

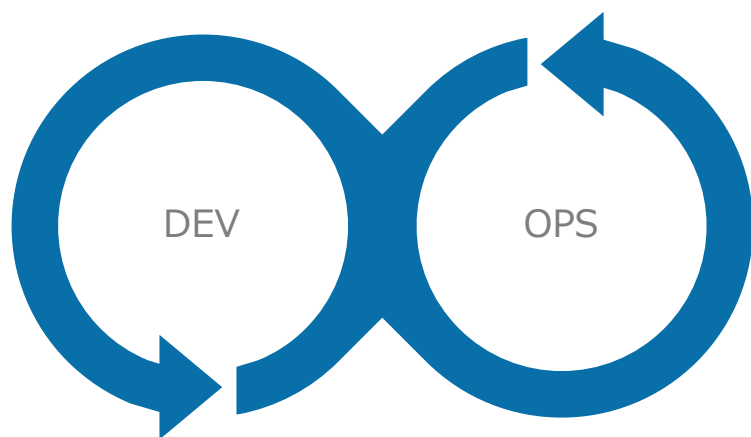
テスト自動化基盤をどうつくるか

TechMatrix Jenkins Platform Package for Java を活用した自動化基盤の構築



DevOps, CI/CD

DevOps



DevOps

DevOps とは、「開発だけでなく、リリース・フィードバックなどを含めたサイクルを早め、製品の価値とチームの力を最大限に高めるための継続的・組織的なカイゼン活動」のことです。システム等の開発や導入のために、開発部門と運用部門が協力し、連携するための手法であり、これまでの開発部門と運用部門の対立した構造ではなく、ともにビジネスの価値を高め、その価値をエンドユーザーに迅速に届けることを目的としています。

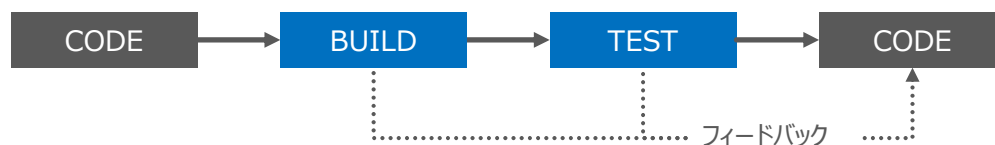
DevOps は、Agile 開発手法に運用部門の観点を追加したものです。そのため、DevOps は Agile 自体やAgile のプラクティスから大きな影響を受けています。DevOps を実現する上で必須となるのが、継続的インテグレーションと継続的デリバリーです。

CI/CD

継続的インテグレーション

継続的にビルドやテストを行う、ソフトウェア開発のプラクティスです。

- コードの変更に対して、自動でビルドや検証(単体テストなど)を行い、フィードバックを行います。
- 影響範囲の小さいうちに問題を修正することで、ソフトウェア開発のリードタイムの短縮を行います。



継続的デリバリー

ビルドやテストだけではなくリリースまでを行う、ソフトウェア開発のプラクティスです。

- より複雑なテスト(UI テスト、API テストなど)を実施し、リリースプロセス全体を自動化します。
- コードの変更からソフトウェアのリリースまでを自動で行うため、より早く新しいソフトウェアを提供できます。



なぜCI/CD？

- 市場をリードする企業のIT組織は、CI/CD, DevOpsなどの一般的な手法を積極的に取り入れ、**ソフトウェアをいかに早く市場に投入するか**に注力しています。
- CI/CD、DevOpsなどの手法を採用することで、従来の開発に比べて圧倒的に早くソフトウェアをリリースできるため、企業の経営層および開発現場の両方に大きなメリットがあります。

競争優位性

顧客のニーズをソフトウェアへすぐに反映
優秀なエンジニアの確保
など

開発効率化

小さい単位でのリリースによるリスク軽減
開発現場の無駄を可視化し改善
など

海外のCI/CDの取り組み例

BOSCH

目的

CI および CD プラクティスの導入により、ますます複雑化する自動車ソフトウェアの開発の簡略化と管理を行い、市場への投入までの期間を短縮する。

成果

自動化とプロジェクトの状態の可視化を促進し、同時にビルドの安定性とソフトウェア品質を改善。結果として、**3日のビルドプロセスを3時間未満に短縮**。

Capital One

目的

最高の品質およびセキュリティ基準を保ったまま、ビジネスアプリケーションのデリバリーを加速する。

成果

安定性の高いスケーラブルなCIプラットフォームを提供し、繰り返し可能なビルドプロセスを自動化するとともに、コードのコミットからデプロイに至るCDパイプラインを管理。結果として、**パイプラインの90%を自動化、デプロイメントの頻度が13倍に増加**。

Netflix

目的

動画ストリーミングおよび大規模なクラウド技術のオープンソース開発への貢献をサポートおよび促進する。

成果

社内のオープンソースプロジェクトをSaaSへ移行。**低テストな保守と、ビルドの即時フィードバック、全ビルドの状態をコミュニティ全体で共有**。最小減のコストでコードをオープンソース化。

某オンライン決済サービス企業

目的

5,000名規模の開発者が存在する中での、従来型の開発手法からの脱却。計画的なダウンタイムの減少、品質が担保されたソフトウェアの早期リリースを実現する。

成果

四半期から隔週ごとのリリースを実現。アジャイルへの転換により、**リリース時の計画的なダウンタイムなし**、開発におけるシェアードサービスやSREなどの組織を新設。

