

システムテストにおけるテスト分析手法の提案

The proposal of test analysis methods in system test

テルモ株式会社 研究開発本部 ME センター システムアーキテクトグループ

Terumo Corporation Product Development Group R&D Headquarters

○永松 康能¹⁾

牧 伸

中村 克也

○Yasuyoshi Nagamatsu¹⁾

Shin Maki

Katsuya Nakamura

Abstract

So far, we have created explicit test protocol specifications as tests based on the software requirements specification. Further, in order to check the implicit specifications, we have extracted test perspective from the past cases, and have been tested while error guessing. However, we were not able to guarantee whether sufficient system test was done because we were not able to verify leakage of software requirements specification in these methods.

So, we have studied test analysis methods in terms of testability of the software requirements specification and created test design guideline. We will present the test analysis methods.

1.はじめに

1.1.背景

弊社は、医療機器の製造・販売を行う企業である。我々はその中で、血圧計や血糖計、画像診断装置などのソフトウェアサブシステム、および血管内画像ビューワー等の単体ソフトウェアの設計、開発および設計検証に従事している。

そして、我々はFDA (Food and Drug Administration 米国食品医薬品局) の定める General Principles of Software Validation^[1] (以下、GPSV と記す)、および IEC62304 医療機器ソフトウェア ソフトウェア ライフサイクルプロセス^[2]を遵守することを念頭におき活動しており、特に GPSV に着目して活動している。なぜなら、GPSV は Validation および Verification (以下 V&V と記す) に言及しているのに対し、IEC62304 は Verification について言及しており、GPSV のほうが、範囲が広いからである。

さて、GPSV では、「ソフトウェアの仕様がユーザのニーズや意図する用途に一致しており、ソフトウェアに実装された特定の要件が一貫して満たされていることを、検査および客観的な証拠の提出により確認すること」^[1]とされているが、具体的にどうやってそれを実現するかは、製造業者に任されている。

1.2.我々の V&V の構造論

GPSV にて、ソフトウェアバリデーションは設計妥当性確認の一部である^[1]と定義されている。そこで、我々は、ソフトウェア製品またはソフトウェアを搭載した機器の設計妥当性確認を行うことをソフトウェアバリデーションと解釈し、図 1-1 のように整理している。

つまり、全ての活動は広義的には設計妥当性確認の活動と位置付けつつ、狭義の設計妥当性確認と設計検証に分類し、対象範囲を明確にしている。さらに設計検証をシステム設計検証とサブシステム設計検証に分類している。ちなみに、ここで言うシステムとは製品を指し、サブシステムとは製品を構成するソフトウェアや電気、メカを指す。

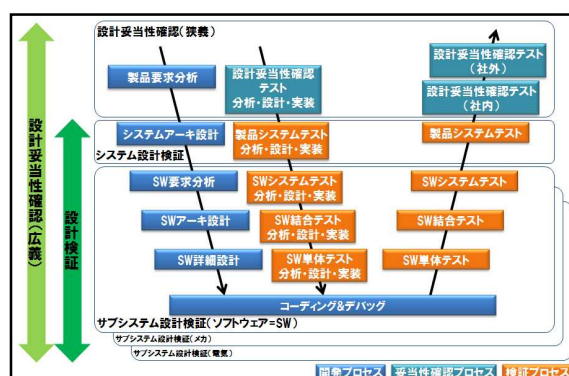


図 1-1. 設計妥当性確認と設計検証の構造

1) テルモ株式会社 湘南センター

〒259-0151 神奈川県足柄上郡中井町井ノ口 1500 電話 0465 (81) 4180

E-mail : Yasuyoshi.Nagamatsu@terumo.co.jp

1.3.テストプロセス^{*1}

前述の図 1-1 に示す検証プロセスは、大きく 4 つのテストに分類することができる。

①製品システムテスト、②SW システムテスト、③SW 結合テスト、④SW 単体テストである。この 4 つのテストそれぞれに、図 1-2 のテストプロセス^[3]がある。(SW はソフトウェアの略)

