DevOpsから見たテスト自動化と価値の見える化（ダイジェスト版）

June/17/2016

Rakuten Inc.
Ecosystem Service Department
Delivery and Reliability Solution Group
Group Manager
荻野恒太郎

http://corp.rakuten.co.jp/careers/engineering/

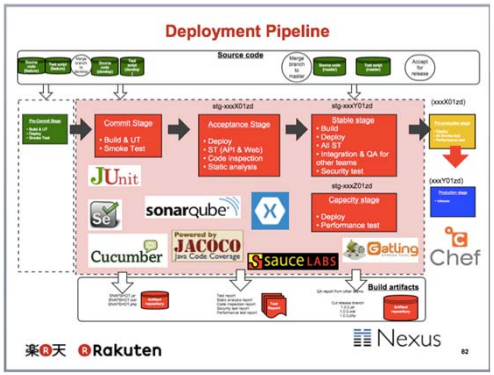
Rakuten

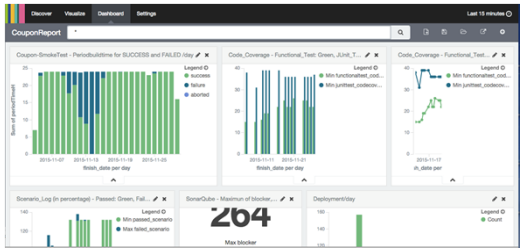
「Delivery and Reliability Solution Group」のミッション

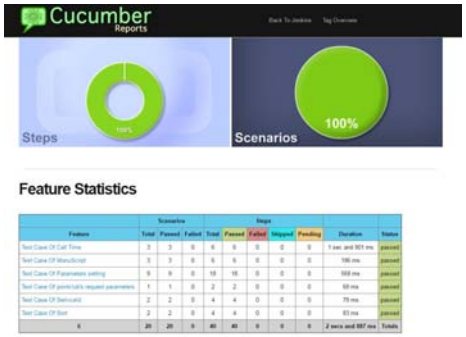
デリバリーとテストのための
自動化プラットフォームの構築

DevOpsの導入・普及

テスト自動化







docker

Azure

テスト自動化の価値

JaSSTの関心

Rakuten

本発表のTake Awayと アウトライン

Take Away:

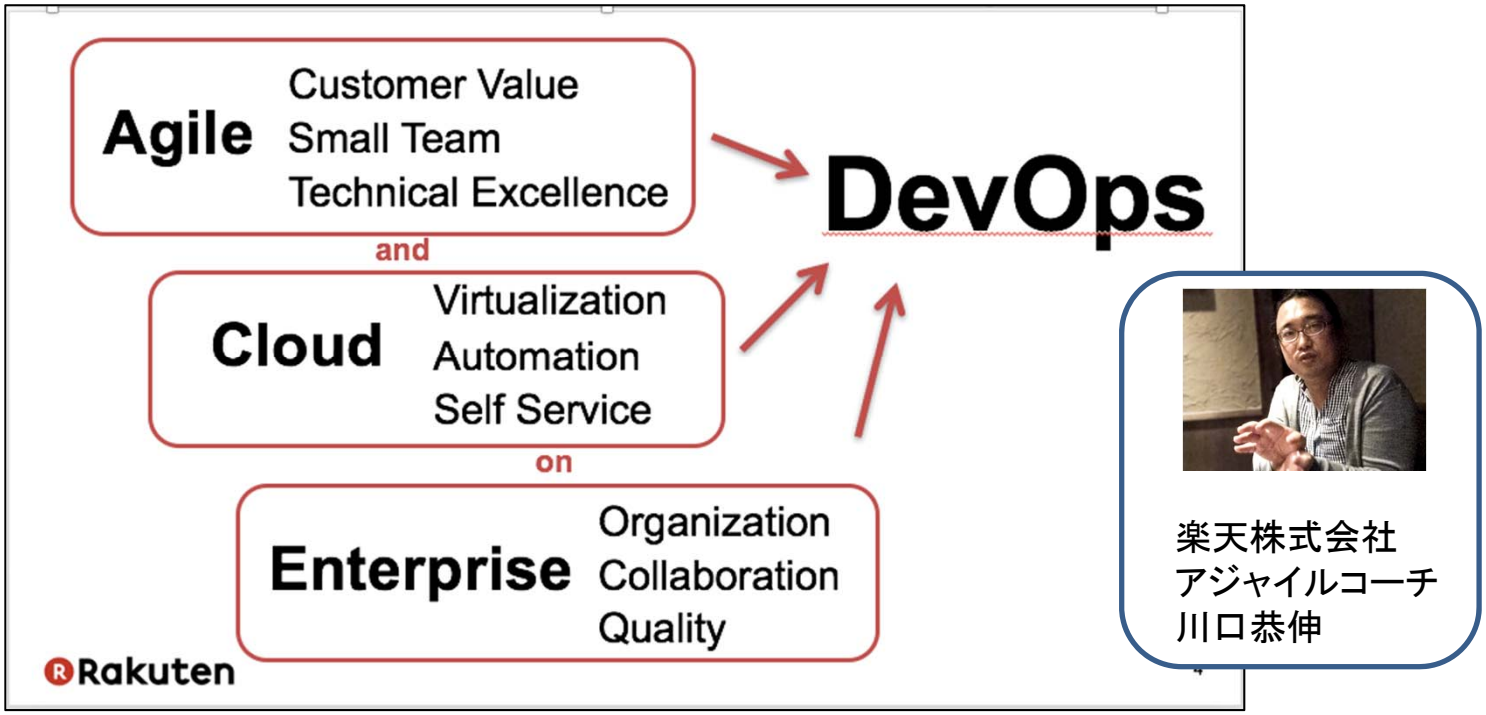
DeliveryとReliabilityというDevOpsの側面から見たテスト自動化の価値とその事例

