

テストのスパイス教えます ～ちょっとした手間で変わる仕上がり～

Powered by TEF-Do

1

How do you like software test?

正式名称：TEF北海道ソフトウェアテスト勉強会
(Testing Engineer's Forum HOKKAIDO)

2006年ごろ発足。

テストをメインにソフトウェアの品質に関わること全般の勉強。

- オンラインで議論
 - テスト設計技法の勉強したり、要求分析を勉強したり、最近の話題はレビュー
- 不定期で勉強会開催
 - ほぼすべて手弁当で対応。会費は無料。
- 他地域勉強会連携
 - 関西からゲストが来たり、東海でSIGしたり

参加メンバーは10名程度で、常時5～6名のメンバーが参加。

メンバは常に募集中。

<http://ameblo.jp/tef-do/tefdoblog@gmail.com>

2

こんなことになっていないですか？

- 後からバグが出て、テストが漏れていたと後悔したことはありませんか？
- テストケースを書いていて不足していないか不安になったことはありませんか？
- コピペでテストを作ってるけど、これでいいのか悩んでいる

TEF道より

あまり工数を使わずに効果的なテストが
実施できる方法をご紹介します

3

こんな感じでテストを作ってませんか？

1. 機能仕様書を準備
2. テストケース仕様書のテンプレートを準備
3. 機能仕様書から「機能」部分をコピー
例) 対象機能
「ボタンAを押下するとトップ画面が表示される」
4. テストケース仕様書に張り付け、語尾を「～すること」に変更する。
例) テストケース
「ボタンAを押下するとトップ画面が表示されること」

これを「Copy&Paste&Modify法 (CPM法)」と呼ぶ人もいます……

どこかでみたことがあるような……

何かが足りないテストケース

5

こんな状況があったとします

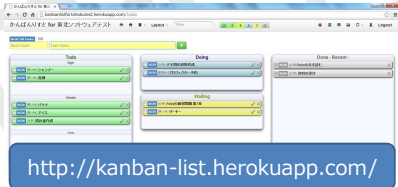
- 皆さんはソフト開発のチームに所属しています。
- あるソフトの「一部の機能」に対するシステムテストを行おうとしています。
- このテストで行うテストケースの作成に新人君をアサインしました。

6

あるソフト=テストベース 「かんぱんりすと」

「かんぱんりすと」って？

- 小玉 直樹 氏 作のWeb上で使える ToDoリストアプリです。
- タッチ対応ですのでスマートフォンやタブレット等でもご利用いただけます。



今回テスト対象機能

テスト対象

- TOP画面
- サインアップ
- ログイン
- タスクの一覧表示
- タスクの登録
- タスクの更新(内容)
- タスクの更新(状態)
- タスクの削除

テスト対象外

- タスクのFilter機能
- Book(カテゴリ)の操作
- レイアウトの変更
- dashboard
- ユーザー情報の変更
- ログアウト
- 付加機能詳細
- ポモドーロタイマー
- Done List一覧表示
- タスク一覧のEmail送信
- 背景イメージの変更
- 自動ロードのON/OFFの切り替え

対象機能の使用については「資料① 要求仕様書」をご覧ください

新人君からこんなテストケースがでてきた

No	大項目	中項目	小項目	テスト内容	結果
1	TOP画面	ユーザー登録ボタン	押下	URLからユーザー登録ボタン押下でユーザー登録画面に遷移できる	
2		ログインボタン	押下	URLからログインボタン押下でログイン画面に遷移できる	
3	サインアップ	Name	入力	Nameが入力できる	
4		Email	英数字入力	Emailに英数字が入力できる	

ワーク①a (個人ワーク)

15分

このテストケースのレビューをしてください。

下記の観点でレビューしてください。

- システムテストとして、テスト内容は十分足りているか？ 不足していないか？
- テストケースの品質として問題が無いのか？（テスト実施やレビューのしやすさなどを含めて）

実際のアプリを動かしてみてもOKです！！



ワーク①b (グループワーク)

