると，実務で提案手法を活用することは価値がある．
- テスト目的としては必要だがレビューにどのように使ったらよいか分からないものが一部あり，レビューの実施時に戸惑うことがあった．そのようなテスト目的の中にはプロジェクトの暗黙知になっているものもあり，本当に仕様書に記載が不要かどうかを検討することには繋がるかもしれない．

4. 考察

4.1 予備調査

表 3 の○に該当する欠陥について，図 3 からテスト目的そのもので欠陥検出可能だったものは，処理の競合による中断，ソフトウェアの処理時間，例外処理といった，平均修正工数が 20h 以上の修正工数が大きい欠陥だった．テスト目的を仕様レビューに利用し，レビューの時点でこれらのテスト目的に関連する欠陥を検出できれば，テストで検出・修正された場合の修正工数を低減することに効果的と考えられる．

表 3 の△に該当する欠陥をレビューで検出できるかどうかが人に依存する理由として，今回のケーススタディでは汎用的な PC アプリケーション向けのテスト目的を利用したことが理由として挙げられる．対象のアプリケーション向けの固有なテスト目的を用意できれば，△に該当する欠陥をレビューで検出しやすくなると考えられる．今回のケーススタディの対象のアプリケーションは，Excel や Word のデータをインポートする，ブラウザの描画モジュールを利用した画面描画を行う，各機能間で作成したデータを再利用する，といった特徴があった．製品固有のテスト目的として以下のようなテスト目的を用意できれば，仕様のレビューでこれらのテスト目的に関する欠陥を検出しやすく

なると考えられる．

- 入力データで考慮すべき条件に関するテスト目的（文字コードなど）
- Excel や Word やブラウザなどの連携するアプリに関するテスト目的
- 各機能で作成したデータの別機能での再利用に関するテスト目的

4.2 レビュー実施による評価

レビューで使用するテスト目的を選択することについて，「戸惑わずに実施できた」との意見を得たことから，今回の評価対象においてはテスト目的の選択は問題なく実施できたと言える．評価を実施したエンジニア 3 名はレビュー対象の製品に対するソフトウェアテストの実務経験があったため，戸惑わずにテスト目的を選択できたと考えられる．ソフトウェアテストの実務経験の浅いエンジニアや，レビュー対象の仕様に詳しくないエンジニアの場合にはテスト目的の選択に戸惑う可能性がある．そのような場合は，テスト実施の工数のデータに基づいて，レビューで使うテスト目的を選択することが望ましい．

テスト目的をレビューに使うことについて，「チェックする内容が明確になり，重大な欠陥がないことをチェックできることがメリットと感じた」との意見があり，実務に携わるエンジニアから好意的な評価を得られたと言える．チェックリストベースレビューに代表される従来のレビュー手法でもチェックする内容を明示化することはできるが，製品固有のチェックリストを用意するには，テスト目的を活用するのは良いアプローチだと考える．

「テスト目的としては必要だがレビューにどのように使ったらよいか分からないものがあった」という意見について，該当のテスト目的はテストには必要だが，仕様書にはそこまで考慮して記載しないような項目があった．仕様書などの文書に記載せずにプロジェクトメンバーの暗黙知になっている知識はどのプロジェクトにも存在するが，暗黙知に関する項目は，わざわざ仕様書に書かないというケースも実際にはある．しかし，経験の浅いエンジニアは，暗黙知に関する項目は考慮が抜けがちなため，テスト目的をチェックリストとして用いてレビューすることには意味があると考える．暗黙知に該当するテスト目的を仕様レビューに利用することで，「本当に仕様書に記載が不要かどうかを検討することには繋がるかもしれない」という意見もあった．テスト目的を仕様のレビューに用いることで，これまでプロジェクト内で暗黙知となっていた仕様を，仕様書などの文書として可視化することに繋がり，暗黙知の理解不足による欠陥の混入を防ぐ効果が期待できる．

4.3 妥当性

今回のケーススタディで利用したテスト目的には，各テスト目的から作成したテストケースのテスト実施にかかる工数は情報として付与されておらず，試行者の主観的な判断でテスト目的を選択した．同様の状況は，提案手法を実際の開発プロジェクトで利用する際にも起こりうる．テストケースのテスト実施にかかる工数の情報が無い場合は，レビューアの経験に基づきテスト目的を選択する必要があり，選択されるテスト目的がレビューアによってばらつく可能性がある．テスト目的を選択するエンジニアが，ソフトウェアテストの実務経験が少ない場合や，類似製品のプロジェクトに携わった経験が少ない場合，レビューに使用するテスト目的を適切に選択できない可能性がある．テスト目的の選定には一定のスキルが必要であるため、実施者に前提を置く必要がある．ただし，テスト実施にかかる工数のデータをテスト目的に紐づけて蓄積し，あてはまるテスト目的を先に選ばずにテスト目的の工数で先に限定してからあてはまるテスト目的を選ぶといった順序で実施することで対応できる可能性がある．テスト実施にかかるデータの蓄積，データに基づきテスト目的を選択する際の閾値の検討，データに基づいてテスト目的を選択した場合とデータに基づかない場合のレビュー結果の比較による効果検証は今後の課題である．