アウトライン

- DevOpsとは?
- 継続的システムテスト
- 事例①: KPIの見える化
- 事例②: Testability
- 事例③: Coverage

Rakuten

DevOpsの潮流①

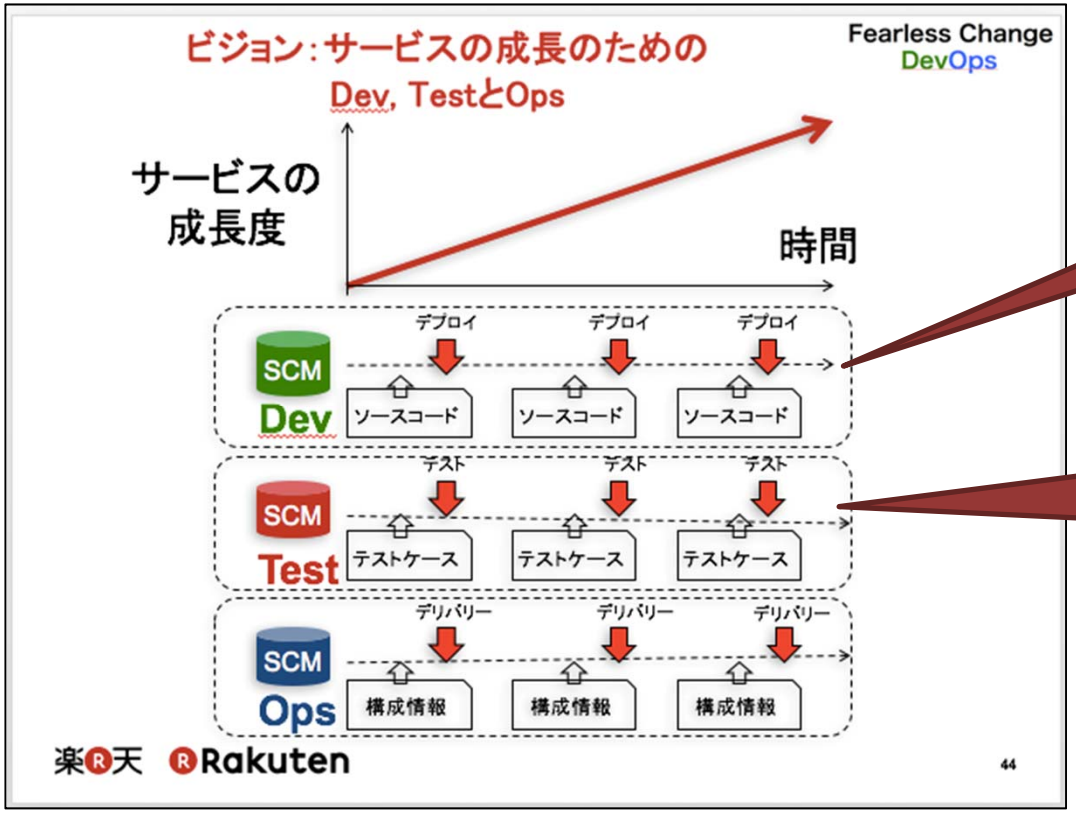


楽天株式会社
アジャイルコーチ
川口恭伸

Rakuten

Agile, Cloud, Enterpriseの流れが、
テスト自動化のみでなく開発・運用スタイルを大きく変えている

DevOpsの潮流②



継続的なサービス成長
= 継続的なコード開発

テストの自動化
= テストのコード化
= 継続的なテスト開発

