

派生開発におけるテスト分析手法の提案

- ラルフチャートを用い、変更前後の差分に着目するテスト分析 -

Proposal of Test Analysis Method for Derivative Software Development

- Test analysis focused on differences before and after change using the Ralph's chart -

主査 : 秋山 浩一 (富士ゼロックス株式会社)
副主査 : 喜多 義弘 (東京工科大学)
リーダー : 池田 克彦 (株式会社アドバンテスト)
研究員 : 北岡 幸治 (AJS 株式会社)
小平 隆彦 (株式会社モバイルインターネットテクノロジー)
榊原 圭一
望月 信昭 (日本ナレッジ株式会社)

研究概要

派生開発のソフトウェア製品/システムに対するテストでは、変更の影響範囲を考慮しテスト観点の抜け漏れを抑えるテスト分析が大きな課題である。我々は、派生開発において、変更の前後のラルフチャート(Ralph's chart)を作成し、両者を比較することで変更の影響範囲を考慮したテスト観点を導出するというテスト分析手法を提案する。

ラルフチャートは、比較的短時間で作成することができる、表現が視覚的でテスト対象の目的機能全体を把握しやすい、ノイズや状態変数を書くことで因子の導出漏れが発見できる、という長所を持つ。ラルフチャートを応用したこの手法は、短時間で習得でき、影響範囲を考慮したテスト分析を行い、テスト観点の検討漏れを防ぐことができる。少数の事例での実験ではあるが、派生開発のテスト分析に活用可能との見通しを得た。

1. はじめに

ソフトウェア開発では、対象製品をゼロからつくりあげる新規開発よりも、既存の製品に手を加える派生開発（改修・保守・再開発・拡張）の割合が高くなっている。開発全体に派生開発が占める割合は表1の通りである^{[1],[2]}。（参考文献[2]では改修・保守，再開発，拡張と分かれているが，本稿ではそれらを合算している）

表1 ソフトウェア開発に占める派生開発の比率

	新規開発 (%)	派生開発 (%)
組み込み系	5.0	95.0
IT系 (全体)	49.5	50.5
IT系 (小規模開発)	35.5	64.5

派生開発では、新規開発に比べ製品出荷後に不具合が生じる傾向が強い。表2は、新規開発・派生開発それぞれにおける出荷後の不具合発生密度を、開発全体と、派生開発に多い小規模開発と比較したものである^[2]（参考文献[2]では、新規開発では40KSLOC未満，派生開発では20KSLOC未満を小規模開発としている）。全体としても、小規模開発に限っても、派生開発の方が出荷後の不具合発生密度が高い。派生開発は新規開発に比べ不具合が流出しやすいことがわかる。

表2 出荷後の不具合発生密度(件数/KSL0C)

	不具合発生密度(全体)	不具合発生密度(小規模開発)
新規開発	0.072	0.075
派生開発	0.104	0.152

ここまで記したことは、我々研究員の所属組織でもあてはまる。研究員5名中4名が派生開発製品の開発ないしテストにたずさわっている。表3は、4名中3名が担当する製品の流出不具合件数を示したものである（1名はデータなし）。過去数年間、毎年不具合の流出がある上、重度の不具合の流出も経験している。

表3 研究員の担当製品(派生開発製品)の流出不具合数

	2013年	2014年	2015年	2016年	合計	重度不具合数
A社	1	23	11	5	41	9
B社	10	4	1	8	23	1
C社	7	8	12	6	33	4

我々は、派生開発の方が新規開発より不具合が流出する傾向が強いのは派生開発で起こりがちな状況に起因していると考えた。派生開発では、複雑な仕様の衝突の発生、対象の全体を把握できていない部分理解に基づく変更の実施という状況が起こりがちだからである^[3]。変更に当たりリソース（メモリ、ファイル）の制約に対応できていなかった例も報告されている^[4]。開発の問題はテストにも影響をおよぼす。すなわち、複雑な仕様の衝突と仕様に対する部分理解から変更の影響範囲に気づきにくくなり^[3]、テスト観点（何をテストするか^[5]）の検討、すなわちテスト分析が不十分になり、結果としてテストで不具合を検出できないという事態が生じる。表3の流出不具合のうち、原因を分析できたものの中にもテスト観点漏れが流出原因と言えるものがあつた。表4に原因分析結果を示す。

