

# レガシーマイグレーションにおける受け入れテストの 生成支援手法の検討

木津谷諒† 山田敬三† 高木正則† 佐々木淳†

岩手県立大学ソフトウェア情報学部†

## 1. はじめに

日本では多くの企業がレガシーシステムを抱えており<sup>[1]</sup>、これらのシステムを更改するためにレガシーマイグレーションが行われている。レガシーシステムは大規模であることが多く、それに伴いテストコードの作成に必要な作業量・コストは膨大なものとなる。そこで本研究では、レガシーマイグレーションにおける受け入れテスト作成の作業量・コストの削減を目的とし、受け入れテストの生成支援手法を提案する。本稿では、システムに必要な機能の利用シナリオを作成し、レガシーシステムでの操作・振る舞いの記録から、受け入れテストが生成できることを示し、これらの作業の支援手法を検討する。

## 2. 現状の手法

### 2.1. 受け入れテスト

受け入れテストは、JSTQB によって“システムが、ユーザーのニーズ、要件、ビジネス・プロセスを満足するかをチェックするための公式のテスト”と定義されている<sup>[2]</sup>。受け入れテストを用いることで、システムが問題なく運用できるかどうかを評価することが可能となる。

従来、受け入れテストの実行は人手によって行われていたが、近年では継続的インテグレーション(CI)の利用の増加に伴って受け入れテストの実行を自動化するケースも増加している。

しかしながら、受け入れテストの作成は依然として人手によって行われている。これには、システムの要件を理解した上でテストの作成を行わなければテスト・システムの品質を保証することができないことが背景にある。

### 2.2. レガシーマイグレーション

レガシーマイグレーションは、老朽化により保守・運用・機能追加が困難なレガシーシステムを更改し、新たなシステムに再構築することである。近年、レガシーシステムを理解するエ

ンジニアが減少し、マイグレーションが困難になりつつある。また、レガシーシステムには企業の基幹システムとして利用される大規模かつ複雑なシステムも多い。

### 2.3. レガシーマイグレーションにおける受け入れテスト

マイグレーションでは、システムの評価や新旧システムの比較のために新システムの受け入れテストが実施される。通常のシステム開発と同様に、受け入れテストの作成は人手により行われる。マイグレーションはエンジニアの減少および大規模なシステムという課題を抱えており、受け入れテストの作成においても作業量・コストの削減が望まれる。

## 3. 手法の検討

### 3.1. 検討手法概要

そこで、マイグレーションにおいてはレガシーシステムを基に新システムの要件を定義することに着目し、同様にレガシーシステムの振る舞いを新システムの受け入れテスト作成に利用することを考えた。以下の3つの手順を順に実行することで、システムの要件を落とし込んだ受け入れテストが生成できると考えられる。

(1) 新システムにおいて必要な機能を網羅した利用シナリオを用意する

(2) レガシーシステム上でシナリオ通りに操作し、振る舞いを抽出する

(3) 抽出した振る舞いからテストを生成する

(1)に関しては、十分な品質を確保するため、人手で行うことが望ましいと考える。(2)(3)に関しては、(1)で品質を確保しているため、レガシーシステム上での操作を除いた全箇所まで自動化が可能であると考えられる。

上記の点を踏まえ、振る舞いの抽出およびテストの生成を行う受け入れテストの生成支援システムを提案する。

### 3.2. システム概要

図1に本研究で提案するシステムの概要図を示す。シナリオ実行者がレガシーシステム上でシナリオを実行し、そのときのキー入力および

A Study on the Supporting Method for Generating Acceptance Test in Migration of Legacy Systems

†Makoto KITSUYA, Keizo YAMADA, Masanori TAKAGI and Jun SASAKI, Faculty of Software and Information Science, Iwate Prefectural University

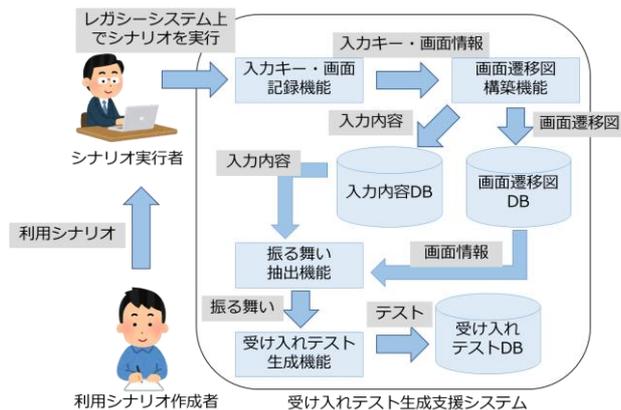


図 1 受け入れテスト生成支援システムの概要図

画面情報をシステムに入力し、これを利用して自動的に受け入れテストの生成を行う。システムの開発には Python を用いる。レガシーシステムには CLI アプリケーションを想定している。また、新システムには Web アプリケーションを、受け入れテストのプラットフォームには Selenium を想定している。

### 3.3. 入力キー・画面記録機能

シナリオ実行者がレガシーシステム上で入力したキーをすべて記録する。キー入力の発生と同時に、画面の記録も行う。

### 3.4. 画面遷移図構築機能

記録された画面情報から、前後の画面との類似度を測定することで画面遷移を検知する。ここで得られた情報を基に画面遷移図を構築し、画面遷移図 DB に格納する。また、構築された画面遷移図において、それぞれの画面で入力されたキーを統合して入力内容とし、入力内容 DB に格納する。

### 3.5. 振る舞い抽出機能

それぞれの画面を OCR で文字列に変換し、入力内容の文字列を検索する。画面内で、一致した文字列の付近を探索し、その情報からレガシーシステムの振る舞いを抽出する。

### 3.6. 受け入れテスト生成機能

抽出した振る舞いを用いて、特定のテストプラットフォーム用のテストコードを生成する。生成した受け入れテストは、受け入れテスト DB に格納する。

## 4. おわりに

本研究では、レガシーマイグレーションにおける受け入れテスト作成の作業量・コストの削減を目的とし、受け入れテストの生成支援手法の検討および支援システムの提案を行った。現段階では、入力キー・画面記録機能のみ完成している。今後は未完成箇所的设计および開発を行

い、レガシーシステム上でのシナリオの実行によって受け入れテストが生成できることを示す。また、生成されたテストの妥当性の検証を行う。

### 参考文献

- [1] 一般社団法人 日本情報システム・ユーザー協会, 野村総合研究所 システムコンサルティング事業本部, デジタル化の取り組みに関する調査, pp. 50, 2018
- [2] Erik van Veenendaal 著, JSTQB 技術委員会訳, ソフトウェアテスト標準用語集 (日本語版), pp. 14, 2014