

60分で学ぶ実践E2Eテスト

2022/3/10 JaSST'22 Tokyo

末村拓也(Autify)・伊藤由貴(ベリサーブ)

セッション概要

内容

Webアプリケーションを題材に、テスト設計からテストコード実装までの自動テスト作成の流れを一気通貫で、実践的に解説していきます。

想定参加者

- ソフトウェアテストの経験がある方
- プログラミングの経験があり、簡単な環境構築くらいは出来るよ！という方

セッションの構成

大きく2部構成です

1. 設計編：どんなテストを自動化するのか、などを考える
2. 実践編：設計編で考えたテストを、ツールを使って実際に自動化する

自己紹介：伊藤由貴

- テスト自動化エヴァンジェリスト
@ベリサーブ
 - テスト自動化の普及活動
- コミュニティ活動
 - JaSST Tohoku実行委員
 - JSTQBテスト自動化エンジニアシラバス翻訳WG



設計編

はじめに：皆さんに質問

「E2Eテストを自動化しよう」と思った場合、何から始めますか？

- 思いつくままに自動化する
- 今ある手動テストを順番に自動化する
- なんとなく大事そうなテストケースから自動化する

アンチパターンです

設計編の目次

1. やみくもにE2Eテストを自動化するとなぜ危険なのか
2. テスト自動化にあたって考える・工夫するべき点
3. 具体例

やみくもにE2Eテストを自動化するとなぜ危険なのか

テスト自動化を行うと、テストケースの追加や実行が（一見）簡単になるため、テストを増やしてしまうことがある。

行っても効果がないテストが増えてしまうと、

- メンテナンスの手間が大きくなる
- 実行時間（期間）が延びる

など、不都合が生じる。（そして自動テストはお蔵入りへ・・・）

"手動でおこなって効果のないテストを自動化しても無駄である"

そもそも、テストプロセス（e.g.テスト分析、テスト設計、テスト実装、報告）、特に**テスト分析、テスト設計が適切に行われていないテストは**、優秀なテスターが手動でテストを実施したところで、テストに期待される動作の保証やバグの検出といった**効果を発揮しない**。いわんや、自動テストにおいておや、である。

テスト自動化研究会 - テスト自動化の8原則 より。強調は伊藤。

自動化の前に考えるべきこと

「やみくもに」はダメ、つまり自動化の前に考えるべきことを考えましょう、ということ。

8原則にもあるように、「テスト分析」と「テスト設計」を適切に行う必要がある。

- テスト分析：「何をテストするか」を決める
- テスト設計：「どのようにテストするか」を決める

参考：[ソフトウェアテスト基礎講座 - 株式会社ベリサーブ](#)

参考：「E2Eテスト」とは

- システムテスト？ユーザー受け入れテスト？UIテスト？
 - 組織によって定義が異なる
 - そして、業界統一の「正しい定義」もない
- 本セッションでは、システムをユーザーと同じように操作して行うテストで、かつユーザーストーリー単位で行うものをE2Eテストと呼ぶことにする

※組織でのE2Eテストの定義が本ページと異なっている場合でも、実践編でテストを自動化する際の技術要素は共通しているため、何らかお役には立つと思います

設計編の目次

1. やみくもにE2Eテストを自動化するとなぜ危険なのか
2. **テスト自動化にあたって考える・工夫するべき点**
3. 具体例

テスト自動化にあたって考える・工夫するべき点

- 何をテストするか
- どのようにテストするか

大前提：「E2Eテストの自動化」に閉じないこと

E2Eテストをどう自動化するかという視点ではなく、開発・テストをどうしたいかを考え、そのためにE2E自動テストがどうあるべきか、を検討しましょう。

- 単体テストや結合テストなど、自組織の各テストが担う役割
 - E2Eテストでは何を担保したいか、単体テスト・結合テスト・E2Eテストの棲み分け
- 理想とする開発サイクル
 - リリース頻度、CI/CDパイプライン、テストにかけられる時間、など

何をテストするか、を考えるポイント

テストベースをもとに、どの機能（フィーチャ・シナリオ）をどこまでテストするか、を検討します。

- E2Eテストで対象とする機能（フィーチャ・シナリオ）
 - テスト対象のコアとなる機能
 - 正常系の基本的なユーザシナリオ
- どこまでテストするか、の考え方
 - もし不具合が発生した場合にビジネス上の影響が大きい範囲
 - 例）人数ベース：IE11を使っているのは3%だから対象外
 - 例）金額ベース：ガラケーを使っているのは全体の5%だが、売上の30%を占めているから対象

