

基幹システムにおける 自動テストの課題と対策

住友電工情報システム(株)

QCD改善推進部 生産技術グループ

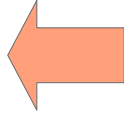
野尻 優輝

2024/10/16

Agenda

1. 背景
2. 課題と対策
3. 自動テスト生成システム
4. まとめ

住友電工情報システム株式会社

- 資本金 : 4.8億円
- 従業員数 : 600名
- 本社 : 新大阪
- 株主 : 住友電気工業株式会社 60%
住友電装株式会社 40%
- 事業内容
 - 住友電工グループ向け 業務システムの開発 
 - 住友電装向け 業務システムの開発
 - パッケージソフト 開発・販売

1.背景

1.1 背景

- 既存システムの保守人材不足
- 全体的な人口減少
- DXによるシステム数の増加



- 今後のシステム保守要員不足は確実
- 開発、保守作業の効率化が必要



自動テスト 導入

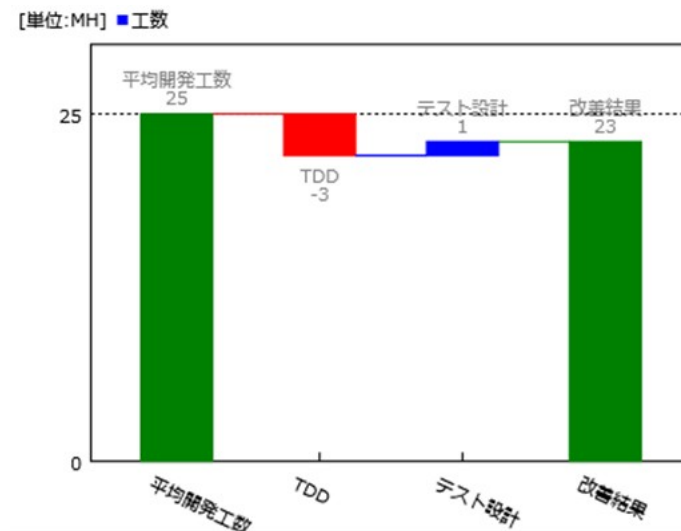
GitLab 導入

仕様書 改善

ノウハウ共有

1.2 目標

- 開発工数2MH削減
 - TDDによるコスト減: ▲3MH
 - テスト設計によるコスト増: 1MH
- 自動テスト生成率60%
 - 弊社標準テスト項目の2/3を生成



1.3 過去の取組

- JUnitを用いた自動テストを試行
 - テスト駆動開発(TDD)導入
 - テストスクリプトのテンプレート出力
 - テストデータテンプレート出力

課題

- 画面操作がテストできない
- プログラム改善の度にテストスクリプト修正が必要
- テストデータを手動で準備する必要がある
- 繰り返しテストが容易に実施できない

登録ボタン押下
遷移先の画面表示値を確認

更新テストの度に
データのリセット

1.4 取組の方針

- 以下を自動生成する
 - 画面操作テスト
 - ビジネスロジックテスト
 - テストに必要なデータ
- 繰り返しテストが容易に実施できる
(テストデータのリセット)

2.課題と対策

2.1 テスト自動生成の課題と対策

- 自動テストの環境

自動テストツール	Selenium
テストスクリプト言語	Javascript

- テストスクリプト例

- 項目の表示値を確認する場合



テスト対象機能 画面イメージ

```
const expect = "aakp0010"
let driver = await new Builder().forBrowser('chrome').build();
const loc = await driver.findElement(By.id('disp.atHeader_1.PGID.1'));
let value = await loc.getText();
if (value == expect){
  return "OK"
}
return "NG"
```

テストスクリプト

HTML IDを指定して表示値を取得
期待値との比較を行う

2.1 テスト自動生成の課題と対策

- 課題 1 : 画面の入出力項目等が実装前にわからない
 - 画面設計がExcelやイメージファイルで作成されている
 - TDDの為、プログラム作成前にテストを作成する必要がある
- 対策 : 画面設計作成機能を開発
 - 画面設計をデータ化 (JSON)
- 課題 2 : 項目やボタンのHTML IDがない
 - 弊社標準のフレームワーク(*1)では、表示項目にHTML IDが振られていなかった
- 対策 : 弊社標準のフレームワークを改善

