

第24回Ques #ques24

はじめてのテスト自動化

2025/05/23

日本電気株式会社 坂下 聡

アジェンダ

1. はじめに
2. テスト自動化のメリット・デメリット
3. どんなテストを自動化すべきか？
4. テスト自動化を始めるためのステップ
5. テスト自動化の失敗例と原因
6. まとめ

1. はじめに

- このセミナーでは、テスト自動化の基本を理解するひとつの方法を紹介します
- 受講対象者は、4月からQA業務（=Quality Assurance業務）を担当しテスト自動化に取り組んでいこうという方や、すでにテスト自動化を経験されているが、基本を再認識されたい方を想定しています
- 本日まで説明する内容は、一般的に公開されている情報をもとに作成していますが、この内容以外にも実施される状況によっては異なるテスト自動化のとらえ方もありますので、ご了承ください

自己紹介

坂下 聡（さかした さとし）

日本電気株式会社

SI変革推進 & エンジニアリング統括部

スペシャリスト

テスト効率化推進担当

■ 最近興味があること

- ジブリ風の画像作成
- 写真
- オープンカー



ChatGPT-4o Image Creationにて作成

左上から10歳→20歳→30歳

左下から40歳→50歳→現在

「テスト自動化」とは

不具合の混入を検知するテスト自動化の目的を実現することで、
テスト実行にかかる時間やコストを削減し、常に実施し続けることでソフトウェアの品質を維持する

■ テスト自動化の重要性

手動テストの課題	テスト自動化による解決策
時間がかかる	テスト時間を大幅に短縮
人為的なミスが起こりやすい	ミスを減らし、正確性を向上
すべてのパターンを網羅できない	多くのテストケースを自動実行
コストがかさむ	長期的なコスト削減

■ テスト自動化がもたらす価値

価値	内容
品質の維持・向上	人為的なミスを減らし、一貫性を高めることが可能
開発のスピードアップ	手動テストに比べて、テストを高速で正確に実行可能
コストの削減	同じテストを何度でも繰り返し実行できるため、コストを削減可能
チームの負担軽減	夜間や休日にテストを実行可能

2. テスト自動化のメリット・デメリット

テスト自動化のメリット/デメリット

テスト自動化には「時間とコストの削減」「品質と信頼性の向上」「開発効率の向上」のメリットがあり、自動化に向かないテストやテストスキル以外が必要なテストにおいて自動化出来ないデメリットがある

メリット

時間とコストの削減

- ・初期投資を**長期的に回収**
- ・**コストかけず**何度も実行
- ・**夜間・週末での実行が可能**となり、他のタスクに集中

品質と信頼性の向上

- ・**一貫性と人為的エラー**のリスク排除
- ・テストカバレッジの**拡大**
- ・常時、品質を確認しながら**早期にバグ**をとらえる

開発効率の向上

- ・効率とスピードを大幅に改善
- ・**少ない時間**でより多くのテストを実施
- ・**同時使用**のシミュレートを容易に実現

デメリット

初期構築・投資コスト

- ・**専門的なツール**が必要で**導入コスト**がかかる
- ・適切なQAエンジニアによる**作成・維持**が必要

学習・メンテナンスコスト

- ・ツールの**学習コスト**と修正やメンテナンスに**時間を費やす**
- ・計画時に自動化に要する**費用や工数、メンテナンス工数**を考慮する必要あり

自動化に適さないテスト

- ・実行回数が**少ない**
- ・ユーザビリティなど**人の観点で判断**が必要
- ・**手動**でのテスト実施が**容易**なもの
- ・物理的な操作がある

テスト以外のスキルも必要

- ・プログラミングスキルを含む**膨大な専門知識**が求められる
- ・**設計内容やコードの確認**が必要な場合がある

3. どんなテストを自動化すべきか？

自動化に向いているテストの特性

繰り返し実行や人的チェックによるミスが発生しやすいテスト、操作項目や手順が事前に決まっているテスト、実行回数が多くてまとめて実行することでテスト範囲を広くカバーするシーンに向いている