参考：一般的にE2Eテストとして自動化されるもの

- テスト対象のコアとなる機能
 - 起動/終了、ログイン/ログアウト、課金・購入、など
- ビジネス上の影響が大きい機能やユーザシナリオ
 - 会員登録、メルマガなど

どのようにテストするか、を考える際のポイント

- 各テストケース（シナリオ）で、どのように「テストしたいこと」を網羅するか
 - テスト技法の活用
- 自動で行う場合の効率
 - テストケース間の依存関係をなるべく減らす
 - テストの実行単位を変える

※この検討の過程で「このテストは手動でやろう」などの判断も発生

設計編の目次

1. やみくもにE2Eテストを自動化するとなぜ危険なのか
2. テスト自動化にあたって考える・工夫するべき点
3. **具体例**

具体例

ここからは具体的なサイトに対してE2Eテストを考えてみましょう

ホテル予約サイトのE2Eテストを考えよう

- テスト対象：[HOTEL PLANISPHERE - テスト自動化練習サイト](#)
- 条件設定
 - サイトは公開済み
 - 機能追加やBugfix後のリリース前にリグレッションテストとしてE2Eテストを行っているが、現在は手動
 - E2Eテストを自動化して効率化したい

ホテル予約サイトの機能を把握する

- ログイン
- ログアウト
- 新規会員登録
- 退会
- アイコン変更
- 宿泊プラン予約

テストしたいこと、を検討する

- 基本的なユーザストーリーは最低限網羅したい
- 予約における金額計算が正しく行えていることを担保したい
- 各会員種別でプラン表示が適切にできているか、を確認したい

各テストレベルで行っていることを確認

- 単体テスト
 - 金額計算ロジックのみ整備
 - hotel-example-site/billing_test.html
- 結合テスト
 - なし

今回対象にする範囲

- 重要機能
 - ログイン・ログアウト、予約、会員登録
 - 特に予約機能はサイトの主たる機能のため、会員種別ごとに予約可否を確認
- E2Eでしかできないテスト
 - 会員種別ごとのプラン表示確認
 - 例) 一般会員向けの画面にプレミアム会員専用のプランが表示されていないこと

ほんとにそれだけでいいの？

- テスト自動化はプロジェクトではない
 - 「重要機能のテストを自動化できたら終わり」ではもちろんない
- 最初から「何をテストするか」のすべてを自動化しようと思うと失敗することもある
 - Spotify社のように、段階的に成熟していくという方法も

選定したユーザーシナリオ6つ

1. 非会員で予約
2. 会員登録→予約→ログアウト
3. プレミアム会員でログイン→予約→ログアウト
4. 一般会員でログイン→予約→ログアウト
5. 一般会員の画面にプレミアム会員限定プランが表示されないこと
6. 非会員の画面に一般・プレミアム会員限定プランが表示されないこと

このうち、1番目の「非会員で予約」のシナリオを、具体的な手順として書き起こします

非会員で予約するシナリオの手順(1/2)

1. <https://hotel.testplanisphere.dev/ja/> を開く
2. メニューから「宿泊予約」を選択
3. 宿泊プラン一覧から「お得な特典付きプラン」の「このプランで予約」を選択
4. 宿泊日を翌月1日に設定
5. 宿泊数を7泊に設定
6. 人数を2に設定
7. 朝食バイキング、昼からチェックインプラン、お得な観光プランを選択
8. 氏名に「テスト太郎」を入力

非会員で予約するシナリオの手順(2/2)

9. 確認のご連絡をメールに設定
10. メールアドレスにhoge@example.comを設定
11. ご要望・ご連絡事項に「テスト」と入力
12. 予約内容を確認するボタンを選択
13. 宿泊予約確認画面で、以下を確認
 - i. 合計金額が123,000円であること
 - ii. 期間、人数、追加プラン、お名前、確認のご連絡、ご要望・ご連絡が入力通りになっていること
14. この内容で予約するボタンを選択し、以下を確認
 - i. 予約が完了しましたダイアログが表示されること

手順化にあたって考えたポイント(1/2)

- 実行のたびに結果が変わらないようにする
 - 休日と平日とで料金が異なるため、単純に翌月1日から宿泊にただけでは金額が可変になってしまう。7泊に設定することで、いつ実行しても同じ金額になる
- 「任意の～」は避ける
 - 自動テストを実装する際に困る

手順化にあたって考えたポイント(2/2)

- どのプランを選択するか
 - 今回は非会員で選択できるプランで、かつ画面上の場所等から最も人気のプラン＝予約できないとビジネス上の影響が大きいプランであると仮定し選択

ここからは、選定 & 手順化したテストケースについて、実際にコードを書いて自動化していきましょう