

会社紹介

株式会社Udzuki



Table of Contents

- 会社概要
- 研究開発体制
- Mission-Vision-Value
- ビジネスモデル
- 事例紹介（受託開発）
 - Webアプリ開発・クラウド導入
 - 機械学習システム構築
 - テスト自動化
 - テスト設計支援
 - LA基盤構築
- プロダクト
 - RevAjaxMutator：Webアプリの自動プログラム修正
- 貴社との連携（協業）

会社概要

項目	説明
会社名	株式会社Udzuki
所在地	東京都千代田区九段南1丁目5番6号りそな九段ビル5F・KSフロア
代表者	前澤 悠太 Ph.D. (情報理工学)(東京大学)
設立日	2018年4月
事業内容	ソフトウェア工学分野の学術研究をベースに ✓ テスト自動化・自動プログラム修正に関する研究開発 ✓ Webアプリケーション開発・クラウド導入（インフラ～UI/UX） ✓ 統計解析・機械学習プログラム実装 ✓ 技術指導・講演
特徴	✓ 世界最先端のソフトウェアテスト・デバッグ技術を保有 ✓ 情報理工学（コンピュータ科学）に関するハイレベルな知識 ✓ 0/1の着想から実装・テスト・公開までSW開発全行程を網羅

社名の由来：SW開発者が創造的な活動に注力できる世界の実現に向けて自動プログラム修正（Automated Program Repair）を最善策と見定めその頭文字APR→Aprilの略→四月（旧名：卯月）→Udzukiと連想し命名

研究開発体制

情報理工学に関するハイレベルな学術研究・ソフトウェア開発技術



代表 前澤悠太
博士 (情報理工学)(東京大学)

フルタイム・パートタイム
エンジニア数名

e.g., WillBooster社代表・坂本一憲



研究

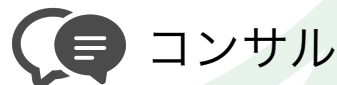


開発

相談役 岡村俊二



営業



コンサル

相談役 末永俊一郎

経験豊富なバックアップ陣

Mission-Vision-Value

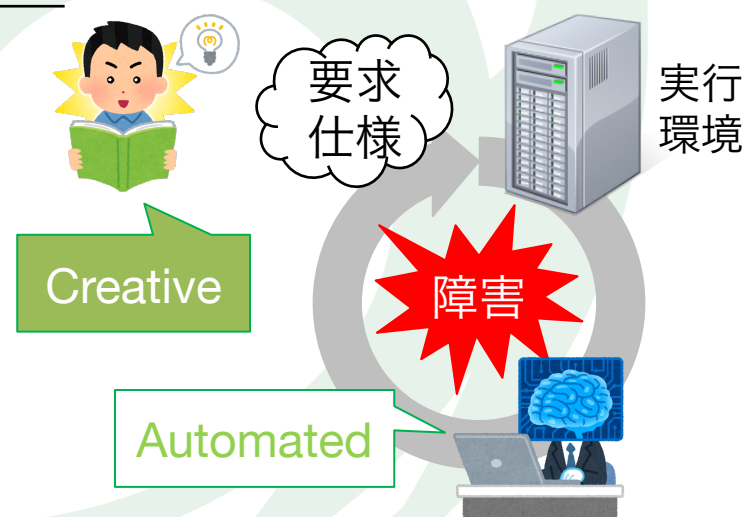
ソフトウェアテスト・デバッグの**自動化**により
開発者が**創造的**な仕事に集中できる社会の実現を目指す