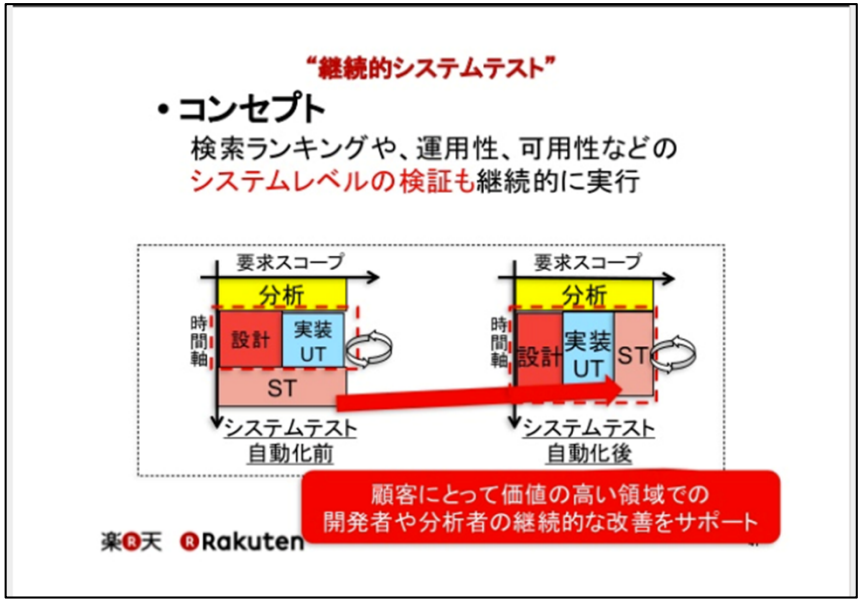
継続的なテスト開発
= 継続的システムテスト

Rakuten

継続的システムテスト

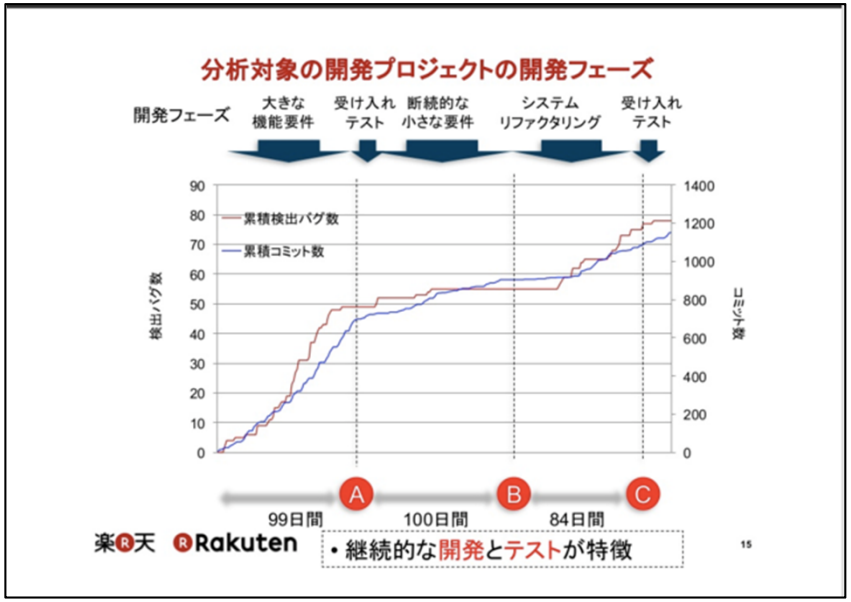
継続的システムテスト:概要

システムテストを開発と並行して実行



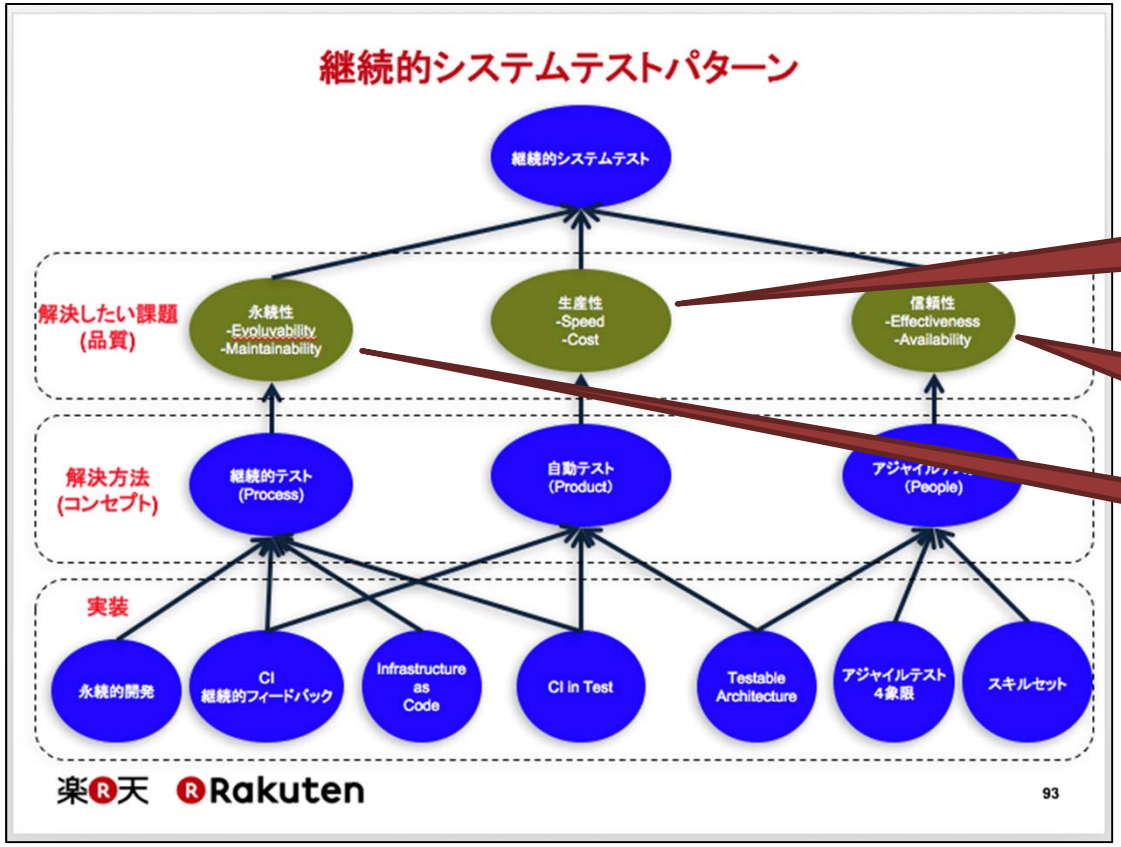
“システムテスト自動化による大規模検索プラットフォームの開発工程改善”, JaSST’Tokyo 2014

累積コミットと検出バグのグラフ



“継続的システムテストについての理解を理解を深めるための開発とバグのメトリクスの分析”,SQiP 2014.

