

1. HAYST 法について

ここでは既存のテスト分析・設計手法である HAYST 法のテストプロセスと、テスト要求分析手法である 6W2H、FV 表についての情報補足を行う。

1.1. HAYST 法のテストプロセスについて

HAYST 法とは、「因子」と「水準」を抽出し、これらの組み合わせを網羅するための直交表に割り付けを行い、これをもとにテストを実施するソフトウェア組み合わせテスト技法である[8]. 「因子」と「水準」を抽出する「テスト分析」と「テスト設計」、組み合わせテスト作成とテスト実施をする「テスト実装」とテスト実施までの全てのプロセスをまとめて HAYST 法と呼ぶ[9]

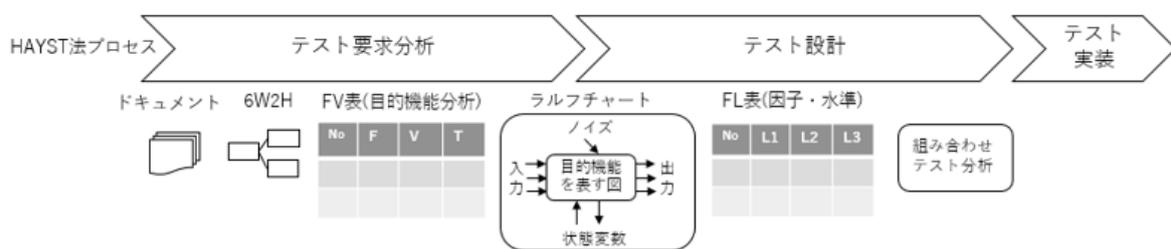


図 1 HAYST 法のプロセス

1.2. HAYST 法の 6W2H について

機能仕様書などのテストベースとなるドキュメントを元に、テスト要求を理解するための方法として「6W2H」のツリーを作成する。「6W2H」ではテスト対象と、テスト目的の両面からテスト観点の詳細化を行う。テスト対象を理解してテストすべきことを明確にし、ユーザーの目的に叶うかどうかを抜け漏れなく洗い出すテスト要求分析となる。

- What? : 機能仕様 (正常, 異常)
- When? : いつ使われるか
- Where? : どこで使われるか
- Who? : 誰が使うのか
- Whom? : 誰のために使うのか
- How? : どのように使うのか
- How much? : 価格はどれくらいのソフトウェアなのか

1.3. HAYST 法の FV 表について

FV 表とは HAYST 法によるテスト要求分析の手法であり「システムを機能が果たす目的の切り口で分割した単位(以降, 目的機能と表記)」で分解・整理を行うことで, 目的機能(F)ごとに検証すべき条件(V)を一覧にまとめた表である。FV 表に「機能」ではなく, 「目的機能」を記入する意味は, 開発とテストの違いに由来する。ソフトウェアテストでは, 機能が仕様通りに「正しく動作」することの確認はもちろん, ユーザーが求める「正しい動作」を確認する必要がある。「正しい動作」とは, その機能が, ユーザーの目的(=したいこと)を達成することができるということである[8].

FV 表の基本のフォーマットは表 1 に示す通り「No.」「目的機能(F)」「検証内容(V)」「テ

スト技法(T)」の4列である。図2にFV表のサンプルイメージを示す。なお、FV表には、「列数が多くならないため扱いやすい」「シンプルであるために共通認識が持ちやすい」「機能が持っている目的ごとにまとめたテスト分析が行える」などのメリットがある。

表1 FV表

No.	目的機能(F)	検証内容(V)	テスト技法(T)

FV表のそれぞれの列の意味については、以下の通りである。

- No. : 機能仕様書の対応 No. かつ レビュー版 FV 表の番号
- 目的機能(F) : お客様は何をしたいのか? (Why?)
- 検証(V) : 何を確認したら機能したといえるのか? (What?)
- テスト技法(T) : どのテスト技法で検証するのか? (How?)

番号	目的機能 (F)	検証 (V)	テスト技法 (T)
1	一人の相手の情報を、1つの連絡先としてまとめて管理したい	<ul style="list-style-type: none"> ・新しく「連絡先の追加」を行い、連絡先情報を項目ごとに新規登録できること ・項目情報が“ある場合/ない場合”を組み合わせ、1つの連絡先として利用できること ・ラベル情報が“ある場合/ない場合”を組み合わせ、1つの連絡先として利用できること 	単機能テスト 無為組み合わせ
2	相手を絞り込むための情報として、名前を登録したい	<ul style="list-style-type: none"> ・入力フィールドに、入力した名前が反映されること ・文字種類や文字数(0~多)などの条件に依り、入力した内容が登録されること ・敬称などの付加情報の有無を組み合わせ、指定した名前が登録されること 	単機能テスト 無為組み合わせ

