

ツールを用いたテスト自動化と 継続的インテグレーションの実現

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

会社紹介



テクマトリックスは、最新のIT技術を活用し、顧客企業のビジネスモデル変革と企業競争力の強化をご支援するITのスペシャリスト集団です。日々技術革新が進むIT分野において、常に新しい技術トレンドを掴み、時代を先駆ける最先端かつ最高のテクノロジーとソリューションをお届けします。

設立	1984年8月30日
上場市場	東京証券取引所市場第一部
従業員数	1,057名〔連結〕（'18年3月末現在）
本社所在地	東京都港区三田3-11-24 国際興業三田第2ビル

ソフトウェア開発の環境の変化

時代の変化が高速化し、ニーズが多様化するにあたり、ソフトウェアの重要性が向上しています。
市場競争力や戦略は、ソフトウェアで実現することが求められています。

2008

2008	
Rank	Company
1	Exxon
2	General Electric
3	Microsoft
4	AT&T
5	Proctor & Gamble
6	Berkshire
7	Google
8	Chevron
9	J&J
10	Walmart

VS

2018

2018	
Rank	Company
1	Apple
2	Google
3	Microsoft
4	Amazon
5	Facebook
6	Berkshire
7	J&J
8	JP Morgan
9	Exxon
10	Bank of America

largest companies by market value in 2018 vs2008

ビジネス＝ソフトウェア



いかにリリースを早くするか？

「企業の競争力＝製品のリリーススピード」となりつつある現代において、品質の高いソフトウェアをいかに早く市場にリリースするかという点が重要になってきています。

1. テスト自体を自動化する

例：JUnitによる単体テスト、Seleniumによるwebアプリテスト

2. ビルド、テスト、デプロイなどの一連の流れを自動化する

継続的インテグレーション(CI)、継続的デプロイメント(CD)

本日はご紹介するツール



UIテストを効率的に自動化
Ranorex



組織的なCI/CDを実現

CloudBees Jenkins Solution

UIテスト自動化 : Ranorex

テスト自動化の目的と効果



コスト削減

- ≫ 再利用・繰り返し実行可能な自動テスト
- ≫ 従業員の工数削減 (手作業でのテスト工数の削減など)
- ≫ 欠陥の早期発見・修正工数の削減



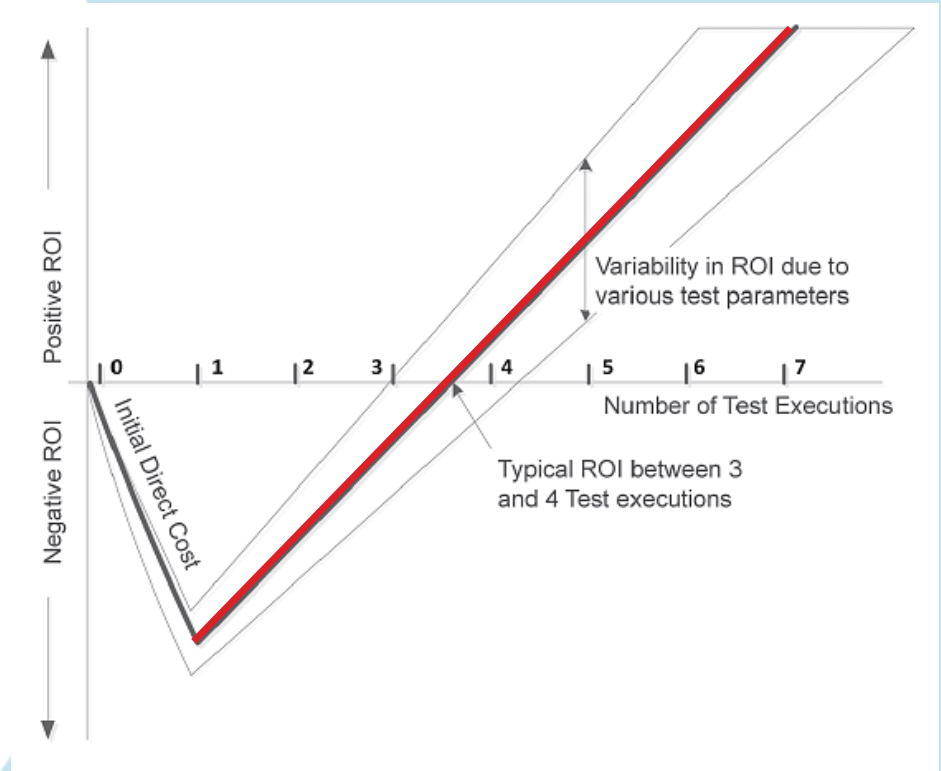
効率向上

- ≫ テスト実行時間の短縮
- ≫ テスト実行回数の増大



品質向上

- ≫ 対象システムの品質向上
- ≫ テストの品質向上 (属人的ミスなどの排除)



UIテスト自動化への様々な課題

- ✓ 複数のプラットフォーム・デバイス・テクノロジーへの対応
- ✓ テスト対象システムの頻繁な変更・リリース
- ✓ テストケース・テストスクリプト作成/保守の負荷
- ✓ 他システム(BTS,SCM,etc.)との連携
- ✓ 担当者のスキルセット・スキルレベルのばらつき
- ✓ ツールの学習コスト

...

Test Automation Made Easy



オールインワンのUIテスト自動化ツール

さまざまな環境・デバイス・ソフトウェアアプリケーションをサポートするテスト自動化ソリューション

Ranorex のテクノロジー

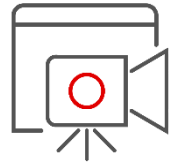


様々なテクノロジーのサポート



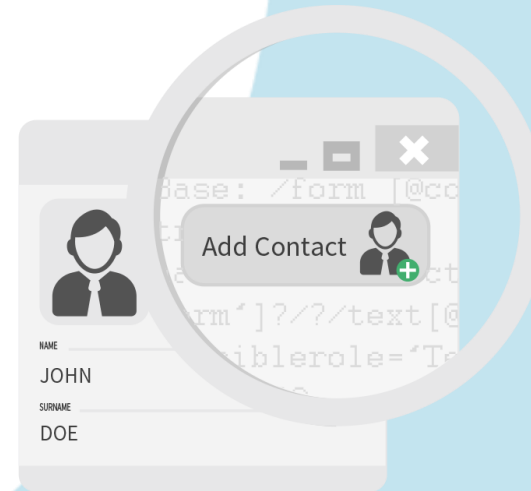
デスクトップ: 最新のWin10の他、多くの3rdパーティ製コントロールをサポート
Web : Flash,AIRなどのRIAをサポートし、クロスブラウザーでのテストをサポート
モバイル : Android/iOSの実デバイス上でのテストをサポート

Ranorex のテクノロジー



キャプチャ & リプレイ

- ≫ マウス・キーボード操作を記録し、テストを作成
- ≫ 記録されたアクション(テストシナリオ)と、識別したオブジェクト情報を分離して管理
- ≫ 記録されたアクションをGUI上で編集可能



