

## 評価補助ツール開発による コードカバレッジ測定の導入と不具合防止に向けた取り組み

- 開発エンジニアによる継続的開発・評価のための仕組み造り-

株式会社ワークスアプリケーションズ COMPANY Web Service Group エキスパートエンジニア 大畠 悠介



### <u>AGENDA</u>

- 弊社でのQC活動取り組み状況
  - ■魚の骨分析図
  - ■対策一覧表
- カバレッジを運用に乗せる為に実施したこと
  - ■オープンソースをベースとしたツール開発
    - ◆ JavaScriptカバレッジ計測ツール
    - ◆案件別カバレッジ結果出力ツール
  - 実際に行っている運用手順
- ■効果測定



## 取り組みの背景

- パッケージ製品の開発を重ねていく度に設定値の組み合わせ量が増加し、 テスト量が増加し続けている。
- お客様より製品に対する機能追加の要望も多いため、<u>短いサイクル</u>での 製品リリースを求められる状況は変わらない。
- 過去にテストケースやドキュメントの整備などいくつかの対策を実施した経緯はあるが、目立った改善は見られなかった。
- → 出荷後に発見される不具合件数はリリースを重ねるごとに増加傾向にあり、 QC活動を通じて、<mark>開発・評価の仕組み</mark>を改善することでプロダクトの品質を 向上しようと本取り組みはスタートした。



## 対策概要の一覧



## カバレッジは費用対効果が高そう!!



## カバレッジ導入にあたっての問題点

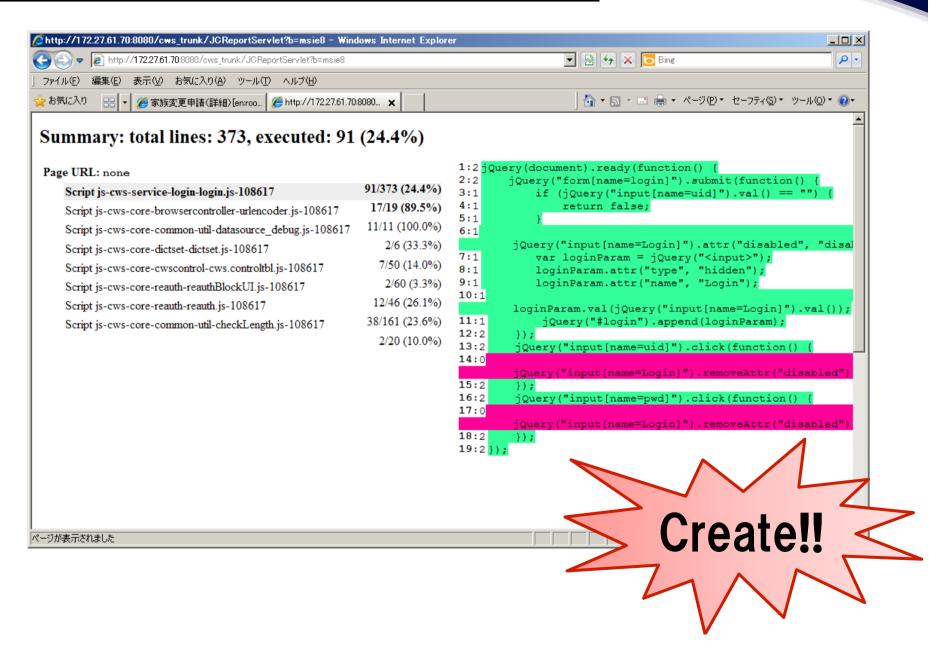
- 主要プログラムの1つであるJavaScriptに関する コードカバレッジが、サポート対象ブラウザにて測定できなかった。
- 10年以上続いている製品の為、デッドコードが存在する箇所も存在し、カバレッジ100%を達成することは不可能であった。
- カバレッジ計測結果の利用方法が分からない開発者や 評価者が大勢いた。
- 明確な利用基準が設けられない為に、過去に1度失敗した 経緯がある。