繰り返し実行されるリグレッションテスト

- 既存機能に対して同じ工程で同じ観点の内容を同じ手順で実行し、同じ結果を求めるためのテスト
- 人が疲弊してミスを呼び起こしやすいテスト

人的チェックではミスをしやすいテスト

- 大量のデータを扱うテスト
- 長時間実施するテスト
- 深夜・早朝に実施する必要があるテスト

操作項目や手順が事前に決まっているテスト

- 特定の画面の入力項目に対するテスト
- 決まった順序でボタンをクリックするテスト

人手よりも高い効果が出そうなテスト

- テストケースを目的や対象ごとにまとめたテストスイートを自動実行する方が、時間とコストを節約
- 多くのテストケースをカバーでき、ソフトウェアの品質を維持

自動化に向いていないテストの特性

一度きりのテストや物理的な作業を伴う手順があるテスト、人の判断が必要だったり複雑でパターン化出来ないテストには向いていない

実施回数が少ないテスト

- 一度きりや数回しか実施しないテスト

物理的な作業をともなうテスト

- 物理的なデバイスの操作や、実際のデバイス接続環境での検証を伴うテスト

テスト担当者の創造性や判断力が求められるテスト

- 探索的テストやアドホックテスト、ユーザビリティテスト
- 主観的な評価が必要となる自動化が困難なテスト

パターン化が難しいテスト

- テストのパターンが多岐にわたるテスト
- 動作が変化するテストや、複雑な検証をおこなうパターン化が難しいテスト

人的チェックが望ましいテスト

- テストの重要度高く、わずかな違いが大きな影響を与える可能性があるテスト
- テスト対象のシステムが頻繁に変更されるテスト

テスト自動化による効果

費用対効果は自動化する大切なポイントであり、必ずコストを意識して実施有無を検討する

テスト自動化の効果

効率の向上

- 繰り返し実行可能で再利用
- 作業員の工数削減
- 欠陥の早期の発見と修正
- 再作業の削減により長期間のコスト削減

コストの削減

- テスト実行時間とスケジュールの短縮によるテスト効率の向上

ソフトウェア品質の向上

- 実行するカバレッジの増加により、潜在欠陥のリスクとコストを低減

(参考) 費用対効果 (ROI)

= (手動テストを続けた場合の運用コスト - 自動テストの初期開発コスト - 自動テストの運用コスト)
/ (自動テストの初期開発コスト + 自動テストの運用コスト)

4. テスト自動化を始めるためのステップ

テスト自動化を始めるために考慮すること

最初に目的やメリットを明確にし、許容可能な内容かどうか判断したのち、導入の準備や手動テストからの移行とテスト自動化の継続運用での考慮点を確認し、実施有無を判断する

目的・メリットの確認

主たる目的・メリット

活用方法の確認

開発モデル別のメリット

導入の準備

適切なプロジェクトを
選定

自動化するスコープの
判断

利用するツールの
選定と評価

自動化への移行・運用検討

自動テストに移行する際の
懸念事項

継続運用するために
考慮すべきこと

目的・メリットの確認

どんな目的で導入するのか、導入した際のメリットはどこになるのかを確認することが大事で、テスト自動化の効果が得られやすい場面とテスト自動化の効果が得られにくい場面を確認しましょう

主たる目的・メリット

- 再利用によるコスト削減
- 迅速なフィードバックによる開発者への支援
- 品質意識の文化を促進

活用方法

- リグレッションテスト
- テストデータ準備とクリーニング作業
- 非機能テスト
- 不具合検出後の確認テスト
- 受け入れテスト

開発モデル別のメリット

- ウォーターフォール開発
 - 再利用での時間短縮効果
- アジャイル開発
 - 全てのテストが自動化され、製品の品質が担保される開発の効率化

テスト自動化の効果が得やすい場面

- 長時間の手動テスト
- テストケースの同期実行
- テストのタイミング精度が必要な場合
- CI/CDパイプラインの利用を前提とした利用
- 大量のログファイル解析、テスト実行、データ入力
- マルチのOS、ブラウザ、デバイスなどのテスト
- 非機能テスト

テスト自動化の効果が得にくい場面

- 人間の主観的なレビューが必要なデザインの検証
- 人の介在が必要
- テスト自動化で操作を行えない技術的な制限
- 時間依存性の高いテスト
- 探索的テスト
- ランダムな操作や値を検証するテスト
- ロボットの操作の回避が必要な機能