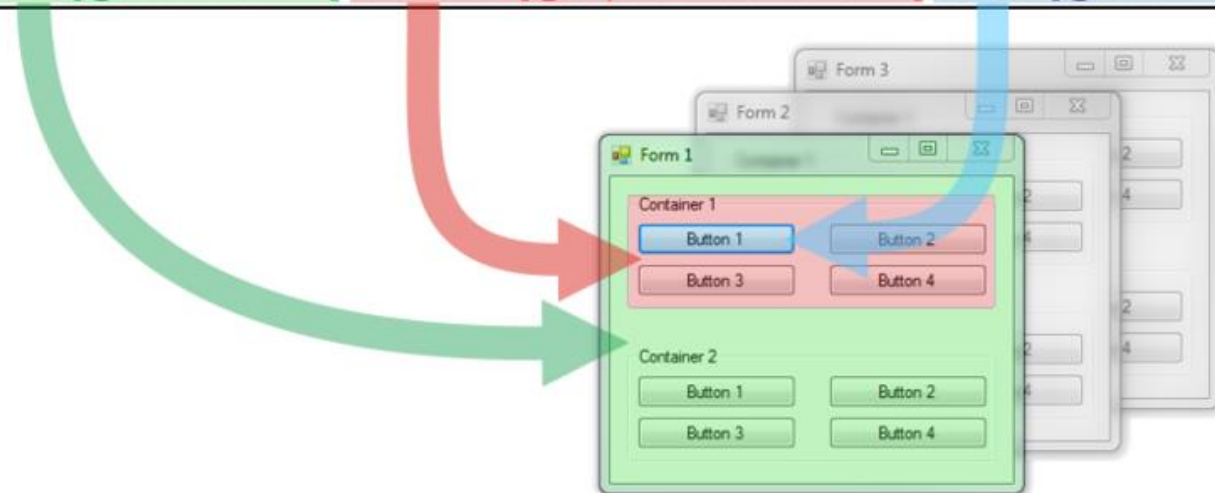
Ranorex のテクノロジー



オブジェクト認識

- » 見た目や座標に依存しない、コード情報ベースの堅牢なオブジェクト認識
- » オブジェクト情報とテストシナリオの分離
- » 動的なユーザーインターフェイスのサポート
- » オブジェクト認識のための **RanoreXPath** テクノロジー

```
/form[@title='Form 1']/container[@caption='Container 1']/button[@text='Button 1']
```

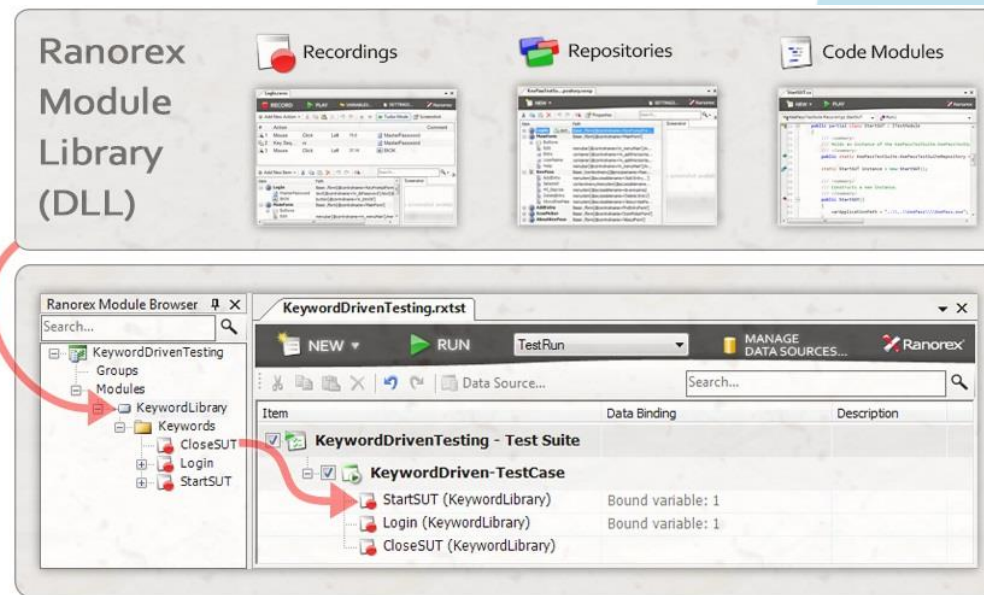


Ranorex のテクノロジー



再利用性の高いモジュール

- ≫ 一連の操作の流れを「モジュール」として管理・再利用
- ≫ ドラッグ&ドロップによる、複数のテストケースでの再利用
- ≫ モジュールの変更は、モジュールを参照しているすべてのテストケースに自動で反映

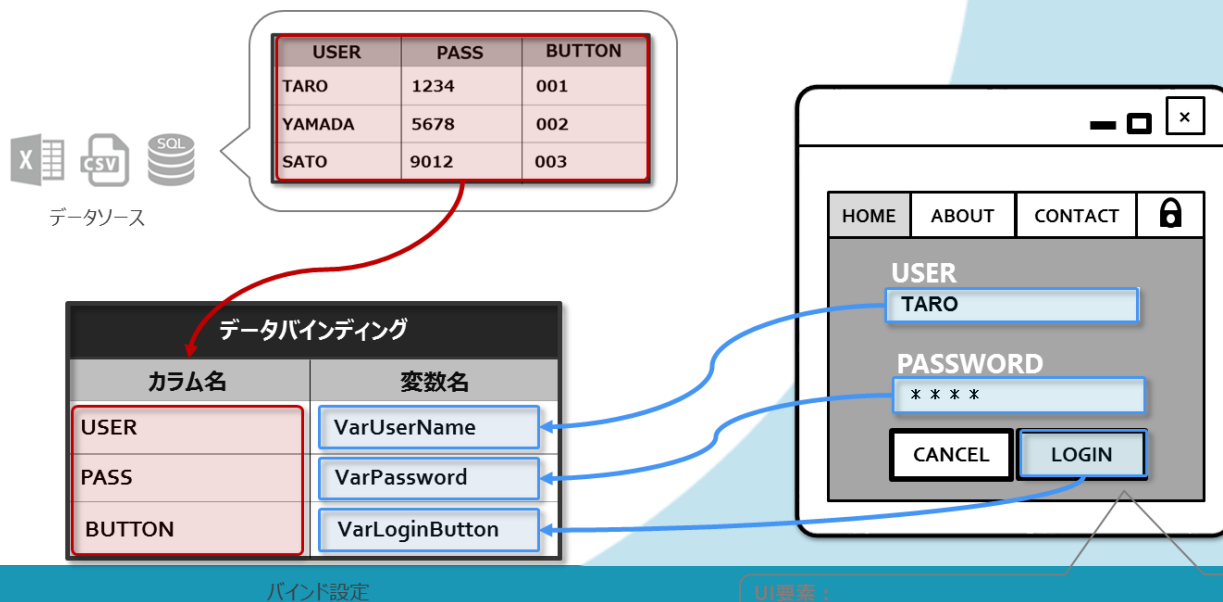


Ranorex のテクノロジー



データ駆動型テスト

- ≫ GUI上から、ソースコードレスでデータソース・データバインドを設定
- ≫ Excel, CSV, データベースをサポート
- ≫ データイテレーションごとのレポート生成



UI要素:

body/div/xxx/xxx/input[@name="varLoginButton"]

