

機能テストの “自動化”

自動+手動テストケース共生による

機能テストの現実的な自動化

2009.01.29 JaSST2009

(株) 永和システムマネジメント

懸田 剛

懸田剛

かけだ たけし



Change Vision, Inc.
Takeshi Kakeda
takeshi.kakeda@change-vision.com

Tangible Bug Tracking Using LEGO Bricks

AgileConference2008
8/5 Sheraton A, LC

Seeing is understanding.



アジェンダ

- ・背景
- ・試行内容
- ・利点
- ・課題
- ・次へのTry
- ・まとめ

”自働化” とは？

不良が発生した際に機械が自動的に停止し、後の工程へ良品のみを送るようにすること、公式ページでは通常の加工が完了したら機械を安全に停止させることも指すとしている。豊田佐吉が発明した自動織機に、稼動中に糸が切れた際に自動で停止する装置が組み込まれていたことに由来している。

半自動化

**Semi-
Automatic**

背景

- IBM JRE 1.5
- Tomcat 5.x
- GWT 1.5
- **IBM Rational Team Concert 1.0**

TRICHORD

Jazz Repository

Koara - Koara Team - 10月までのイテレーション

TRICHORD



Todo

- 582 負荷、パフォーマンスのチューニング
- 592 負荷パフォーマンスのテスト(新)
- 583 回帰テスト(1)
- 584 回帰のバグフィックス
- 585 回帰テスト(2)
- 590 ログイン画面からデモユーザーの記述を削除する

Doing

- 580 起動スクリプト作成 (Win/Linux)

DONE

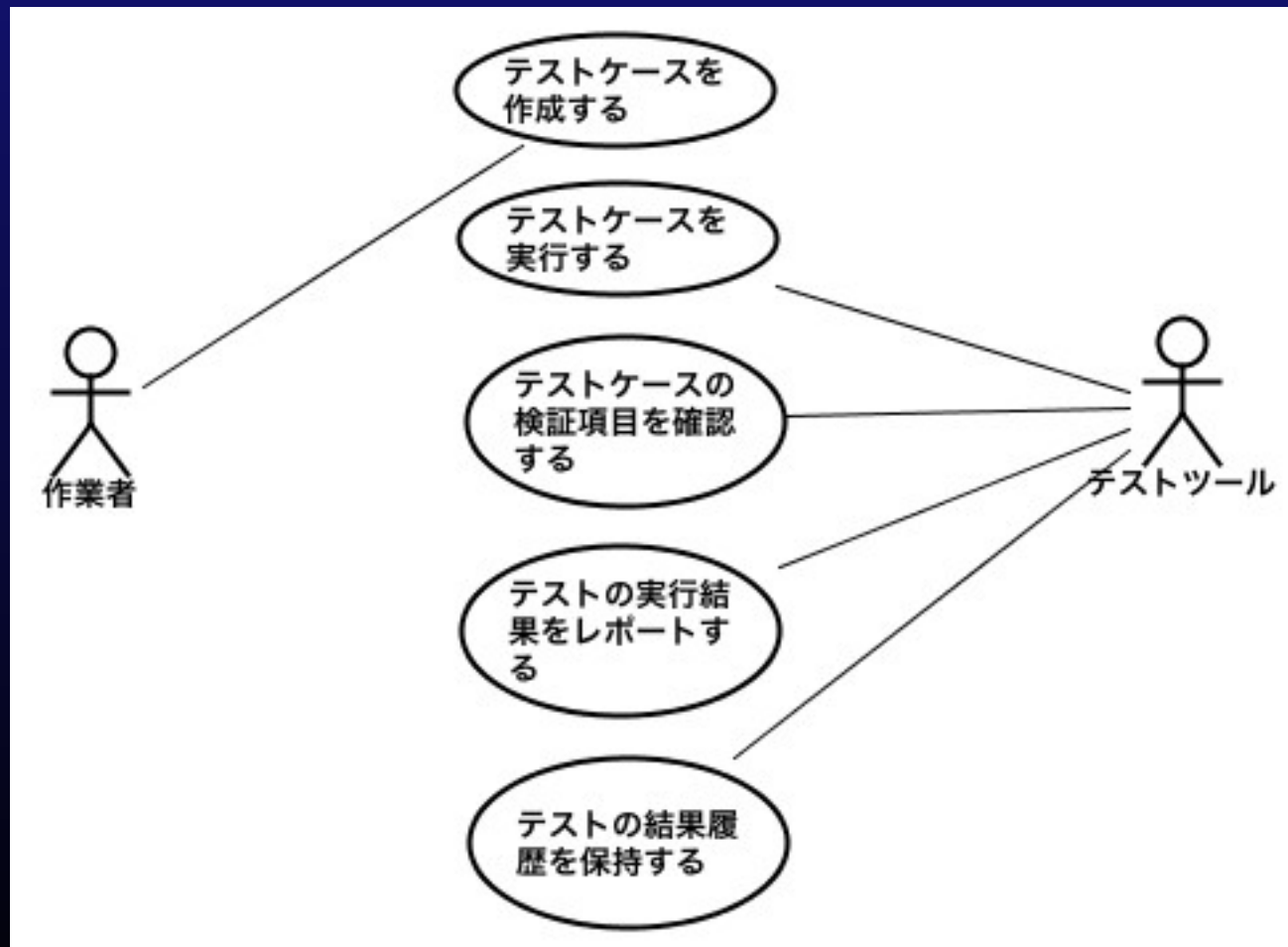
- 588 Proでないでアップロードできないようにする
- 578 同時にアップロードしても不正な状態にならないよう
- 581 ライセンス置き場
- 577 バージョンやライセンス情報を確認できるようにする
- 589 配布物からライセンスを削除する
- 591 負荷パフォーマンス
- 576 サンプル

