

テストマトリックス

作成日: 2017年2月11日
作成者: 森 健太

■ テストマトリックスとは

テストマトリックスとは、テストを行う際に利用する表で、テスト対象となる項目（カラム）を全て列挙し、全組み合わせを網羅的に表現した碁盤の目のような（Excelやテーブル表のような）ものです。

一般的に表内の升目には、「○」「×」「-」などテスト対象かそうでないかを表す文字や、「OK」「NG」の結果を記載するなどして利用します。

	ID	パスワード	名前	出身	結果	判定
1	○	○	○	○	○	OK
2	○	○	○	×	○	OK
3	○	○	×	×	○	NG
4	○	×	×	×	×	OK
5	×	×	×	×	×	OK

■ テスト設計

テスト設計とは「**どのようにテストを行うか**」を検討し設計することです。

テスト戦略やプログラム特性やデータ分析を元にテスト方針を定め、テストケースを作成します。

テスト設計には、3つのポイントがあります。

- ・ **効率の良さ**（テストケースの少なさ）
- ・ **漏れのない網羅性**
- ・ **バグの検出率**

テストマトリックスはこの中の「**漏れのない網羅性**」を担保するために利用されます。

■ メリデメ

○メリット

- ・ 網羅的に行うので漏れを最小化できる。
- ・ 組み合わせの整理ができる。

○デメリット

- ・ 項目数×項目数になるので、ケースが膨大になる。
- ・ 膨大になればなるほど、記載方法によっては漏れが発生する。
- ・ 記載方法によっては網羅されているのか、漏れているのかの確認が煩雑になる。

■ マトリックス記法の重要性

マトリックスは項目（縦横の軸）が増えればその分テストケースが増えます。そのため、**書き方によっては漏れが発生する**だけでなく、それが正しく網羅されているのかも不明瞭になってしまいます。

つまり、マトリックスの記載方法がいい加減であればあるほど、そのマトリックスは意味を失い、本来の利用価値である「網羅性」が担保されなくなります。

網羅性が担保された記載方法は非常に単純です。簡単なルールさえ覚えれば記載できるので、次ページからの例をみて確認してください。

■ 良いマトリックス悪いマトリックス

○例 1 : 簡単な組み合わせ

No	性別	血液型
1	0:男性	1:A型
2	1:女性	2:B型
3	0:男性	3:AB型
4	1:女性	4:O型
5	1:女性	1:A型
6	0:男性	2:B型
7	1:女性	3:AB型
8	0:男性	4:O型

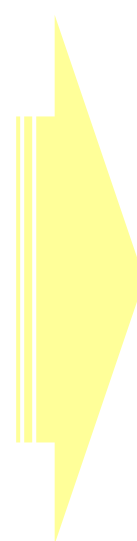
並びに規則性がなく
漏れがわかりづらい。

No	性別	血液型
1	0:男性	1:A型
2	0:男性	2:B型
3	0:男性	3:AB型
4	0:男性	4:O型
5	1:女性	1:A型
6	1:女性	2:B型
7	1:女性	3:AB型
8	1:女性	4:O型

規則性はでたが
一見ではわかりづらい。

No	性別		血液型			
	男性	女性	O型	AB型	B型	A型
1	○	×	×	×	×	○
2	○	×	×	×	○	×
3	○	×	×	○	×	×
4	○	×	○	×	×	×
5	×	○	×	×	×	○
6	×	○	×	×	○	×
7	×	○	×	○	×	×
8	×	○	○	×	×	×

まだ見辛い。



No	性別		血液型			
	男性	女性	O型	AB型	B型	A型
1	●	—	●	—	—	—
2	●	—	—	●	—	—
3	●	—	—	—	●	—
4	●	—	—	—	—	●
5	—	●	●	—	—	—
6	—	●	—	●	—	—
7	—	●	—	—	●	—
8	—	●	—	—	—	●

これで漸く規則性がで
組み合わせが明確に。



No	性別		血液型			
	男性	女性	O型	AB型	B型	A型
1	●	—	●	—	—	—
2	●	—	—	●	—	—
3	●	—	—	—	●	—
4	●	—	—	—	—	●
5	—	●	●	—	—	—
6	—	●	—	●	—	—
7	—	●	—	—	●	—
8	—	●	—	—	—	●

①：対象群が縦に揃うようにする。
左から右にズラして記載していく。

②：左から粒度の大きい（駆動項目）順で並べる。

③：視点は左から右、上から下へ向かうので右下下がりにする。

■ 良いマトリックス悪いマトリックス

○例2：複数パターンマッチ

No	名前入力チェック			
	必須	半角	全角	桁
1	●	—	—	—
2	—	●	—	—
3	—	—	●	—
4	—	—	—	●
5	—	●	●	—
7	—	—	●	●
8	—	●	●	●

一見網羅されてそうだが...