10分

グループでレビュー結果を共有してください。

- どんなテストケースが必要か？
- 質の高いテストケースって何だろう？
- このテストケースの作り方において、考えが足りていない事はなんだろう？

どうでしたか？

- ほぼ仕様書からのコピペでしたよね？
- 足りないテストがありましたよね？
- たとえば、こんな指摘が生まれませんでしたか？
 - 数値項目の境界値のバリエーションが不足している
 - 入力可能項目が列挙されている機能の、入力内容が不明
 - テスト実施不可or実施方法が不明なテストが存在している
 - 対応ブラウザやセキュリティなどの非機能部分の考慮がされていない
 - 異常系はテストしなくて良いのか？

理想的なテストってなんだろう？

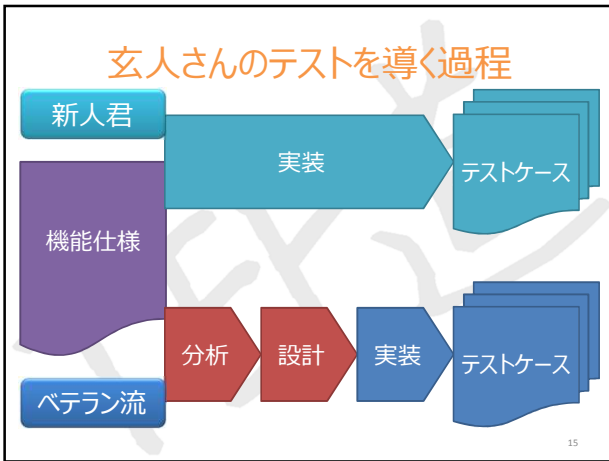
網羅的で抜けの少ないテストケース

13

テストケースの比較

新人君のダメな点をもっと明確にするために、ベテランさんのテストの作り方を見てみましょう。

14



玄人さんの分析

ユーザ分析してユーザゴールを考える

想定ユーザー像

IT企業の開発関連者

代表的なユーザー像

- 30～40代
- 業務の改善を日ごろから考えており、その推進に対してもある程度の影響力をもつ。
- 客先やオフショア先で作業を行う事が多いので、外出先やスマートフォンで自分のタスク管理を行う必要性を感じている。

ユーザーゴール

テストの最終目的

WEB上でプロジェクトに関するTODOや備忘録を管理することで、作業環境に依存せずプロジェクトに関するタスク情報を把握できる。

16

玄人さんの分析

ユーザゴールから保証すべき品質を導く

<p>テストの最終目的</p> <p>ユーザーゴール</p> <p>WEB上でプロジェクトに関するTODOや備忘録を管理することで、作業環境に依存せずプロジェクトに関する情報を把握できる。</p>	<p>ゴールの要素分解</p> <p>ユーザー要求 (満足性)</p> <p>プロジェクトに関するTODOなどのタスクを、いつでもどこでも把握したい。</p> <p>アプリケーションのインストールには社内手帳が必要のため、特別なプラットフォームやアプリインストールを不要にしたい。</p> <p>思いついたタスクをすぐに登録したいので、起動や操作を迅速に行いたい。</p> <p>社内情報を扱うので、機密情報を漏らさない仕組みが必要</p>	<p>品質特性 (抜粋)</p> <table border="1"> <tr> <th>機能適合性</th> <th>機能完全性</th> <th>機能正確性</th> <th>機能適切性</th> </tr> <tr> <td>製品やシステムが定められた利用状況下において、決められたニーズを実現する機能を提供する適合</td> <td>タスクを完了させるために必要な機能すべてがユーザの目的に一致している場合</td> <td>システムソフトがユーザの目的に一致している場合</td> <td>システムソフトがユーザの目的に一致している場合</td> </tr> </table> <p>ユーザ要求に対して、システム側で考慮する必要がある要素</p>	機能適合性	機能完全性	機能正確性	機能適切性	製品やシステムが定められた利用状況下において、決められたニーズを実現する機能を提供する適合	タスクを完了させるために必要な機能すべてがユーザの目的に一致している場合	システムソフトがユーザの目的に一致している場合	システムソフトがユーザの目的に一致している場合
機能適合性	機能完全性	機能正確性	機能適切性							
製品やシステムが定められた利用状況下において、決められたニーズを実現する機能を提供する適合	タスクを完了させるために必要な機能すべてがユーザの目的に一致している場合	システムソフトがユーザの目的に一致している場合	システムソフトがユーザの目的に一致している場合							