*1 (楽々Framework II 以降、FW II)

2.2 ビジネスロジックの分析

- 課題：テストケースが分からない

ビジネスロジックとは？

- 機能固有の処理
 - 管理者権限が無ければエラー
 - 出荷数<受注数の場合、区分:”分納”



ソースコードのどこに表される？
社内のプログラムと仕様書を分析

分岐条件

- エラーチェック
- SQLのWHERE句
- 更新項目表(*2)

日本語で記載
システムが利用できない

*2 (データベース登録更新における
各項目に対する値加工条件の表)

2.2 ビジネスロジックの分析

- 課題：テストケースが分からない
 - 分岐条件が日本語表記かつ、データ化されていない
- 対策：エラーチェック,SQL,更新項目表の設計機能作成
 - 各設計機能上で分岐条件を登録

The screenshot displays a software interface for error checking design. The main window is titled "[D.A.] エラーチェック設計 - [画面] 計画登録 [登録]". Below the title bar, there is a menu bar with options like "文書Top", "画面設計", "エラーチェック設計", etc. The main content area shows "Screen 4" with a list of 6 items. A table lists the input checks, and a pop-up window shows the details for a specific check rule (ChkE#001).

No.	I/O	画面項目名称	画面項目名	入力チェック
1	IN	計画納期	plan_due	[ChkE#001]Try登録日以前の日付は入力できません。× 追加
2	IN	推定工数	est_mh	[ChkE#003]下限値0以上を入力して下さい。× 追加
3	IN	計画内容	plan_desc	追加
4	IN	備考	plan_rem	追加
5	IN	実施担当者ID	userid	[ChkE#002]チー 追加
6	IN	実施担当者氏名	user_nm	追加

チェック内容[-]

All ChkE#001 ChkE#002 ChkE#003

ChkE#001

ID	ChkE#001
チェック対象項目	plan_due
エラーメッセージ	Try登録日以前の日付は入力できません。
表示位置	<input type="radio"/> 画面上部 <input type="radio"/> Form上部 <input checked="" type="radio"/> 項目
実施条件	編集
表示項目	item
項目	項目
チェック内容	[エラー条件] • plan_dueが納期 < Try登録日
編集	編集

2.3 データ自動生成の課題と対策

- 課題：どの様なデータが必要かわからない
 - 区分値、数量等のバリエーション
 - 複数の区分等、テストに必要な組合せがわからない
- 対策：各種仕様で設計した分岐条件を満たす値、満たさない値の2種類を作成

受注(先行イベント)			出荷(後続イベント)			
受注番号	顧客	受注数	出荷番号	受注番号	出荷数	分完区分
1000	A社	20	2000	1000	10	分納
1001	B社	50	2010	1000	5	分納
1002	C社	30	2020	1001	50	完納