継続的システムテストの目的



DevOpsの世界では
継続的なサービス成長の為

・生産性の高い開発

・信頼性の高いデリバリを

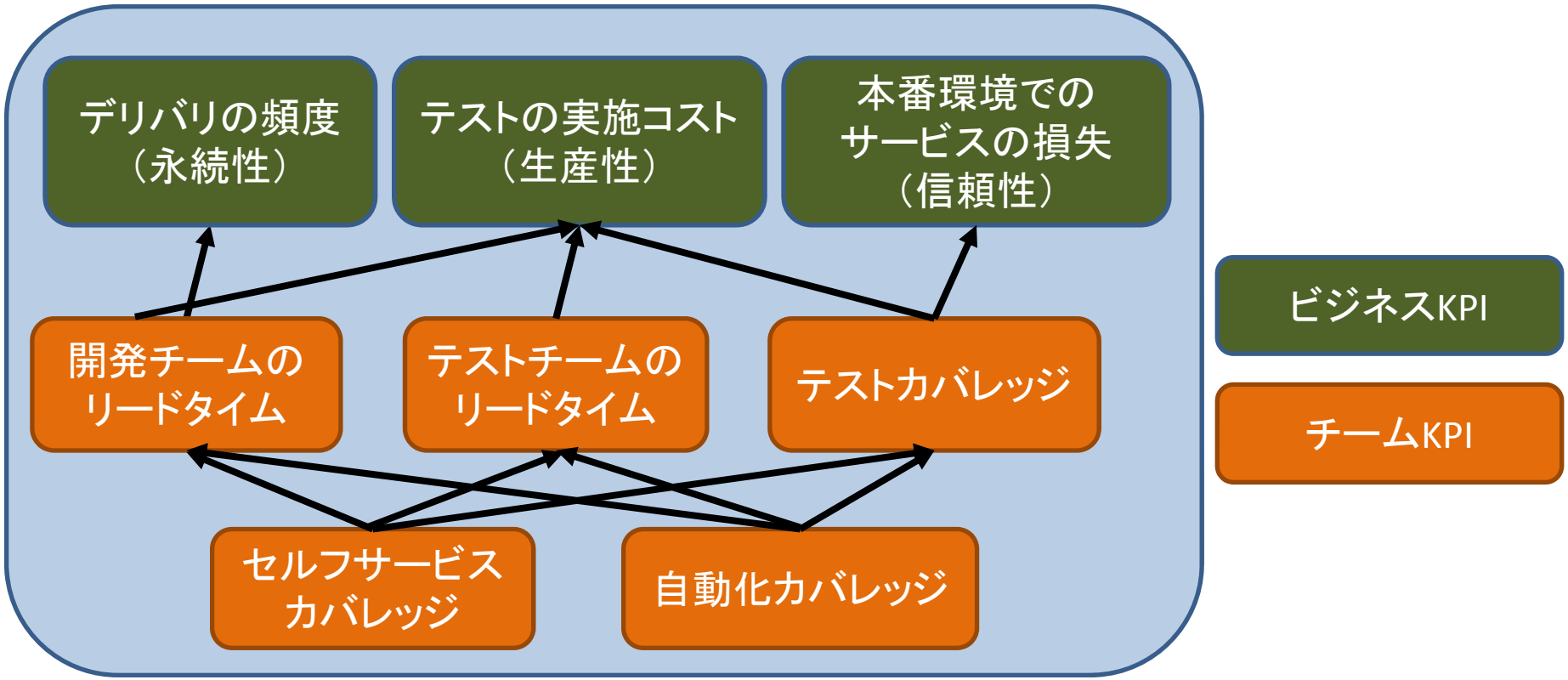
・永続的に

実行していきたい！！！

Rakuten

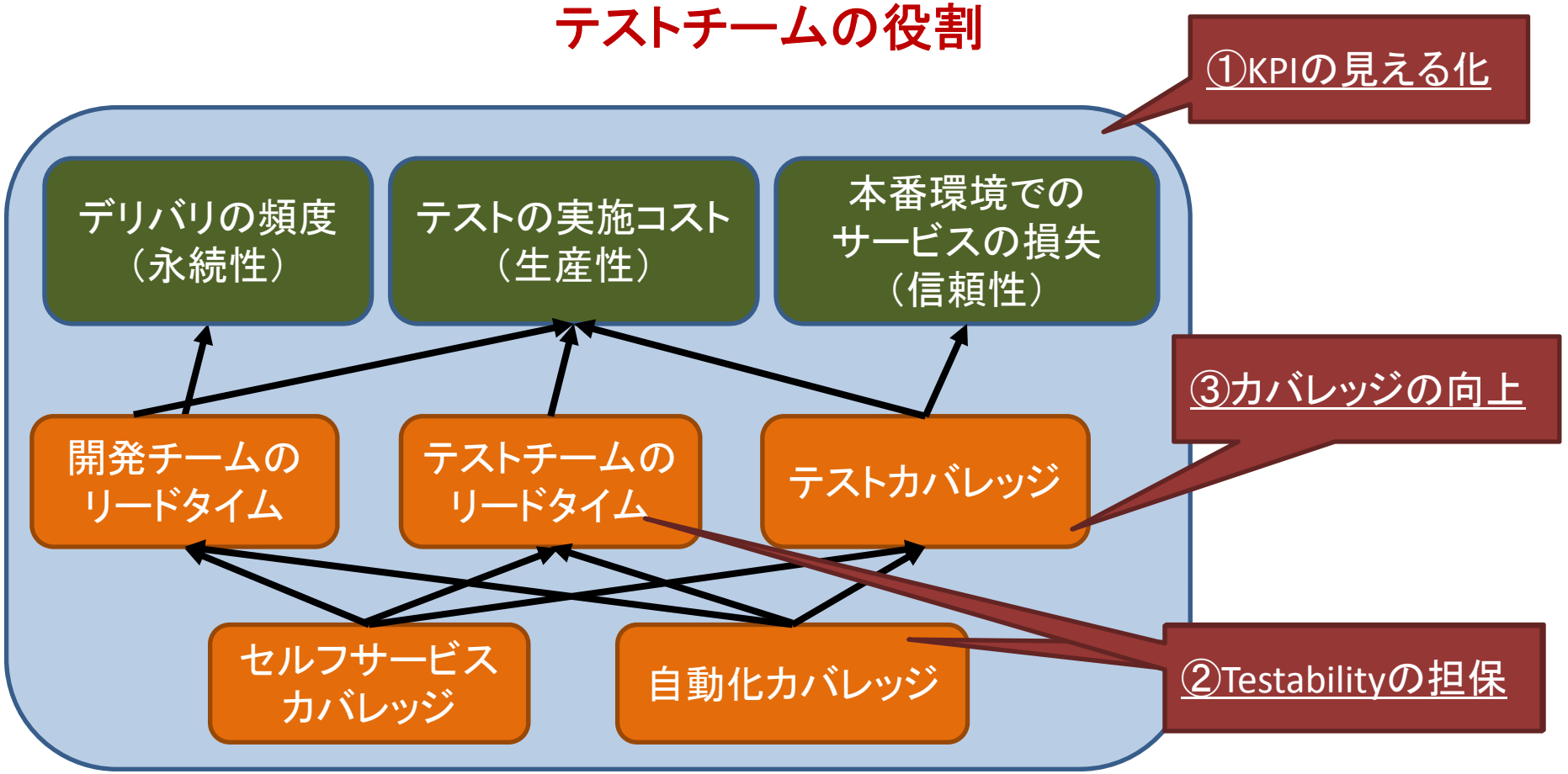
“楽天の品質改善を加速する継続的システムテストパターン”
http://www.slideshare.net/kotaroogino/2015-stac2015