Ranorex のテクノロジー

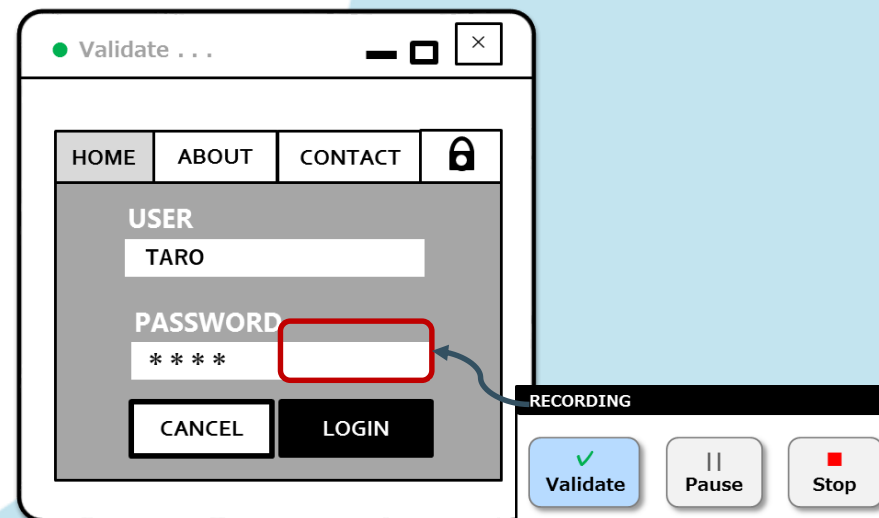


豊富なバリデーション(検証)

- ≫ フィールド値、イメージ、ツールチップ、その他オブジェクト属性
- ≫ オブジェクトの存在検証、オブジェクトの属性検証、オブジェクトの画像比較
- ≫ 指定、データソース値、正規表現で指定

バリデーションの種類

- **オブジェクトの存在検証**
 - ・ オブジェクトの存在の有無を検証します。
- **オブジェクトの属性検証**
 - ・ オブジェクトの属性値が期待値に一致するかを検証します。
正規表現や部分一致での検証をおこなうことができます。
- **UI要素の画面ショット比較**
 - ・ オブジェクトの画面ショットが期待する画面ショットに一致するかを検証します。類似度の指定や、画像フィルター(エッジ抽出やモノクロ化)を行っての比較もできます。



Ranorex のテクノロジー



より高度なテストスクリプトの作成

- » 標準的なプログラミング言語 (C#, VB.NET) のサポート
- » Ranorex API による高い拡張性
- » キャプチャベースのテストシナリオとの互換性

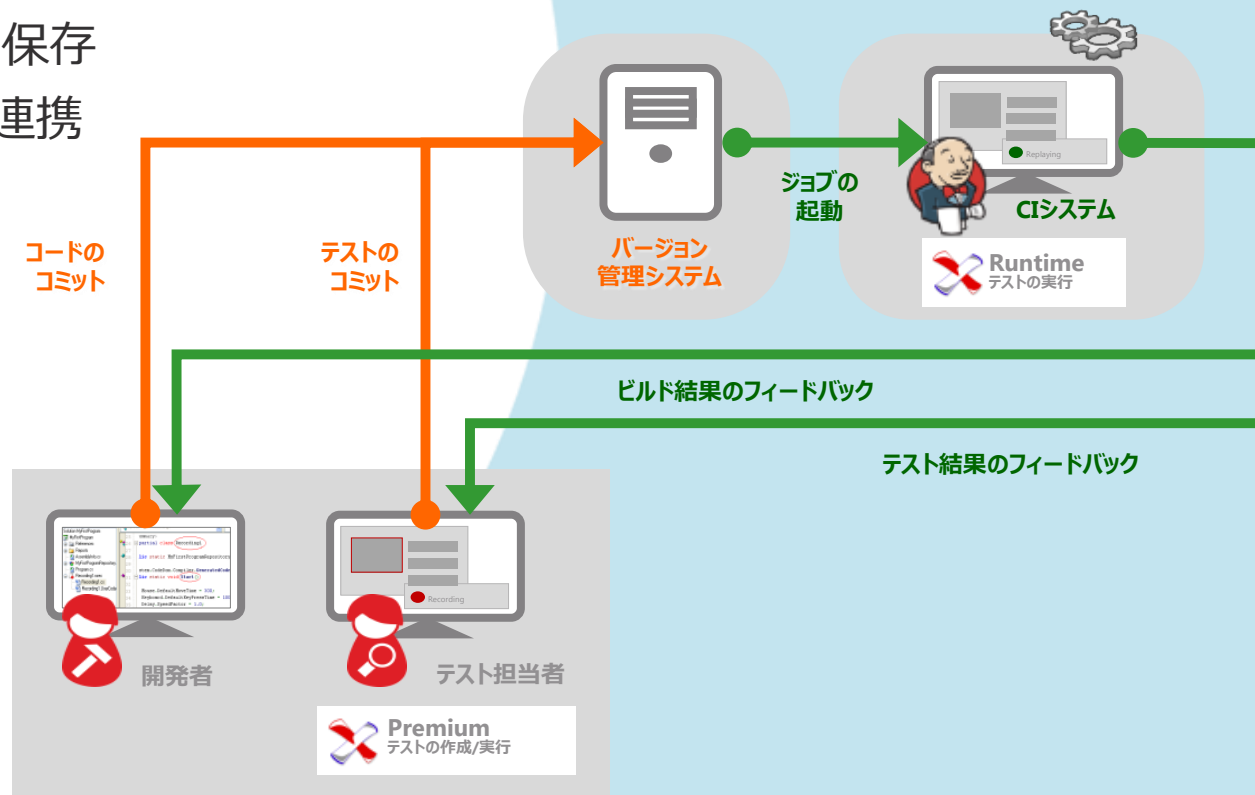
```
98  
99 void ITestModule.Run()  
100 {  
101     Mouse.DefaultMoveTime = 300;  
102     Keyboard.DefaultKeyPressTime = 100;  
103     Delay.SpeedFactor = 1.0;  
104  
105     // Click 'Add Entry' Button MainMenu  
106     MyRepo.MainForm.Edit.Click();  
107     MyRepo.KeePass.AddEntry.Click();  
108  
109     // Set text fields  
110     MyRepo.AddEntry.Title.TextValue = varTitle;  
111     MyRepo.AddEntry.UserName.TextValue = varUsername;  
112     MyRepo.AddEntry.Password.TextValue = varPassword;  
113     MyRepo.AddEntry.Repeat.TextValue = varPassword;  
114     MyRepo.AddEntry.URL  
115  
116     // Choose an icon  
117     MyRepo.AddEntry.M  
118     MyRepo.IconPicker.L1.Icon.Click(Location.CenterLeft);  
119     MyRepo.IconPicker.ButtonClose.Click();  
120  
121     // Set Expires
```