導入の準備（1/2）

小さく始めて成功体験を積み重ね、段階的に適用範囲を広げていくために、自動化を導入する条件を理解しましょう

テスト自動化を初めて導入する場合に 適切なプロジェクトを選定

既存システムの追加機能開発のような開発保守案件

コードの変更やテスト環境が変化しないプロダクト

単純な機能で他の環境や機能の依存関係が少ない

パイロットで得られた知見を適用

パイロットでの問題をビジネスへの影響が最小限に
抑えられるプロジェクトの選択

自動化するスコープの判断

テストケースを自動
化する方法の有無

繰り返し可能性

技術的な課題

メンテナンス性

投資対効果（ROI）

ビジネスフローの
カバー範囲

実行頻度の価値

テストケースと
データの再利用性

テストの種類

導入の準備（2/2）

小さく始めて成功体験を積み重ね、段階的に適用範囲を広げていくために、自動化を導入する条件を理解しましょう

■ テストツールを選定する際に考慮すべき点

考慮点	内容
予算	ツール費用の有無にかかわらず十分な予算が見積られているか
プロジェクト要件	テスト対象のアプリの種類、動作OSやブラウザーなどのバージョン
スキルセット	プログラミングの専門知識の有無
メンテナンスと再利用性	テスト対象変更時のヒーリング機能やテストステップの再利用可否
レポート作成	テスト結果の分析や報告への利用可否
他ツールとの連携	CI/CDプロセスの導入可否
テクニカルサポート	フォーラム、コミュニティ、有償サポートの有無

自動化への移行・運用

自動化への移行・運用を成功させるためには、テストだけではなく継続運用を考慮に入れることが重要です

手動テストを自動テストに移行する際の懸念事項

移行前のテストが正しいテストであることを保証

移行時に一時的にコストが増加

同じ機能・ステップのスク립トを共通化

共有の依存を減らして個々のテストが独立実行

パイプラインなどで依存性を下げる仕組み

前提条件を構築する自動化スク립トと実行管理

テスト自動化を継続運用するために考慮すべきこと

変更時は、テスト実施後にスク립トをデプロイ

スク립トの改修検討や工数見積、即時対応の判断

プロダクトのバージョンに対応したスク립ト管理

OSやブラウザのソフトウェアのバージョンアップ

CI/CDの組み込み

結果分析や状態を評価するダッシュボード検討

定期的な実行時間のモニタリング

自動化への移行・運用（結果を分析し、改善していく）

テスト自動化を始めるためのステップを実施後、次の実施に向けた改善を行うスキームを取り入れましょう

テスト結果の レポート作成

- テスト結果の情報(不具合件数、テスト項目の消化状況など)をもとに**テスト実施結果を振り返る**
- テスト結果の共有や**管理の効率化とテストプロセス改善**

ボトルネックの 特定と改善

- インシデントの分析による**不具合の特徴や今後の改善に向けた推奨事項**を整理
- テスト実施時の**問題点や良かった点**などの振り返り

テスト自動化戦略の 見直し (PDCAサイクル)

- テストプロセスやテストツールの**改善提案**、今後のテスト戦略への提言
- テストサマリでの**知見**を次プロジェクトに活用
- QA業務の**効率化や品質の向上**を図る

5. テスト自動化の失敗例と原因

自動テストが失敗する5つの要因

5つの要因に当てはまる場合、自動テストへの過剰な期待を持ちすぎるあまり、初めて実施した時に思った結果が出ないことで、自動テストは難しくて非効率という印象で、導入に失敗したと感ずることになります