完了

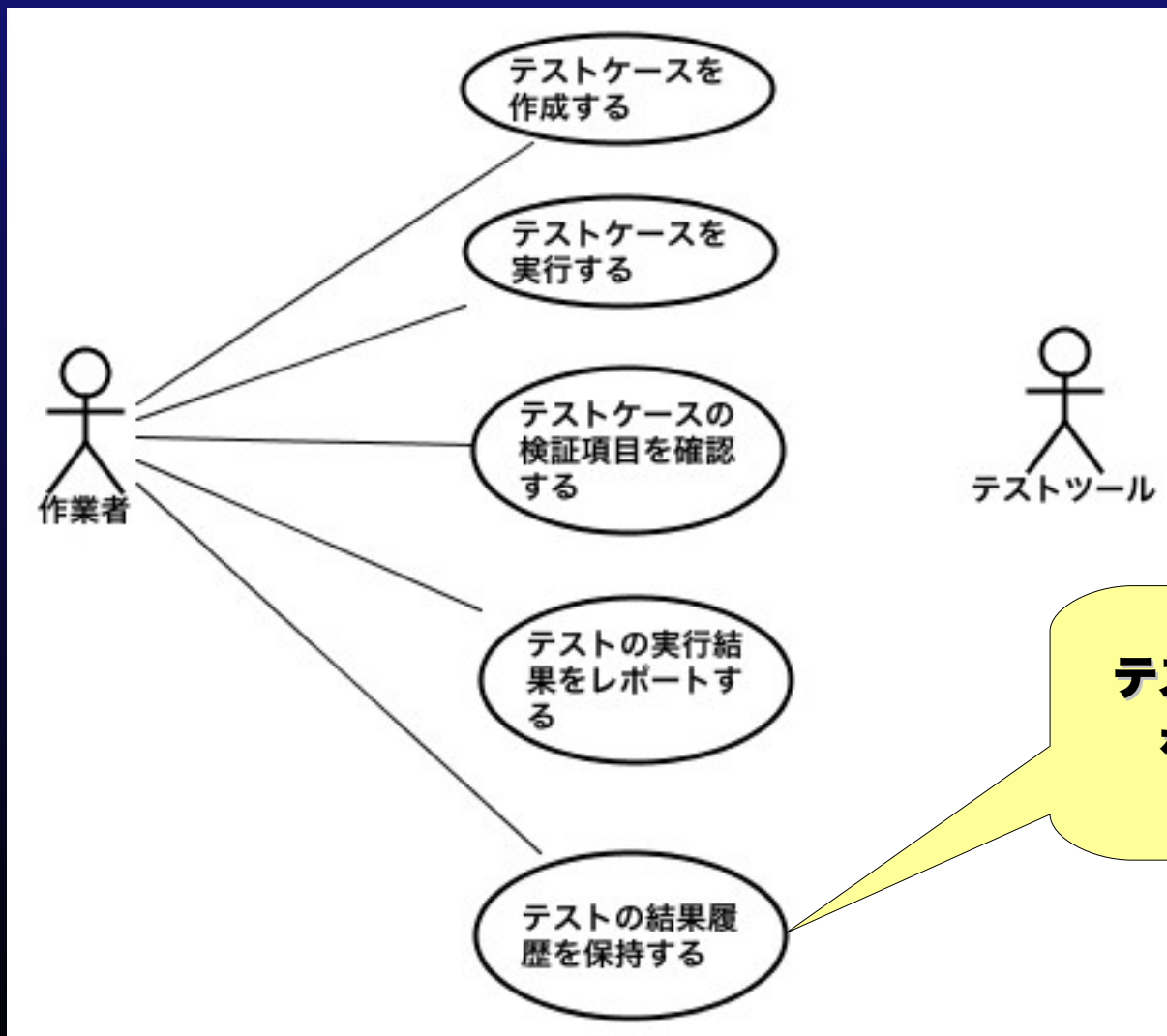
ilove:9543

41

自動



手動



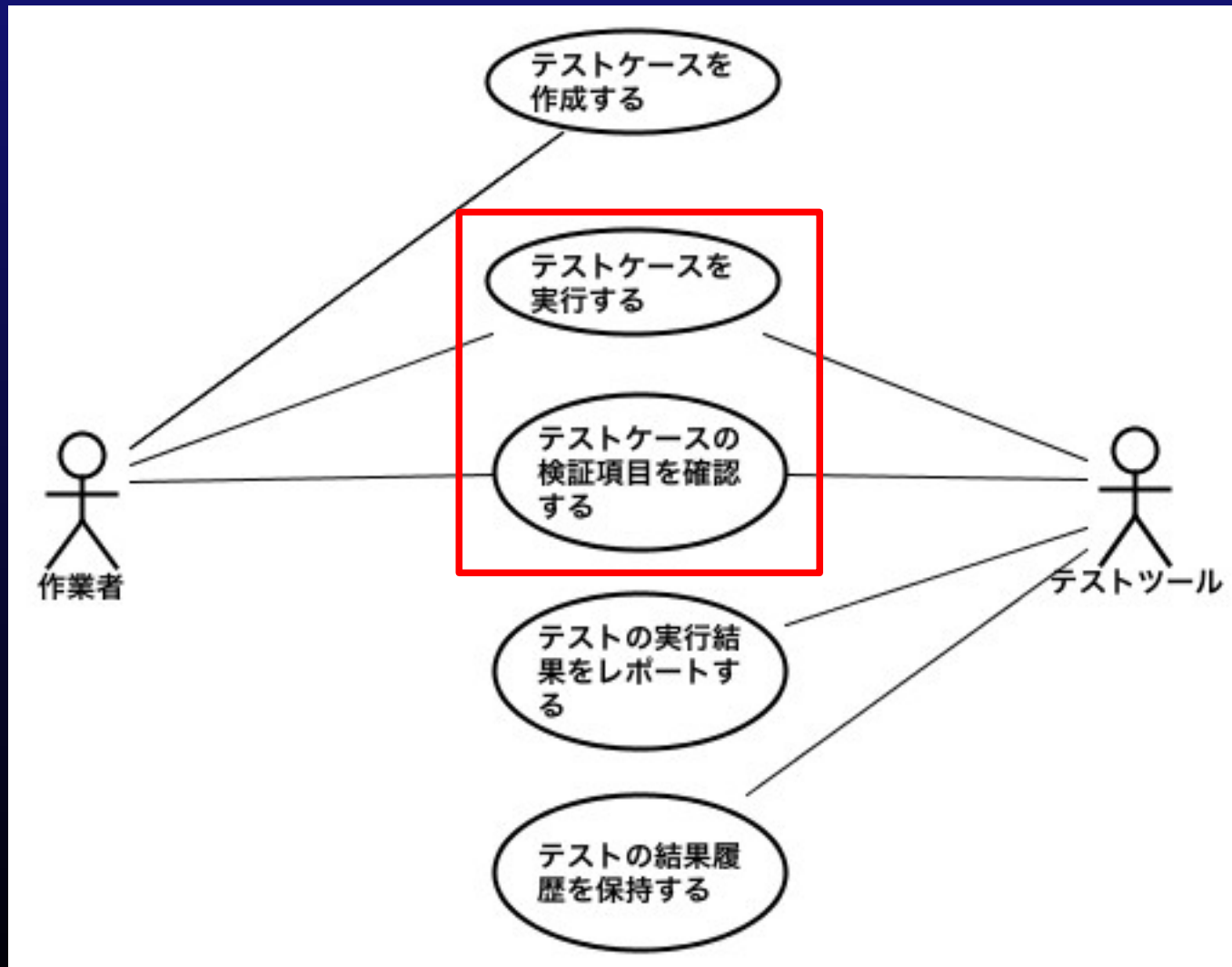
背景 - 課題

自動テストと
手動テストの
管理が分散

手動テストの
実施コスト大

自動化が
困難なケース

自働（半自動）



試行内容

.assertManuallyというメソッドを作成

- ・ 人がアサーションする
- ・ JUnit+SeleniumRC内で利用

.AssertManuallyの振舞

- ・ 1.テスターに目視する内容を提示する
- ・ 2.テストする時間待つ
- ・ 3.テストの結果がどうだったかをテスターに問う
 - ・ テストがまだ終わっていない場合は再度2へ
 - ・ アサーションがOKなら次のステップへ
 - ・ アサーションがNGなら失敗で例外

.設定で(自動+手動)、(自動のみ)を選択可能

手動 + 自動テストの例 (1)

```
Public class AcceptanceXXXTest extends TestCase {  
    protected void setUp() {  
        do_something(); // 事前処理  
    }  