Ranorex のテクノロジー



他ツールとのシームレスな連携

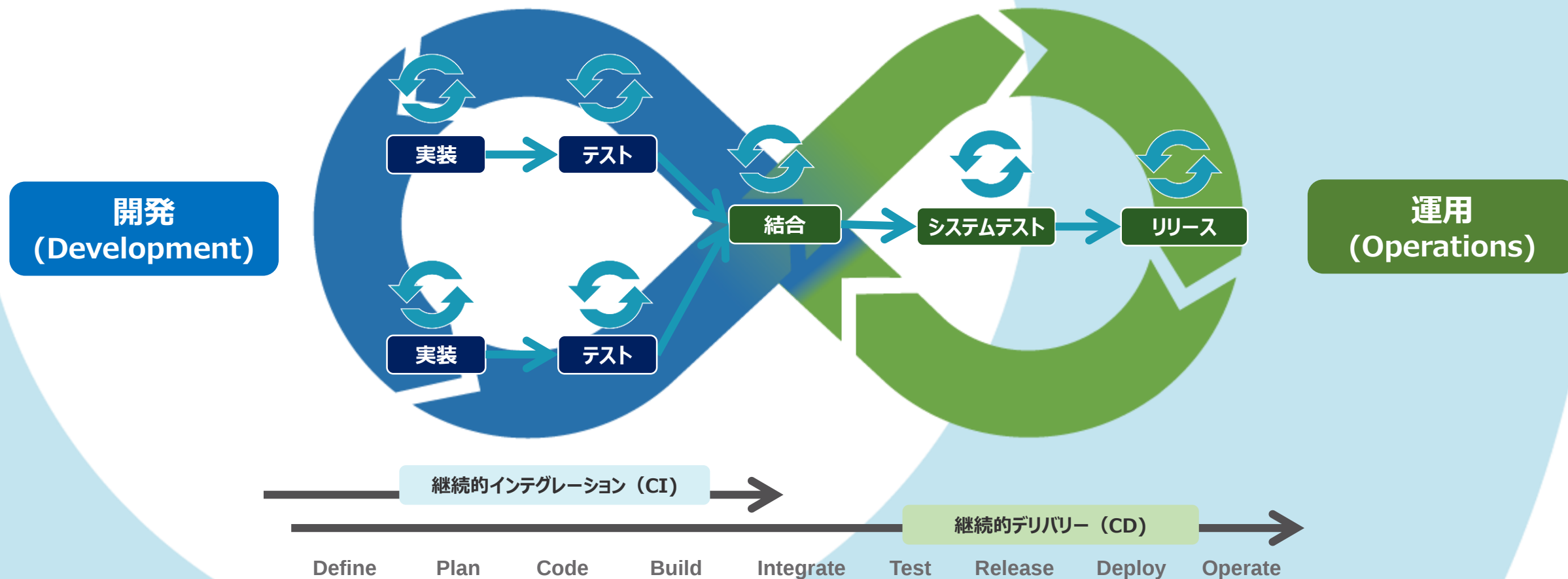
- ≫ テストをテスト実行ファイル形式(*.exe)で保存
- ≫ **JenkinsなどのCI環境**とのシンプルな連携



CI/CDの実現 : CloudBees Jenkins Solution

注目されている「DevOps」

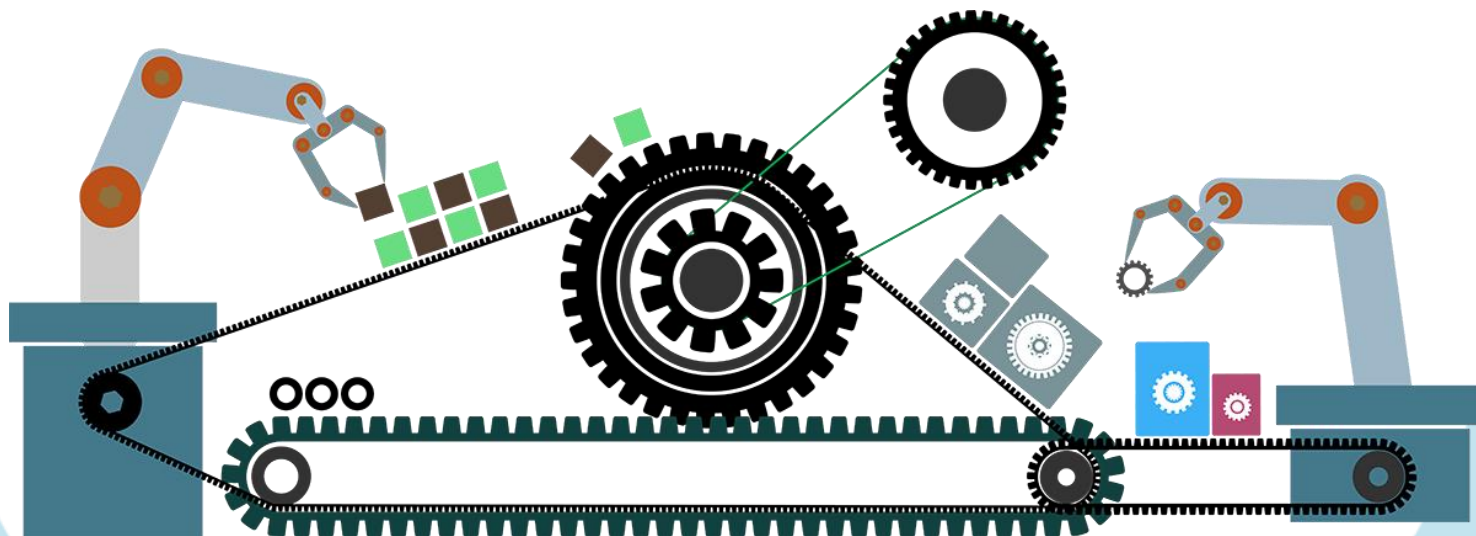
DevOpsとは、開発(Development)と運用(Operations)を組み合わせた言葉です。これまで組織的に分かれていた、開発チーム(Dev)と運用チーム(Ops)を1つのチームとして統合を行うことでソフトウェアのリリースサイクルを短くすることを目的としています。



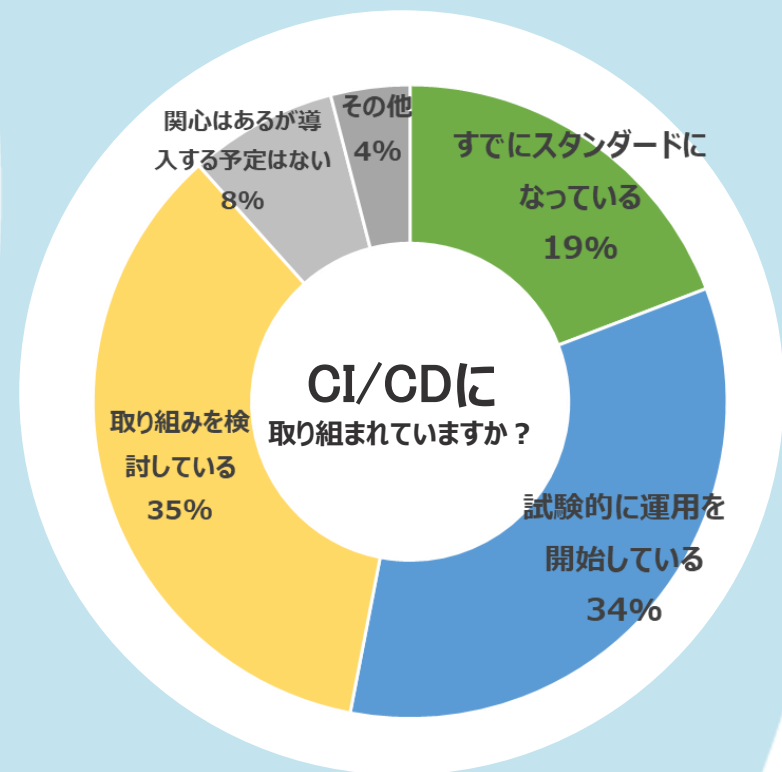
CI/CDの普及は？

CI/CDの導入は、「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」が必要となります。

CI/CD実現に向けては、開発チームと運用チームの間の障壁を取り除く必要があります。会社の組織改革が必要!!



