

バグから始める テスト設計

2017/09/08

株式会社 ウェブレッジ

小田部 健

アジェンダ

1. 一般的なテスト設計の問題点
2. 初心者向けのテスト設計とは？
3. 逆さFujiメソッドの解説
4. 逆さFujiメソッドまとめ

1. 一般的なテスト設計の問題点

テスト設計コンテストで紹介されているテスト設計プロセス

テスト
要求分析

アーキテクチャ
テスト
設計

テスト
詳細設計

テスト
実装

テスト
実施

1. 一般的なテスト設計の問題点

- ・ 実施に必要なスキルレベルと作業量で表現すると...



1. 一般的なテスト設計の問題点

問題点：

ハイレベルのスキルと大量の作業が必要になる

- 一般的なテスト設計の難しさ
 - 最初のプロセスで高度なスキルと知識を要求され、困る
 - テスト空間を広く深くテスト設計している内に、迷う
 - 成果物同士のトレーサビリティを確保しようとして、絡まる

結論：

初級テストエンジニアに全く優しくない



2. 初級テストエンジニアに優しい テスト設計とは

- 設計のハードルが低い
 - 最低限必要なスキルや知識のレベルが低い
- 作業時に迷いにくい
 - 厳選された作業の種類と量
 - 適切なテスト設計サポート
- 設計スタイルが柔軟
 - 自分のスキルレベルに合わせて工夫を加えやすい

3. 逆さFujiメソッド ～バグから始めるテスト設計～

1. テスト対象のバグを想定する
2. 検出すべきバグを決定する
3. バグの発生原因を分析する
4. テストケースを仮実装する
5. 不足を補完して本実装する

バグの
想定

バグの
決定

発生原因の
分析

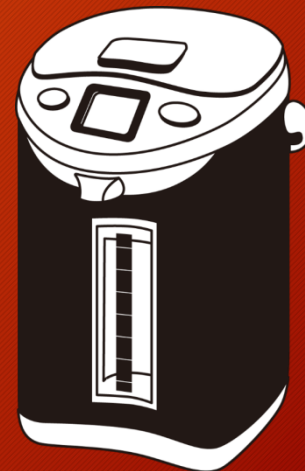
テスト
仮実装

テスト
本実装

※逆さFujiメソッド（及びFujiメソッド）は紙印テスト倶楽部で提唱しているテスト設計手法です

3. 電気ポットで学ぶ 逆さFujiメソッド

- 電気ポット仕様
 - 給湯温度は高温または中温に設定可能
 - 切替えボタン、高温ランプ、中温ランプ
 - 加熱と保温が可能
 - 加熱ランプ、保温ランプ
 - 沸騰後に設定温度まで冷ます
 - 給湯ボタンで給湯
 - 給湯ロック付き



3. バグの想定

Step
1/5

発生しそうなバグを想定してみる

• 外見や操作関連のバグ

- 表示と実際の動作が異なる
- 表記の意味が分かりにくい
- ボタンが固い

• 内部の構造や処理のバグ

- 加熱処理が遅い
- センサの感度が悪い
- 水が入って無くても加熱する

• 環境が影響するバグ

- 高所（低気圧）での使用で沸点が下がる
- 電源が海外の規格に非対応

• 利用者が原因のバグ

- オシャレな部屋に合わないデザイン
- 子供でも簡単にロック解除できてしまう

3.検出すべきバグの決定

Step
2/5

リスク等を勘案して
検出すべきバグを決定する

検出対象：
「給湯したお湯が設定温度と異なる」

※抽象度高めのバグが逆さFujiメソッド向きです

3. バグの発生原因の分析

Step
3/5

- どの様な原因で「給湯したお湯が設定温度と異なる」事象が発生するか分析する
- **バグは1つでも発生原因は複数ありえる**
 - 全ての発生原因を解決しないとバグが発生するリスクが残る
- **表示上の原因**
 - 表示上の設定温度と内部ステータスの設定温度が異なっていた
 - 表記が分かりにくく設定を間違えていた
- **ハード上の原因**
 - 熱センサの感度が鈍くて、検知した温度と実際の温度が異なっていた
 - 容器の保温性が悪く、保温時に設定温度を下回っていた
- **内部処理上の原因**
 - 加熱アルゴリズムの条件判定ミスで設定温度になる前に保温モードに遷移していた
 - 加熱中でも給湯ボタンを押したら給湯可能だった
- **環境上の原因**
 - 熱源が近くにあり、加熱終了後もお湯が設定温度まで中々下がらなかった
 - 電源が不安定で加熱が十分でなかった
- **使用上の原因**
 - コンセントを抜いたら設定が初期化されていた
 - 速くお湯を使いたくて設定温度になるまで待てなかった
- **テスト実施上の原因**
 - テストの実施手順が曖昧で、お湯の温度を測り間違えていた
 - テストで使用した温度計が壊れていた

3. テストケースの仮実装と本実装

Step
4 & 5/5

- **発生原因からテストケースを仮実装**してみる
 - バグの発生原因が分かれば、どうテストすべきかも大体分かる
 - 発生原因:「表示上の設定温度と内部ステータスの設定温度が異なっていた」
 - 仮実装:「設定温度の表示ランプと内部ステータスが一致するか確認する」
- 不足情報を**開発者等に確認してテストケースを本実装**する
 - テストケースを仮実装すると不足点が明確になる
 - 不足点:「設定温度の内部ステータスの確認方法は？」
- **これでテスト設計完了！**

4. 逆さFujiメソッドまとめ

一般的なテスト設計	逆さFujiメソッド
要求分析から始める	バグの想定から始める
網羅的に設計を進める	ピンポイントかつ探索的に設計を進める
設計に必要なスキルが多様かつ高度	バグを想定できれば設計を始められる
作業の量や種類が多い	作業の量や種類が発散しにくい
作成した成果物同士の関連づけが必要	「検出すべきバグ」で成果物同士が関連づけられている
段階的にテストレベルを進めて設計する	全テストレベルを平行して設計する
全プロセスの実施に多くの工数が必要	仮実装までなら1時間程度で設計可能

※逆さFujiメソッドは一般的なテスト設計に置き換わるものではなく、両者を適切に用いることでテスト設計全体の完成度が向上します

参考資料

- 「テスト設計入門 2017年度版」
 - Fujiメソッド及び逆さFujiメソッドその他の解説書
- テスト設計支援ツール：「シン・バグッぱ！」
 - ガイドワードと強制連関法を用いたバグの想定および分析ツール