自動化する切り分け基準があいまい

- どこが自動化出来るかという観点のみで切り分け
- 何度もテストする回数が少なくなり、必要性もなくなる

ツールの選定を間違えた

- ツール再選定と実装しなおしが発生

仕様変更などで発生するメンテナンス工数を考えていない

- スクリプト変更は必ずあるのに、変更の可能性を考えていない

自動化することが目的となっていた

- 自動実行に感動してしまい満足
- 必要でないテストは工数の無駄

担当者にテスト技術が無かった

- 開発主導で自動化を導入すると起こりがち
- テスト計画や設計に前提条件・手順・期待結果を定義したうえでスクリプトを作成していない

(参考) テスト自動化の失敗例

事象	要因	ポイント
自動化する目的を誤ってしまった	自動化が目的	- 自動化しやすいところを自動化することが目的にした 「優先度の高い試験項目を最大5回以上は自動化で実施」とする計画を立てる
テストケースの手順、条件、期待結果が明確に記載されていない	テスト技術が無い 自動化が目的	操作が完了すればいいというわけではなく、スクリプトで実施手順・事前条件や期待結果を具体的な設定値で明確にすることが必要
自動テストの役割を誤ってしまった	自動化することが目的 切り分け基準があいまい	- 自動テストで品質を上げること（不具合を出すこと）を目的にした 「自動テストは、期待結果が決まっていて何度も実施するテストをおこなう」「手動テストは、創造力が必要な不具合を出す試験をおこなう」のような棲み分けをおこなう
工数削減を自動化担当者以外考えない	自動化することが目的 テスト技術が無い	自動テストは工数削減の1手段であり、工数削減はチーム全体で取り組む必要あり
テストの技術が無い 開発担当者が自動化する	切り分け基準があいまい テスト技術が無い	意図するテスト観点が含まれていないのはだめで、テスト設計されテスト観点が明確なテストケースをテスト担当者が用意し、開発者にスクリプトを作成依頼することが必要
何度も自動テストを実施する テストが無い	切り分け基準があいまい メンテナンス工数考慮漏れ	自動化を意識しすぎるため、1回実施するだけのテストも自動化してしまい、自動化するためのコストが発生して効率化出来ない
時間が無いから自動テストが 評価で終わってしまう	ツール選定誤り テスト技術が無い 切り分け基準があいまい メンテナンス工数考慮漏れ	- 試験的に自動テストを実施するものの、導入には時間がかかるため、現状の作業が忙しいメンバーは徐々に自動テストの作業を行わなくなり、結果的に失敗する - 自動テストの専門チームを作り、自動テストを担当し進めながら徐々に範囲を広げていくことが必要
自動テストの運用経験が無い	テスト技術が無い	自動テストを成功させるための近道は、自動テストの運用を経験し、自動テストの計画・設計スキルを高めること