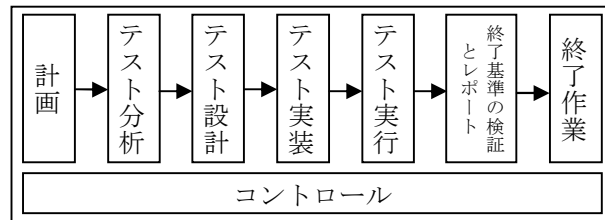


図 1-2.テストプロセス

1.4.従来のソフトウェアサブシステムテストにおける課題

我々はこれまで、ソフトウェア要求仕様書（以下、SRS と記す）をテストベース^[4]として、SRS とテスト手順書を 1 対 1 にトレースを取り、かつ、その 1 項目の記述に条件の漏れがないか検証するため、デシジョンテーブルを用いて曖昧な条件を具体化し、テスト手順書を作成してきた。また、SRS に記載されていない機能を押さえるために、過去事例からテスト観点を抽出し、エラー推測^[4]しながらテストを実施してきた。しかし、開発者が次工程で設計しやすいように正規化された機能が、SRS には記載されているため、その機能はユースケースの順序に並んでいなかった。そのため、製品の振舞いがイメージしづらく、SRS に抜け漏れがあっても見つけることが難しかった。その結果、例えば、電源 ON・OFF を複数回繰り返した後の通信が正しくできるかなど、実際の運用を意識した機能間の繋がりを確認するテスト手順書を作成することが難しかった。そこで、運用を意識したテスト手順書を作成するために、ユースケースに着目した。

また、従来 SRS に記載されているユースケースと機能要求仕様は、プロジェクト毎に構造と粒度が異なるため、我々が求めるソフトウェアシステムテスト（以下、SW システムテストと記す）に使える SRS を得ることはできなかつた。そこで、SRS の記載方法の如何を問わず、SW システムテストに必要な情報を抽出および補完できるテスト分析手法を目指した。

1.5.今回提案する内容

SRS をテスト性（テストできるかどうか）の観点で分析する手法を研究し、検証設計ガイドラインとして手順化した。今回はこのテスト分析手法について説明する。また、本論に入る前に、今回説明するテスト分析の対象範囲を以下に示す。

(1) どのテストレベルで実施するのか？

今回提案するテスト分析手法を実施するテストレベルは、SW システムテストである。前述の図 1-1 に示す検証プロセスの「SW システムテスト設計・開発」に該当する。

(2) テストベースは何か？

前述の図 1-1 に示す開発プロセスの「SW 要求分析」で作成される SRS である。

(3) テスト分析観点は何か？

SRS をベースに SW システムテストのテスト手順書を作成する上で十分な情報を整理できること。十分な情報とは何かについては、2 章にて後述する。

2.提案するテスト分析手法

2.1.ユースケースの定義

ユースケースは、ISTQB のソフトウェアテスト標準用語集^[4]では下記に記す (1) のように定義されている。一方、我々は、前述の図 1-1 の「設計妥当性確認テスト」で利用する妥当性確認用ユースケースと、「製品システムテスト」や「SW システムテスト」で利用する設計検証用ユースケースを区別し、下記に記す (2)、(3) のように定義している。なお、今回本稿で説明するユースケースは、設計検証用ユースケースであるが、本稿では単にユースケースと記す。

(1) ユースケース^[4]

アクターとコンポーネント又はシステムとの間の対話における一連のトランザクション。視覚できる結果を伴う。アクターは、ユーザ又はシステムと情報交換するあらゆるものになりうる。

(2) 設計妥当性確認用ユースケース

製品要求仕様がユーザのニーズや意図する用途に一致していることを確認するためのユースケースである。例えば、看護師が医師の指示を受けて普段どのように行動しているかを分

析し、その行動を模したテストをすることを指す。

(3) 設計検証用ユースケース

SRS の要求仕様が、製品やソフトウェアサブシステムに正しく実装されていることを確認するためのユースケースである。例えば、看護師が設計妥当性確認用ユースケースで定義されている行動をしたときに、製品やソフトウェアサブシステムの機能はどの順序で呼ばれるのかを分析し、その機能の振舞いをテストすることを指す。