No	名前入力チェック			
	必須	半角	全角	桁
1	●	—	—	—
2	—	●	—	—
3	—	—	●	—
4	—	—	—	●
5	—	●	●	—
6	—	●	—	●
7	—	—	●	●
8	—	●	●	●

組み合わせとしては漏れている。

No	名前入力チェック			
	必須	半角	全角	桁
1	●	—	—	—
2	●	●	—	—
3	●	●	●	—
4	●	●	●	●
5	●	—	●	—
7	●	—	●	●
8	●	—	—	●
9	—	●	—	—
10	—	●	●	—
11	—	●	●	●
12	—	●	—	●
13	—	—	●	—
14	—	—	●	●
15	—	—	—	●

簡単な組み合わせを元に整理して記載する。
発生しえないパターンについても省略せずに
記載することで網羅性が一目で確認できる。



No	名前入力チェック				備考
	必須	半角	全角	桁	
1	●	—	—	—	
2	●	●	—	—	No.1で未入力チェック済みのため
3	●	●	●	—	No.1で未入力チェック済みのため
4	●	●	●	●	No.1で未入力チェック済みのため
5	●	—	●	—	No.1で未入力チェック済みのため
7	●	—	●	●	No.1で未入力チェック済みのため
8	●	—	—	●	No.1で未入力チェック済みのため
9	—	●	—	—	
10	—	●	●	—	
11	—	●	●	●	
12	—	●	—	●	
13	—	—	●	—	
14	—	—	●	●	
15	—	—	—	●	No.11でチェック済みのため

ありえない(発生しえない)パターンは備考に記載してグレーアウトする。その他、複雑なパターンなどがあれば備考に記載するようにする。

■ グレーアウトする

前ページで

「ありえない(発生しえない)パターンは備考に記載してグレーアウトする。」と、記載しました。

画面などでは特にそうですが、1項目の挙動に連動して入力などができない場合があります、確認しようにもできないものがあります。当然、確認できないのでグレーアウトします。

それ以外にも、1パターンを確認すれば他パターンと通過するロジックが明らかに同じ場合などはグレーアウトします。

この様にして、テスト設計のポイントである**網羅性と効率化と言う相反する課題をクリア**していきます。

何でもグレーアウトして本来の意味を失わないように注意しましょう。

■ 単体テストに反映する

これまで確認したことを踏まえて単体テストに活用する。

No	ID	パス	入力チェック							結果		実施状況			備考
			必須		半角		全角		桁	OK	エラー	担当者	実施日	結果	
			あり	なし	あり	なし	あり	なし							
1	●	—	●	—	●	—	●	—	●	—					
2	●	—	●	—	●	—	●	—	—	—	●				
3	●	—	●	—	●	—	—	●	●	—					
4	●	—	●	—	●	—	—	●	—	—	●				
5	●	—	●	—	—	●	●	—	●	—					
7	●	—	●	—	—	●	●	—	—	—	●				
8	●	—	●	—	—	●	—	●	●	—					

■ まとめ

テストはシステムの規模が大きくなればなるほど、非常に煩雑で面倒で工数がかかる作業です。

特にテストマトリックスの作成は、項目数や確認項目が増えると気が滅入るほどのボリューム感になります。

ですが、テストマトリックスを作成することによって、確認すべき項目が網羅的に洗い出されるだけでなく、どうすれば効率的に確認できるかの整理にも利用できます。

■ まとめ

そもそも、何故テストが必要なのか。
当然、バグや仕様漏れがないかを確認するためです。

しかし、それだけではなく、経験が浅いうちは特にですが
テストはシステムを理解する格好の教材になります。

テストパターンなどを「**正確に作成できる = 正確な設計ができる**」
なので、その辺りを意識して作業するだけで大きく変わってきます。

これは単体・結合・総合など、そのテストフェーズに関係なく
**機能的・業務的それぞれの観点を意識することで、システム構成・
業務知識・アルベキなどの経験則の精度が磨かれる**ということなのです。

テストの意義を再度見つめなおし、システムへの造詣を深めましょう！