- 「事故前提社会」への対応力強化†
 - ソフトウェア障害発生を前提 → 迅速かつ適切な保守が重要

Before



After



芸術的活動としてのソフトウェア開発

ビジネスモデル

学術研究の成果としてプロダクト開発→最先端技術を世界に適用
学術研究を通して獲得したドメイン知識→世界の既存課題を解決



事例紹介 - 受託開発

事例紹介 - Webアプリ開発・クラウド導入

企画・ディレクション～実装・テスト（全開発行程を網羅）



フルスクラッチ開発

デモサイト構築

SSL・レスポンス性等
モダンな設計にも対応

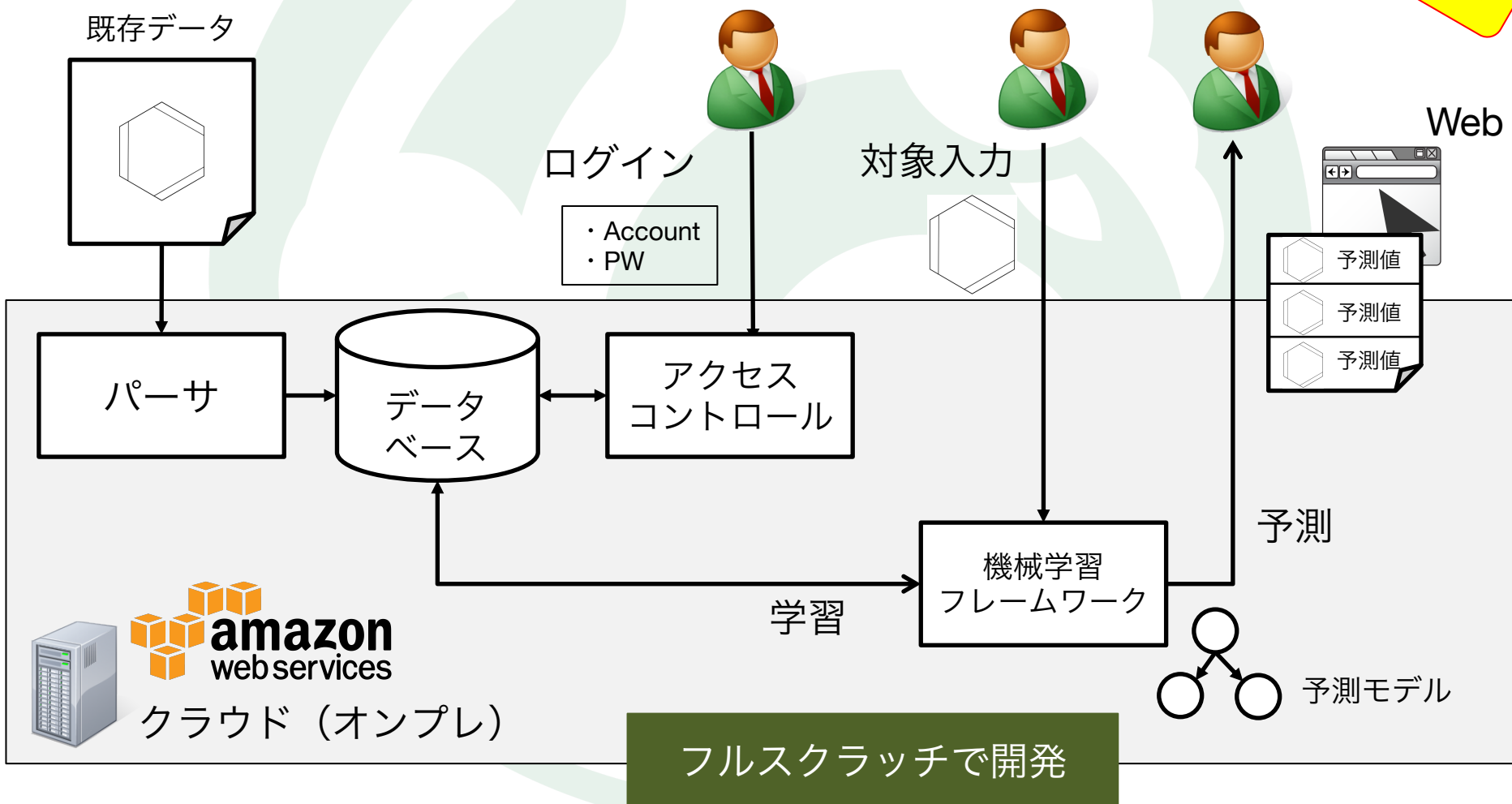
ご相談から3週間で納品



事例紹介 - 機械学習システム構築

数学的背景知識の解釈・適用可能技術の選定

Sample



事例紹介 - テスト自動化

テスト要件分析・設計・実装・CI/CD

(ハイレベルな専門的知識[†]に基づき実施)