継続的システムテストのKPI（の一部）



Rakuten

テストチームの役割



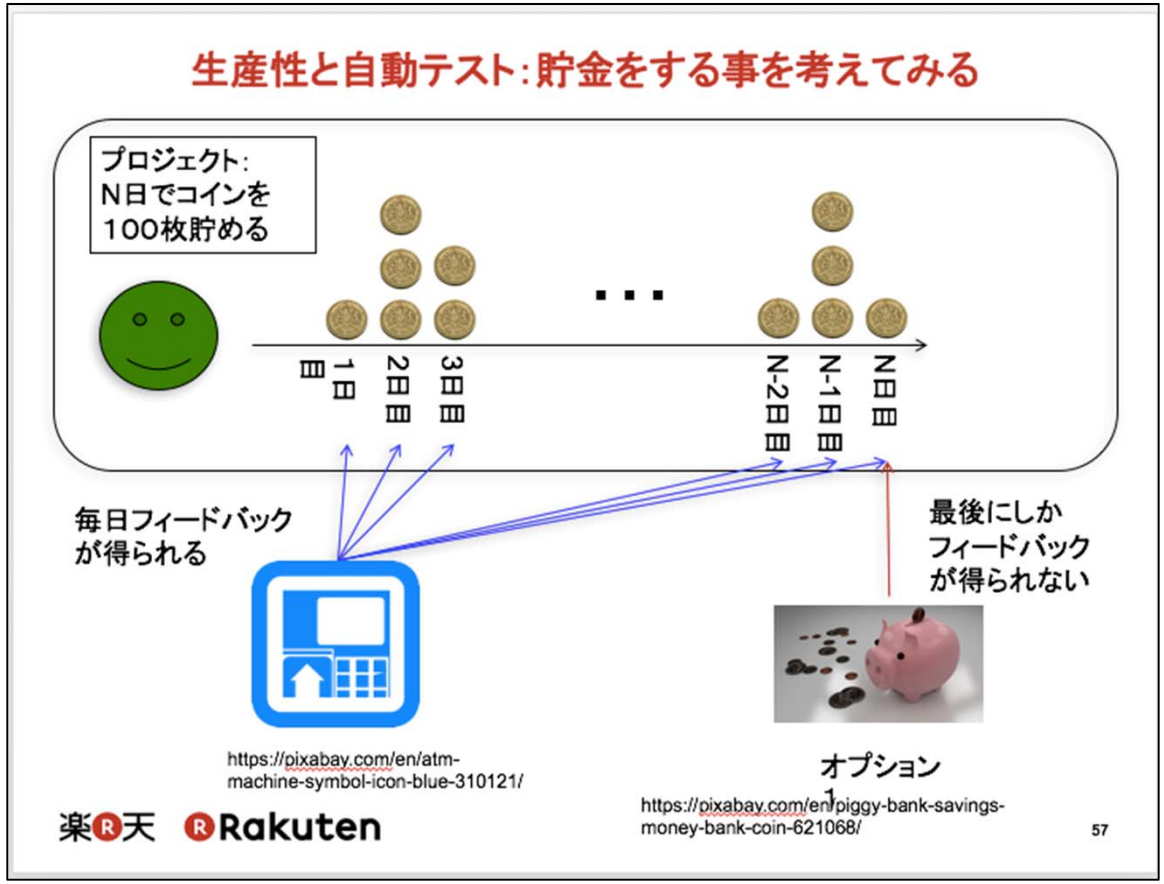
Rakuten

事例の紹介

- KPIの見える化
- Testability
- Coverage

Rakuten

KPIの見える化：背景1/2



KPIの見える化：背景2/2

生産性と自動テスト：貯金をする事を考えてみる

1回こっきりのテストのイメージ

貯金箱を壊して枚数を確認する
→足らなかったらわざわざ貯金箱を
買ってきていれなおす必要w

自動テストにより継続的にテスト実行するイメージ

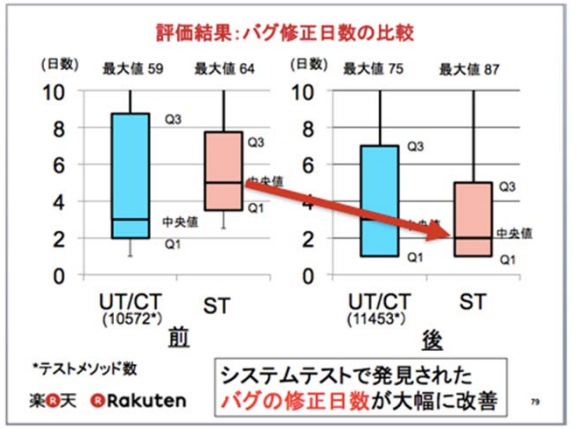
コインを入れたら、
現在の貯金額を教えてくれる
→ 今日足らなさそうだったら、
ちょっと多めに入れてみる