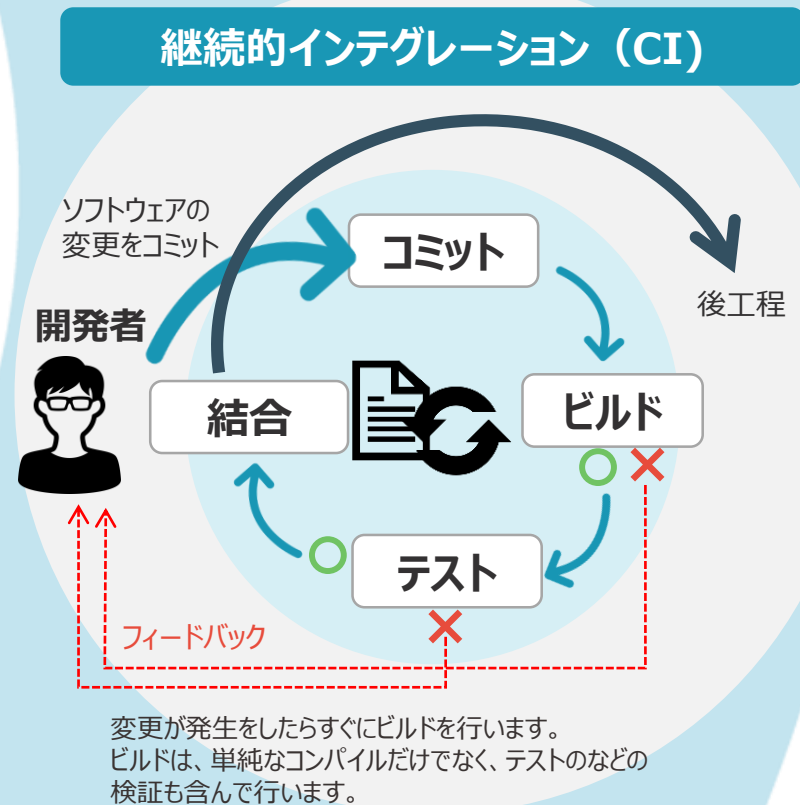
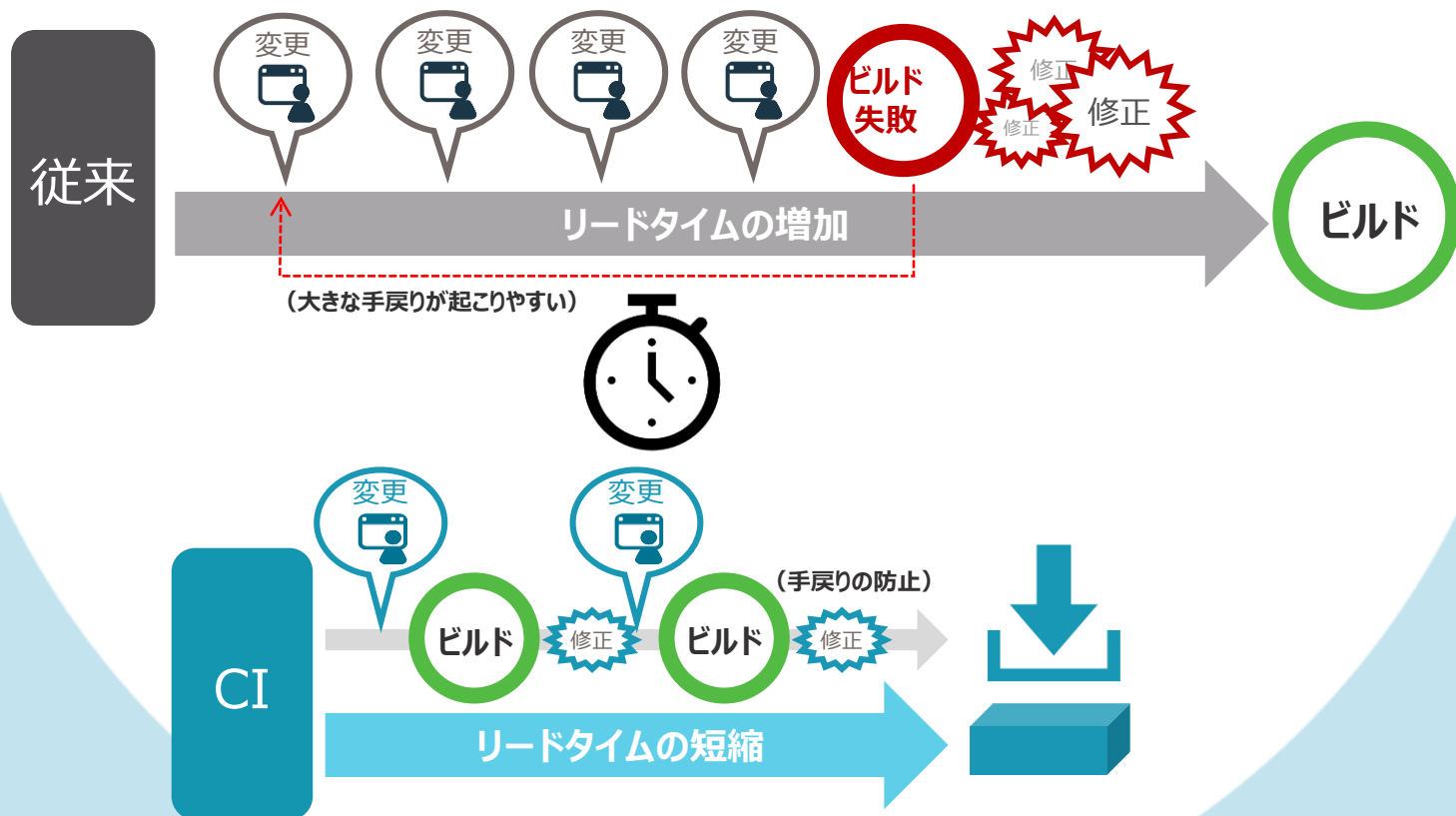
- ・ プロセスや文化を変える必要もありますし、人材も育てなければならない。
- ・ 学習コストもかかる。自動化の仕組みやツールにもお金がかかる。
- ・ ソフトウェアのアーキテクチャを変えないといけない。
- ・ 自動テストを書かないといけない。



データ：Jenkins Day Japan 2018のアンケート (n=283)

まずは、継続的インテグレーション（CI）からをお勧めします!!

継続的インテグレーション、CI（英: continuous integration）とは、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。変更に対して、自動でビルドや検証を行い、問題があれば影響範囲の小さいうちに修正することで後工程での手戻りを排除してソフトウェア開発のリードタイムの短縮を行います。