2.2 テスト分析手法の特長

我々は、テスト分析におけるテストタイプ^[4]を機能テスト、非機能テストと大きく分類している。今回本稿では、機能テストにおける工夫点を示す。

(1) 機能テストにおける 4 つのテスト分析観点

機能テストのテスト分析観点を、次の 4 つに分類した。①「ユースケースの洗い出し」、②「ユースケースフローの洗い出し」、③「単機能の洗い出し」、④「状態、イベント、結果の洗い出し」

(2) テスト分析に用いる 4 つのテンプレート

先のテスト分析観点到に合わせて、4 つのテンプレートを開発した。a. 「ユースケース一覧表」、b. 「ユースケースフロー表」、c. 「ユースケースシーケンス表」、d. 「状態・イベント・結果一覧表」

(3) テスト分析観点とテンプレートの関係性

4 つのテスト分析観点和 4 つのテンプレートの関係を下表に示す。

表 2-1. テスト分析観点和テンプレートの関係性

テストタイプ	テストベース	テスト分析観点	テンプレート			
			a. ユースケース一覧表	b. ユースケースフロー表	c. ユースケースシーケンス表	d. 状態・イベント・結果一覧表
機能テスト	SRS 内のユースケース記載箇所	① ユースケース洗い出し	●	-	-	-
		② ユースケースフロー洗い出し	-	●	-	-
	SRS 内の機能要求仕様記載箇所	③ 単機能洗い出し	-	-	●	-
		④ 状態・イベント・結果洗い出し	-	-	-	●

(4) ユースケースの体系化とそのメリット

SRS に記載されているユースケースと機能要求仕様は、プロジェクト毎に構造と粒度が異なるのが常である。そこで、各プロジェクトの影響を受けずにテスト分析できるように、図 2-1 のとおりユースケースを体系化した。これにより、プロジェクトの影響を受けずにユースケースを整理できるようになった。また、設計仕様に寄りかちな機能要求仕様をユーザが使う機能（我々は、「ユーザ機能」と呼んでいる）で整理できるようになった。つまり、これによりその結果、ユースケーステストのシナリオをイメージしやすくなり、シナリオに SRS の機能要求仕様が漏れていないか検証できるようになった。なお、図 2-1 に登場する用語の定義を表 2-2 に示す。

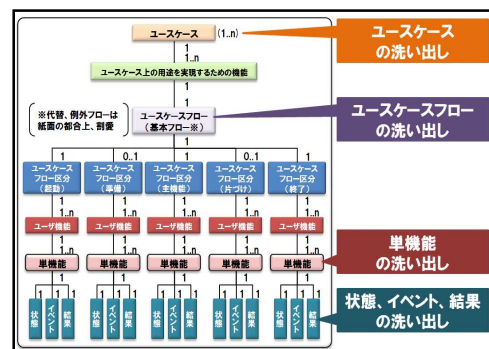


図 2-1. ユースケースの体系図

表 2-2. 用語の定義 (オリジナル)

ユースケースの構造	用語の定義
ユースケース	アクターと利用シーン、および用途が記述されたもの
ユースケース上の用途を実現するための機能	アクターが特定の利用シーンで特定の用途を満たすために行う機能
ユースケースフロー	ユースケース上の用途を実現するための機能の基本フロー、代替フロー、例外フローの総称
ユースケースフロー区分	ユースケースフローを時系列に起動、準備、主機能、片づけ、終了に分類したもの
ユーザ機能	ユースケース区分を実現する最小単位の目的を実現する機能
単機能	状態、イベント、結果で記述される 1 行

2.3. テスト分析のプロセス

(1) ユースケースの洗い出し方法

ユースケース一覧表というテンプレートを用意し、横軸に『ユースケース ID』、『機体種別』、『アクター』、『利用シーン』、『用途』、『ユースケース上の用途を実現するための機能』（フィーチャ^[4]）、縦軸に「市販前」、「市販後」というカテゴリを置き、SRS 内のユースケースに記載されている『アクター』、『利用シーン』、『用途』、『ユースケース上の用途を実現するための機能』をプロットすることで、ユースケースに抜け漏れが無いことを検証できるようにした。