回帰テストで
工数削減

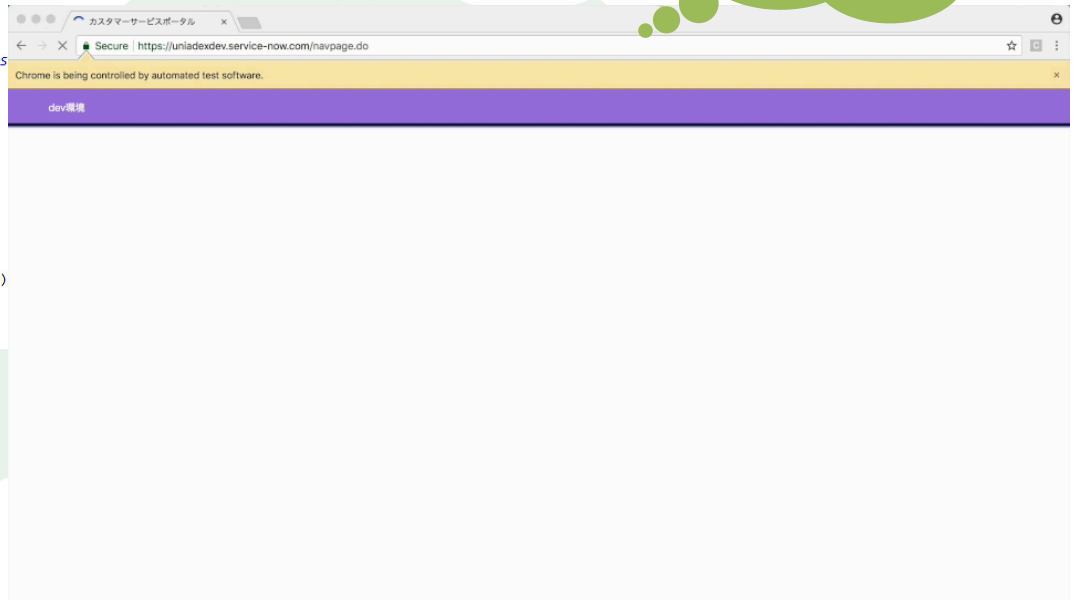
```
GoogleSearch.java
//private boolean acceptNextAlert = true;
private StringBuffer verificationErrors = new StringBuffer();

@Before
public void setUp() throws Exception {
    driver = new FirefoxDriver();
    baseUrl = "https://www.google.com/";
    driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
}

@Test
public void testGoogleSearch() throws Exception {
    driver.get(baseUrl + "/");
    driver.findElement(By.id("gbqfq")).clear();
    driver.findElement(By.id("gbqfq")).sendKeys("Google");
    driver.findElement(By.id("gbqfb")).click();
}

@After
public void tearDown() throws Exception {
    driver.quit();
    String verificationErrorString = verificationErrors.toString();
    if (!"".equals(verificationErrorString)) {
        fail(verificationErrorString);
    }
}
```