表4 研究員の所属組織の不具合事例中、テスト観点漏れの件数

	分析対象不具合数	テスト観点漏れの件数
A社	5	5
B社	3	1
C社	17	9

変更の影響範囲を考慮し、必要なテスト観点を導出することのできるテスト分析方法があれば、表4の不具合をテストで検出し、流出を防げた可能性がある。ただし、この要求を満たす方法であれば何でもよいわけではない。我々は開発期間や予算といった制約のもとで開発を行っている。テスト分析手法の習得や適用に時間がかかったり、莫大な費用が必要であったり、特別な専門スキルが必要な手法は導入しにくい。また開発プロセスや開発手法の変更が必要な手法も導入しにくい。

既存の研究を調査したところ、派生開発に限定することなく、不具合の流出を防ぐテストについての研究があつた^[6]。不具合流出の防止策をテストに求める点では我々と同じであるが、テスト技法の適用（テスト設計）に焦点を当てており、テスト分析（テスト観点を導出）とは異なる。また、派生開発に特化した開発手法であるXDDP^[7]におけるテストの研究がある^{[8],[9]}。いずれも派生開発による影響範囲を考慮したテスト分析の方法を論じているが、XDDPを前提としている点は、XDDPを採用していない組織にとっては導入の障壁になる。

以上から、我々は、下記の条件を満たすテスト分析手法の開発が課題であると考えた。

第5分科会（品質チーム）

- 派生開発において、変更の影響範囲を考慮できる手法
- 手法の習得や適用に時間がかからない手法（導入容易性が高いこと）
- 開発手法に依存しない手法

第2章では、本稿で提案するテスト分析手法について、提案の経緯と手法の実施手順、分析の着目点を説明する。第3章で手法の有効性を実証する実験と評価の方法、および評価結果を述べる。第4章で本提案の結論と今後の課題を論ずる。

2. 提案

1章で述べた課題の解決に当たり、我々は、既存のテスト分析手法やそこで用いられる表記法の中でラルフチャート^[10]が適していると考えた（ラルフチャートの概要、および具体例については付録を参照）。ただし我々はラルフチャートを単独で使用する従来の方法でなく、派生開発による変更前後のラルフチャートを作成し、その差分に着目することで、変更の影響範囲を考慮したテスト観点を導出する「ラルフチャート差分比較法」を提案する。

ラルフチャートは HAYST 法というテスト設計手法で用いられる表記法であり、「目的機能」の詳細な分析を行ない、テストのための因子や水準を導出するために用いる^{[11], [12]}。比較的短時間で作成することができる、表現が視覚的でテスト対象の目的機能全体を把握しやすい、ノイズや状態変数を書くことで因子の導出漏れが発見できる、という長所を持つ。この長所は、派生開発における変更の影響範囲を考慮したテスト分析に役立つと考える。さらに、この表記法は特定の開発手法に依存しない。

2.1では、ラルフチャートの導入容易性と、ラルフチャートが派生開発のテスト分析に適しているかどうかを評価した結果を論じる。2.2で、2.1で見つかった課題を解決し、派生開発のテスト分析に適用できるようにした「ラルフチャート差分比較法」の分析手順を述べる。2.3では「ラルフチャート差分比較法」における分析の着目点を述べる。

2.1 ラルフチャートの評価

研究員の所属組織の不具合流出事例でラルフチャートを作成して、ラルフチャートが派生開発製品のテスト分析に適しているかどうかを評価した。この評価のためにラルフチャートを習得したが、6時間程度で習得できた。なお、研究員は全員実務経験が10年以上の設計者やテスト担当者であった。表5に評価結果を示す。

表5 過去事例への適用によりラルフチャートを評価した結果

利点	課題
(1) 開発規模によるが、6W2H、FV表、ラルフチャートまでなら1日くらいあれば書ける。	(1) 開発規模が小さいと、取り上げる対象の範囲が狭くなり、入力/出力/状態変数/ノイズ/振舞いを書き出すのが難しい。
(2) ラルフチャートが書けるかどうか、または入力を洗い出すための6W2Hを書けるかどうか、対象の理解が十分か、要求が明確かどうかのチェックになる。	(2) 事例の不具合を見つけられるようにしようとするとチャートの記述が細かくなり過ぎてしまう。