表 2-3. ユースケース一覧表イメージ

	ユースケース ID	機体種別	アクター		利用シーン	用途	ユースケース上の用途を実現するための機能
			使用者	被使用者			

市販前	01	H	使用者 1	N/A	生産工程	出荷検査	出荷検査用チェック機能
市販後	02	H	使用者 2	N/A	院内 1	設定 1	設定 1 機能
	⋮						
	15	K	使用者 3	被使用者 1	院内 2	救急	緊急●●機能

(2) ユースケースフローの洗い出し方法

ユースケースフロー表というテンプレートを用意し、横軸に『ユーザ機能』、『SRS に記載されている機能』、縦軸にユースケースフロー区分（『起動』、『準備』、『主機能』、『片づけ』、『終了』）というカテゴリを置き、SRS 内のユースケースに記載されているワークフローを、起動から終了に分ける。その後、SRS 内の機能要求仕様に記載されている機能をプロットすることで、ユースケースフローに抜け漏れが無いことを検証できるようにした。

なお、ユースケースフローとして、基本フロー、代替フロー、例外フローを有識者や開発者を交えて、洗い出すことが特に重要である。

表 2-4.ユースケースフロー表イメージ (1/2)

ユースケースフロー区分	ユーザ機能	SRS-ID	SRS 項番	SRS に記載されている機能
A : 起動	起動する	608	1.1.1	電源 ON
		1002	2.1.1.2	セルフチェック
		998	2.1.1.1	CPU 初期化
B : 準備	付属品セット	620	1.2	ガイダンス
		634	1.2.1	ガイダンス開始
		638	1.2.2	ガイダンス終了
		646	1.3	付属品切替
	パラメータセット	668	1.5.1	パラメータ 1
		690	1.5.5	パラメータ 2
		721	1.6.2	調整 1
調整	704	1.5.8	調整 2	
	C : 主機能	●●開始	712	1.6
704			1.5.8	パラメータ 1
701			1.5.7	パラメータ 2
計画通り●●して停止		782	1.6.4	停止前状態
		1669	5.2	警報
		1684	5.2.2	警報一覧
		791	1.6.5	停止
D : 片づけ	付属品を外す	620	1.2	ガイダンス
		634	1.2.1	ガイダンス開始
		638	1.2.2	ガイダンス終了
E : 終了	終了する	611	1.1.2	電源 OFF
		1027	2.2.4	シャットダウン

なお、実際にはユースケース一覧表とのトレースを取るための ID やユースケースフロー自体に基本フロー、代替フロー、例外フローとのトレースを取るため、ユースケースフロー表には、下表のような表が付いている。

表 2-4.ユースケースフロー表イメージ (2/2)

ユースケース ID	06					
ユースケースフローID	06-H-01-01	06-H-02-01	06-H-02-02	06-H-02-03	06-H-03-01	06-H-03-02
基本/代替/例外	基本	代替 1	代替 2	代替 3	例外 1	例外 2

(3) 単機能の洗い出し方法

1) 単機能の定義

単機能とは、「状態」、「操作」、「処理」、「結果」という流れで一文を作成することにより、1つのテストケースにすることを指す。これは、我々の作った造語である。

2) 単機能の洗い出し方法の説明

ユースケースシーケンス表というテンプレートを用意し、ユースケースフロー毎に『SRS に記載されている機能の各行』、『ユーザ』、『システム』に分け、ユーザが何かしたら、システムが処理をし、ユーザへフィードバックするという、一連の流れ（「状態」、「操作」、「処理」、「結果」）になるように、SRS に記載されている機能の各行をプロットする。こうすることで、SRS の各行を「状態」、「操作」、「処理」、「結果」という流れの一行に抜け漏れがないことを検証できるようにした。

また、ユースケースシーケンス表を使うことで、SRS の各行を「状態」、「操作」、「処理」、「結果」に分解し、テスト設計で利用しない「処理」を除外できるようになった。