* 詳細は <https://cloudbees.techmatrix.jp/resources/> から必要事項を入力後、ダウンロードください

CI/CDにおける 国内の状況

Jenkinsの生みの親 川口氏のコメント

- 弊社で2017年に開催したカンファレンスにて、Jenkinsの生みの親である川口氏は以下のように語っています。



川口耕介氏
CloudBees, Inc. CTO
Jenkinsの生みの親

日本企業はソフトウェア開発への危機感が足りない

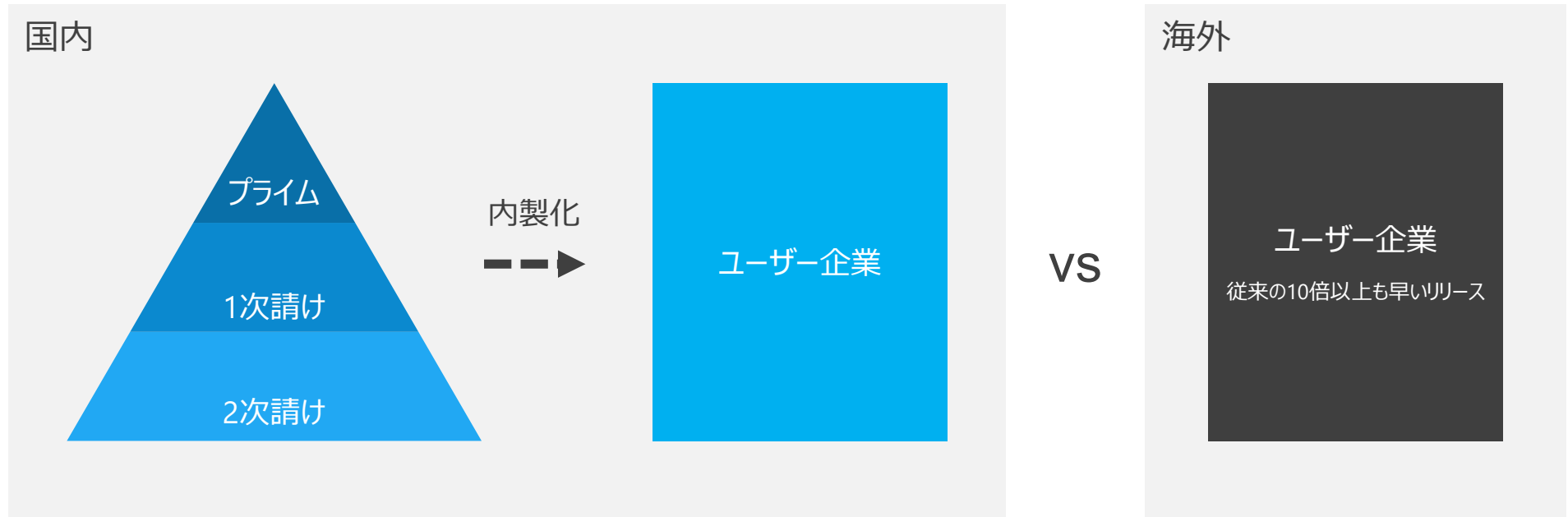
ドイツに行ったときに、自動車関連のサプライヤーが集まっていて、彼らが社内の取り組みを見せてくれた。彼らの会社には、CI/CDの専任のチームがあり、効率が10倍上がるという話をしてくる。

海外は次の10倍の効率化に取り組んでいる状態なのに、日本では効率化が進んでいない。日本が世界で競争していくためには、ソフトウェア開発者が中心となって、すぐに取り組みを始めないといけない。

10年後には日本が「第3世界」になってしまう。

Jenkins Day Japan, 2017/11/09

どうということか？



内製化が進むため、従来型のSIはなくなるが減る。また、ソフト開発 + αを提供する、ITサービス化が進む。
そこへ、既に日本企業よりも何倍も早いスピードでソフトウェアを投入する海外企業が競合する。

弊社調べ：国内のCI/CD普及率

- 国内では試験的な運用を含めて**16%**の普及率（約8割が未導入）です。

Q. CI/CDを実施していますか？



Jenkins Day Japan, 2017/11/09 (N=258)

競合する海外の企業はすでに生産性 10 倍を実現し、さらなる取り組みを実施しており、日本の企業が競合する海外の企業と競争するためには、海外と同様に生産性の向上が必須です。しかし、多くの開発現場では、未だに「取り組みを検討中・関心はある」という状況であり、CI/CD の効果を頭では理解しているものの、実際に自身の開発プロジェクトに導入するためにはどうすればよいかわからないという方が多くいることがわかっています。

どう導入すればよいかわからない

- CI を導入しようとしている方からは、様々な課題をお聞きます。



何を自動化すべきか

ツール評価や構築に
時間がかかる

標準化した
自動化プロセスの
策定が困難

これらの課題を海外の企業はどう解決したのでしょうか？

実は、海外の取り組みには**共通点**があり、テスト自動化をうまく推進するためのステップが存在します。

CI/CDの導入ステップ

1. BOTTOM-UP

- 現場で興味のある人が自動化を始める
- 現場レベルで自動化のメリットが理解される
- 徐々に周りに広がっていく

3. SCALE

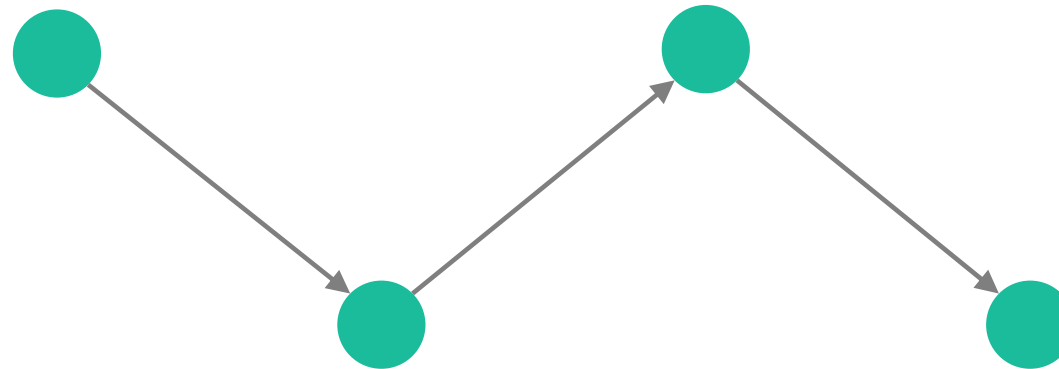
- サイロを均一化し、改善の効果を上げる
- 自動化の専門チームを組織する
- 共通基盤を構築する