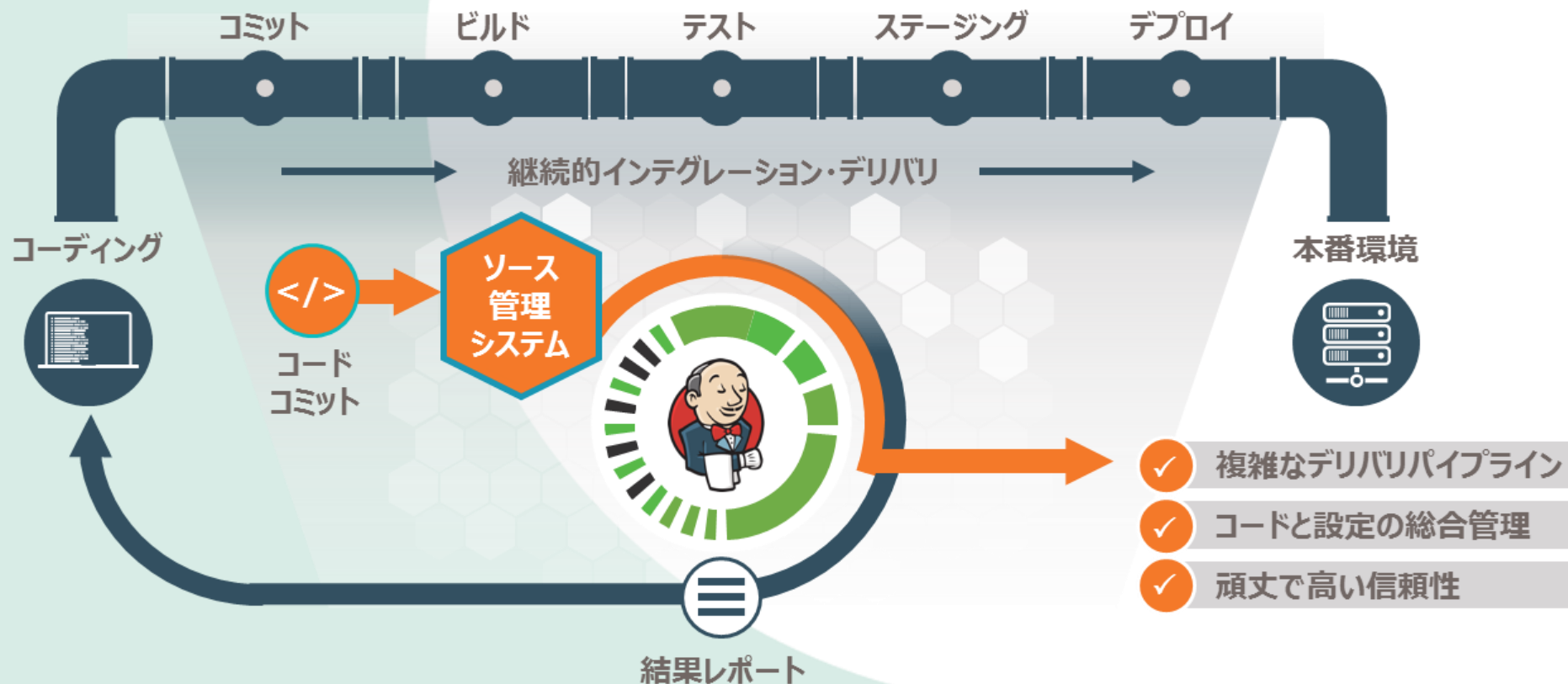
CI/CDツール「Jenkins」とは？



Jenkins

CI/CD実現のためのシェアNo.1ツール「Jenkins」

Jenkins (ジェンキンス)とは、ビルドパイプラインにおける「継続的インテグレーション・継続的デリバリー支援ツール」として利用されています。Jenkinsは汎用性が高く、基本的にJavaが動く環境であればLinuxでもWindowsでも動作させることができます。また、ビルドやテストに限らずどんなスクリプトでも実行することができるので、ビルドプロセスに合わせてビルドからリリースまで柔軟に自動化を行うことができます。



Jenkins導入以前の課題

ソフトウェア開発では、ソフトウェアのビルドやテストの作業のたびに、社内の中にいるビルド、テストを行う担当者に依頼を行う必要があり、ソフトウェアの開発の生産性に問題あるのではないのか？という動機から、生まれたのがJenkinsです。

Jenkinsは、開発者にとって面倒なビルド、テスト、品質検査、サーバーへの展開などを自動化を行い、ソフトウェアの開発プロセスにおける省力化を目指したものです。

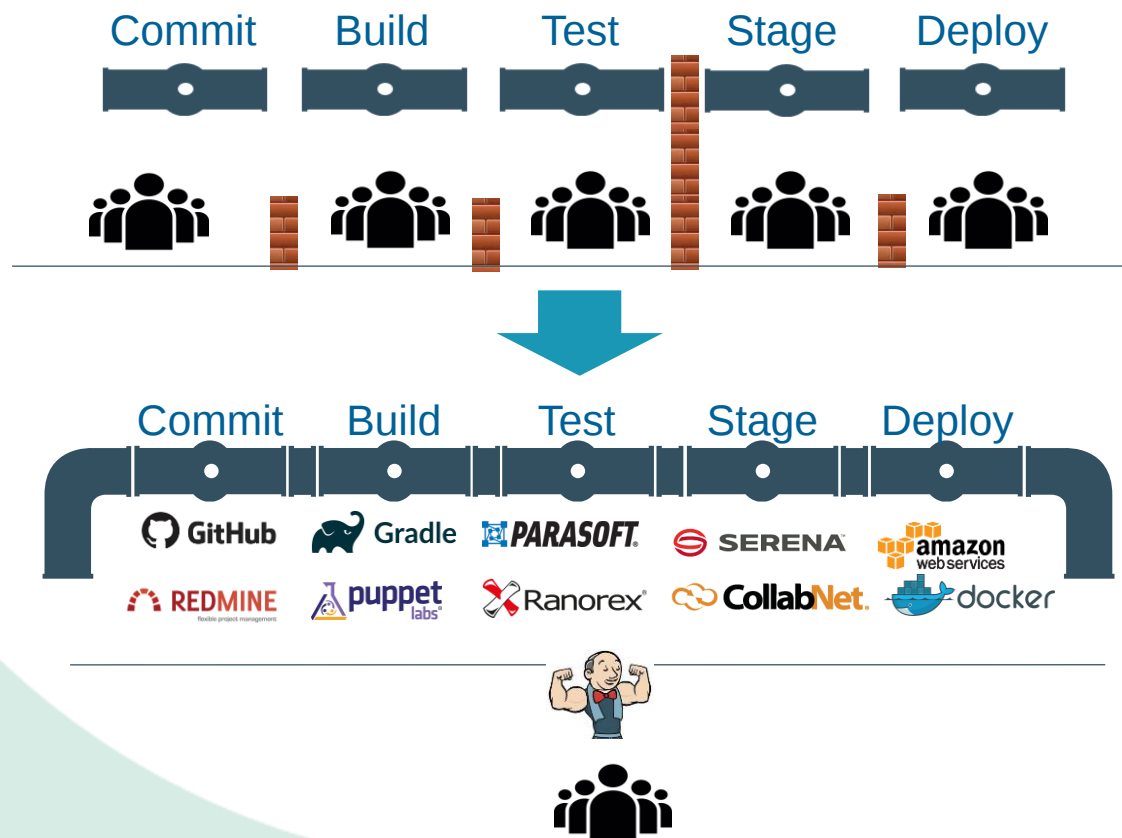
【導入以前】

- 手作業のプロセス
- 手作業による人的ミス
- ビルドからデプロイまでの主導による繰り返しタスク
- 時間の制約
- 部門間の開発速度の障壁
- 矛盾する目標
開発者：開発のスピード
オペレーション：安定性
- コミットからデプロイまでの透過性がない



Jenkins

開発者にとって
面倒な作業をしてくれる
執事



Jenkinsコミュニティ

Jenkinsには、コミュニティが大きいことから、プラグインも豊富にあり、1400以上のプラグインを利用することができます。利用者が多いため、日本でも勉強会が開かれていたり、日本語の情報がすぐに手に入るのも広く使われています。