表 2-5.ユースケースシーケンス表イメージ (1/2)

ユースケース フロー区分	単機能 ID	SRS に記載されている機能の各行	種別	対象	SRS-ID	シーケンス	
						ユーザ	システム
A 起動	1	電源 OFF 状態	状態	●	609		←
		電源スイッチを 1 秒間押し続ける	操作	●	609	→	
		CPU 初期化は 3 秒以内に完了	処理	×	999		■
		セルフチェックは 3 秒以内に完了する	処理	×	999		■
		セルフチェック画面が表示される	結果	●	新規追加		←
		警報なし	結果	●	新規追加		←
		メッセージ表示なし	結果	●	新規追加		←
		セルフチェック音が鳴動される	結果	●	新規追加		←
		LED が赤・緑と交互点滅される	結果	●	新規追加		←
		モータが停止される	結果	●	新規追加		←
	2	セルフチェック中	状態	●	新規追加		←
		セルフチェック完了まで待機する	操作	●	新規追加	→	
		●●停止中状態に遷移する	処理	×	新規追加		■
		パラメータ 1・2 が●色背景で表示される	結果	●	新規追加		←
		警報なし	結果	●	新規追加		←
		メッセージ表示なし	結果	●	新規追加		←
		ブザー鳴動無し	結果	●	新規追加		←
		LED が消灯される	結果	●	新規追加		←
		モータが停止される	結果	●	新規追加		←
		B 準備	3	●●中状態	状態	●	621,622,623,625
●●開	操作			●	621,622,623,625	→	
ガイダンス画面を表示	結果			●	621,622,623,625		←
●●をアニメーション表示	結果			●	621,622,623,625		←
●●未装着を表示	結果			●	621,622,623,625		←
●●ガイド、●●未装着をアニメーション表示	結果			●	621,622,623,625		←
警報なし	結果			●	新規追加		←
●●を閉める旨のメッセージが表示	結果			●	新規追加		←
ブザー鳴動無し	結果			●	新規追加		←
LED が消灯される	結果			●	新規追加		←
モータが停止される	結果	●	新規追加		←		

※実際のユースケースシーケンス表には、E 終了まで記載されているが、紙面の都合で割愛する。

なお、実際にはユースケースフロー表とのトレースを取るため、ユースケースシーケンス表には、下表のような表が付いている。

表 2-5.ユースケースシーケンス表イメージ (2/2)

ユースケース ID	06
ユースケースフローID	06-H-01-01
基本/代替/例外	基本

(4) 状態、イベント、結果の洗い出し方法

状態・イベント・結果一覧表というテンプレートを用意し、ユースケースシーケンス表で整理した単機能を『ある状態のときに』（状態）、『誰かが何かをしたら』（イベント）、『ある結果になる。』（結果）に分けてプロットし、さらにテストする上で曖昧な部分を具体化する。これにより、状態、イベント、結果に抜け漏れがないことを検証できるようにした。

表 2-6.状態・イベント・結果一覧表イメージ (1/3)

ユースケース フロー区分	単機能 ID	単機能			SRS -ID
		状態	イベント	結果	
A 起動	1	電源 OFF 状態	電源スイッチを 1 秒間押し続ける	セルフチェックを行う	***
	2	セルフチェック中	セルフチェック完了まで待機する	●●中状態に遷移する	***
B 準備	3	●●中状態	●●開	ガイダンス画面を表示	***

※実際の状態・イベント・結果一覧表には、E 終了まで記載されているが、紙面の都合で割愛する。また、SRS-ID は前述のユースケースシーケンス表から紐づけるので、複数になるため、ここでは***と表記する。

前述した曖昧な部分を具体化するには、例えば、「電源 OFF 状態」とか「セルフチェック中」という記述があった場合に、外側からソフトウェアサブシステムのどこを見ると確認（認識）できるかを明確にすることである。そのため、状態の判断基準も下表のように整理している。

