

# JaSST15

## エンジニアの幸せの形

私の自動テストの歴史

2015.2.18

# 自己紹介

## ❖ 所属

- ❑ 株式会社OSK R&D本部 品質保証部 品質管理2課
- ❑ 小井土 亨

## ❖ 担当

- ❑ パッケージソフトの開発に関する業務のシステム化
- ❑ システムテストの自動化のためのツールやスクリプト開発
- ❑ パッケージソフトの品質管理及び技術支援

## ❖ 経験

- ❑ 入社後、一貫して業務パッケージソフトの開発
- ❑ 近年は、パッケージシステムのアーキテクチャ策定とソフトウェアコンポーネントやフレームワークの開発

# 私の自動テストの歴史 セルフテスト

## ❖ セルフテスト時代(1985年)

### □ 対象

- MS-DOS3.1対応の業務パッケージソフト

### □ テスト方法

- パッケージの持つ、マクロ機能を使って自動テストを実行

### □ ヒント

- マクロのテストをしていて、他の機能の自動テストとなることに気が付いた

### □ 効果

- 検査の効率化

### □ 問題

- テストできる範囲が限られる
- 実行もチェックもマクロで行うため、実行が安定しない
- 利用できるシステムが限られる

# 私の自動テストの歴史 トレーステスト

## ❖ トレーステスト時代(1992年)

### □ 対象

- Microsoft Windows3.0対応の業務パッケージソフト

### □ テスト方法

- OSのトレース機能と再生機能を使って自動テストを実行

### □ ヒント

- オートデモをOSの機能を使って作成、これを応用して自動テストを実現

### □ 効果

- 検査の効率化

### □ 問題

- 変更にも弱く、安定して実行できない
- テストケースを作成するトレースのコストが掛かる
  - 操作ミスや無駄な操作もトレースされる
  - 検査対象が正しく動作しないとトレースできない

# 私の自動テストの歴史 テストファースト

## ❖ eXtremeProgrammingテスト技法(2001年)

### □ 対象

- Web対応の業務パッケージソフト用ライブラリ
- 単体テストの自動実行
- 常時結合
  - ソース管理、ビルドツール、ユニットテストを連携

### □ テスト方法

- ソースの更新をトリガーとして、コンパイルとユニットテストを実行しエラーがなければ、モジュールを公開

### □ 効果

- 品質の維持
- 開発のリズムと素早いフィードバック

### □ 問題

- 対象が単体テスト

# 私の自動テストの歴史 キーワードテスト

## ❖ キーワードテスト(2007年)

### □ 対象

- .Net Framework 対応の業務パッケージ

### □ テスト方法

- Excelで、ルールに従ってテストケースを作成し、専用の実行モジュールで実行
- 環境構築や結果の収集と判定はスクリプト言語で行う

### □ ヒント

- 2005年のMicrosoft UI オートメーション ライブラリの記事
  - セシビリティ専用機能として設計されているが、UI テスト自動化で利用可能

### □ 効果

- テストケースの作成効率アップ
- PICTなどでの、自動生成も可能
- 比較的变化に強い

# 私の自動テストの歴史 現在

## ❖ 自動テストの管理と運用(2014年)

### □ テストシナリオ

- 約5000のテストシナリオ、実行時間は1000時間

### □ 管理

- モジュールとテストシナリオの関係をシステム化
- 変更されたモジュールに関するテストシナリオの選択と実行をほぼ自動化

### □ 運用

- 並列実行環境をスクリプト(PowerShell)を利用して構築
- 結果判断も自動(画面、PDF、DBの比較を自動化)
- 実行結果の収集も、ほぼ自動化
- カバー率を収集し未実行部分を可視化して、テストシナリオを追加

### □ 課題

- 効率的なテストの運用
- テストでNGとなった場合の調査作業の効率化

# 自動テストで重要なこと

## ❖ 昔からこれは変わらない

- 自動テストを効果的に利用するためには、改善が必要
  - 自動化することで効果や必要なことが明確になり、改善できる
- 自動テストの保守・運用コストを考える
  - テストの作成と実行だけではなく、テストを継続的に運用するために必要な保守・運用コストも考える

## ❖ こう思うようになってきた/気づいた

- 開発プロセスの視点で、テストの自動化を考える
- システムテストの自動化は、チーム全体で考える
- 複数のツールを組合わせて連携させることを考える
  - 単一のツールを導入するだけではなく、自作ツールや市販ツール(含むフリーソフト)などを組合わせて、効果的にテスト環境を構築する

# 最近の自動化事例

## ❖ 自動検査用フレームワーク型スクリプト

### □ 役割分割(レースをメタファー)

- ランナー(テストを実行する)
  - 各クラスの組み合わせを行う
- コントローラー(テストをコントロールする)
  - テストの制御/テスト結果の書き出し/情報の収集
- マネージャー(テストを管理)
  - 実行するテストスイートの管理
- メカニック(テストの準備)
  - 初期化/テストデータのリストア/必要のないファイルの削除
- ドライバー(テストの実行)
  - テストの実行/テスト結果の収集
- チェッカー(テストの確認)
  - テストが成功したか、失敗したかを判断

# 取り組んでいること 自動テストの品質向上

- ❖ システムテストの自動テストとアーキテクチャ
  - 自動テストのシステム構成
    - 二つのシステム
      - テスト対象システム
      - 自動テストシステム
    - 自動テストシステムのサブシステム
      - テストケースの作成
      - テストケースの実行
      - 自動テストの運用
  - システムごとに異なった関心事
  - システム単位で要求される品質特性を明確化
  - 品質特性を達成するためのパターン
    - パターンの組合わせが、品質特性を達成する近道
  - 公開スライド
    - <http://www.slideshare.net/koido1961/ss-39837146>