170万

ユーザー以上の導入実績

1400

以上のプラグイン
あらゆる局面をサポート



コード・コミット

ビルド・構成管理

テスト・スキャン

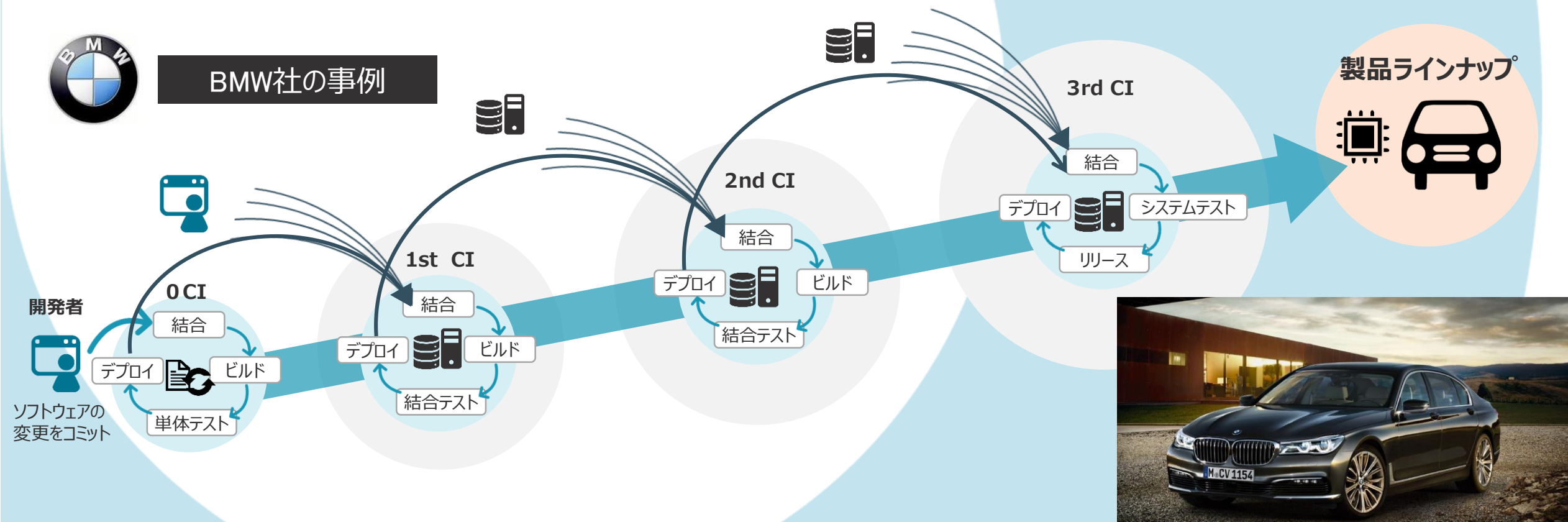
リリース

デプロイ



BMW社事例：段階的な継続的インテグレーションを実現

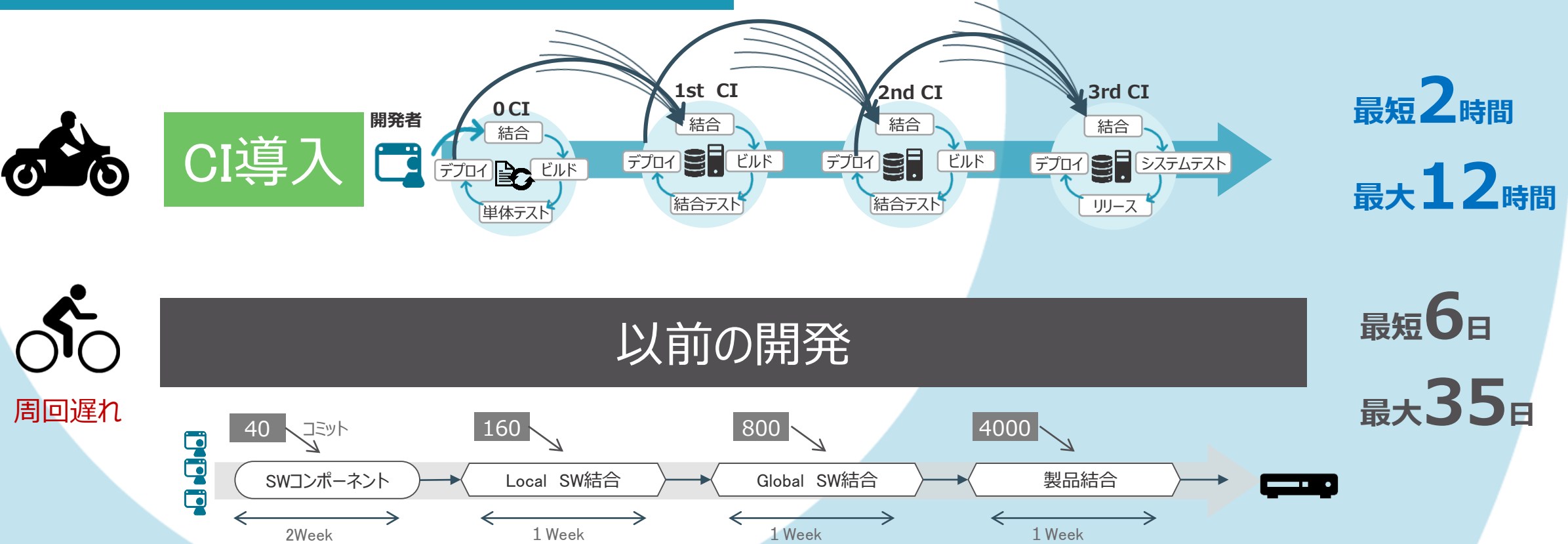
BMW社は、大規模のソフトウェア開発における段階的なCIシステムを導入している事例になります。BMW社の事例を実現するためには、コードがコミットされたら自動的にリリースまで行われるビルドパイプラインが整備されていることが必要になり、開発者の働き方の変更も必要になります。ですが、CI導入の基本としては、ビルドパイプラインは最初は小さな単位でビルドと検証を行いながら、フェーズが進みにつれて大きな単位での検証が行われるように設計します。そうすることで、エラーフィードバックが早くなり、影響範囲の小さなうちに修正されるようなプロセスとなります。



理想なソフトウェア開発では、製品のリリースを70倍速く

理想的なソフトウェア開発で、多段階CIシステムを導入した場合では、変更した内容が反映されるまでの時間に70倍の違いが生まれてきます。70倍の違いがある場合、目的地まで『6日』かかる乗り物と、目的時まで『2時間』で着く乗り物があるとしたら、どれだけの差が生まれるでしょうか？また、どれだけの費用対効果が得られるのでしょうか？

例：1行のソースコードの変更が最終的な製品に送り込むまでの時間



CI（継続的インテグレーション）のメリット

新しいビジネスに対して、スピード感のあるサービスの創出を目指すためには、システム運用や利用現場からのフィードバックを迅速に反映できる仕組みが必要です。CI導入により、これまでプロジェクトの最終段階で見つかったエラーが、早く小さなコストで修正されるようになるため、開発チームの生産性が向上し、品質の良いソフトウェアの開発に繋がります。