表 2-6.状態・イベント・結果一覧表イメージ (2/3) : 状態の判断基準

状態	画面	ブザー	LED	モータ
電源 OFF (AC なし)	画面出力無し	ブザー鳴動無し	消灯	モータが停止
電源 OFF	液晶表示部が暗いグレー表示	ブザー鳴動無し	消灯	モータが停止
	液晶表示部に●●アイコンを表示			
	液晶表示部に●●アイコンを表示			
セルフチェック	セルフチェック画面が表示	セルフチェック音が鳴動	赤・緑と交互点滅	モータが停止

●●停止中	パラメータ1・2が●色背景で表示	ブザー鳴動無し	消灯	モータが停止
●●中	パラメータ1・2が●色背景で表示	操作音が鳴動	緑点滅	モータが駆動
●●終了	パラメータ1・2が●色背景で表示	停止遷移音が鳴動	消灯	モータが停止

※上記以外にも警報やメッセージなど複数存在するが、紙面の都合で割愛する。

なお、実際にはユースケースシーケンス表とのトレースを取るため、状態・イベント・結果一覧表には、下表のような表が付いている。

表 2-6.状態・イベント・結果一覧表イメージ (3/3)

ユースケース ID	06
ユースケースフローID	06-H-01-01
基本/代替/例外	基本

3.結果と考察

3.1.SRS の抜け漏れ抽出

前章で説明した一連のテスト分析手順の基礎検討を行った結果を下記マトリクス^[4] (1) ~ (3) として集計した。(1) SRS に記載されている機能の網羅率は、SRS に記載されている機能が漏れなくテスト分析対象になっていることを証明するための値である。(2) SW システムテストに利用できる SRS の行の網羅率は、SRS に記載されている行が漏れなくテスト分析対象になっていることを証明するための値である。一方、(3) SRS に記載されていない項目数は、本テスト分析手法でどの程度抜け漏れを抽出できているかの度合いであり、本テスト分析の効果を示す値である。

マトリクスの話に入る前に、まず各成果物のボリュームを表 3-1 に示す。

表 3-1.今回作成した成果物のボリューム

成果物	ボリューム	備考
ユースケース一覧表	15 ユースケース	特になし
ユースケースフロー表	150 個	15 ユースケース×10 ユースケースフロー ※ユースケースフローの内訳 基本フロー×1、代替フロー×6、例外フロー×3
ユースケースシーケンス表	150 個	特になし
状態・イベント・結果一覧表	150 個	特になし

(1) SRS に記載されている機能の網羅率は 92.2%

本マトリクスは、SRS に記載されている機能（文書構造上の章、節、項^[5]に該当）を完全にテスト分析していることを保証するための値である。表 3-2 にマトリクス検出方法と目標値を示す。

表 3-2. SRS に記載されている機能の網羅率の検出方法と目標値

マトリクス検出対象テンプレート	ユースケースフロー表
抜け漏れのチェック方法	SRS に記載されている機能が、ユースケースフロー表に記載されている基本フロー、代替フロー、例外フローの何れかに一度以上登場しているかどうかをチェックする。
計算式	ユースケースフロー表に登場している SRS 上の機能の割合 =ユースケースフロー表に登場している SRS 上の機能数/SRS 上の機能数
目標	100%

SRS に記載されている機能のうち、ユースケースフロー表に登場した機能は 92.2%であった。

ユースケース未登場の機能について、どのような機能がユースケースから漏れたか整理したところ、出荷前の生産工程と出荷後すぐの院内設定のユースケースが曖昧であったため、SRS に記載されている機能を紐づけられなかったことが判明した。このユースケース未登場の機能については、再度ユースケースを確認し、ユースケースフローに登場するように修正する。

(2) SW システムテストに利用できる SRS の行の網羅率は 100%

本マトリクスは、SRS に記載されている行（文書構造上の文章^[5]に該当）を完全にテスト分析していることを保証するための値である。表 3-3 にマトリクス検出方法と目標値を示す。

表 3-3. SW システムテストに利用できる SRS の行の網羅率の検出方法と目標値

マトリクス検出対象テンプレート	ユースケースシーケンス表
抜け漏れのチェック方法	SW システムテストに利用できる SRS の行が、ユースケースシーケンス表に全て登場しているかどうかをチェックする。
計算式	ユースケースシーケンス表に登場している SRS の行数の割合 =ユースケースシーケンス表に登場している SRS の行数/SW システムテストに利用できる SRS の行数
目標	100%