    Protected void tearDown() {  
        do_something(); // 事後処理  
    }  
}
```

```
public void test_ログインが成功時ダイアログがアニメーションして表示されること() {  
    // 画面遷移などの自動処理
```

```
    open_login();  
    input_id_and_password("user", "pass");  
    push_ok();  
    assertEquals(" ログイン成功", get_message());
```

通常のSeleniumRCの
テストのように自動処理

```
    // マニュアルモードなら手動実施
```

```
    If (isManual()) {
```

```
        assertManually(" ダイアログがアニメーションして表示しましたか?",  
                        "(失敗時)ダイアログがアニメーションしませんでした");
```

```
    }
```

```
}
```

```
}
```

手動で目視確認する内容を記述。
テストケースがテストターに
確認すべき内容を知らせる

手動 + 自動テストの例 (2)

```
Public class AcceptanceXXXTest extends TestCase {  
    protected void setUp() {  
        do_something(); // 事前処理(データロードなど)  
    }  
    Protected void tearDown() {  
        do_something(); // 事後処理  
    }  
}
```

```
public void test_ログインが成功時ダイアログがアニメーションして表示されること() {  
    // 画面遷移など  
    open_login();  
    input_id_and_password("user", "pass");  
    push_ok();
```

シナリオ部分も手動で
テスターに実行してもらう

// マニュアルなら実行する

```
If (isManual()) {  
    showMessage(" 画像をドラッグして領域外にドロップしてください");  
    assertManually(" カースルのアイコンが禁止アイコンになりましたか?",  
        " アイコンが禁止アイコンになりませんでした");  
}
```