毎日品質についてフィードバックするツールの一つ

楽天 Rakuten 58

自動テストが生産性を向上する例

バグを作ってからすぐに修正可能なので
バグの修正日数が早くなる



楽天 Rakuten "システムテスト自動化による大規模検索
プラットフォームの開発工程改善", JaSST'Tokyo 2014 59

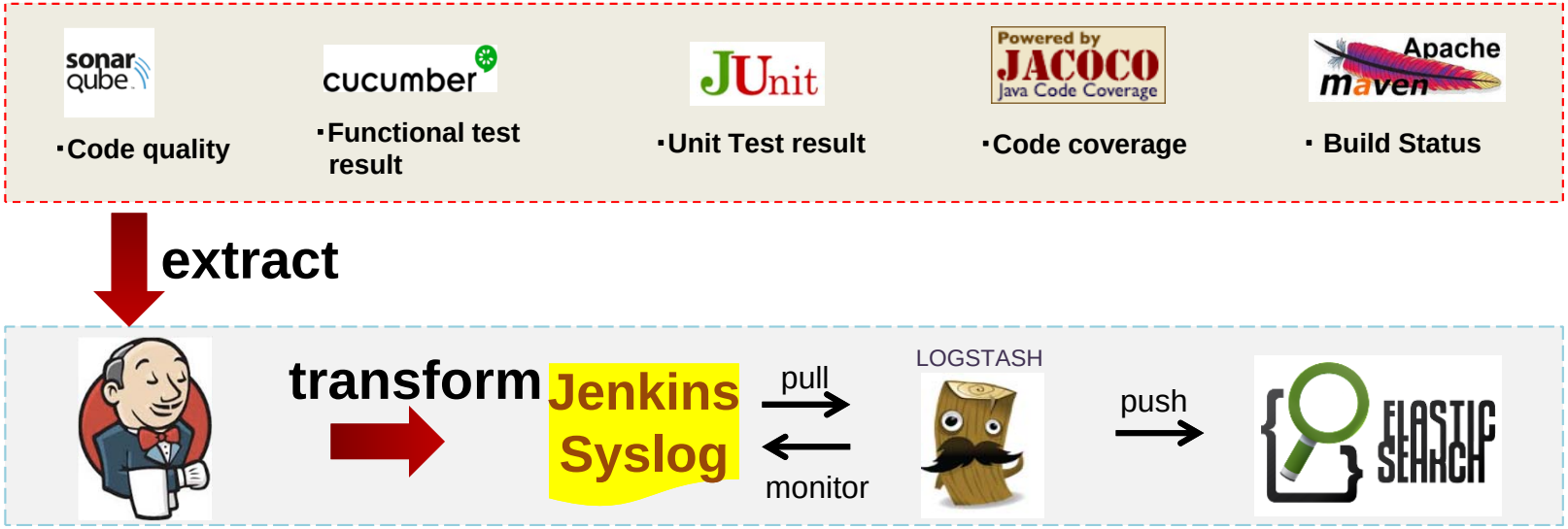
迅速なKPIのフィードバックが、
自律的な改善活動を促進する！！

Rakuten

KPIの見える化：Reportの自動化

Rakuten

KPIの見える化：アーキテクチャ



-Jenkinsを使ってすべて自動化
-見たい情報がすぐ手に入る