SRS の行の中には、SW システムテストに利用できる行（有効行）と SW システムテストに利用できない行（無効行）が存在する。無効行とは、下位の設計文書の作成に必要な補足説明やデータ定義など SW システムテストに不要な行を指す。なお、SW システムテストに利用できる行を有効行と捉えてマトリクスを収集したところ、網羅率は 100%であった。

(3) SRSに記載されていない項目数は880項目^{*2}

本メトリクスは、SRSに記載されていない内容（文書構造上の文章^[5]に該当）をどの程度抽出しているか、つまり、テスト分析の効果を示す値である。表3-4にメトリクス検出方法と目標値を示す。

表3-4.SRSに記載されていない項目数の検出方法と目標値

メトリクス検出対象テンプレート	状態・イベント・結果一覧表
抜け漏れのチェック方法	SRSの機能の振舞いを説明している行を、状態、イベント、結果に分解したときに、どの項目が漏れているか、また曖昧かをチェックする。
計算式	状態・イベント・結果一覧表の空白箇所に対する指摘数
目標	N/A（初の試みのため、目標値は未設定。今後のベンチマークとする。）

状態・イベント・結果一覧表全2013行中880項目の抜け漏れ、または曖昧な箇所を見つけた。

3.2.考察

(1) SRSの記載方法を問わず、テストに必要な情報の抽出および補完が可能

従来、開発者が次工程で設計しやすいように正規化された機能が、SRSには記載されていないため、その機能はユースケースの順序に並んでいなかった。そのため、製品の振舞いがイメージしづらく、SRSに抜け漏れがあっても見つけることが難しかった。また、SRSに記載されているユースケースと機能要求仕様は、プロジェクト毎に構造と粒度が異なった。そこで、各プロジェクトの影響を受けずにテスト分析できるように、前述の図2-1のとおりユースケースを体系化した。

今回ユースケースに着目し、SRSに記載されているユースケースを4つのテンプレートを使い分析した結果、単機能毎の状態・イベント・結果の抜け漏れを見つけられるようになった。また、SRSに記載されている機能のうちユースケースフローに登場していないもの（今回のテスト分析では7.8%の機能がユースケース未登場）を見つけられるようになった。

以上のことから、このテスト分析手法を用いることで、SRSの記載方法の如何を問わず、SWシステムテストに必要な情報を抽出および補完できるようになり、ユースケースの抜け漏れの発見にも役立つようになったと言える。無論、その客観的な効果の程度については、今後測定する欠陥除去率が目標値の80%を達成できるかどうかで判断する予定である。とは言え、過去に実施した今回と同規模のシステムにおけるテスト分析の指摘件数が78件^{*3}であったのに対し、今回880項目^{*2}の指摘を行うことができたことと、特に従来曖昧だった「状態」を明確にできるようになったことは事実であり、テスト分析の精度は従来よりも上がっている。その結果、テスト実施時に見つけられる故障^[4]の精度も上がるものと予想する。

(2) 設計品質の向上にも有効

SWシステムテストに必要な情報を補完できると前述したが、これは状態（＝結果）、操作を明確にすることができることを指している。例えば、状態を明確にするために画面の仕様を整理していくと、画面とその中の部位を明確に定義することができる。つまり、これ自体が次の設計工程へ向けた情報の整理になるので、設計品質の向上にも役立つと言える。

(3) テスト分析からテストケース設計の自動化

テスト分析で作成した状態・イベント・結果一覧表の情報（状態、イベント、結果）をデシジョンテーブルおよび状態遷移表のインプットとして利用できるようになったため、テスト分析からテストケース設計の自動化が可能になった。

4.まとめ

4.1.SWシステムテストにおけるテスト分析の十分性について

一定の教育を施したメンバーであれば、今回手順化したテスト分析手法により、SRSからテストに必要な情報の抽出および補完することが可能である。

4.2.今回判明した課題

(1) 欠陥除去率の収集と評価

諸事情により欠陥除去率（DRE）^[6]が取得できなかった。そのため、テスト分析が検証プロセス全体でどの程度SRSの欠陥抽出に貢献したかを定量的に示すことができなかった。引き続き欠