2. TOP-DOWN

- 組織的な自動化に取り組むことをコミットする
- 経営者・管理職の優先順位を上げるために非常に高い目標を設定することを推奨
- 自動化が組織的に広がっていく

4. INSIGHTS

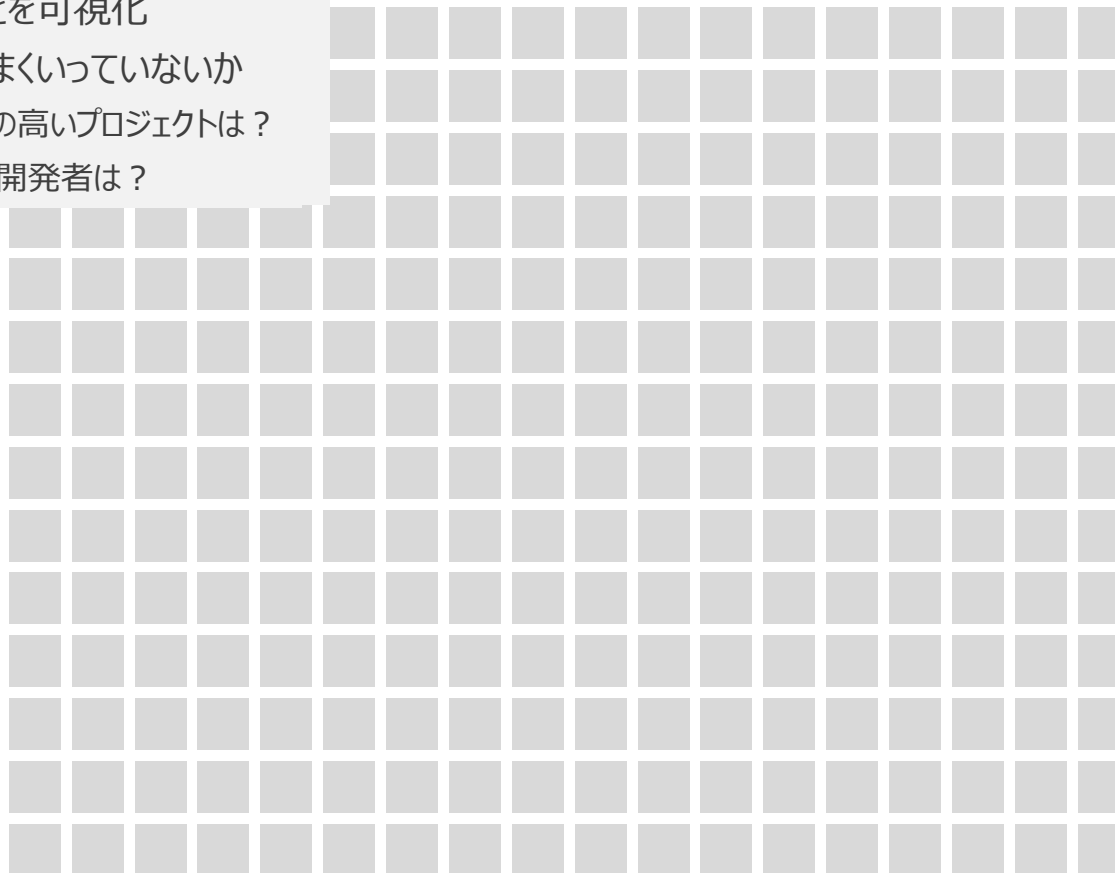
- 開発現場で起きていることを可視化
- どこがうまくいて、どこがうまくいっていないか
 - ビルドに失敗する頻度の高いプロジェクトは？
 - ビルドを壊すことの多い開発者は？



CI/CDの導入ステップ

4. INSIGHTS

- 開発現場で起きていることを可視化
- どこがうまくいって、どこがうまくいっていないか
 - ビルドに失敗する頻度の高いプロジェクトは？
 - ビルドを壊すことの多い開発者は？



日本におけるCI/CDは…

1. BOTTOM-UP

- 現場で興味のある人が自動化を始める
- 現場レベルで自動化のメリットが理解される
- 徐々に周りに広がっていく

3. SCALE

- サイロを均一化し、改善の効果を上げる
- 自動化の専門チームを組織する
- 共通基盤を構築する

海外事例は
CI/CDでココ

国内では、CIのみであっても
約8割が“1. BOTTOM-UP”にすら到達していない

2. TOP-DOWN

- 組織的な自動化に取り組むことをコミットする
- 経営者・管理職の優先順位を上げるために非常に高い目標を設定することを推奨
- 自動化が組織的に広がっていく

4. INSIGHTS

- 開発現場で起きていることを可視化
- どこがうまくいて、どこがうまくいないか
 - ビルドに失敗する頻度の高いプロジェクトは？
 - ビルドを壊すことの多い開発者は？