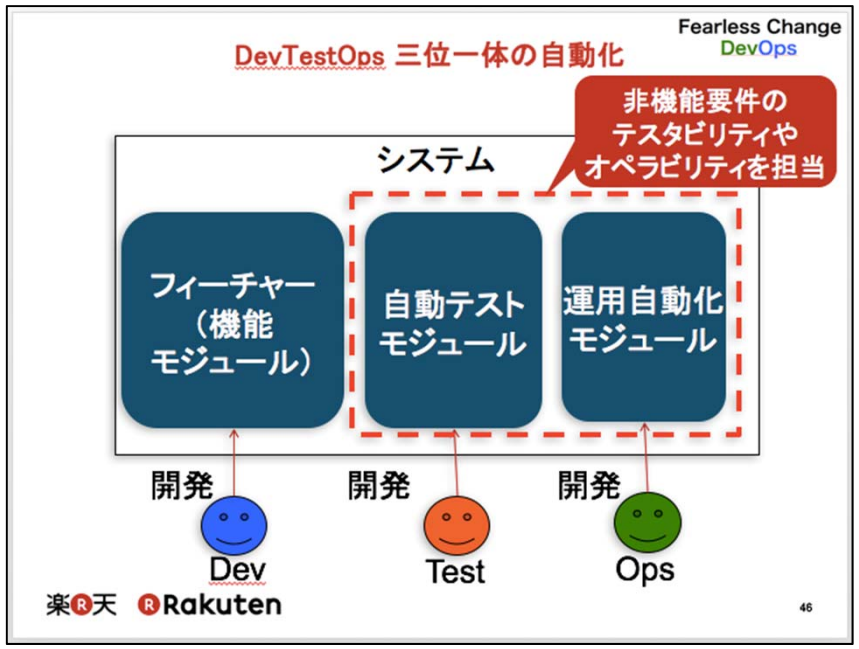
事例の紹介

- KPIの見える化
- Testability
- Coverage

Rakuten

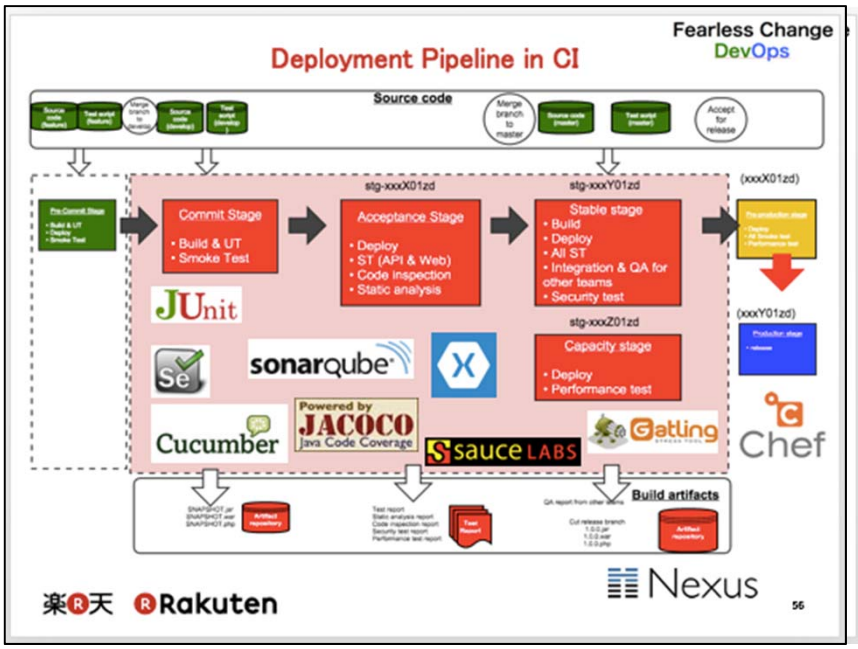
Testability

テスト対象がテスト容易な設計か？



設計へのフィードバック

テストチームはテスト対象のテストを容易に自動化可能か？

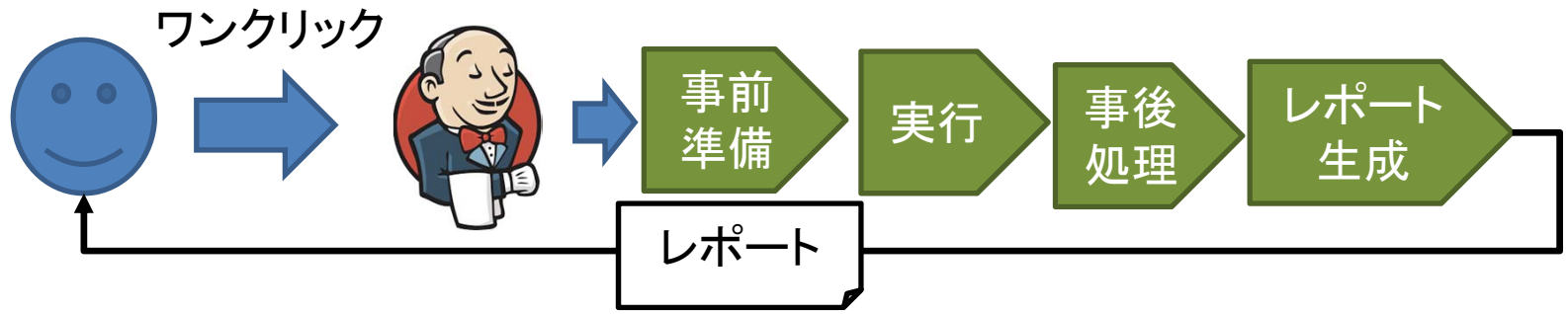


ツールは? CIに組み込めるか?
準備期間は? メンテナンスコストは？

Rakuten

Testability: テストチームは容易に自動化可能か?