図2 FV表サンプルイメージ

2. テスト分析・設計の有無の有効性の評価 実験結果

ここでは本論文4章にて行った実験結果のうち、付録となる表2~6の情報を示す。

2.1 実験結果

表2は、被験者別にそれぞれ合計工数とテスト実装数の比較を示している。[A. テスト分析・設計なし]では合計工数は最小7.7hから最大27h、テスト実装数は最小167ケースから最大612ケースと被験者により大幅に差異がある。また[B. テスト分析・設計あり]では、合計工数は最小25hから最大32h、テスト実装数は最小668ケースから最大704ケースとテスト設計を行っていない場合よりも差異の幅が小さい結果となった。

表3は、レビュー版FV表にて、被験者別にそれぞれ抽出した目的機能の件数と、レビューを行い対象外とした目的機能を示す。被験者によっては「目的機能」を「要求」や「要求から導き出される期待動作」を書いていたケースがあり、期待動作を記載された場合、抽出も細かくなり件数が多く記載されている傾向が出た。なおレビューにて、研究員全員で合意した目的機能数は[26]である。

表4および表5については、被験者別にそれぞれ機能別のテスト実装数の比較した内容である。表4の[A. テスト分析・設計なし]では[全体機能1~3]や[誕生日]、[追加項目1]のテスト実装数が0件である箇所が見受けられた。また[住所]や[追加項目2]のテスト実装数の標準偏差は50以上と、目的機能別でも大きく差異が発生している結果となった。表5の[A. テスト分析・設計あり]ではテスト実装数が0件となる機能は存在せず、また機能ごとの標準偏差についてもテスト設計を行っていない場合よりも差異の幅が小さい結果となった。

表6は、表4にて[A. テスト分析・設計なし]でテスト実装数が0件となった箇所の原因を機能別に洗い出し、一覧化した表である。利用者視点の考慮不足や、テストケースを間引く際に評価項目を落としてしまったケースが目立つ結果となった。

表 2 被験者別の工数とテスト実装数の合計値比較

	A. 分析・設計なし		B. 分析・設計あり	
	合計工数 単 位:時	テスト実装数 単 位:個	合計工数 単 位:時	テスト実装数 単 位:個
被験者 1	27.0	507	32.0	704
被験者 2	8.0	612	31.5	682
被験者 3	7.7	181	29.0	692
被験者 4	10.8	167	25.5	722
被験者 5	15.0	212	30.0	668
平均	13.7	336	29.6	694
標準偏差	7.1	186.2	2.3	18.5

表 3 レビュー版 FV 表の 被験者別の抽出した目的機能数・対象外数

レビュー版 FV 表	被験者 1	被験者 2	被験者 3	被験者 4	被験者 5
各人が抽出した目的機能の数	20	34	28	54	15
対象外にした目的機能数の数	0	14	10	28	4

確定した目的機能数: 26

表 4 「A. 分析・設計なし」 被験者別の機能別のテスト実装数比較

機能	被験者 1	被験者 2	被験者 3	被験者 4	被験者 5	標準	
						平均値	偏差
全体機能 1	0	0	8	0	0	1.6	3.2
全体機能 2	76	0	0	3	12	18.2	29.2
全体機能 3	0	0	0	0	0	0	0
画像	31	9	5	39	10	18.8	13.6
名前	21	37	53	10	16	27.4	15.6
メール	19	24	10	8	16	15.4	5.9
電話番号	31	72	16	8	50	35.4	23.2
住所	29	150	56	18	12	53	50.8
誕生日	0	21	14	12	0	9.4	8.2
URL	21	40	13	8	12	18.8	11.4
メモ	1	6	6	1	8	4.4	2.9
追加項目 1	0	0	0	0	12	2.4	4.8
追加項目 2	278	253	0	60	64	131	112.4

表5 「B. 分析・設計あり」 被験者別の機能別のテスト実装数比較

機能	被験者 1	被験者 2	被験者 3	被験者 4	被験者 5	平均値	標準 偏差
全体機能 1	80	79	80	80	74	78.6	2.3
全体機能 2	88	74	75	74	80	78.2	5.4
全体機能 3	32	32	18	21	32	27	6.2
画像	24	20	22	29	26	24.2	3.1
名前	102	92	103	92	96	97	4.7
メール	25	24	22	25	20	23.2	1.9
電話番号	46	44	46	45	40	44.2	2.2
住所	58	58	58	61	58	58.6	1.2
誕生日	9	11	11	14	10	11	1.7
URL	84	84	84	87	84	84.6	1.2
メモ	49	49	49	46	42	47	2.8
追加項目 1	21	23	24	24	20	22.4	1.6
追加項目 2	86	92	100	124	86	97.6	14.2

表6 「A. 分析・設計なし」機能別のテスト実装数が[0]となった箇所の原因

テスト実装を漏らした理由	漏れた目的機能数
各項目の機能に目を向けすぎて製品全体としての考慮不足	7
見た目の操作、動作にとらわれすぎユーザーの視点の考慮不足	4
利用時のシナリオテストでカバーする内容が考慮されていない	3
評価項目を個人のさじ加減で間引き、結果的に漏れに繋がった	2
システム環境条件、入力規則の条件に対する考慮漏れ	2