JUnit
annotations
used in
Selenium
WebDriver
Test code



JUnit



Jenkins



Travis CI

自動化テストの実装・保守を担当

servicenow

[†] Udzuki代表・前澤はソフトウェアテストに関して博士号(情報理工学)(東京大学)を取得

事例紹介 – テスト設計支援

- 大規模な既存システム（国際基準）をカスタマイズして実装

研究データ管理システム

Sample

- 要求される機能はドメイン知識により把握可能

- ソフトウェアテストに関する専門知識の不足

- 外部に説明できるテスト設計
- 効果的・効率的なテスト設計

- テストコンサルを依頼

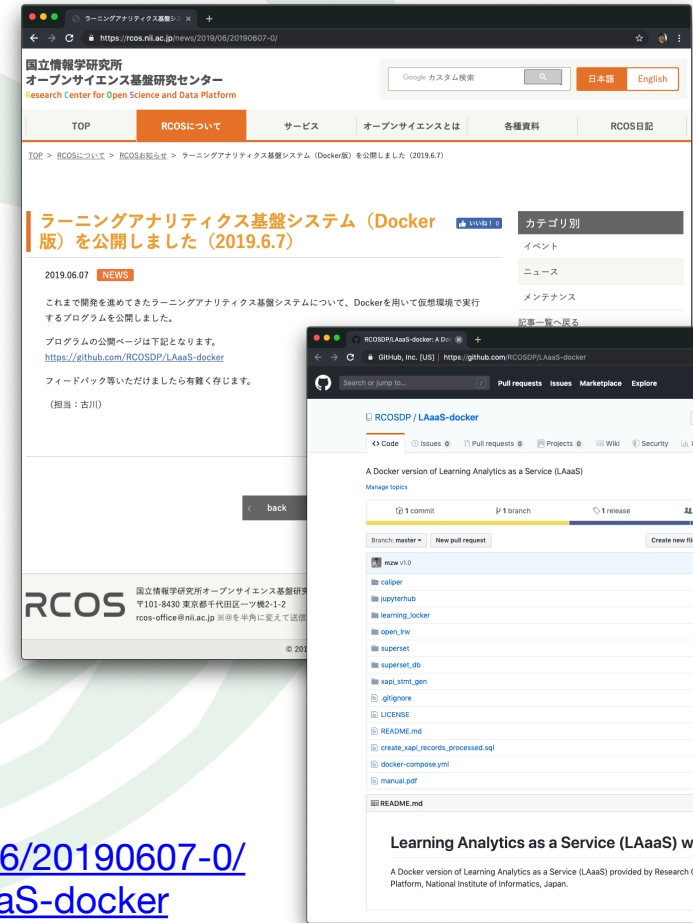
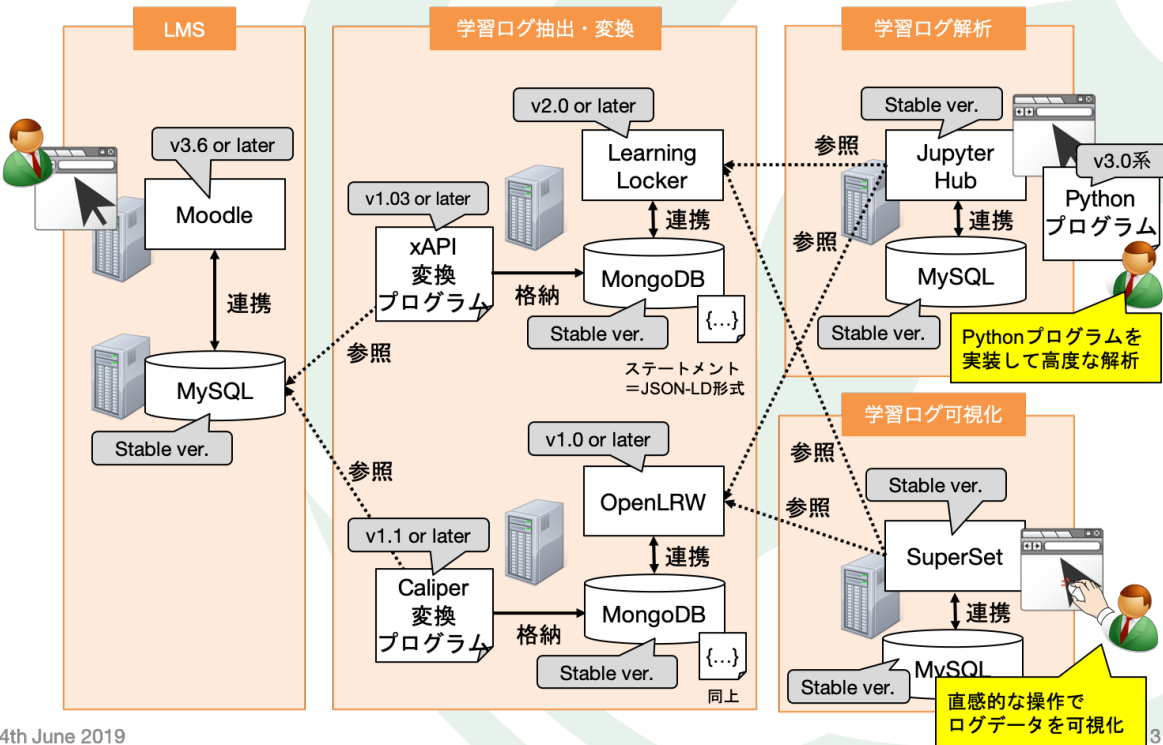
- 既存システム・カスタマイズ部分を精査
- 基礎的なソフトウェアテスト技術の適用
 - テスト手法選択（状態遷移テスト）やカバレッジ選択（C0/C1/C2）etc.