提案手法では過去に作成したテスト目的を流用してレビューを行うため，未知の新製品や新機能に対してはそのままは適用できない可能性がある．今回のケーススタディで利用した汎用的なアプリケーション向けのテスト目的のように，対象製品を抽象化して汎用的なテスト目的を蓄積しておけば適用範囲は広がるが，その分エンジニアがレビュー時にチェック内容を具体化する必要があり，レビュー結果がエンジニアの能力に依存しやすくなる．どの程度の抽象度でテスト目的を蓄積すると仕様のレビューに流用しやすいかの検証は今後の課題である．

5. おわりに

テスト目的を用いて，テストで検出・修正された場合の修正工数の大きい欠陥を，レビューで検出する方法を提案した．提案手法では，まず，過去に作成したテスト目的からレビュー対象においても効果があるテスト目的を選定する．次に，選定したテスト目的のうち実施工数が大きいものを選択し，レビューで確認する項目とし，レビューを実施する．

過去のプロジェクトで検出済みの欠陥を用いたケーススタディにより，テスト目的をレビューに用いて実際に欠陥の検出が可能であることを確認した．仕様書レビューの業務に携わるエンジニアに提案手法を試用してもらい，インタビューにより，提案手法を実務で活用することに対して好意的な意見を得た．

実際のプロジェクトへ適用しての効果の確認，テスト目的に紐づくテストケースの件数やテスト実施の工数に関する実際のデータに基づいてテスト目的を選択する方法を今後検討していく．

謝辞 評価にご協力頂いた皆様に，謹んで感謝の意を表する．

参考文献

[1] Fagan, M.E.. Design and Code Inspection to Reduce Errors in Program Development. IBM System Journal. 1976, vol. 15, no. 3, p. 182-211.
[2]Chernak, Y.. A Statistical Approach to the Inspection Checklist Formal Synthesis and Improvement. IEEE Transactions on Software Engineering. 1996, vol. 22, no. 12, p. 866-874.
[3]Shihab, E., Mockus, A., Kamei, Y., Adams, B. and Hassan, A. E.. High-impact Defects: A Study of Breakage and Surprise Defects. In Proceedings of the 19th Symposium and the 13th European Conference on Foundations of Software Engineering. 2011, p. 300-310.
[4]森崎修司, 松本健一. 業務観点でのレビューを目指した不具合情報の分析. 情報処理学会 ソフトウェア工学研究報告, 2013, vol.2013-SE-179, no.35, p.1-8.
[5]森崎 修司, 久保 匡, 荻野 利彦, 阪本 太志, 山田 淳. 過去の不具合の修正工数を考慮したソフトウェアレビュー手法. 電子情報通信学会論文誌 D, 2012, vol. J95-D, no.8, p. 1623-1632.
[6]Kasubuchi, K., Morisaki, S., Yoshida, A. and Ogawa, C.. An Empirical Evaluation of The Effectiveness of Inspection Scenarios Developed from A Defect Repository. In Proceedings of 2015 IEEE International Conference on Software Maintenance and Evolution. 2015, p. 439-448.
[7]Chillarege, R., Bhandari, I. S., Chaar, J. K., Halliday, M.J., Moebus, D. S., Ray, B. K. and Wong, M. Y.. Orthogonal Defect Classification - A Concept for In-process Measurements. IEEE Transactions on software Engineering. 1992, vol. 18, no. 11, p. 943-956.
[8]Leszak, M., Perry, D. E. and Stoll, D.. Classification and Evaluation of Defects in A Project Retrospective. Journal of Systems and Software. 2002, vol. 61, no. 3, p. 173-187.
[9] Gorschek,T. and Dzamashvili-Fogelström, N.. Test-Case Driven Inspection of Pre-Project Requirements - Process Proposal and Industry Experience Report. in Proceedings of the Requirements Engineering Decision Support Workshop held in conjunction with the 13th IEEE International Conference on Requirements Engineering. 2005.
[10]Ali, S., Mriand, L.C., Hemmati, H. and Panesar-walawege, R. K.. A Ststematic Review of the Application and Empirical Investigation of Search-based Test-Case Generation. IEEE Transactions on Software Engineering. 2010, vol. 36, no. 6, p. 742-762.
[11]Nishi, Y.. Design principles in test suite architecture. 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops. 2015, p. 1-4.
[12]池田暁, 鈴木三紀夫. [改訂新版]マインドマップから始めるソフトウェアテスト. 技術評論社, 2019.