手動で目視確認する内容を記述。
テストケースがテスターに
確認すべき内容を知らせる

3つの利点

管理
一元化

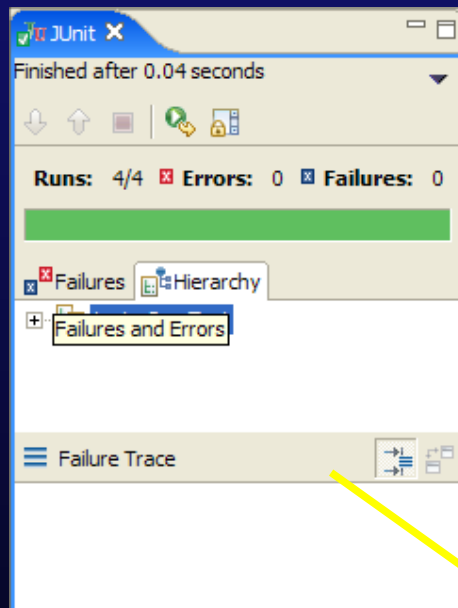
一部(自動|手動)
による
効率化

段階的
自動化

管理一元化

自動テスト

手動テスト



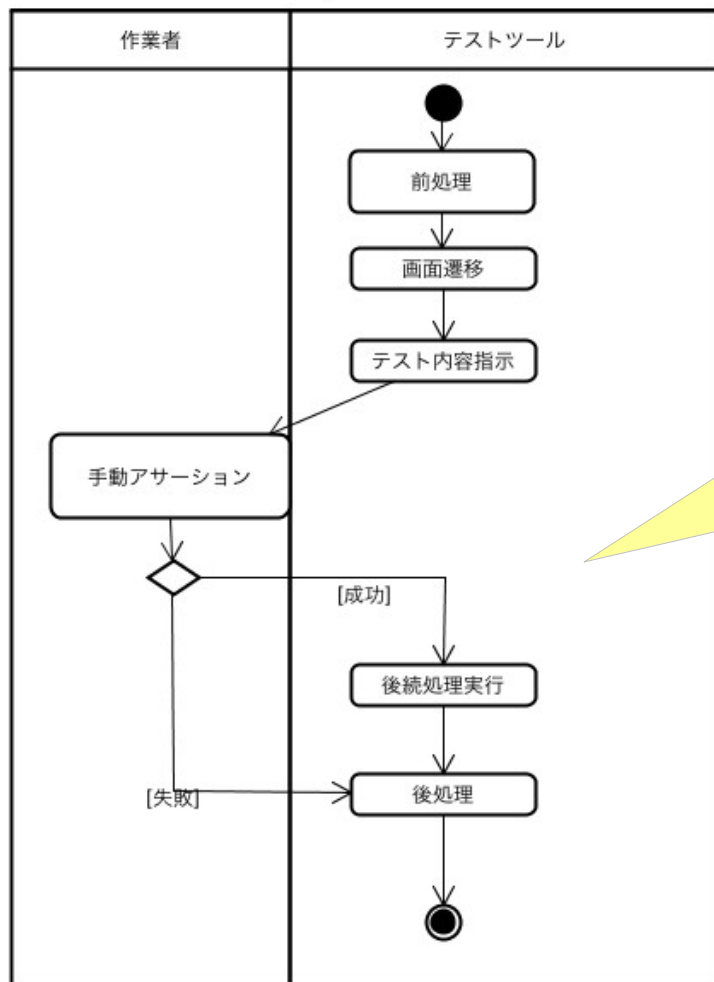
Microsoft Excel - 結合試験仕様書.xls

システム名		サブシステム名				作成会社		作成者	
販売・調達管理システム		販売管理システム				SI		梅田	
テーブル	項目	パターンA	パターンB	パターンC	パターンD				
部門マスタ	部門コード	11	11	12	12				
顧客マスタ	顧客コード	001	002	001	002				
商品マスタ	商品コード	P01	P02	P01	P02				
在庫管理	在庫引当	する	しない	する	しない				
受注データ	受注日	2005/12/01	2005/12/01	2005/12/01	2005/12/01				
	受注部門コード	E01	E01	E01	E01				
	受注担当者コード	P01	P01	P01	P01				
	数量	10	10	10	10				
	商品コード	P02	P02	P02	P02				
	数量	10	10	10	10				