利点(1)はテスト分析の作業効率の点で、利点(2)はテスト分析の精度の点で、それぞれラルフチャートの有用性を確認できた。課題を分析した結果、課題(1)は、機能のうち変更対象となる部分にのみ着目したことが原因であった。課題(2)は、変更の内容や影響をすべて同一のチャートに記述しようとしたことが原因であった。これらの課題に対策を打てれ

第5分科会（品質チーム）

ば、派生開発のテスト分析にラルフチャートを適用可能と考えた。

対策として、課題(1)については、変更部分だけを取り上げるのではなく、変更部分を含む目的機能全体を記述することを考えた。課題(2)については、変更前と変更後を別のチャートに分けることで、各チャートの記載内容を分離することを考えた。

変更前後のラルフチャートがあれば、両者を比較することで、影響範囲を考慮することが可能になる。これを「ラルフチャート差分比較法」と名づけた。

2.2 ラルフチャート差分比較法の手順

図1にラルフチャート差分比較法によるテスト分析フローを示す(表記にはPFD(Process Flow Diagram)を使用)。

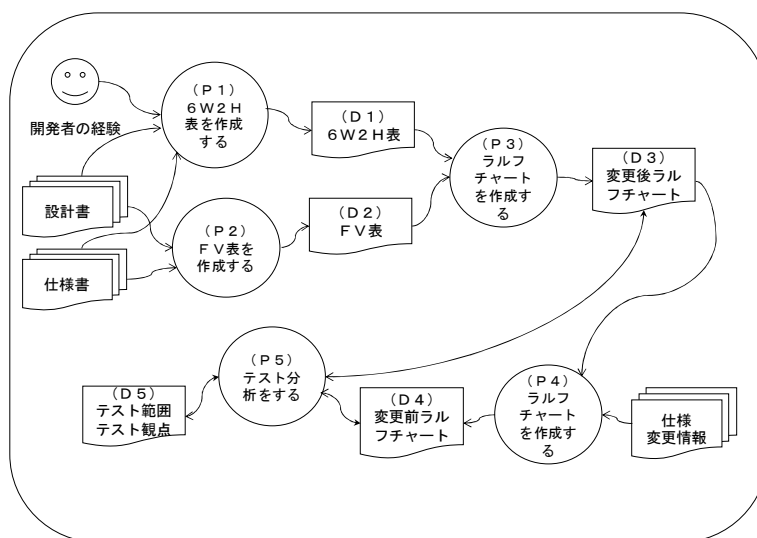


図1 ラルフチャート差分比較法によるテスト分析フロー

テスト分析手順は以下の通りである。

- ① 変更後の目的機能に対して、6W2H, FV表^[12]を作成する。(P1, P2)
- ② 変更後のラルフチャートを作成する。(P3)
目的機能に関係する入出力・状態変数・ノイズを検討し、記入する。
- ③ 変更前のラルフチャートを作成する。(P4)
- ④ 各ステップの成果物(D3, D4, D1, D2)を用いてテスト対象を分析し、テスト観点を導出する。(P5)
次の場合には②に戻る。
 - 入出力、状態変数、ノイズの見落としに気づいた
 - 目的機能のボックス内の振舞いや内部ロジックに差分があることに気づいた

2.3 ラルフチャート差分比較法の分析の着目点

2.2の④でテスト対象である目的機能を分析する際の着目点を以下に記す。なお、目的機能の振舞いの変更は、本手法によらなくともテスト観点になる。

a. 変更前後の要素の差分

要素（入力・状態変数・ノイズ・アクティブノイズ）が、変更前と比較して、要素自体の増減はないか、各要素の値に変化はないか。要素の増減や値の変化が、振舞いや内部ロジックの変更にどう関係しているか

b. 入力・状態変数・ノイズと、状態遷移の関係

ボックス内に状態遷移が現れる場合、入力・状態変数・ノイズが状態遷移のトリガー

第5分科会（品質チーム）

やタイミングに影響しないか

分析の結果から、以下のようにテスト観点を導出する。

- 要素（入力・状態変数・ノイズ・アクティブノイズ）の増加や値の変化が振舞いや内部ロジックの変更に関係している場合、要素の増加や値の変化をテスト観点に加える
- 入力・状態変数・ノイズがボックス内の状態遷移に影響を与えている場合、それらをテスト観点に加える