海外事例は
CI/CDココ

テスト自動化基盤パッケージを活用した基盤の構築



TechMatrix Jenkins

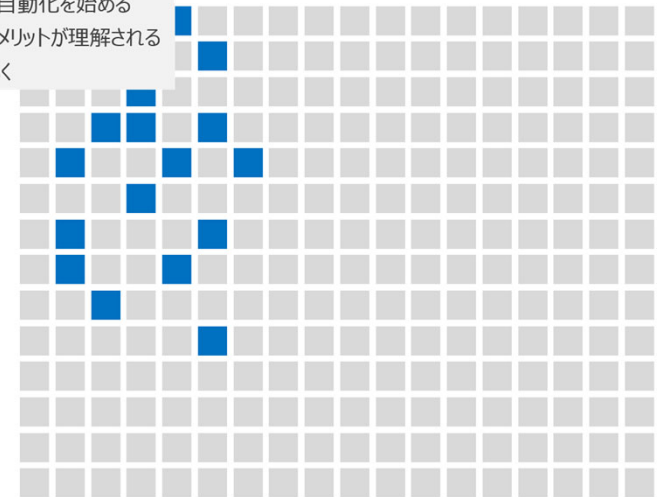
Platform Package for Java

開発現場における自動化の第一歩を支援

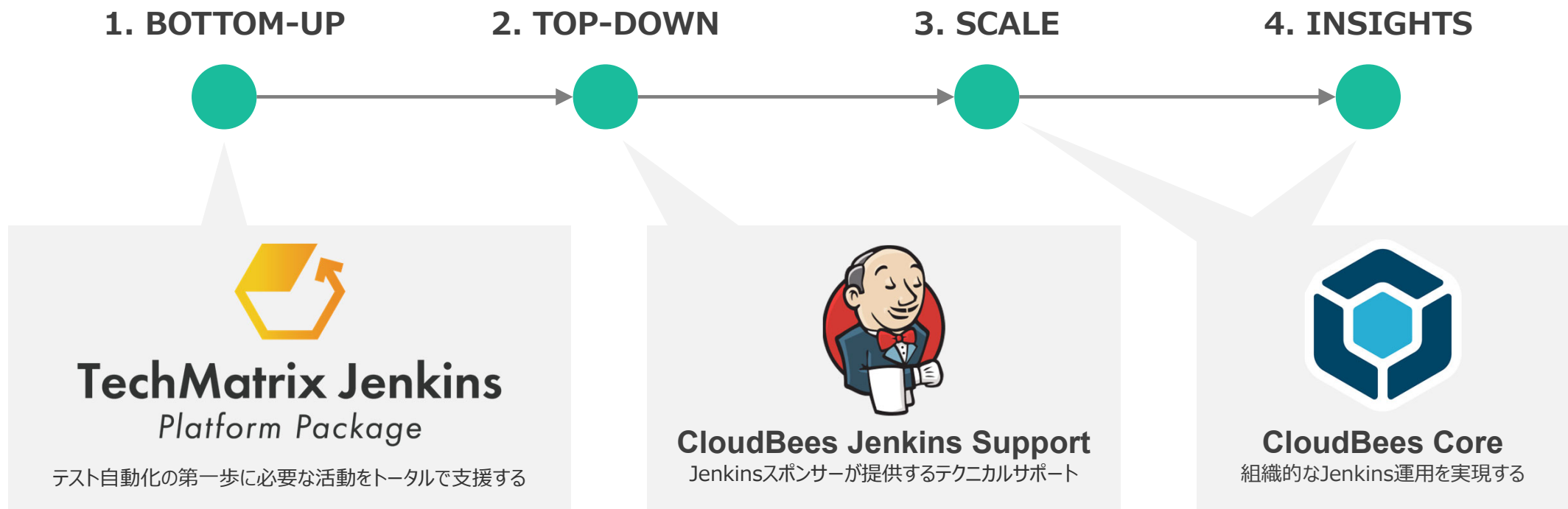
- **Jenkins**を用いた自動化基盤の構築
- 静的解析ツール**Jtest** によるバグ検出
- 構築手順書、構築内容のレクチャー
- 構築した環境のサポート
- 豊富なオプション

1. BOTTOM-UP

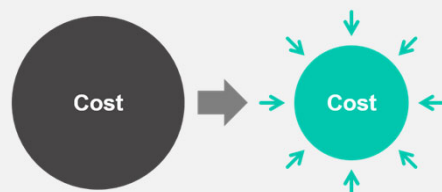
- 現場で興味のある人が自動化を始める
- 現場レベルで自動化のメリットが理解される
- 徐々に周りに広がっていく



位置づけ

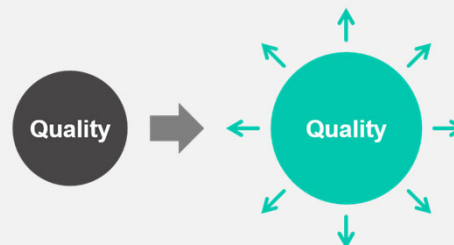


開発チームにおける導入のメリット



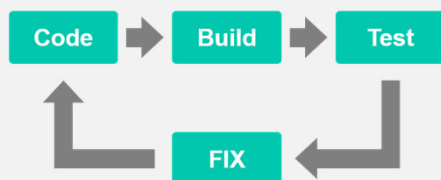
開発工数の削減

ソースコードの早期チェックにより、バグ解決数の向上やコードレビューの工数が削減される。



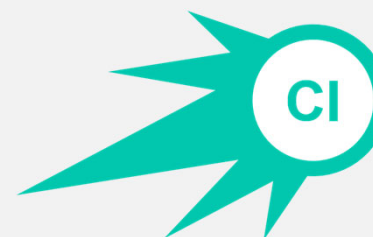
作業品質の向上

ビルドやソースコードのチェックの定期実行により、開発されるコードの品質が均一化される。



運用の標準化

自動化や構成管理が適切に行われるようになり、運用のルール化が進む。



自動化の促進

手間のかかる構築・教育を支援することにより、開発現場に自動化基盤が楽に導入される。

自動化の基盤を手に入れた後に、単体テストの実施などの「自動化する活動の追加(テストの追加)」や他の追加案件プロジェクトでの利用などの「対象の開発プロジェクトの追加(範囲の拡大)」を実施できるようになり、現在の開発現場のより大きな改善が期待できます。