集約

効率化

act 手動アサーションによるテスト



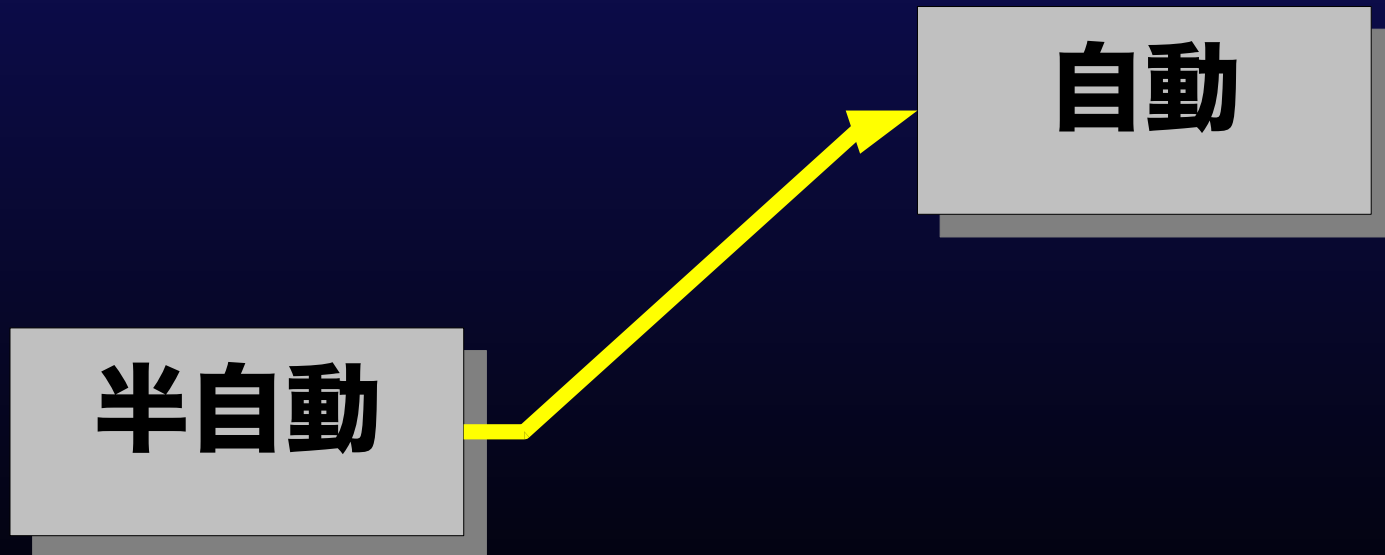
**自動/手動の
混在**

半自動向け

- ・ 自動アサーションが難しい部分
 - ・ 視覚効果
 - ・ 形
 - ・ レイアウト
 - etc
- ・ 自動化できない処理
 - ・ 外部環境への操作
- ・ 変更周波数の大きい箇所

段階的自動化

- ・ 最初は半自動
 - ・ 難易度の高い部分は手動検証
- ・ 検証後に自動化



課題

- ・ 同一画面上に指示ダイアログを表示してしまう
 - ・ alert/confirmを何とかすべき
- ・ 途中から、単一実行に難
 - ・ 手動ミスのリカバリ
 - ・ バラバラに実行した時のテスト結果の扱い
- ・ テスト開始時にテスト内容が不明
 - ・ そのテストがどういうテストなのか直前まで分からない

同一画面上に指示ダイアログ

```
Public class AcceptanceXXXTest extends TestCase {  
    protected void setUp() {  
        do_something(); // 事前処理(データロードなど)  
    }  
    Protected void tearDown() {  
        do_something(); // 事後処理  
    }  
}
```

```
public void test_ログインが成功時ダイアログがアニメーションして表示されること() {  
    // 画面遷移など  
    open_login();  
    input_id_and_password("user", "pass");  
    push_ok();
```

実行時にダイアログが
表示してしまう

```
// マニュアルなら実行する
```

```
If (isManual()) {  
    showMessage(" 画像をドラッグして領域外にドロップしてください");  
    assertManually(" カースルのアイコンが禁止アイコンになりましたか?",  
        " アイコンが禁止アイコンになりませんでした");  
}
```

```
}
```

次への Try

- ・ チートシートウィンドウ
 - ・ テスト手順はそちらに表示する
- ・ テスト結果のReRun対応
- ・ 実行前にテストのタイトルを表示する
- ・ 汎用化
 - ・ ツール、フレームワーク化

まとめ

- **自動+手動=半自動(自働)**

- コンピュータは単純作業を実施する
- 人は複雑な作業と不良の確認を担当する

- **現実的なオートメーション**

- **過度な自動化、妥協した手動化を防ぐ**
- **もう一工夫必要**

御静聴

ありがとう

ございました