3. 実験/評価

2章で提案した手法について、本手法を考案した研究員自身が3.1に示す要領で実験・評価を行った。3.2に実験結果を例示する。3.3で実験の評価結果を記す。

3.1 実験・評価の方法

研究員のうち3名を実験者として、本手法を、2.1と同様、研究員の所属組織の不具合事例に対して適用し、実験を行った。

評価基準には、1章で挙げたテスト分析手法の条件を満たすかどうかを適用した。ただし、開発手法に依存しないことについては2章で確認できているので、以下の2点を評価した。

- 派生開発において、変更の影響範囲を考慮できるか。これについては、実験者への聞き取りを行うこととした。
- 手法の習得や適用に時間がかからないか（導入容易性が高いか）。これについては、実務への適用を考慮し、実験に要した時間を測定することとした。

3.2 実験結果

実験結果を示す。例示する派生開発要件の概略は以下の通りである。

- 要件： 半導体製造装置動作中のドアロック解除機能の追加
- 仕様： 装置に以下の機能を追加する。
 - 従来装置停止中のみ可能だったドアロック解除を装置動作中も可能にする。
 - ただし、ロック解除は手が入る危険部の動作が停止した後に行う。
 - また、ドア閉後は自動で危険部の動作を再開する。

手順①. 変更後の目的機能について6W2H, FV表を作成する（図2, 図3）。

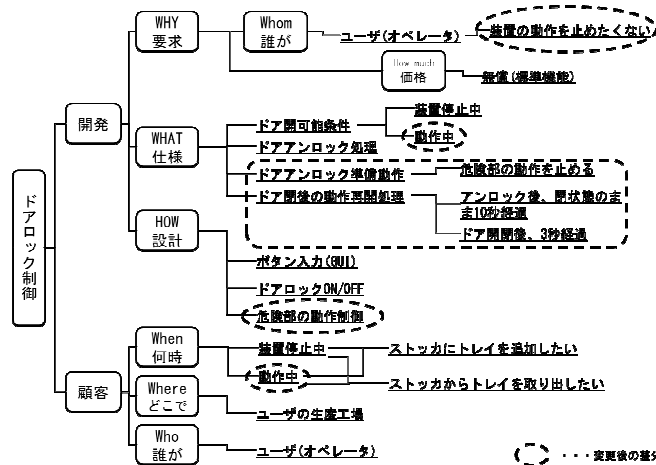


図2 手順① 6W2H

No.	目的機能（F）	検証（V）	テスト技法（T）
	ユーザ（オペレータ）が空になったストックにトレイを追加するため、動作中にドアを開け、トレイ追加作業を行い、ドアを閉める。	<ul style="list-style-type: none"> 動作中にドアがアンロックできるか ドアがアンロックされた時危険部は動作停止し、危険部以外は動作を継続するか ドアを閉めた後、自動で危険部の動作が再開するか ドアを開けなかった場合に自動で危険部の動作が再開するか 	シナリオテスト

図3 手順① FV表

手順②. 6W2H, FV表を基に, 変更後のラルフチャートを作成する (図4).

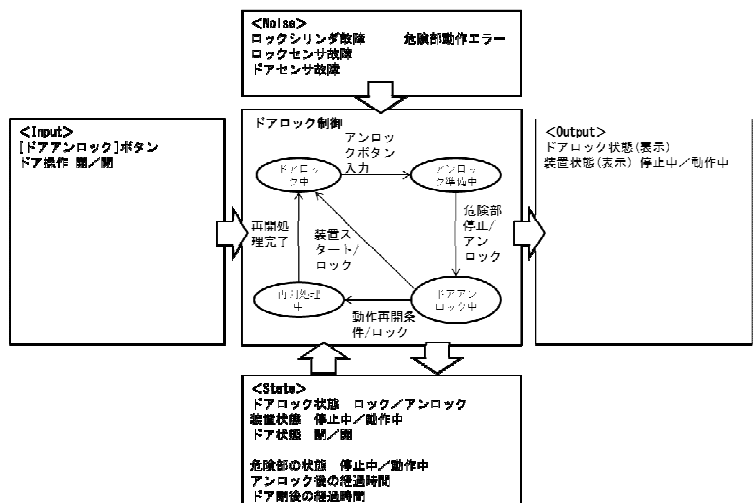


図4 手順② 変更後のラルフチャート

手順③. 変更前のラルフチャートを作成する (図5).