17

玄人さんの分析結果 -1

- ユーザーゴール
 - WEB上でプロジェクトに関するTODOや備忘録を管理することで、作業環境に依存せずプロジェクトに関する情報を把握できる。
- ユーザー要求 (満足性)
 - プロジェクトに関するTODOなどのタスクを、いつでもどこでも把握したい。
 - 重要** アプリケーションのインストールには手続きが必要のため、特別なプラットフォームやアプリインストールを不要にしたい
 - 思いついたタスクをすぐに登録したいので、起動や操作を迅速に行いたい
 - 重要** 社内情報を扱うので、機密情報を漏らさない仕組みが必要

18

ワーク②a (個人ワーク&グループワーク)

15分

CPM+αの「テストケース仕様書（資料②）」のテストを「テストケース分析シート（資料③）」に当てはめてください。

- テストケースの「テスト対象機能」毎に、「どんなテストが実施されているか」を分析します
- マトリクスの縦軸は「機能」です。テストケースの「テスト対象機能」と（ほぼ）一致しています。
- 横軸のテストカテゴリをヒントに、テストケースから「テストの目的」を読み取ります
- 一致する箇所に○をつけていきます
- レビュー指摘で導いたテストケースについては×をつけます

25

ワーク②a (個人ワーク&グループワーク)

15分

CPM+αの「テストケース仕様書（資料②）」のテストを「テストケース分析シート（資料③）」に当てはめてください。

- テストケースの「テスト対象機能」毎に、「どんなテストが実施されているか」を分析します
- マトリクスの縦軸は「機能」です。テストケースの「テスト対象機能」と（ほぼ）一致しています。
- 横軸のテストカテゴリをヒントに、テストケースから「テストの目的」を読み取ります
- 一致する箇所に○をつけていきます
- レビュー指摘で導いたテストケースについては×をつけます

26

どうでしたでしょうか？

CPM+αでは……

- 機能テスト…それも目に見える部分に偏っていませんか？
- 非機能部分やセキュリティの考慮は出来ていないですね？
- 異常系や環境などの考慮も…
- そもそも、仕様に書いていない内容は一切考慮できていない…

分析、設計を行って導いた確認すべき項目とCPM+αの項目を比較してみてください。
違いがあった場合、その違いはCPMやレビューから導くことができるでしょうか？グループ内で議論してみましょう。

27

CPMの特徴

- 良いところ
 - 早い
 - 簡単
- 悪いところ
 - 書いていない事はテストできない
 - 機能組合せや状態遷移などの、機能間の連携の漏れが生じる
 - 非機能やセキュリティや異常系が漏れやすい
 - レビューがしづらい

28

(少なくともCPMよりは) 理想的なプロセスの特徴

- 良いところ
 - 機能仕様だけではなく、利用者観点からの分析が行える
 - 中間成果物を明確にしているので、プロセス単位での内容合意が可能
 - 全体を俯瞰できるので、レビューが容易
 - 抜け漏れを防止しやすい
- 悪いところ
 - 時間がかかる
 - ある程度の技術が必要

29

ちゃんとテストプロセスを やらないといけないのはわかっているが……

- 理想的には横軸（＝ヒントワード）にあたる部分を導くプロセスが必要です。
- でも、そこまでできない、何やって良いかわからない、ということがあられると思われ、結局CPMにもどる…。
- そんな風にならないための最初の一歩的に、比較的汎用的に使えるヒントワードをご紹介します。