各層10個弱のコンポーネントで構成

事例紹介 - LA基盤構築

- 多様な学習管理システム (LMS) で生じる学習データを標準化・一元管理・分析する基盤



4th June 2019

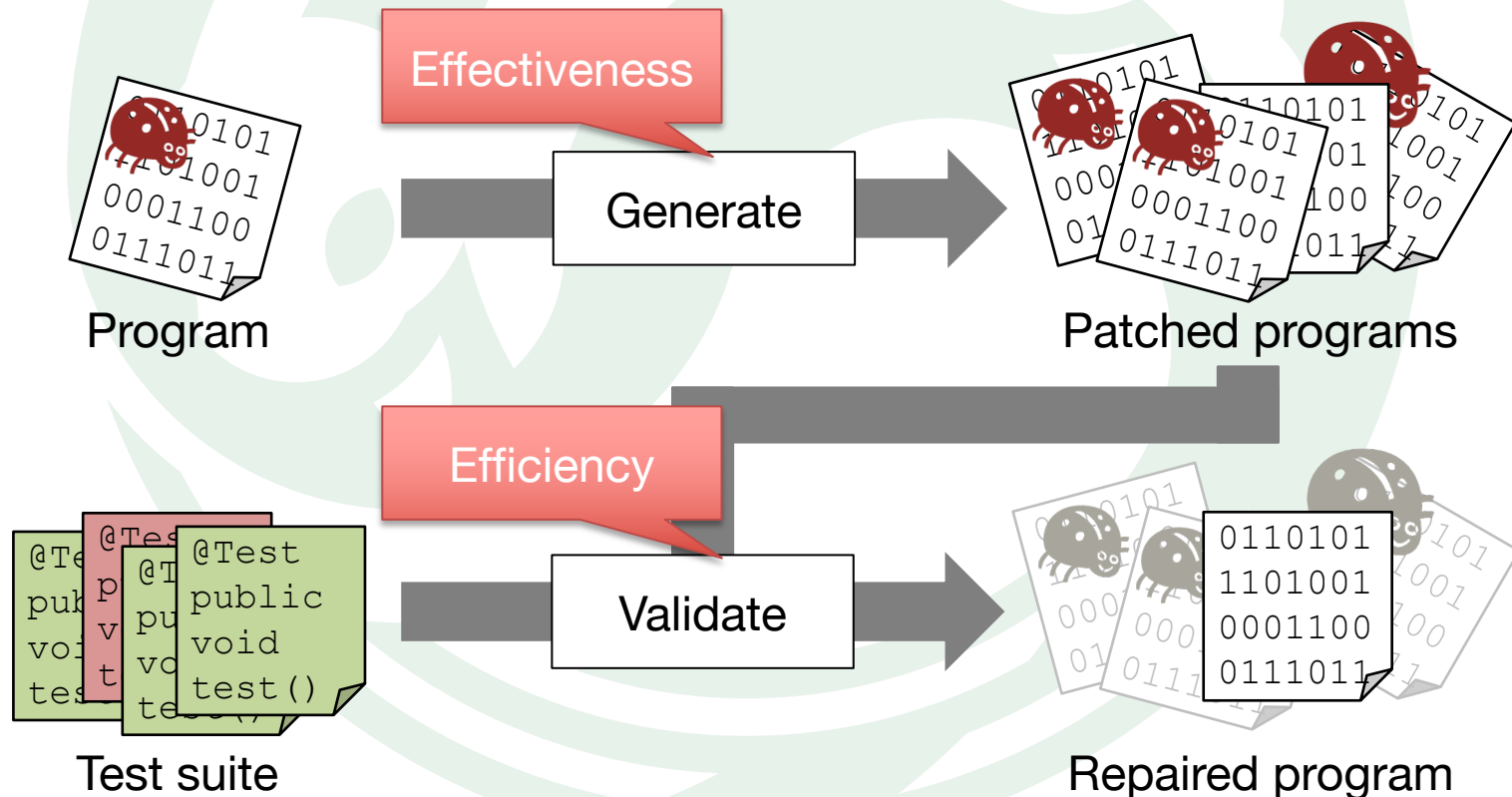
国立情報学研究所でDocker版を公開
 cf. <https://rcos.nii.ac.jp/news/2019/06/20190607-0/>
 cf. <https://github.com/RCOSDP/LAaaS-docker>