陥除去率の収集をしていく。なお、収集しようとしているメトリクスは表 4-1 の通りである。

表 4-1.欠陥除去率の集計結果

検証プロセス	SRS の欠陥	アーキの欠陥	SDD の欠陥	コードの欠陥	合計	DRE	目標 DRE
SRS のテスト分析	880	---	---	---	880	N/A	80%
SW 単体テスト	未実施のため収集できず				0	N/A	95%
SW 結合テスト					0	N/A	
SW システムテスト					0	N/A	
製品システムテスト					0	N/A	
設計妥当性確認テスト					0	N/A	
合計	880	0	0	0	880	N/A	95%

(2) メトリクスが取り易いテスト分析手順とテンプレートを考える

メトリクス収集の側面からみると、指摘の重複を集計しない、また逆に抜け漏れと曖昧な箇所を別々に集計できる、というようなテスト分析手順およびテンプレートの改良が必要である。

(3) 状態・イベント・結果一覧表の自動生成

状態・イベント・結果一覧表にて、状態、イベント、結果の洗い出しを行う手順としていたが、実際にテスト分析していると、ユースケースシーケンス表でも洗い出せそうであることに気付いた。もしこれが可能であれば、ユースケースシーケンス表を作成した時点で、状態・イベント・結果一覧表を自動生成できるようになるので、効率化を図れる。

(4) テスト分析で「入出力範囲」をどう扱うか

状態・イベント・結果一覧表作成時に、状態および結果にそれぞれ入力範囲と出力範囲を設定できれば、入出力テストもテスト分析時に考慮することができるようになる。

(5) 設計妥当性確認用ユースケースの作成手順の標準化

今回設計検証用ユースケースをテストベースにテスト分析したが、設計妥当性確認用ユースケースの質によって、設計検証用ユースケースの質も決まることを実感した。そのため、今後は如何に設計妥当性確認用ユースケースを作るかを標準化することが重要であると感じた。

4.3.今後の展開

前項に整理した課題のうち (1) ~ (4) は、テスト分析のブラッシュアップと効果の裏付けであるので、今後も継続して取り組んでいく。また現在、テスト分析からテストケース設計、テスト実装、テスト実施、テストレポートまでの自動化を研究中である。この一連の自動化実現後は、非機能テストおよび前項に整理した課題の (5) 設計妥当性確認用ユースケースの作成手順の標準化を予定している。

参考文献

- [1] General Principles of Software Validation; Final Guidance for Industry and FDA Staff ソフトウェアバリデーションの一般原則 ~業界および FDA 職員用ガイダンス~, 2002 年 1 月 11 日
- [2] IEC62304 医療機器ソフトウェア ソフトウェアライフサイクルプロセス, 第 1 版, 2006 年 5 月
- [3] テスト技術者資格制度 Advanced Level シラバス日本語版 テストマネージャ, Version 2012.J03, pp.8-15
- [4] ISTQB 用語集作業班, ソフトウェアテスト標準用語集 日本語版, Version 2.2.J03
- [5] 合同会社イオタクラフト, 「ドキュメンテーション基礎スキル」テキスト, pp.37, 2013 年 7 月初版
- [6] リンダ・M・ライド+M・キャロル・ブレナン著, 演習で学ぶソフトウェアメトリクスの基礎 日経 BP 社, pp.187-190, 2009 年

補足説明

*1 正確には、検証プロセスにはテスト以外の活動が存在するが、その説明を入れると却って煩雑になるため、検証プロセスの中のテストプロセスのみを説明することにした。

*2 状態、イベント、結果に分解したときの何れかの項目を指す。例えば、単機能 1 行の漏れを見つけた場合は、状態、イベント、結果の漏れとなるので 3 項目と数える。

*3 当時は、状態・イベント・結果で抽出していなかったもので、単純な指摘数を意味する。