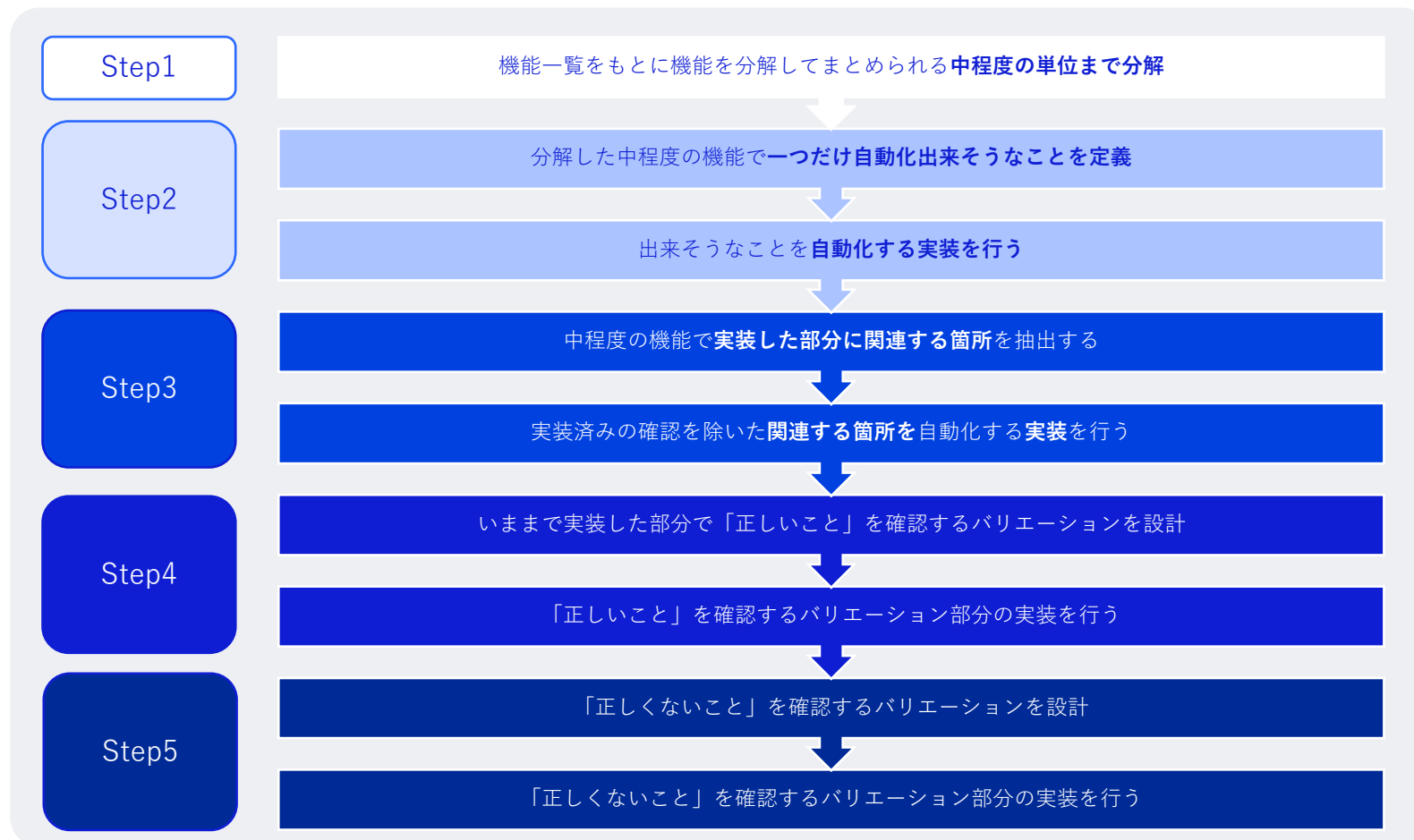
(参考) テスト自動化の成功例

機能が多いプロダクトにおいて、機能を分割して段階を追って網羅する範囲を増加していったスモールスタートの事例

- まず、1 機能 1 ケースで自動化できる機能を網羅していく
- 次の段階で 1 機能数ケースでスモークテストとして使えるように増やす
- 段階を追っていくことで、手動のテストケースを流用するよりも効率的にテストを行うことができる

ポイント！

手動テストのテストケースをそのまま自動化すると難易度が高いので、自動テスト用のテストケースを作成しながら進めるのがコツ



6. まとめ

工数削減の先にある真の自動テストによる効果

自動化は1つの手段であり、その手段を用いた削減効果で発生した時間を一番大切な作業に割り当てられるようになるのが真の効果

自動化で削減できた工数で品質を上げる

- 現行のメンバーで最大の品質を上げるために、今までやっていたテストを自動化し、**カバーできていない範囲のテストを実施**することが可能になる

チームメンバー全体でテスト効率化の意識が高められる

- 工数削減は**自動化だけで実現するものではない**
- **試験環境の改善、不要な試験項目の削除、効率の良い不具合検出**など、本来注力すべきことはたくさんあるので、**注力すべきことに時間を費やすことが重要**

本セミナーのまとめ

01

はじめてテスト自動化に取り組むための概要

- テスト自動化の重要性とその導入に向けたステップ
- 時間やコストを削減し、品質と信頼性を向上させる効果
- テスト自動化に適したテストの特性
- 逆に自動化が難しいテストの特徴
- 具体的なプロジェクト選定やスコープ判断のポイント
- 失敗要因や失敗例

02

テスト自動化は継続的な取り組みであり、長期的な視点で活動を

- テスト自動化は計画的な取り組みと継続的な改善が必要
- 自動化自体が目的化しないように注意
- チーム全体で協力し、テスト自動化を進めることが重要
- 最初から大規模な自動化を目指すのではなく、簡単なテストから始めて、徐々に自動化の範囲を広げていく
- 必ず振り返りをおこない、改善し続けることが継続の秘訣

ご清聴ありがとうございました

NEC

\Orchestrating a brighter world