プロダクト

自動プログラム修正

- Generate-and-Validate Technique (生成・検証手法)

Given a program and its test suite with at least one failing test case, create a patch that makes the whole test suite passing. †



† M. Monperrus, "Automatic Software Repair: a Bibliography", ACM Computing Survey, 2017

プロダクト : RevAjaxMutator

<https://www.youtube.com/watch?v=prpYW9t-des>

学術研究・プロダクト開発の背景

- 研究資金（抜粋）
 - 前澤PI：JSPS科研費（若手B）
 - 前澤PI：JSPS科研費（研スタ）

研究体制

連携大学院生

- 東京大学
大学院情報理工学系研究科

研究生・インターン

- 株式会社日立製作所
- University College London（英国）
- INESC-ID, Lisbon（ポルトガル）
- University of Kent（英国）
- Paul Sabatier University（フランス）

Ajaxアプリケーションのテスト自動化・支援ツールの業務適用に関する評価
日立水戸エンジニアリング株式会社 星敬一郎 keiichiro.hoshi.ej@hitachi.com

MVC Webアプリケーションの機能縮退を考慮した振舞い検証
(株)日立製作所 インフラシステム社 澤野 宏貴 hiroki.sawano.dg@hitachi.com

Webアプリケーションにおけるミューテーション解析の効率化
(株)日立製作所 伊藤隼平 shumpei.ito.fn@hitachi.com

開発における問題点
Webアプリケーションに適用可能なミューテーション解析ツールであるAjaxMutatorは、Webアプリケーションのテストを高品質化することが可能である。しかし、このツールは、テスト実行処理がボトルネックとなり、解析時間が長く、特に大規模なWebアプリケーションに適用した場合、実用的でない恐れがある。

手法・ツールの適用による解決
ミューテーション解析処理で大きなボトルネックとなるテスト実行処理を以下3つの手法の適用により効率化する。
①テスト実行の並列化
②カバレッジ情報を用いたテスト実行の効率化
③同ステートメントのミュータント数の削減

Webアプリのミューテーション解析ツール (AjaxMutator)

ボトルネック
AjaxMutator
Webアプリケーション
テスト実行
評価実験

効果
Webアプリのテスト高品質化を支援
テスト実行処理の時間が長い

課題
①イベント要素に対するミュータントのイベントネブに對するミュータント
②コールバック量に対するミュータント
どれも発生する確率は同じのため
検査対象を3種類→1種類とする
検査するミュータントの数を削減