## Java Script Coverage



## IE/FirefoxでのJSカバレッジ計測ツール





```
var ctr = 0;
document.body.onclick = function() {
    ctr = ctr + 1;
    if (ctr > 4) {
        alert(ctr);
    } else {
        alert(ctr);
    }
}
```



```
//BRT_BLOCK_BEGIN:1
var ctr = 0:
document.body.onclick = function () {
         //BRT_BLOCK_BEGIN:2
        ctr = ctr + 1:
        if (ctr > 4) {
                 //BRT_BLOCK_BEGIN:3
                 alert (ctr);
                  //BRT_BLOCK_END:3
        } else {
                 //BRT_BLOCK_BEGIN:4
                 alert (ctr);
                  //BRT_BLOCK_END:4
         //BRT_BLOCK_END:2
//BRT_BLOCK_END:1
```



```
//BRT_BLOCK_BEGIN:1
var ctr = 0:
document.body.onclick = function () {
         //BRT_BLOCK_BEGIN:2
        ctr = ctr + 1:
        if (ctr > 4) {
                 //BRT_BLOCK_BEGIN:3
                 alert (ctr);
                  //BRT_BLOCK_END:3
        } else {
                 //BRT_BLOCK_BEGIN:4
                 alert (ctr);
                  //BRT_BLOCK_END:4
         //BRT_BLOCK_END:2
//BRT_BLOCK_END:1
```



```
//BRT_BLOCK_BEGIN:1
var ctr = 0:
document.body.onclick = function () {
//BRT_BLOCK_BEGIN:2
  ctr = ctr + 1:
  if (ctr > 4) {
  //BRT_BLOCK_BEGIN:3
     alert (ctr):
  //BRT_BLOCK_END:3
  } else {
  //BRT_BLOCK_BEGIN:4
     alert (ctr);
  //BRT_BLOCK_END:4
//BRT_BLOCK_END:2
//BRT_BLOCK_END:1
```



JsCoverageの仕組み window.scriptObjects [scoverage-a.js-O].executedBlock [1] = (window.scriptObjects ['scoveragea.is-0'].executedBlock[1] ? window.scriptObjects['scoverage-a.js-0'].executedBlock[1] + 1:1): var ctr = 0: document.body.onclick = function () { //BRT\_BLOCK\_BEGIN:2 window.scriptObjects ['scoverage-a.js-0'].executedBlock [2] = (window.scriptObjects ['scoveragea.is-0'].executedBlock [2] ? window.scriptObjects ['scoverage-a.js-0'].executedBlock [2] + 1 : 1); ctr = ctr + 1: if (ctr > 4) { //BRT BLOCK BEGIN:3 window.scriptObjects ['scoverage-a.js-0'].executedBlock [3] = (window.scriptObjects ['scoveragea.is-0'].executedBlock[3] ? window.scriptObjects ['scoverage-a.js-0'].executedBlock [3] + 1 : 1); alert (ctr): //BRT BLOCK END:3 } else { //BRT BLOCK BEGIN:4 window.scriptObjects ['scoverage-a.is-0'].executedBlock [4] = (window.scriptObjects ['scoveragea.is-0'].executedBlock [4] ? window.scriptObjects ['scoverage-a.js-0'].executedBlock [4] + 1 : 1); alert (ctr): //BRT BLOCK END:4 //BRT\_BLOCK\_END:2 *}:* 

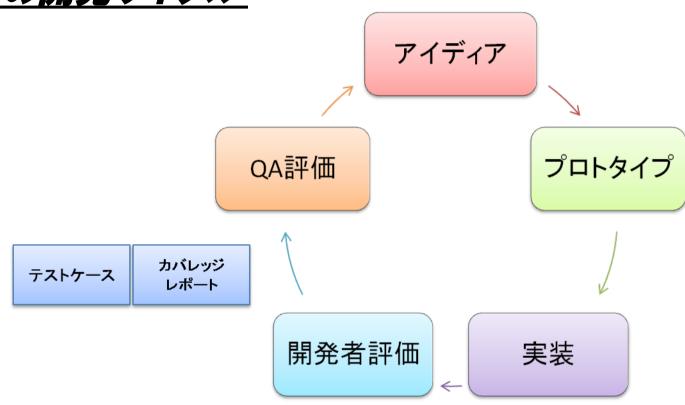
//BRT BLOCK END:1



# 運用手順



## カバレッジ運用に関するご説明の前に・・ 部署の開発サイクル



- ・開発評価の完了基準はカバレッジが100%通っていること
- ・品質保証部門はカバレッジ100%のテストケースを閲覧できるため、 テスト観点の抜けをチェック可能



## カバレッジを運用に乗せるまでの課題

## 課題1:「カバレッジ率90%」の善し悪しが判定できない

→ スモールスタートとして、カバレッジ対象ソースコードを 前回リリースからの差分のみとし、<u>100%</u>を完了基準にした。

### 課題2: カバレッジ測定方法が分からない

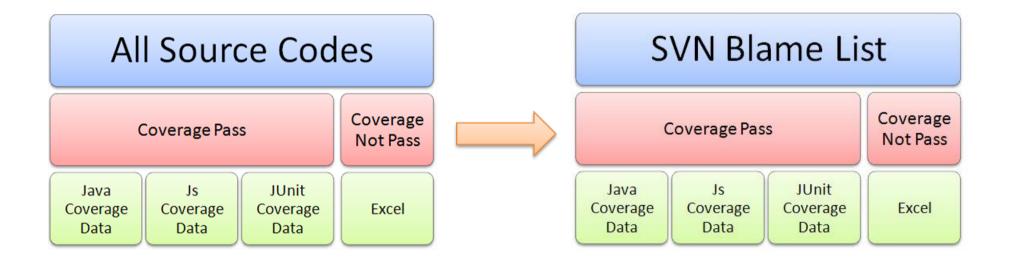
→ ツールが利用できる環境をクラウド (Amazon EC2) 上に整備し、 利用時はインスタンスをコピーするだけの状態とした。