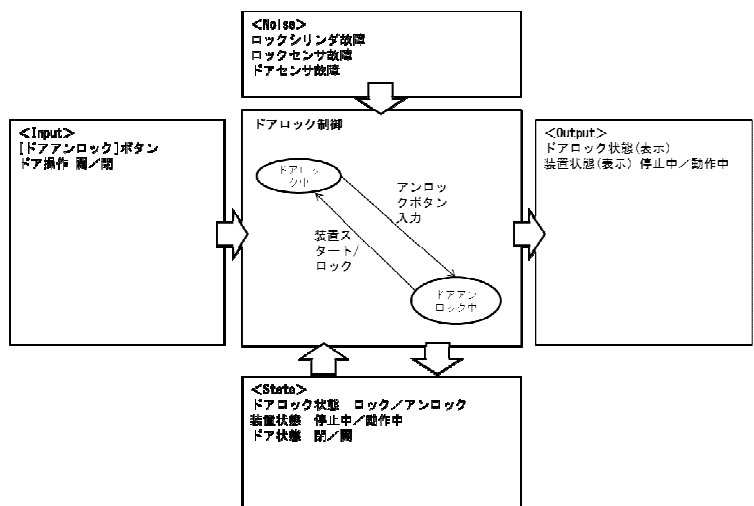


図5 手順③ 変更前のラルフチャート

手順④. 図4と図5の差分を比較し, テスト観点を検討する (図6).

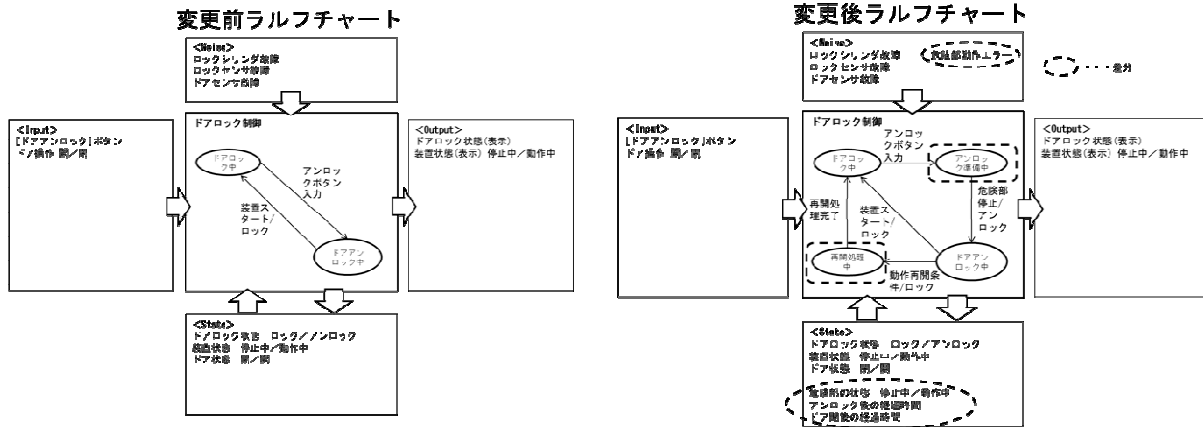


図6 手順④ 変更前後のラルフチャート

状態変数に「危険部の状態」を追加したことで、目的機能の振舞いは2つの状態変数の組合せとなることが分かった。さらに追加された状態からの遷移の条件が入力とは非同期であるため、「組合せ状態の遷移のタイミングに着眼したテスト」が新たなテスト観点として導出できた。

3.3 評価結果

表6に評価結果を示す。

表6 実験に要した時間と実験者への聞き取り結果

実験に要した時間	影響範囲の考慮
実験者(1) 6時間	できなかった。実験しやすそうなものを選んだせいか、差分比較の効果は実感できなかった。
実験者(2) 6時間	この手法を用いることで、流出した不具合を検出するテストはできるものだったことを確認した。
実験者(3) 4時間	差分をラルフチャートで表すことでドアロック状態の遷移と危険部の状態との組み合わせのテストが必要であることが判明した。