テスト実行処理効率化のための適用手法

1. 抽出 入力 MVCの各コード
2. 制限 入力 MVCの各コード
3. 検証 入力 MVCの各コード

⇒ 並列化により複数同時テスト ⇒ カバレッジ情報を用いて、テスト実行の最適化 ⇒ 検査するミュータントの数を削減

評価実験
実験内容
・手法適用時の実行時間測定
対象
・実在する11個のWebアプリ

評価結果
平均で50.9%テスト実行時間が削減
最大で約17時間半→約4時間に削減

NII 国立情報学研究所 National Institute of Informatics
トップエスイー
～サイエンスによる知的ものづくり教育プログラム～

貴社との連携（協業）

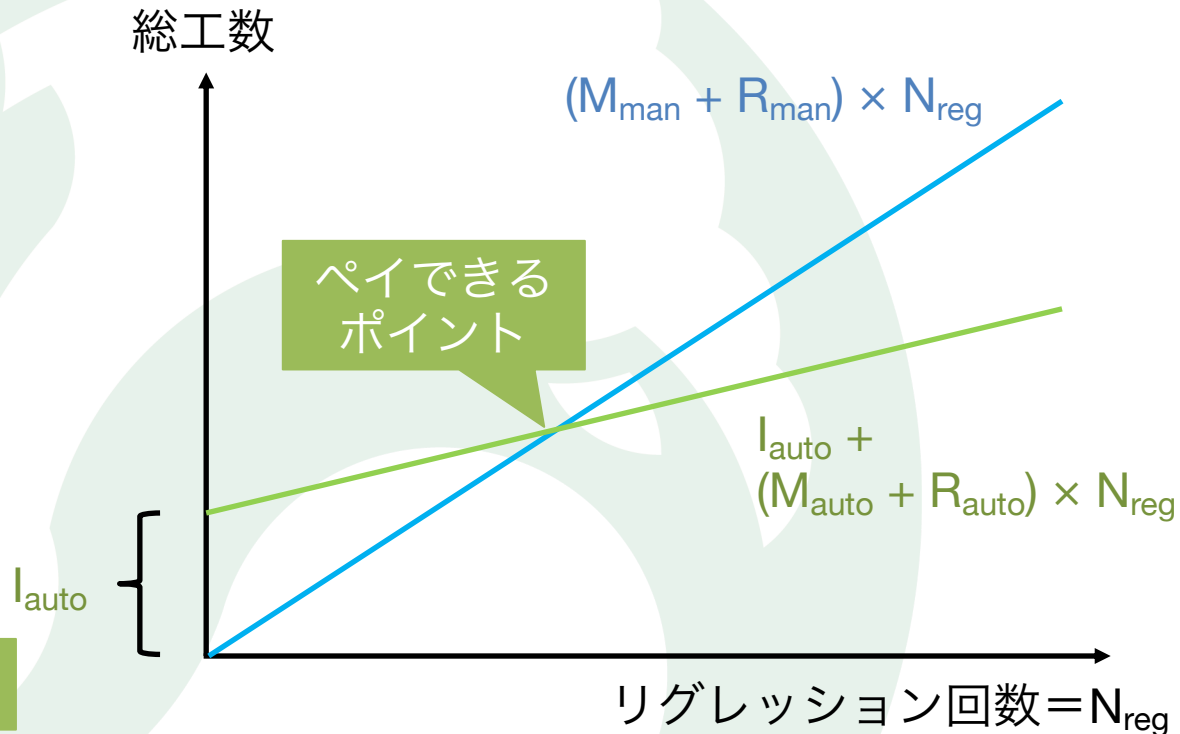
連携（協業）の可能性について議論させていただき
関連部署をご紹介いただければ幸いです

- 貴社のメリット
 - 既存技術では解決できない課題に対して
専門的・先端的な視点から解決策を模索できる
(※ 社内・お客様側にご提案可能)
- 弊社のメリット
 - 安定的 and/or 保有技術を活用できる
商流を獲得できる

テスト自動化のメリット（概算） 1/3

- 手動テスト
 - 実行： R_{man}
 - 保守： M_{man}

- 自動テスト
 - 初期実装： I_{auto}
 - 実行： R_{auto}
 - 保守： M_{auto}



$I_{\text{auto}} < R_{\text{man}} \times N_{\text{reg}}$ が見込める場合は自動化を検討

例えば、

- ・自動テストの初期実装 (I_{auto}) に「4人月（4人チームで1月）」を要する見込み
- ・現在、1回の手動テスト (R_{man}) に「0.5人月（2人週・2人で1週間）」を要している
- ・リグレッション (N_{reg}) を8回以上「（4半期アップデートを2年継続）」する計画の場合
テスト自動化を検討するとよい

テスト自動化のメリット（概算） 2/3

- ペイできるポイントの計算
 - 総工数（手動テスト）
= 総工数（自動テスト）

$$(M_{man} + R_{man}) \times N_{reg} = I_{auto} + (M_{auto} + R_{auto}) \times N_{reg}$$

