

ソフトウェア実行情報を活用した回帰テストの自動化

倉 林 利 行^{†1}

テスト品質を保ちながら回帰テストの稼働の大幅削減を実現する方法について議論する。

Regression Testing Automation based on Software Information Analysis

TOSHIYUKI KURABAYASHI^{†1}

The authors would like to discuss on how to reduce the manual works of regression testing with keeping test quality.

1. はじめに

ソフトウェア開発の半数以上を占める保守改造案件¹⁾では、保守・改造のたびに行われる回帰テストの稼働削減が求められている。回帰テストは従来機能の劣化の有無を確認するためのテストである。本ワークショップにおいては、テスト品質を保ちながら回帰テストの稼働の大幅削減を実現する方法について議論したい。

2. 従来の回帰テスト実行方法

回帰テストは通常、図1のタスクフローで行われる。

(a) テスト環境準備について

テストを行う上で必要となる環境（ハードウェア、OS、DB、NW など）を準備する。

(b) テストケース準備について

本タスクでは (c) テスト実行において必要なテストケースの準備を行う。テストケースはテストシナリオ、入力値、事前条件、期待結果などを含む。実行するテストの数を減らしたい場合は前回開発で仕様したテストケースに基づいてテストケース選定を、テスト実行を自動化したい場合はテストスクリプトの作成を行う。

(c) テスト実行について

本タスクでは、テストケースに書かれたテストを実行する。テストケース準備のタスクにてテストスクリプトが用意されている場合はスクリプトを実行し、用意されていない場合はテストケースに書かれたテスト



図1 回帰テストのタスクフロー

シナリオに沿ってテスターが人手で実行する。

(d) 合否判定について

本タスクではテスト実行によって得られた結果（画面結果やDBの値など）に基づいて、合格または不合格を判定する。回帰テストの場合はデグレードの有無の確認が目的となるので、以前のバージョンと今回のバージョンの結果を比較し差異がなければ合格とする。

3. 着目する回帰テストのタスク

本ワークショップでは回帰テストの稼働削減を目指しているため、回帰テスト全体の稼働削減に最も寄与する (b) テストケース準備に着目する。テストケース準備の効率化に対する需要は高いが、現状では支援ツールの使用率は1割程度である²⁾。テストケース準備に対する実用的な支援技術を樹立できれば、回帰テストの稼働削減が狙える。

回帰テストの稼働を削減するために、テストケース準備で行える工夫として「テストケーススクリプト化」という方法が存在する。本手法では準備したテストケースをスクリプト化することによってテスト実行のタスクを自動化し、稼働を削減することが狙いである。テストケースには入力値や事前準備する入力以外のデータの値とテスト実行の手順の情報が記載されている。これらをすべてスクリプト化することで、テスト実行の自動化が可能になり、回帰テストの度に使い

^{†1} NTT ソフトウェアイノベーションセンタ
NTT Software Innovation Center

まわすことができる。

上記の方法は図2の①で表される。

4. テストケーススクリプト化の課題と原因

4.1 課題

テストケーススクリプト化は必ずしも稼働削減に効果があるとは限らないという課題がある。これはテストケーススクリプト化自体に稼働がかかるためである。具体的には以下の2種類の稼働がかかる。

・テストスクリプト作成の稼働

テストケースは1つずつ手でスクリプト化する必要がある。テストケーススクリプト化の支援として、Selenium IDE等のCapture and Replayと呼ばれるツールが存在する³⁾。Selenium IDEは、画面操作を保存し、自動的にスクリプト化するツールであり、スクリプトを手で作成する手間を軽減できる。しかし本ツールを用いても、画面操作自体は人手で実行しなければならない上に、ツール操作やスクリプト修正の稼働もかかるため、稼働のオーバーヘッドが大きい。

・テストスクリプトのメンテナンスの稼働

テストケースをすべてスクリプト化した後も、ソフトウェアの改造内容によってはスクリプトがそのまま実行できないこともあるため、スクリプトのメンテナンスが必要となり余分な稼働がかかる。

上記2種類の稼働により、一般的に3回以上回帰テストを実行して初めて、スクリプト化による稼働削減効果が得られると言われている。

4.2 原因

テストケーススクリプト化に稼働がかかる原因は、前回開発のときに作成したテストケースを利用していることにある。前回開発のときに作成したテストケースを用いるという前提では、自然言語かつフリーフォーマットで書かれているテストケースを全自動でスクリプト化することは不可能なため、支援ツールを利用しても何らかの形で必ず人の稼働を要することになる。また前回開発で用いたテストスクリプトを流用しているためスクリプトのメンテナンスも必要となる。

5. 提案手法

前章の課題を解決するために本ワークショップでは、回帰テストにおいて前回のテストケースを用いるという固定概念を捨て、「ソフトウェア実行情報を活用した結合テストレベルのテストケース自動生成による回帰テストの稼働削減手法」を提案する。ソフトウェア実行情報はソースコードや実行ログのことを表す。本提案手法のフローは図2の②で表される。本提案手

法によってテストケース準備にかかる稼働は以下のように改善される。

・テストスクリプト作成の稼働

現場に実際に存在する資材（ソースコードや実行ログ）からテストケースをスクリプトの状態で作成するため、不要となる。

・テストスクリプトのメンテナンスの稼働

回帰テストの度に新しいスクリプトが生成されるため、メンテナンスの必要がなく、不要となる。

①従来の回帰テストのフロー



②提案する回帰テストのフロー

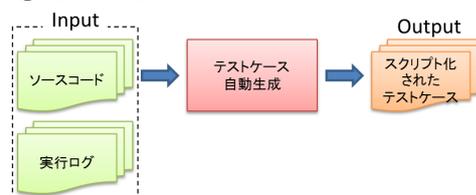


図2 従来と提案する回帰テストのフロー

6. おわりに

本ワークショップでは「ソフトウェア実行情報を活用した結合テストレベルのテストケース自動生成による回帰テストの稼働削減手法」を提案した。これにより回帰テストを少ない稼働で実行することが可能となる。本手法によって今まで回帰テストという非生産的な作業に費やしていた時間をソフトウェアの新規開発や新機能追加などの価値創造的な活動に費やすことができるため、事業者のサービス開発力が向上し、社会全体が恩恵を受けることができる。今後は、ソースコードなどの現場の資材からテストケースを自動生成する技術について検討して参りたい。

参考文献

- 1) 独立行政法人情報処理推進機構. ソフトウェア開発データ白書 2014-2015. 独立行政法人情報処理推進機構, 2014.
- 2) ソフトウェアメトリックス調査 2012 報告書. <http://www.juas.or.jp/servey/library/pdf/12swm.pdf>.
- 3) Selenium ide plugins. <http://www.selenium/hq.org/projects/ide/>.