顧客テーブルに存在する値

自動採番

受注テーブルと紐づくキー項目

出荷数<受注数の場合
分納

エラーチェック
分完区分：完納の受注データを変更した場合エラー

データのバリエーション イメージ

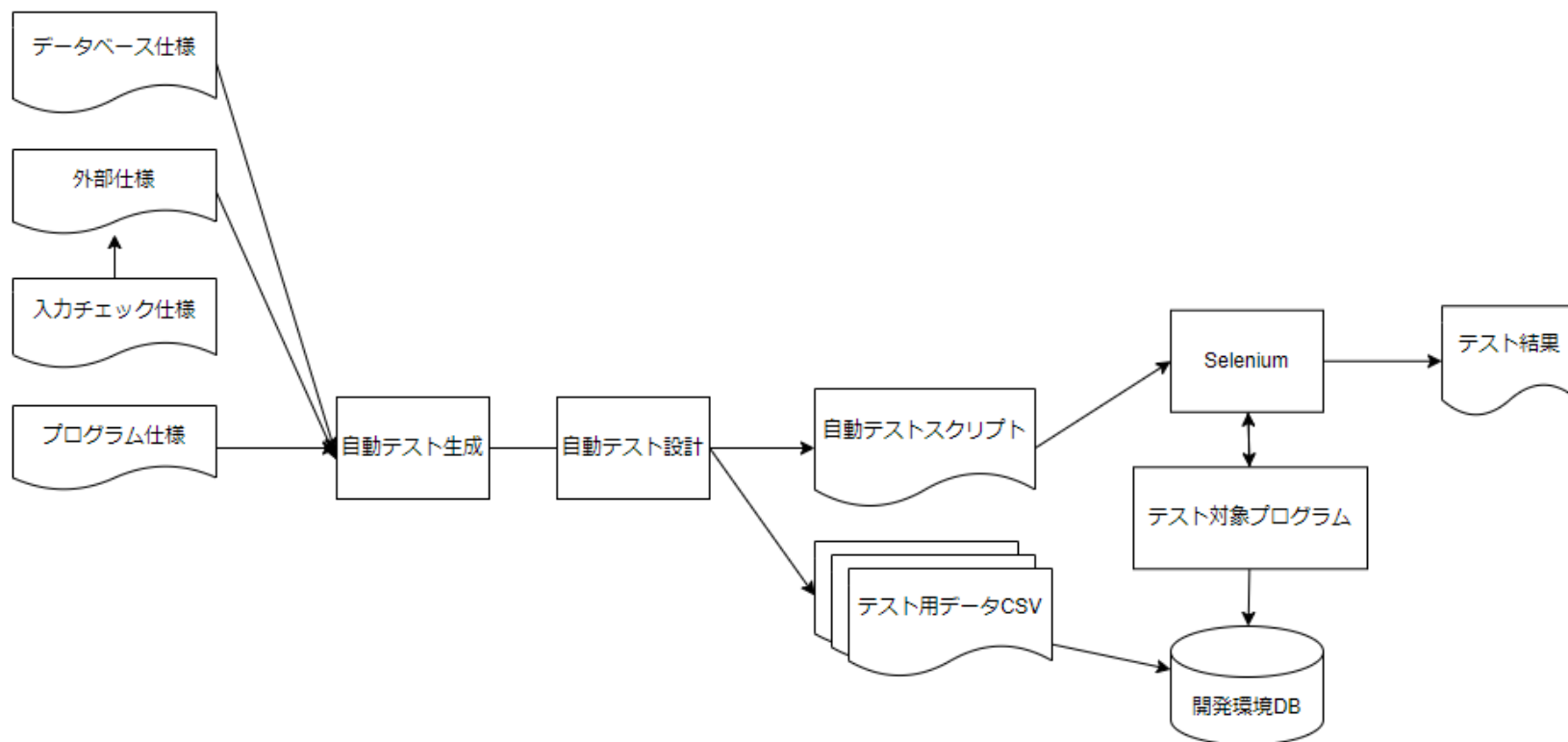
2.4 繰り返しテスト実行の課題と対策

- 課題：データのリセットに手間がかかる
 - 複数の開発者が1つのDBで開発しているケースが多い
 - 対象テスト用のデータと紐づくデータのみリセットしたい
- 対策：データリセット機能を作成
 - 生成したテスト用データと紐づく後続イベントデータを削除するSQL生成
 - 生成データの再登録、削除SQL実行をワンボタンで実行

3. 自動テスト生成システム

3.1 設計情報との連携

- 設計情報と自動テスト生成システムの関係



3.2 自動テスト設計画面

■ テスト設計[-]

All | 検索テスト01 | 検索テスト02 | 登録テスト01 | 更新テスト01 | 削除テスト01 | + | Delete

登録テスト01

概要

テストケースID: TC_Ent01 | テストケース名: 登録テスト01

テスト内容

パターン: PtnEntRFa PtnInqRDI PtnUpdRFa PtnDelRDIC

開始画面: 登録(np:20) 検索(np:0) 一覧(np:15)