キーワード:ワンクリックですべて実行可能か?



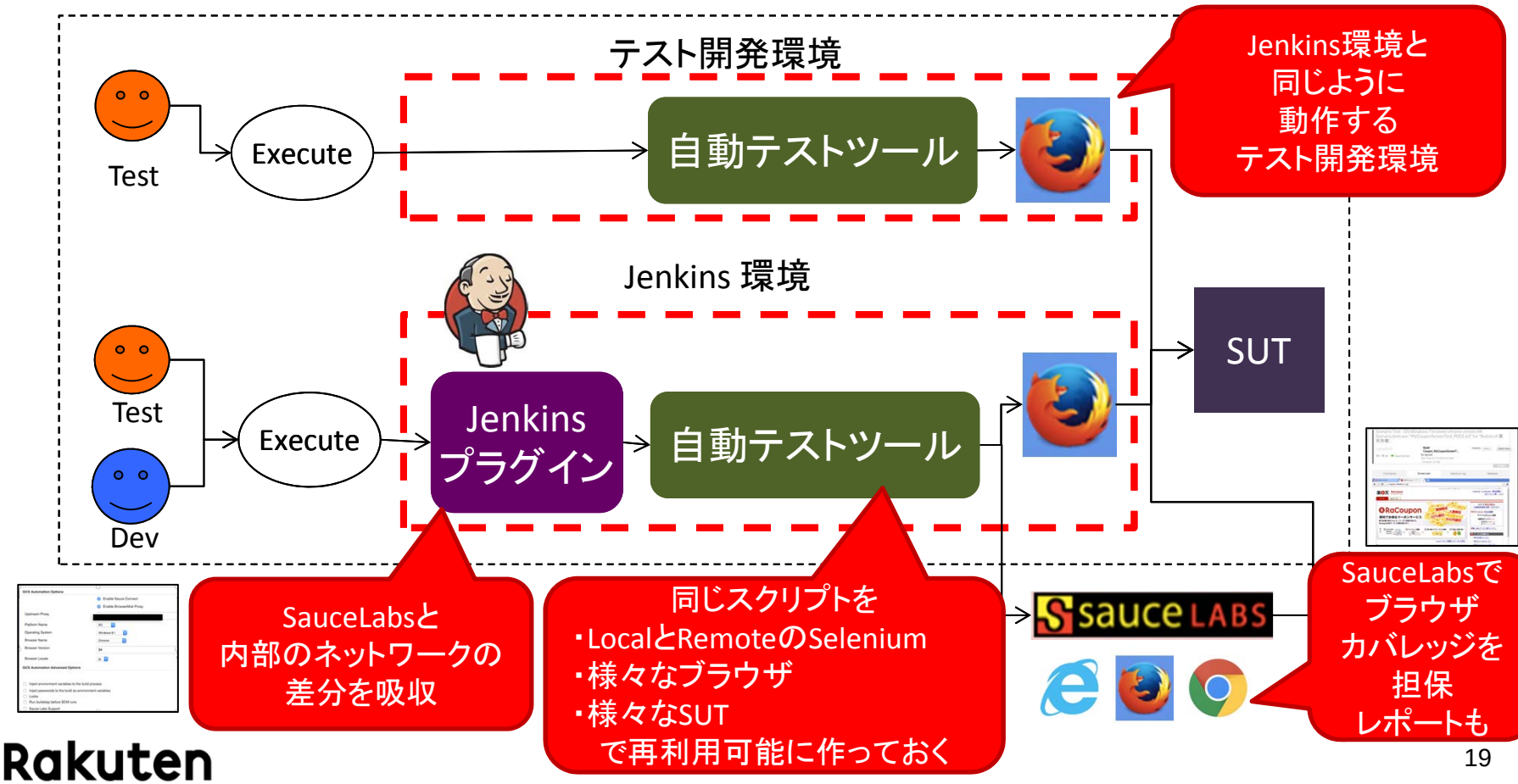
ワンクリックでテストできるように

- -機能テスト
- -パフォーマンステスト
- -セキュリティテスト
- -マルチブラウザテスト
- …

ありがちな自動化の落とし穴

- -テストの開発環境がないので、デバッグが大変
- -事前準備での環境差異の吸収が手動
- -SUTのホスト名やテスト対象のブラウザなどを外から与えられない
- -レポートが自動で生成されない

Testability:例）マルチブラウザテストのアーキテクチャ



事例の紹介

- KPIの見える化
- Testability
- Coverage

Coverage：コードカバレッジの使い方

- アジャイル開発での有効なテスト品質指標は?
  - -バグ密度?
  - -テスト密度?
  - -Permutation Score?
- コードカバレッジだけでの評価は難しい
  - -ユースケースに対するカバレッジ?
  - -新しいバグを発見する能力?
  - -バグの本番環境でのインパクト
- 上記の問題点を考慮した上でコードカバレッジを使う
  - - テスト観点との追跡性
  - - 他のメトリクスとの組み合わせ

まとめ

- DevOpsな開発プロジェクトでは、
  「自動テストモジュール」をシステム内に開発
- 継続的システムテストのフレームワークは
  それをサポート
- 以下の事例
  - KPIレポートの自動化
  - Testability
  - Coverage

- http://corp.rakuten.co.jp/careers/engineering/