

繰り返し実施を要する試験における省力化に関する提案

○ 小宮 崇[†], 楓 仁志[‡]

三菱電機株式会社 情報技術総合研究所

1. はじめに

社会の情報化により、多くの企業では IT 化による業務の効率化が進められており、それを実現するために、日々多くのソフトウェア開発が行われている。本論文では、そのソフトウェア開発そのものではなく、完了後に繰り返し必要となる回帰試験の実施方法について提案する。

2. 背景

ソフトウェア開発において、試験は大きなウェイトを占めており、コストを削減しつつ、確実に効率よく行うための手法が多く用いられている[1]。しかし、開発が完了し、リリースが終わった後にも、サポートする OS やミドルウェア、ブラウザ等の更新がある度に、繰り返し回帰試験が必要となる。多くの場合、この回帰試験は軽視されがちであり、開発時のような潤沢な予算は確保されていないケースが多い。しかし、試験自体は重要なものであり、ここで障害を検出できないと、フィールド障害が発生し大きな損失を招く結果となる。

そこで、この問題を解決するために、回帰試験を低予算で確実に実施できる方法について、検討を行った。

3. 課題

回帰試験における作業コスト増大の要因について検討を行い、以下の課題を抽出した。

① 試験環境毎の試験ツールの構築

試験環境を構築する際、試験対象のソフトウェアについては CI ツールからの自動デプロイ等による構築が可能となっている場合が多い。しかし、試験に使用するツール類については考慮されていない場合が多く、別途構築作業が必要になるため、追加の作業コストが必要となる。また、環境に合わせて異なるライブラリ・設定が必要となる場合には、そのための学習と準備が必要となり、動作確認も含め、作業コストが更に膨らむ。

常設の試験環境がある場合には、試験に使用するツール類のインストールは 1 回で済み、その後はツールの更新がある場合の作業のみとなるが、最近は VM やコンテナを用いて新しく環境

をいくつも作るケースも多くなっており、試験のためのツール類の導入の作業コストは無視できない大きさとなっている。

② 試験内容の把握とメンテナンス

多くの試験では、自動化の試みが行われており、試験のためのツールやライブラリが用いられている。その際に、試験内容をプログラミング言語等で記述すると、可読性が悪く内容の把握が難しくなり、障害検出時にどこで問題があるのか分かりづらい場合もある。また、メンテナンス性も低くなるため、試験内容の変更や追加が必要となった場合に、シナリオの変更や追加が困難となる。

4. 解決方法の検討

課題の解決への取り組みについて説明する。

① 試験環境の試験ツールの構築

構築担当者の作業コストを削減する方法として、試験ツールの導入を自動化する方式を検討した。

この方式は、試験環境から環境情報(OS 等)を入手するためのモジュールを提供し、それを用いて入手した環境情報に応じて必要なモジュール・シェル等を判断して提供するサービスを配置するものである。環境情報を取得するモジュールもこのサービスから提供する。概要を図 1 に示す。

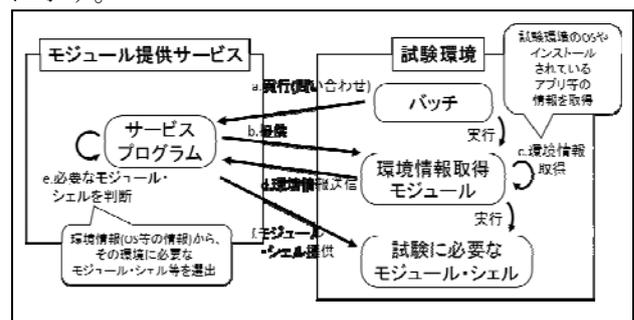


図1 試験に必要なモジュールの提供

試験環境上でバッチを実行(a)し、モジュール提供サービスから環境情報取得モジュールを取得(b)する。次にそのモジュールを実行して環境情報を取得(c)し、それをモジュール提供サービスへ送信(d)する。モジュール提供サービスは、受け取った環境情報から、その環境に必要なモジュールやシェル等を判断(e)して提供(f)する。最後に、必要であれば試験環境側で提供を受け