$$(M_{man} + R_{man}) = \frac{I_{auto}}{N_{reg}} + (M_{auto} + R_{auto})$$

$$(M_{man} + R_{man}) - (M_{auto} + R_{auto}) = \frac{I_{auto}}{N_{reg}}$$

$$(M_{man} - M_{auto}) + (R_{man} - R_{auto}) = \frac{I_{auto}}{N_{reg}}$$

- ここで、下記2点を仮定
 - 手動も自動も保守（データ更新等）コストは同程度
 - 手動に比べて自動は実行コストが十分小さい

$$M_{man} \approx M_{auto}$$

$$M_{man} - M_{auto} \approx 0$$

$$R_{man} \gg R_{auto}$$

$$R_{man} - R_{auto} \approx R_{man}$$

- すると、

$$(M_{man} - M_{auto}) + (R_{man} - R_{auto}) = \frac{I_{auto}}{N_{reg}}$$

$$0 + R_{man} \approx \frac{I_{auto}}{N_{reg}}$$

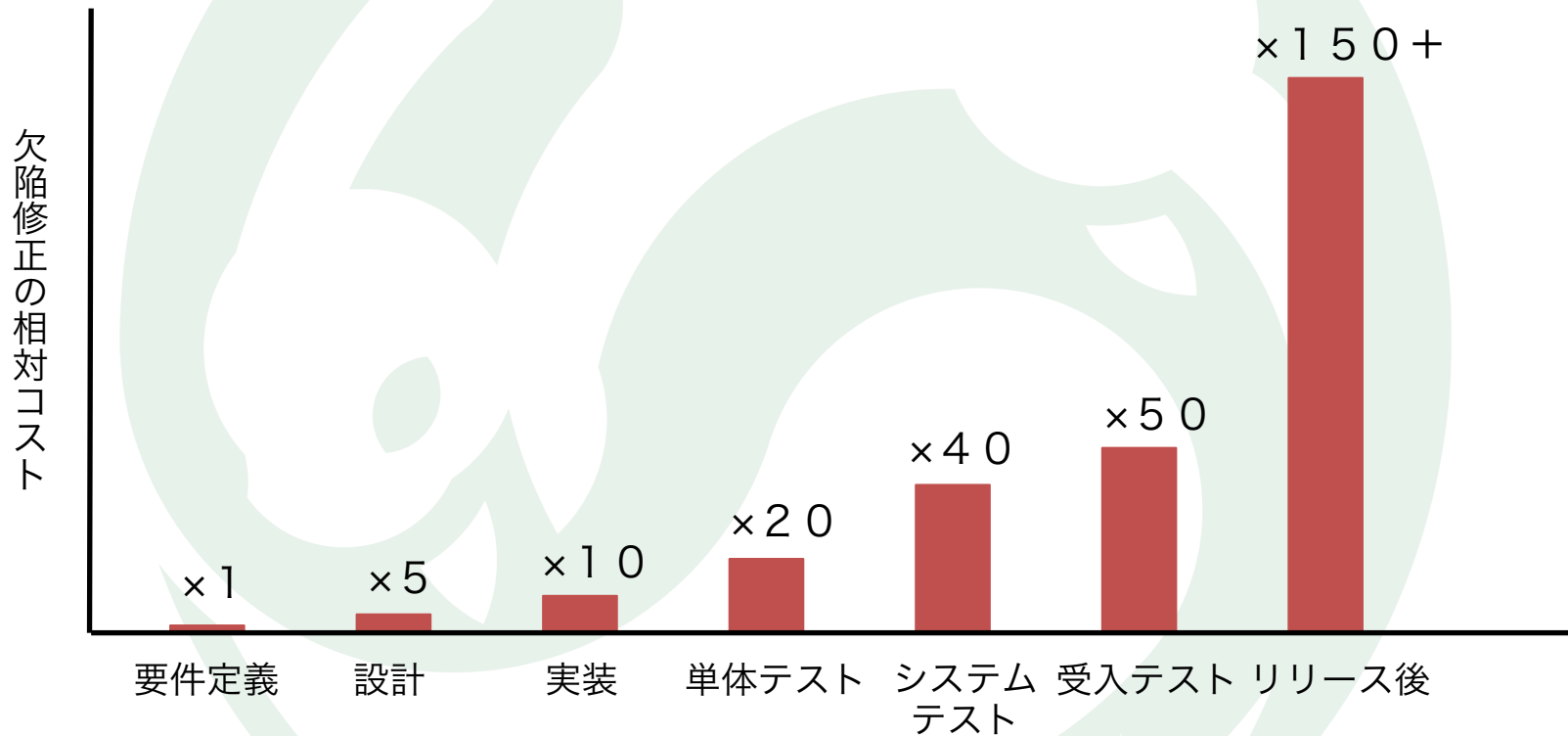
$$I_{auto} \approx R_{man} \times N_{reg}$$

属人性の排除・早期発見による欠陥修正コストの削減等も見込める

反面、テスト技術の習得・人によるエラー発見も要考慮

テスト自動化のメリット（概算） 3/3

- テストに早く着手せよ



障害発見が遅れるほど欠陥修正コストがかかる