ご提供内容

1 Jtest Engine ノードロックライセンス 1本

- バグや保守性の問題を検出するためのJava対応テストツールです。
- 米国Parasoft社にて開発され、国内導入実績は5,000ライセンスを超えます。

2 バグ検出 アセスメントサービス

- エンジニアが貴社へ訪問し、Jtestにて貴社の実プロジェクトのバグを検出、内容を説明いたします。
- バグ検出の性能評価のために行います。

3 Jenkinsを用いた自動化環境の構築

- オープンソースのJenkinsを用いた、ビルド・バグ検出の自動化環境を構築します。
- 必要となるJenkins, Jtestおよび関連ツールのインストール、動作確認を行います。

4 環境構築手順書の作成

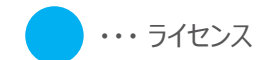
- 構築した自動化環境の設定内容をドキュメントとして作成します。

5 環境構築のノウハウをレクチャー

- 自動化環境の構築手順書の内容や設定変更の手順などをエンジニアから貴社担当者様へ説明いたします。

6 構築した環境の保守サポート(3ヶ月間)

- 環境構築後、3ヶ月間の保守サービスを提供します。
- 構築した環境に関するお問い合わせをメールにて対応します。



… ライセンス



… サービス

A Jenkins 1 Day トレーニング 推奨

- CloudBees社認定エンジニアによるJenkinsのハンズオントレーニングです。
- 継続的インテグレーションの基礎から、Jenkinsの基本的な使い方(ビルド、デプロイ、プラグイン管理)、さらにはJenkins2.0で導入されたPipelineまで、実践的な演習を通じて習得できます。

B CloudBees Jenkins Support

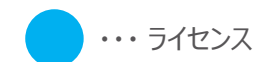
- プラグインの互換性の確保と、Jenkinsエキスパートによるベストプラクティスを提供します。
- クリティカルなCDパイプラインから不要なビジネスリスクが削除され、開発チームは安心してJenkinsを利用することが出来ます。

C Jtest バグ検出のルール選定サービス

- 実際の実プロジェクトの解析結果の分析、および貴社のご要望をもとに、Jtestの適切なルールを選定します。

D Ranorex によるUI テスト自動化

- UIテスト自動化ツール Ranorexを使い、Jenkins上でUIテストの実行が呼び出されるよう、設定を行います。※ テストシナリオの作成は含まれません。



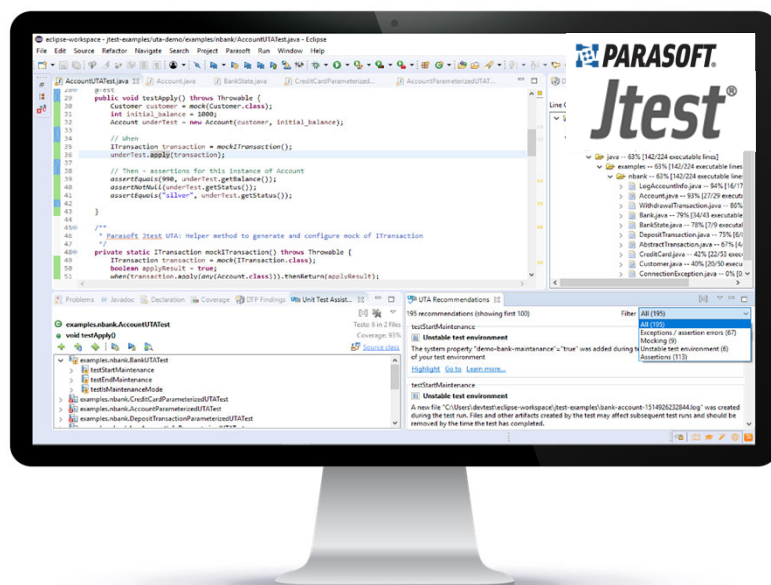
… ライセンス



… サービス

ソースコードの問題を検出する

- TechMatrix Jenkins Platform Package for Javaでは、**Parasoft Jtest** を利用します。
- 約1,000種類のコーディングルールやバグ検出のルールを用いて、ソースコードの問題を検出します。