3.4 考察

実験者の考察を記す。この手法の利点や提案として以下のものが挙げられた。

- ノイズを意識することで非機能テストの必要性を認識できた。
- ラルフチャートの目的視点により、他機能への意識が喚起され、全体的な視点で分析をすることができるのではないかと。
- この手法を適用してみて、仕様に違和感を覚えた。（実験で取り上げた仕様は、不具合流出がきっかけとなって仕様・設計を見直すことになった）
- 中央にテスト対象となるシステムや機能がイメージできる構造を描く。これにより問題が起こりそうな入出力や状態の組合せを予見しやすくなった。
- ラルフチャート作成中に気づき生まれ、手順をさかのぼって6W2Hの「What」に追記を行うことがあった。必要に応じ前の手順に戻ることで分析の精度が高まりそうだと感じた。
- 変更前後の差分の表し方が実験者によって異なった。ラルフチャートの表現方法の統一が必要ではないかと。

4. 結論と展望

ラルフチャート差分比較法を実験した結果、影響範囲を考慮でき、不具合流出時には挙げられなかったテスト観点を導出できた。導入の容易さについては、習得にかかる時間が6時間程度、派生開発のテスト分析にかかる時間が4時間ないし6時間であり、実務への適用が容易であることがわかった。特定の開発プロセスや開発手法に依存しない方法であることは、ラルフチャート自体の特徴である。以上、少数の事例での実験ではあるが、本手法は十分有効であるとの見通しを得たと考える。

一方、3名のうち1名が「差分比較の効果は実感できなかった」としている。これは機能が単純なものを取り上げたためであった。このことは、本手法の適用には対象の粒度（規模、複雑さ）が重要であり得ることを示唆している。対象の粒度を適切に決めることは、テスト観pointsの抜け漏れを抑えるためにも重要である。また、今回の研究では過去の不具合事例に適用する実験のみであり、評価は定性的なものに留まった。若手や未熟な技術者が習得・適用する場合の効率や効果も研究する必要がある。以下は今後の課題とする。

- 過去事例でなく、新規の派生開発案件への適用による有用性の確認。取り上げる対象の適切な粒度の見極めを含む
- 定量的な評価基準の策定と、定量的な評価
- さまざまな種類の案件への適用。
今回実験で取り上げた事例は、いずれも「機能の追加」であった。派生開発には他にも、他システムとの接続、OSやライブラリの入替え、不具合の修正などがある。そうした案件に適用して有用性を確認することが必要である。
- 若手技術者への教育方法の検討

参考文献

- [1] 独立行政法人情報処理推進機構 (IPA) 技術本部ソフトウェア高信頼化センター (SEC), “組込みソフトウェア開発データ白書2015”, 独立行政法人情報処理推進機構, 2015.
- [2] 独立行政法人情報処理推進機構 (IPA) 技術本部ソフトウェア高信頼化センター (SEC), “ソフトウェア開発データ白書2016-2017”, 独立行政法人情報処理推進機構, 2016.
- [3] 清水吉男, “派生開発における母体由来するバグとその対応”, JaSST' 09 Tokyo, pp. 5-21, 2009.
- [4] SQiP25期第6分科会Bグループ, “派生開発におけるモレ・ムダのないテスト設計”, SQiP研究会, pp. 1-11, 2009.
- [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”, JaSST' 13 Tokyo, pp. 17-20, 2013.
- [6] SQiP27期第5分科会Aグループ, “バグの流出防止を考えるーどんなテストをすればバグを見つけられたのか?ー”, SQiP研究会, pp. 1-8, 2011.
- [7] 清水吉男, “「派生開発」を成功させるプロセス改善の技術と極意”, 技術評論社, 2007.
- [8] SQiP25期第5分科会(テスト設計グループ), “派生開発における影響箇所の把握改善によるテスト範囲の特定方法の提案”, SQiP研究会, pp. 1-8, 2009.
- [9] 派生開発推進協議会T4研究会, “「T型マトリクス」を用いたXDDPとテストプロセスの接続”, 派生開発カンファレンス2012, pp. 1-38, 2012.
- [10] 吉澤正孝, 秋山浩一, 仙石太郎, “ソフトウェアテストHAYST法入門 品質と生産性がアップする直交表の使い方”, 日科技連出版社, 2007.
- [11] 秋山浩一, “事例とツールで学ぶHAYST法ーソフトウェアテストの考え方と上達のポイント”, 日科技連出版社, 2014.
- [12] 秋山浩一, “ソフトウェアテスト技法ドリルーテスト設計の考え方と実際”, 日科技連出版社, 2010.