### 課題3: カバレッジ測定が面倒くさい

→ 案件番号を入力するだけで担当分のカバレッジ未通過行一覧が、EXCELで閲覧可能なツールを開発した。



## 前回リリースからの差分のみって?



#### カバレッジ未通過のカウント方法を変更し、

#### と考えた。

※SVN等にてソース管理をしていると、ソースの1行1行に対してリビジョンを持っている為、特定(前回リリース番号等)のリビジョンより前の行は「通過行」としてカウント可能にできる!



## 未通過行一覧作成ツール 概要

1. カバレッジを測定する

**Coverage Pass** 

**Not Pass** 

2. 計測結果を取得

Java.xml

Js.xml

3. ソースと突合する

Source Codes

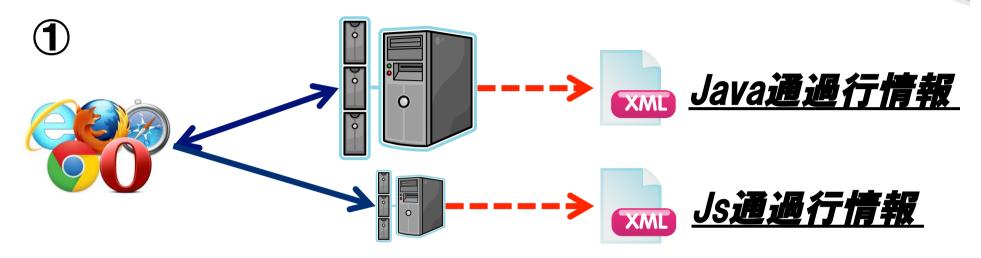
4. 未通過行がエクセルに出力される

Excel

5. 未通過一覧を利用して、再テスト



## カバレッジ計測 全体図









## 未通過行一覧の使い方

revision	class_name	row_num	ber u	ıser	line
99828	jp/co/worksap/companyweb/sys/CwsSessionManager.java	<u>414</u>	oł	hata_y	trace("Single-Sign-On By Unknown method!!");
99828	jp/co/worksap/companyweb/sys/CwsSessionManager.java	<u>435</u>	oł	hata_y	trace("Single-Sign-On By APServer EnvVal user-id = " + uid);
99828	jp/co/worksap/companyweb/sys/CwsSessionManager.java	<u>441</u>	oł	hata_y	trace("Single-Sign-On By RemoteUser user-id = " + uid);
99828	jp/co/worksap/companyweb/sys/CwsSessionManager.java	<u>455</u>	oł	hata_y	trace("Single-Sign-On By Cookie user-id = " + uid);
99828	jp/co/worksap/companyweb/sys/CwsSessionManager.java	<u>475</u>	ol	hata_y	trace("Single-Sign-On By Parameter user-id = " + uid);
99828	jp/co/worksap/companyweb/sys/CwsSessionManager.java	<u>481</u>	oł	hata_y	trace("Single-Sign-On By ServletRequest attribute user-id = " + uid);
99828	jp/co/worksap/companyweb/sys/CwsSessionManager.java 🦯	<u>488</u>	oł	hata_y	trace("Single-Sign-On By SAML user-id = " + uid);
99828	jp/co/worksap/companyweb/sys/CwsSessionManager.java	<u>496</u>	ol	hata y	trace("Single-Sign-On By Unknown method!!");
	99828 99828 99828 99828 99828 99828 99828	99828 jp/co/worksap/companyweb/sys/CwsSessionManager.java 99828 jp/co/worksap/companyweb/sys/CwsSessionManager.java 99828 jp/co/worksap/companyweb/sys/CwsSessionManager.java 99828 jp/co/worksap/companyweb/sys/CwsSessionManager.java 99828 jp/co/worksap/companyweb/sys/CwsSessionManager.java 99828 jp/co/worksap/companyweb/sys/CwsSessionManager.java	99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         414           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         435           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         441           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         455           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         475           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         481           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         488	99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         414         c           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         435         c           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         441         c           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         475         c           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         481         c           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         481         c           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         488         c	99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         414         ohata_y           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         435         ohata_y           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         441         ohata_y           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         455         ohata_y           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         475         ohata_y           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         481         ohata_y           99828         jp/co/worksap/companyweb/sys/CwsSessionManager.java         488         ohata_y

```
CwsSessionManager.java - Windows Internet Explorer
(△) ▼ B http://uranus/trunk/jp.co.worksap.companyweb.sys/CwsSessionManager.java.html! ▼ ★ X Io Bing
 ファイル(E) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルブ(H)
 🏠 お気に入り
                                                              🏠 • 🗟 ፣ 📑 🛖 • ページ(P) • セーフ
            CwsSessionManager.java
                 return req.getSession().getAttribute(AUTH PRM) != null;
 242.
 243.
              return false;
 244.
 245.
 246.
          private boolean isAuthenticatedRequest (HttpServletRequest req, HttpServletResponse r
 247.
                 throws Exception {
 248.
              CompanySession session = getSession(req);
 249.
              //そもそもセッションがない場合
 250.
              if (session == null) {
 251.
                 return false;
 252.
 253.
 254.
              // セッションが存在しても以下のケースでは場合強制的にログアウトを行う
 255.
              // 以下の処理はもともとORで結合されていたため処理順序は問わない
 256.
 257.
              // ケース1 ユーザIDとパスワードが送信され現在のログオンユーザと異なる場合
 258.
              String userId = session.getUser().getUserid();
 259. 4
              if (session.getUser().transFormMode()) { //CWS420100 成りすましモードの場合
 260.
                 userId = session.getUser().getOriginalUser().getUserid();
 261.
 262.
```