たファイルを実行する。

これにより、導入の手順を簡素化するとともに、知識の無い試験担当者でも確実に短時間で構築可能とする。

数回の利用では構築コストの方が大きいと思われるが、一度に複数の試験環境を使用する、試験ツールの更新頻度が高いといったケースには、効果を発揮すると思われる。

② 試験内容の把握とメンテナンス

試験内容を分かりやすく、また変更や追加を簡単に行えるようにするために、キーワード駆動方式の導入を検討した。概要を図2に示す。

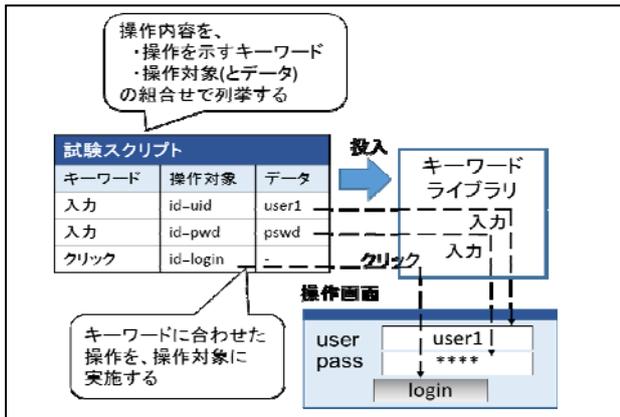


図2 キーワード駆動方式

この方式は、操作内容を、操作を表現する「キーワード」と「操作対象」で表現して外部ファイルに列挙し、それを読み込んで順番通りに操作を実現するものである。キーワードを実現するためのプログラミング等は別途必要になるが、試験スクリプトをキーワードと試験対象の列挙で記述できるため、可読性が高く、試験内容の把握が比較的容易となり、メンテナンス性の向上も期待できる。シナリオの変更や追加についても、比較的簡単に実施可能となる。

5. 自動化の実現

前述の方式の評価として、それを適用した試験自動化ツールを試作し、既存の回帰試験の一部の再現をできるか確認を行った。その内容について説明する。

(1) 再現する回帰試験

Webアプリケーションの回帰試験であり、IE、Chrome等のWebブラウザに対して特定の操作を行い、その結果を確認するものである。概要を図3に示す。

(2) 試作した試験ツール

Seleniumを用いてブラウザの操作を行う試験ツールを試作した。本ツールは、試験端末から、各試験環境のブラウザをSelenium Grid経由で、試験スクリプトに従って操作することで試験を

実現するものである。これへの取り組みの導入内容を説明する。

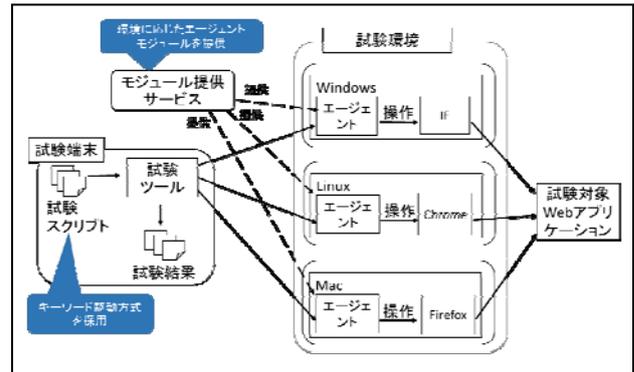


図3 回帰試験概要

① 試験環境の試験ツールの構築

実行された試験環境のOSとブラウザの情報をサーバへ送信するクライアントプログラムと、その情報を受けてその環境に必要なライブラリや設定ファイルを提供するモジュール提供サービスプログラムを作成した。

今回の試作では、これを用いてその試験環境にあったモジュール(OS・ブラウザによってブラウザ操作に必要なドライバファイルが異なる)を配置し、Selenium GridのNodeを構築した。

② 試験内容の把握とメンテナンス

クリック、値入力等のブラウザへの操作の単位でキーワードを指定し、それを実現(Seleniumでブラウザを操作)するライブラリを作成し、試験スクリプトから操作内容をキーワードで指定できるようにした。

6. 結果と考察

回帰試験を、一から構築して実施する場合よりも少ない作業コストで実現できることを確認した。

試験ツールの構築については、構築はできたが、サーバ側のモジュールや設定等の管理面について工夫が必要という課題が抽出できた。

キーワード駆動方式の導入についても、試験スクリプトの可読性が担保できることが確認できたが、その作成に知識と時間が必要となったことから、試験スクリプトの生成方法についても検討が必要という課題が抽出できた。

7. おわりに

今回は、回帰試験を対象に、省力化を実現する方法について検討・試作を行った。今後は、今回抽出できた新たな課題について、解決方法を検討・試作する予定である。

参考文献

[1] 情報処理推進機構, 「つながる世界の品質確保に向けた手引き」, 情報処理推進機構, 2018.