備考

All | 登録_画面表示確認 | 登録_入力操作 | 登録_画面表示確認 | 登録_DB更新確認 | 登録_メッセージ確認 | + | Delete

登録_画面表示確認

概要

ステップ名: 登録_画面表示確認

備考

画面確認 ▶ 11件

No.	削除	テスト種類	対象	期待値	備考
1	<input type="checkbox"/>	画面遷移先	NP	20	
2	<input type="checkbox"/>	画面項目	表示Table 1 項目名称 [チーム名] 列 0 行 1	空値	
3	<input type="checkbox"/>	画面項目	表示Table 1 項目名称 [チーム区分] 列 0 行 1	空値	

登録_入力操作

概要

ステップ名: 登録_入力操作

備考

入力 ▶ 2件

No.	削除	表示Table	項目名称	項目ID	行	入力値
1	<input type="checkbox"/>	1		team_nr	1	オンショア10
2	<input type="checkbox"/>	2		userid_rf	1	

削除

操作

ボタン [登録] をクリック

[画面] チーム・メンバー登録 [登録]

メニュー 新規登録 検索 全件一覧

登録

チーム名称

チーム区分 xxx

大分類名

中分類名

小分類名

仕様書より登録、更新、削除等
パターン別テストシナリオが
自動生成される

自動生成されたテスト項目を
確認し、適宜修正する

3.3 自動テスト実行

All テスト実行 テスト選択 FAQ

■ テスト実行[-]

テスト設計No.	19084	改訂内容	
作成者	aah03852	作成日時	2024-08-07 17:21:32

■ テストケース

Eclipse環境にスクリプトとデータをコピー

▶ 5件 CSV

No.	実行 <input type="checkbox"/>	データ(PmsWeb)	TestCase	タイトル	テスト内容
1	<input type="checkbox"/>	<input type="button" value="登録"/> <input type="button" value="削除"/>	TC_Inq01	検索テスト01	
2	<input type="checkbox"/>	<input type="button" value="登録"/> <input type="button" value="削除"/>	TC_Inq02	検索テスト02	
3	<input type="checkbox"/>	<input type="button" value="登録"/> <input type="button" value="削除"/>	TC_Ent01	登録テスト01	
4	<input type="checkbox"/>	<input type="button" value="登録"/> <input type="button" value="削除"/>	TC_Upd01	更新テスト01	
5	<input type="checkbox"/>	<input type="button" value="登録"/> <input type="button" value="削除"/>	TC_Del01	削除テスト01	

■ 生成条件

▶ 8件 CSV

No.	分類	文書ID	タイトル	Rev.
1	ED	a.2.1.1	[画面] チーム・メンバー登録 [登録]	Rev.003
2	ED	a.2.1.2	[画面] チーム・メンバー登録 [検索]	Rev.003

設計した自動テストケースの一覧

テストケース用データを登録してテスト準備

実行するテストを選択して実行

4. まとめ

4.1 自動テストの課題と対策

分類	課題	対策
自動生成	入出力項目が 実装前に特定できない	画面設計機能作成
自動生成	項目やボタンの HTML IDがない	標準開発ツールの改善
テストケース	分岐条件がデータ化 されていない	分岐条件が存在する 各種仕様の設計機能作成
データ自動生成	必要なデータの バリエーションが わからない	分岐条件からデータの バリエーションを生成
繰り返しテスト (データリセット)	データベース内の 特定データのみを 削除するのは手間	ワンボタンでデータ削除と 再登録を行う機能を作成

4.2 成果

不十分

成果	目標	状態
-	開発工数2MH削減	未試行
△	自動テスト生成率60%	生成率30%

現時点では定量的に有効、無効を判断できるレベルには達していない
しかし、実現できればプログラム開発、保守に与えるインパクトは大きい
試行に向けて調整、改善を行っていきたい

今後の課題

- データ化された設計情報と自動テスト生成の連携不足
- データ化後の仕様書を用いた実プロジェクトでの開発試行
- 自動テストを用いたテスト駆動開発の実地運用 etc.

The END