# 環境準備



## JaCoCo (Java Code Coverage Library)

- javaagentを利用したon-the-fly方式
- 起動引数に1行追加するだけ!

-javaagent: [yourpath/] jacocoagent.jar= [option1] = [value1], [option2] = [value2]

#### 例)

WebSphereでは管理コンソールから「アプリケーションサーバ > (サーバー名) > プロセス定義 > Java 仮想マシン」の「汎用JVM 引数」に下記の様に追記します。

-javaagent:F:\footnote{\textit{F:} IBM\footnote{\textit{WebSphere}} AppServer\footnote{\textit{java}} jre\footnote{\textit{Ib}} ext\footnote{\textit{jacocoagent.jar=destfile=F:} IBM \footnote{\textit{YWebSphere}} AppServer\footnote{\textit{profiles}} AppSrvO1\footnote{\textit{Jogs}} exec, includes=jp/co/worksap/\footnote{\textit{worksap}}



## **JsCoverage**

- JSファイルのInstrument(カバレッジデータ埋め込み) ディレクトリを指定して、bat/antで実行するだけ
- Webアプリケーションへの配置
  WEB-INF/libにjs-coverage-tool.jarを設置
  web.xmlに数行追記
- カバレッジデータの収集(レポート出力) ディレクトリを指定して、bat/antで実行するだけ



## 未通過一覧作成ツール

- 以下をINIファイルに記載して実行
  - SVN認証情報・ソースのディレクトリ
  - カバレッジ通過XMLのディレクトリ
  - 対象としたい開発者の名称など
  - ■リビジョン範囲



- 結果ファイル中身
  - チェックイン時のコメント
  - ■修正リビジョン番号
  - クラス名称・行番号
  - 開発者名称
  - 行の内容



# 効果測定



## 効果測定

## <u> 定量的効果</u>

カバレッジ対策にてフォーカスした要因に対しての不具合件数がリリース毎の不具合数比較にて約7割減少した

## 

開発時にカバレッジ測定を意識したソースコードを作成する開発者が増加 自主的にカバレッジ測定を開始した部署が出てきた

→品質に対する意識が高まったと考えることができる



## こんな場合はぜひカバレッジを!

- ・組み合わせテストが多いので、Pairwiseなどを利用しているが正しく利用できているかどうか、不安が残っている。
- ・協力会社等のプロジェクト関係者が多いが、<u>自社に関係する部分</u> だけに絞ったカバレッジを測定したい
- ・特定期間(特に締切間際など)における、コミュニケーションロス (修正内容把握不足)が心配だ
- ・開発者が勝手に修正してしまった部分を検知したい



## 今後の展望

■ カバレッジ測定の利用を社外にも広めるため、今回開発した 測定ツールをオープンソースとして公開します!

URL

https://github.com/WorksApplications (今後、QC活動にて作成したツールは上記にてどんどん公開していきます!)



## Thank You!!

ご清聴ありがとうございました。

ご質問・ご意見などは

ohata y@worksap.co.jp までお願い致します。