例

バグの可能性

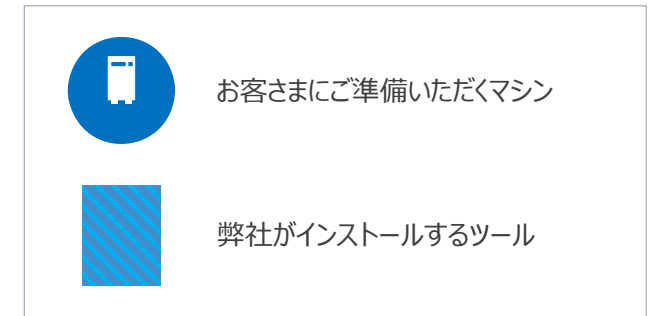
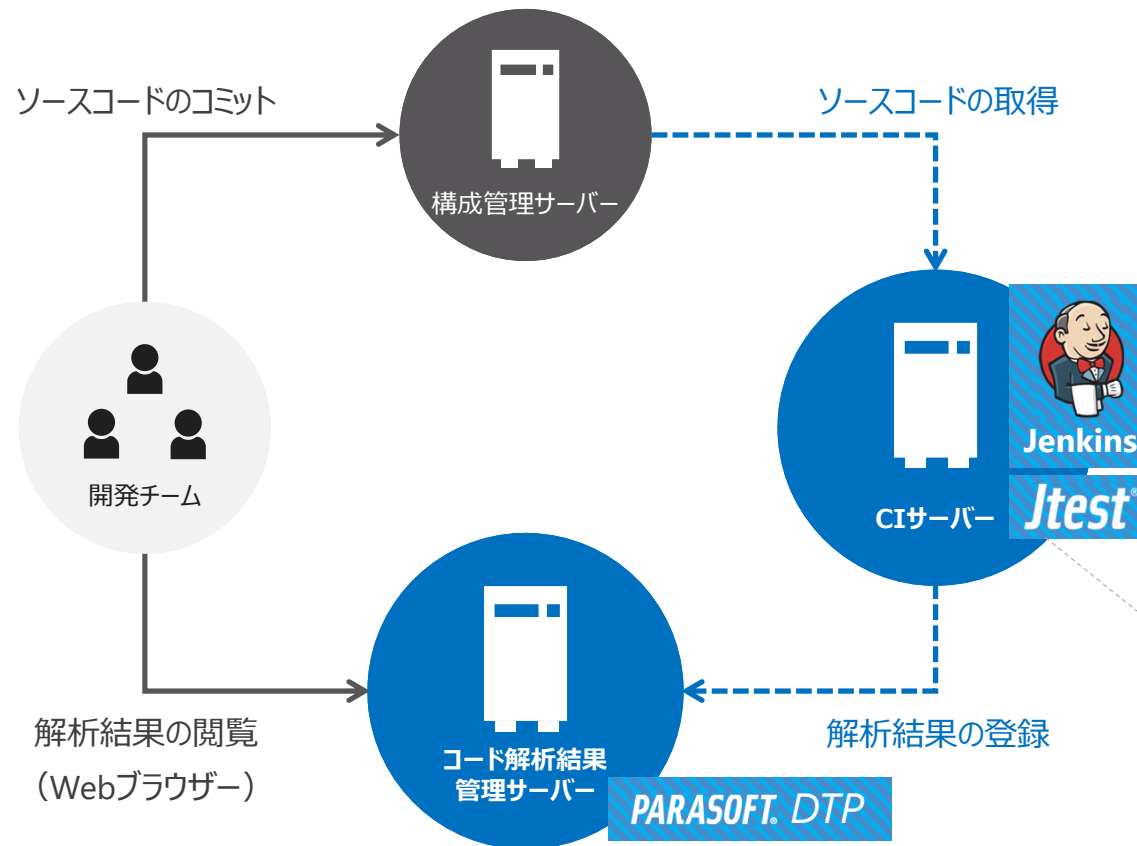
- 初期化されていない変数を使用しない
- リソース解放漏れはないか
- 一連の処理ロジックにおいて、除数がゼロとなる処理（ゼロ除算）を行わない
- アプリケーションがデッドロックするような処理はないか
- 必要な可能性のあるシステムリソースを解放しない

保守性

- ハード コーディングされた定数を使用しない
- 重複コードが含まれていないか

※ 本パッケージのJtestには、解析結果をWebブラウザ上で確認するためのDTP Standardが付属します。

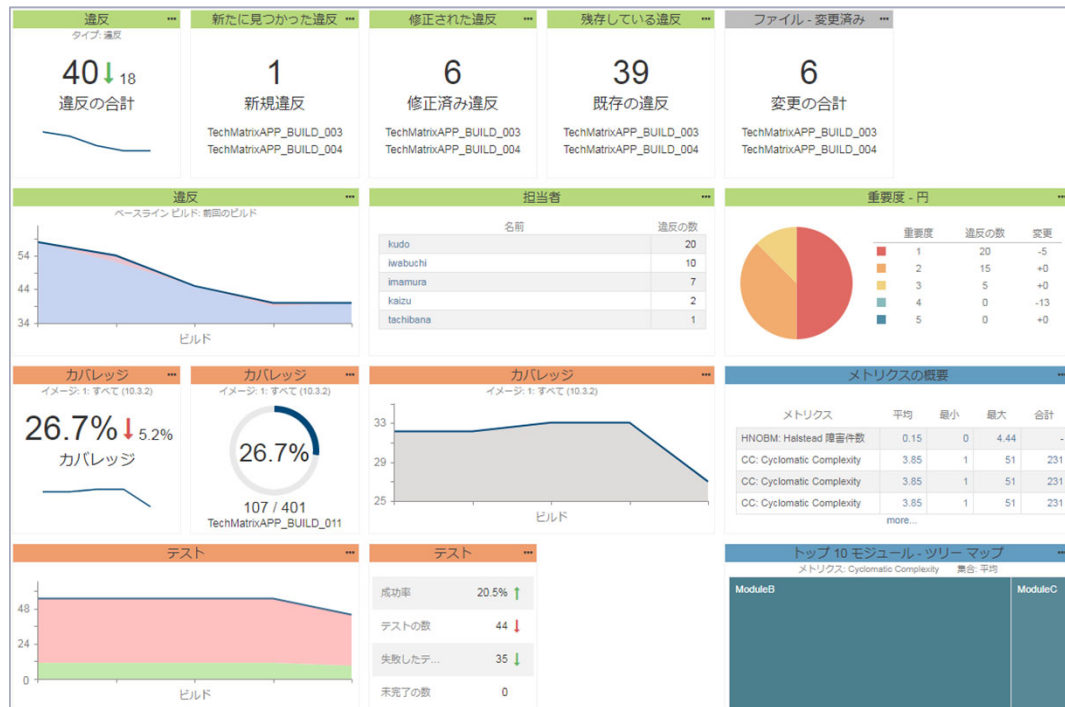
マシンおよびツールの構成



JenkinsのJobを実行

1. 構成管理からソースコードを取得
2. ビルドを実行
3. **Jtest**によりバグおよび保守性の問題を検出
4. **Jtest**の解析結果を管理用サーバー(DTP)へ登録

Parasoft DTPによる結果の確認



The interface shows a list of violations with columns: 期間 (Period), ファイル (File), 行 (Line), メッセージ (Message), 重要度 (Severity), 担当者 (Assignee), 優先度 (Priority), アクション (Action), リソース (Resource), カタログ (Catalog), ルール (Rule).