30

ヒントワード・例

- それはあたる
(書籍「事例とツールで学ぶHAYST法—ソフトウェアテストの考え方と上達のポイント」より)
- 非機能カテゴリ
 - 排他/同期/容量/メモリ量/タイミング…など
- 意地悪漢字
 - 「再」「特」「略」「誤」…など
(<http://www.jasst.jp/archives/jasst10s/pdf/S3-9.pdf>)
- エラー推測
 - 出やすいバグを推測する(メモリーク/競合/繰り返し/…など色々)
- HAZOPのガイドワード
 - 化学プロセスにおける危険シナリオ分析手法をソフトウェアに応用
- 品質特性の応用
(ISO/IEC 25010 <http://kikakunui.com/x25/X25010-2013-01.html>)
- TEF道作成の「ぶっちゃけ品質特性」
 - 魅力性⇒「かっこいい、かわいい」
 - 環境適応性⇒「いろんなところで使える」 …など、ぶっちゃける

31

少しでも理想に近づきたい

一歩進んだテストケース

32

一例として「それはあたる」の紹介



- コーナーケースの入力値を探す方法。
- ①れ (例示) : 怪しい部分の例示を考える
 - ②あ (間) : 例示したデータの間を考える
 - ③た (対称) : 例示したデータに軸足を置いて、もう片方の足の降ろし場所を探す
 - ④る (類推) : 例示から離れてテスト対象を眺める
 - ⑤そ (外側) : テスト対象から外や補集合を探す
 - ⑥は (破壊) : テスト対象が破壊する場合を探す

33

「それはあたるの」例題

西暦⇒和暦の変換WEBプログラム

西暦⇒和暦の変換

西暦 2014 年 8 月

変換

和暦 平成 24 年

<仕様>
西暦部に「西暦での年」と「月」を入力し、「変換」ボタンを押下すると和暦部に対応した「年号」と「和暦での年」が表示される。

①仕様の「怪しい箇所」にチェック(できれば色つきボールペンで)して、怪しい内容をメモしてください。

- <例>
- ・「西暦での年」⇒入力範囲は？入力内容のチェックは？
 - ・「月」⇒入力範囲は？入力内容のチェックは？
 - ・「日」は無くても良いのか？(月途中で年号が切り替わる場合は？)
 - ・「和暦」⇒元号は、どこまでが対象なのか？明治から？それ以前は？
 - ・「変換ボタン」は、マウスのみ？キーボード対応している？

34

ワーク③ (個人ワーク⇒グループワーク) 20分

- 「それはあたる」を使って、CPMでは導出できないテストの視点を考えてみましょう
 - 演習① データの入力に関して (要求3)
 - 演習② 操作に関して (要求7)

35

どうでしょうか？

- CPM + aで導かれなかったテストの視点ができませんか？
 - このヒントワードを使うと、主に異常系の視点が(強制的に)得られると思います

36

まとめ（スパイスとは？）

- はじめの一步として、汎用的なヒントワードをつかう一時間で、CPM + aのaが強化されるでしょう
↑一味変える「スパイス」を、まずはお試しあれ
- ただし、同じことをやりつづけると一時間の効果がなくなります。（殺虫剤のパラドクス）
- 理想的な方法で導いた横軸のように、テーラーリングして育てることは将来的に必要です。
↑「秘伝のスパイス」を作り上げてください

37

理想を求めて

本ワークで伝えきれていない事

38

横軸について

横軸の「テストカテゴリ=ヒントワード」は、対象製品や組織で変わります。実はこれを大事に育てることが大切です。

【育て方の例】

- 担当者のノウハウや暗黙知を盛り込む
- プロジェクトの過去のデータの集計結果を盛り込む
- 出荷後の不具合などの是正内容をフィードバックして盛り込む
- 上記は、いったん抽象化して盛り込む

39

縦軸について

今回の縦軸は、異なるモデリングのうちの一部にすぎません。本来であれば、縦軸は対象システム分析を行い、再構築が必要なのです。

【モデリングの例】

- 機能モデル
利用者から見たシステムの機能を表すモデル
- 静的モデル
問題領域やシステムの静的構造を表すモデル
- 動的モデル
振る舞いを表すモデル

40

参考文献

- ソフトウェア・テストPRESS Vol.10
-「今こそ聞きたい テストの上流設計」
(湯本 剛氏)
- 事例とツールで学ぶHAYST法—ソフトウェアテストの考え方と上達のポイント
(秋山 浩一氏)



How do you like software test?

ありがとうございました！

42