

# テスト設計コンテスト 発表資料 (JaSST'11 Tokyo)

Team 66-Kashiwa  
(山上 直宏)

## 目次

1. はじめに
2. テスト設計の目的
3. テスト設計の方向性(基本方針)
4. テスト技法
5. 優先順位
6. カバレッジ
7. 粒度(どこまで設計するか)
8. 工数
9. テスト容易性 / 環境・ツール
10. テスト設計詳細(アウトプット)
11. 同値クラス
12. レビュー

## 1. はじめに

### ■参加経緯

TEF MLのメールから、つい出来心で参加してしまいました。

### ■普段の仕事

複合機に内蔵されるプリントエンジン部のソフトウェアテスト

### ■手法の成り立ち(経緯)

ある日突然、他部署で手に負えなくなったソフトウェアを持ち込まれ、「1週間でテストして全部たたきだして！」という、非常に苛酷なストレス環境下で生まれました。

## 2. テスト設計の目的

インプットが「要求仕様書」のみということから、対象を“機能確認”に絞った。

### ■その理由

例えば・・・

ユーザビリティ⇒ユーザ層や設置環境など市場環境に関する情報が必要(商品企画書など)。

信頼性⇒市場で想定される使われ方、これまでの出荷実績から推定される必要なMTBF等が必要。

⇒これらを含めると仮定・想定が多くなってしまいうため今回は除外した。

## 3. テスト設計の方向性

### ■基本的にはこれまでの「おさらい」と考える。

- 要求仕様と合致しているか？(正常系)
- 要求仕様の行間(隙間)を埋められているか？(未記述・異常系)

### ■設計時の留意点

なるべくシンプルに工数をかけ過ぎないようにする。

(ソフトウェア自体がシンプルなため、テスト設計自体に工数をかけるのは?)

## 4. テスト技法

方向性のうち、「要求仕様の行間を埋められているか？(未記述・異常系)」を満足するため、状態遷移表を使用する。

状態遷移表

イベント	E <sub>1</sub>	E <sub>2</sub>	...	E <sub>n</sub>
状態 S <sub>1</sub>	-	A <sub>1</sub> /S <sub>1</sub>	-	-
S <sub>2</sub>	-	-	-	A <sub>n</sub> /S <sub>1</sub>
S <sub>m</sub>	A <sub>2</sub> /S <sub>2</sub>	-	-	-

(S: 状態, E: イベント, A: 動作, -: 不正な遷移)

出典: Wikipedia (状態遷移表)

### 状態遷移表の特徴(概要)

メリット: モレ・抜けを顕在化しやすい。

デメリット: 状態・イベントが増えると項目数が膨大になる。

## 5. 優先順位

以下の観点から優先順位を付ける。  
(あまり複雑にすると、順位付けに工数がかかるため、シンプルに)

1. 正常系(通常操作範囲)
2. 異常系(故障・エラー等動作継続できなくなると想定されるもの)
3. 異常系(故障・エラー等動作継続できなくなると想定されないもの)

## 6. カバレッジ

機能確認を主目的にしているため、  
「要求仕様網羅率:100%」を目標とした。

具体的には要求仕様書の以下の項目に記載されている機能に関して網羅するように設計した。

2. 操作要求仕様
3. 温度制御行為
4. 温度制御方式
5. エラー検知

## 7. 粒度

どの程度まで細かく設計するか？  
⇒一番悩みました。

段階的詳細化の目的の一つである、  
「再利用性の向上」を意識し、以下のように設定した。



類似製品(ソフトウェア)であれば設計を小改造で流用できる粒度。

具体的には・・・  
状態遷移表の状態・イベントを割り出すところまで行う。  
実際の結果・実施順序決定はテスト実装にて行う。

## 8. 工数

以下の方法により、概算工数(2.5人日)を算出した。

全体項目数を状態・イベント数から算出 ⇒  
有効/無効操作比率を概算 ⇒  
上記の数値を想定工数に掛け合わせる

項目数  
状態数:31、イベント数:29 ⇒全体項目数:899  
時間当たり消化項目数(所要時間)  
有効操作:1項目/分(1.00分)  
無効操作:6項目/分(0.17分)  
有効操作/無効操作比率  
(例)状態遷移表全体から概算:54:46  
故に概算工数:約6時間(1人日)  
実装工数:約9時間(1.5人日)・・・前提:有効1項目/分、無効4項目/分

## 9. テスト容易性 / ツール・環境

■容易性  
操作部(ボタン・表示機能)は目視で確認できるため特に問題はない。  
温度制御等内部の動作は直接目視できないため、後述するツール・治具があると良い。

■ツール・環境  
以下のようなツール・治具があるとテストが容易になる。  
ログ出力機能  
⇒内部状態、サーミスタ検知温度の確認に必要。  
サーミスタ温度設定治具  
⇒温度制御、温度表示、エラー検知機能の確認に必要。  
水位センサ設定治具  
⇒水位表示、空焚き防止機能の確認に必要。

## 10. テスト設計詳細(1/3)

要求仕様から状態とイベントを割り出す作業。

⇒ただし通常の状態遷移表ではテストに不都合なため、仕様書に無い状態・イベントを拡張している。

例えば・・・  
一定周期でエラーチェックを行う場合  
周期タイマイベントで遷移する「エラー条件判断状態」を追加する。  
など。