違反のリスト (List of Violations):

期間	ファイル	行	メッセージ	重要度	担当者	優先度	アクション	リソース	カタログ	ルール	
2017-06-21	useBeforeInit.c	40	ファイルがクローズされていない file	1	kudo	緊急	修正	大	2017-06-21	リソース	BO-RES-L...
2017-06-21	csv_check.c	150	ディレクトリストリームがクローズされていない dir	1	iwabuchi	緊急	修正	大	2017-06-21	リソース	BO-RES-L...
2017-06-21	fileLeak.c	49	ファイルがクローズされていない file	1	iwabuchi	緊急	修正	大	2017-06-21	リソース	BO-RES-L...
2017-06-21	memoryLeak.c	40	メモリが解放されていない pointer	1	iwabuchi	緊急	修正	大	2017-06-21	リソース	BO-RES-L...
2017-06-21	unreachableCode.c	67	"cur" が参照参照の後に null チェックされている	2	kudo	緊急	レビュー	中	2017-06-21	バグの可能性	BO-PB-DE...
2017-06-23	csv_check.c	156	配列 cRead_fname_tbl[0..1] の境界外のインデックス A...	2	imamura	緊急	修正	大	2017-06-23	バグの可能性	BO-PB-AR...
2017-06-23	csv_check.c	1215	ファイルがクローズされていない fp	1	kudo	緊急	修正	大	2017-06-23	リソース	BO-RES-L...

違反の詳細 (Violation Details): ファイル (違反): csv_check.c

```

147- {
148-     case 1: /* カレントディレクトリオープン */
149-         dir = opendir(".");
150-         while ( (dir_p = readdir(dir)) != NULL )
151-         {
152-             /* ディレクトリ内ファイル読み込み */
153-             while ( (dir_p = readdir(dir)) != NULL )
154-             {
155-                 /* 読み込み対象となるファイルパスを探索するループ */
156-                 for ( i = 0; strcmp(cread_fname_tbl[i], ".") != 0; i++)
157-                 {
158-                     /* *.csv があれば、カウントメニュー */
159-                     if ( strstr(dir_p->d_name, cread_fname_tbl[i]) != NULL )
160-                     {
161-                         read_flg = CSV_CHECK_ON; /* ファイル有無フラグ ON */
162-                         break;
163-                     }
164-                 }
165-                 /* カウント対象ファイルが存在したらファイル読み込みループを抜ける */
166-                 if ( read_flg == CSV_CHECK_ON )
167-                 {
168-                     break;
169-                 }
170-             }
171-             /* カレントディレクトリに読み込むファイルがある場合 */
172-             if ( read_flg == CSV_CHECK_ON )
173-             {
174-                 read_cnt = GetFilename(); /* 読み込みファイル名 */
175-                 read_cnt = GetFilename(); /* 読み込みファイル名 */
176-             }
177-         }
178-     }
179- }
180- 
```

違反の選択 (Violation Selection): 違反が選択されています: 1

違反の属性 (Violation Attributes):

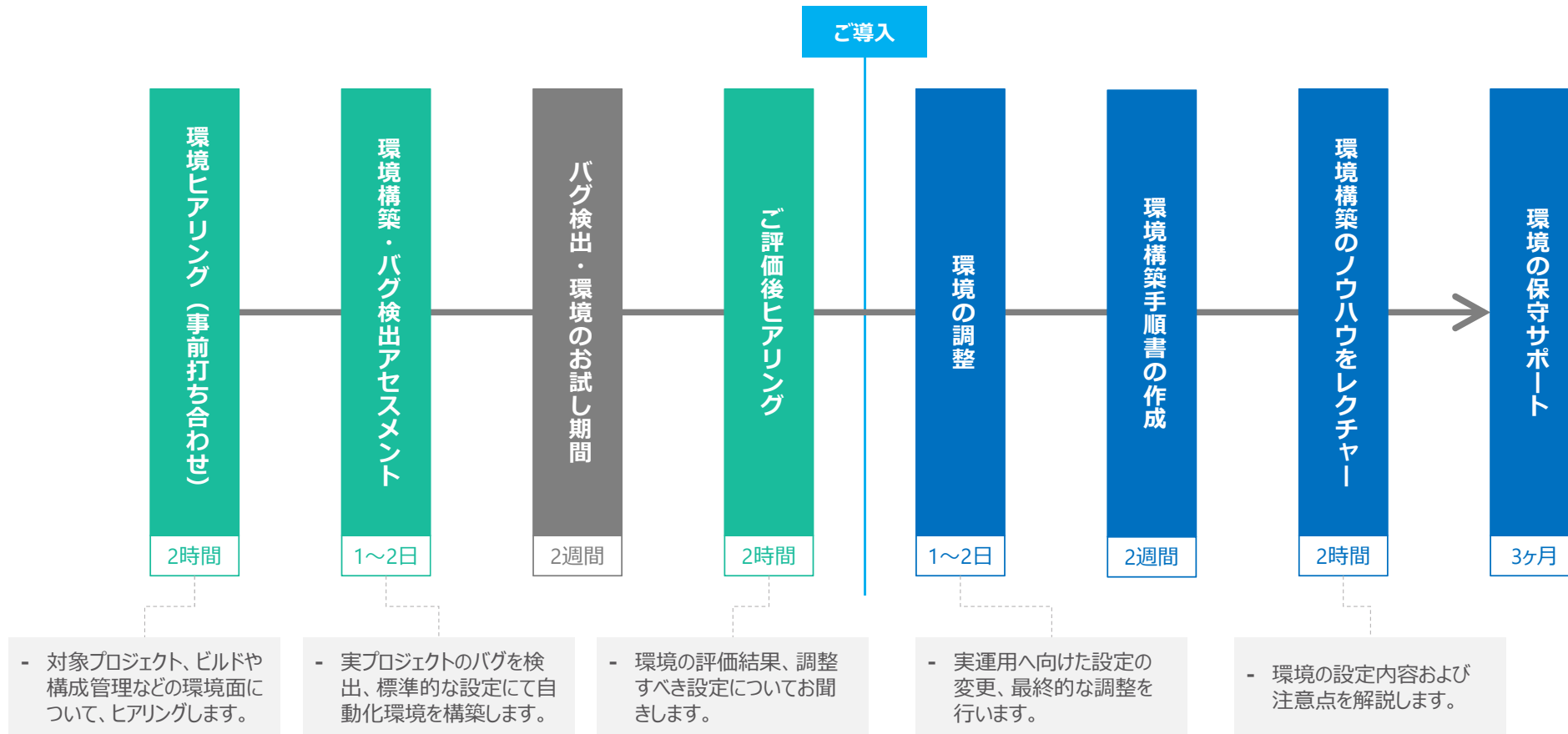
- 重要度: 1
- 作成者: kudo
- 担当者: iwabuchi
- 優先度: 緊急

フロー解析トレース (Flow Analysis Trace):

```

e csv_check.c (121): int menu_no; /* 入力されたメニュー */
e csv_check.c (122): int read_flg; /* ファイル有無フラグ */
e csv_check.c (123): DIR *dir; /* ディレクトリストリーム */
e csv_check.c (124): struct dirent *dir_p; /* ディレクトリエントリ */
e csv_check.c (125): int i; /* ループカウンタ */
e csv_check.c (126): int read_cnt; /* 読み込みファイル名 */
181- 
```

サービスの流れ



一式 **¥4,500,000** (税抜)

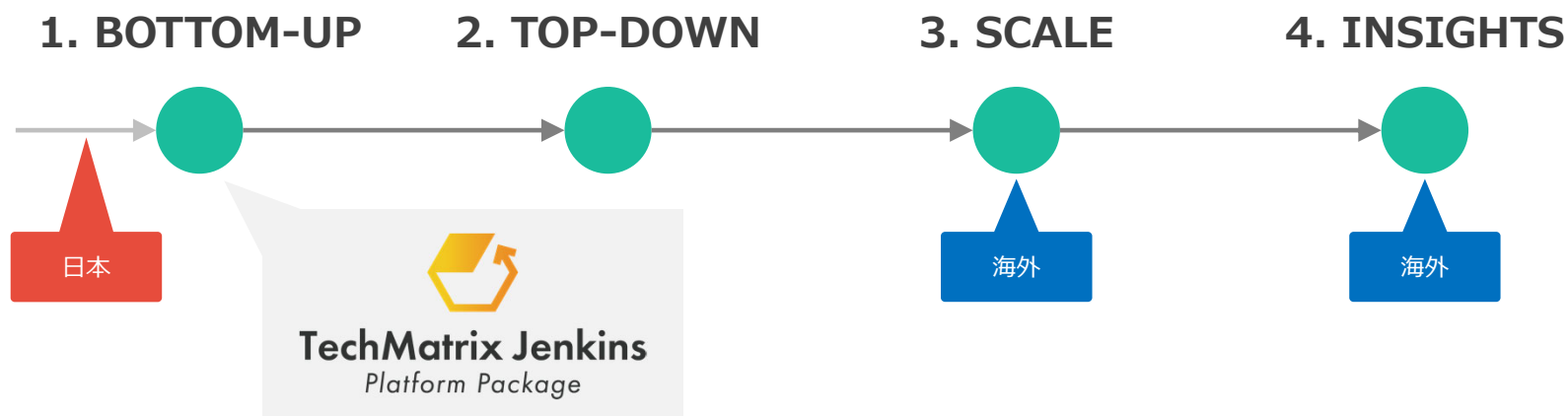
- 1 Jtest Engine ノードロックライセンス 1本
- 2 バグ検出 アセスメントサービス
- 3 Jenkinsを用いた自動化環境の構築
- 4 環境構築手順書の作成
- 5 環境構築のノウハウをレクチャー
- 6 構築した環境の保守サポート(3ヶ月間)

- 構築する環境は1環境となります。
- 3ヶ月間の保守サポート後は、Jtest Engineに関する保守サポートが購入から1年間付きます。(次年度更新費用：¥600,000)

まとめ

まとめ

- 海外では自動化の取り組みが当たり前であり、日本ではまだ普及していません。
- 海外の自動化の取り組みには共通する4つのステップがあり、まずは開発チームで自動化をはじめることからスタートしています。




- 弊社が提供する、ツール導入や環境構築、教育やサポートの一括提供により、開発現場における自動化導入の障壁を取り除くことで、競争優位性や開発効率化の実現を支援します。




テクマトリックス株式会社

システムエンジニアリング事業部

ソフトウェアエンジニアリング営業部

 03-4405-7853

 03-6436-3553

 se-info@techmatrix.co.jp

 <https://www.techmatrix.co.jp/product/jppj/>