- 影響規模の小さいうちにバグを修正

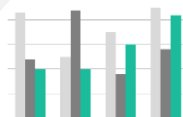
生産性の向上

- 安定版を確保する

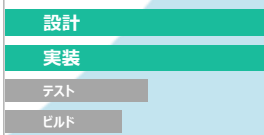


品質の良いソフトウェアの開発

- 同じ環境・手順・設定でテストする
- 知的な作業に集中する
 - 開発データを蓄積する
- 手作業の削減



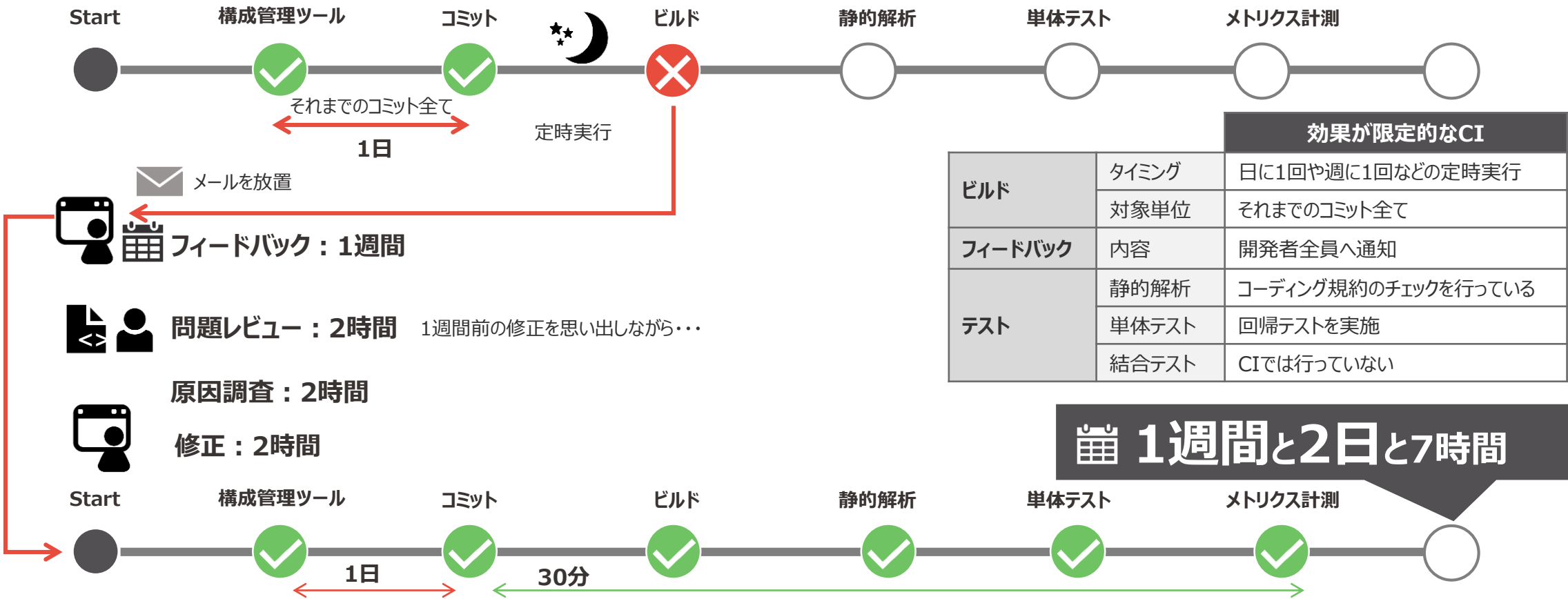
開発の競争力



効果が限定的なCIの問題

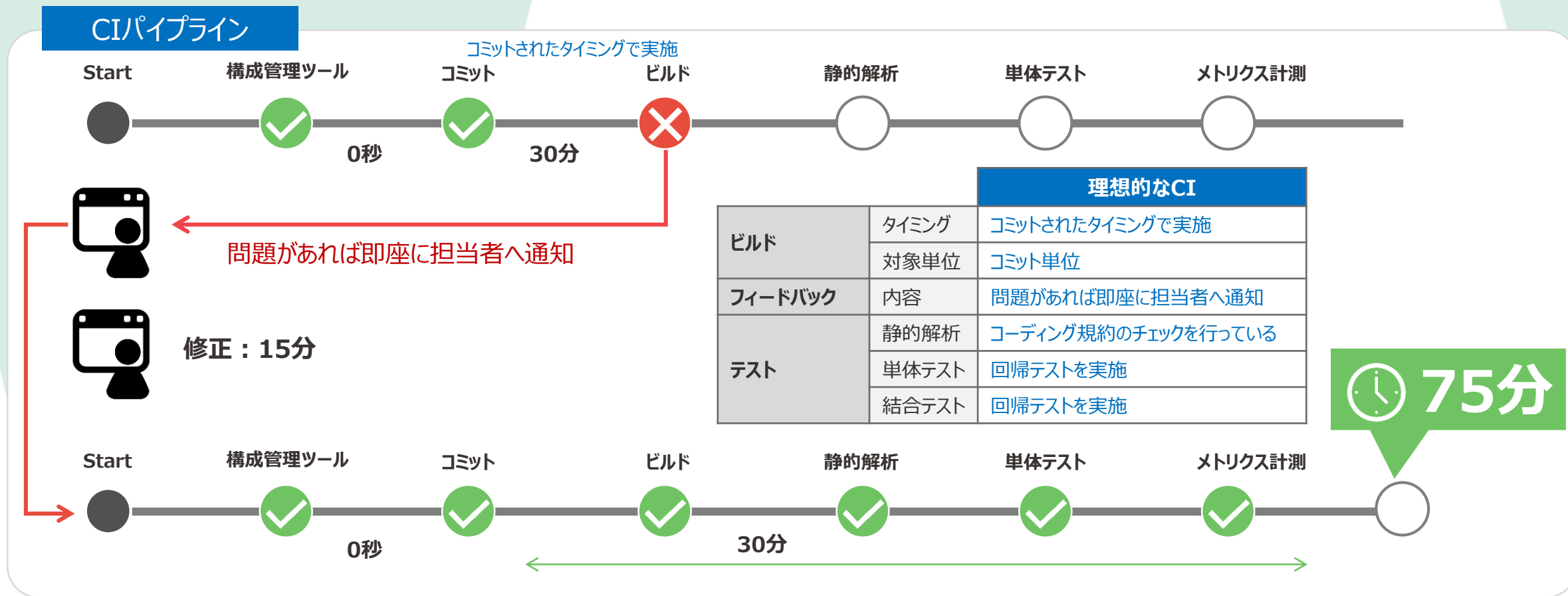
Jenkinsを使ってCIを行っているが、効果が限定的になってしまう問題があります。

効果が限定的なCI



理想的な開発プロセスへの「CIパイプライン」

コミットされた変更によりCIで行われる一連の流れをCIパイプラインと呼びます。Jenkinsを利用して、自動化されたビルドパイプラインの設計が重要となります。



JenkinsでCI/CDをはじめる前に

Jenkinsを利用したいと思っても、自分で環境構築をするには、時間も労力も必要です。

こんなお悩みないでしょうか？

環境構築の時間が
取れない

興味はあるが、
どうやって始めたらいい
かわからない

環境を作る
ノウハウがない

自分たちの開発環境
に合った環境を
作りたい

そこで、これからJenkinsをはじめる方に！

Jenkins担当者をアサインする余裕がない
簡単に、そして今すぐにJenkinsの環境がほしい！



CI環境構築サービス



自社の環境にあった
Jenkins環境を獲得！

Jenkinsの導入だけではなく、
単体テストや機能テストなどもCI/CDに
取り入れたい



TechMatrix Jenkins Platform Package



開発に必要な解析、テストを備えた
総合的なCI/CDソリューション



自社の環境にあったJenkins環境を獲得

CI環境構築サービス

Jenkinsを明日から始められる「CI環境構築サービス」

CI環境構築サービスは、「Certified CloudBees Jenkins Engineer」が貴社の環境に合わせて、Jenkinsのインストールからジョブの作成・保守を行います。

1

OSS Jenkins のインストール

- OSS Jenkinsのインストールおよび設定を行います。

2

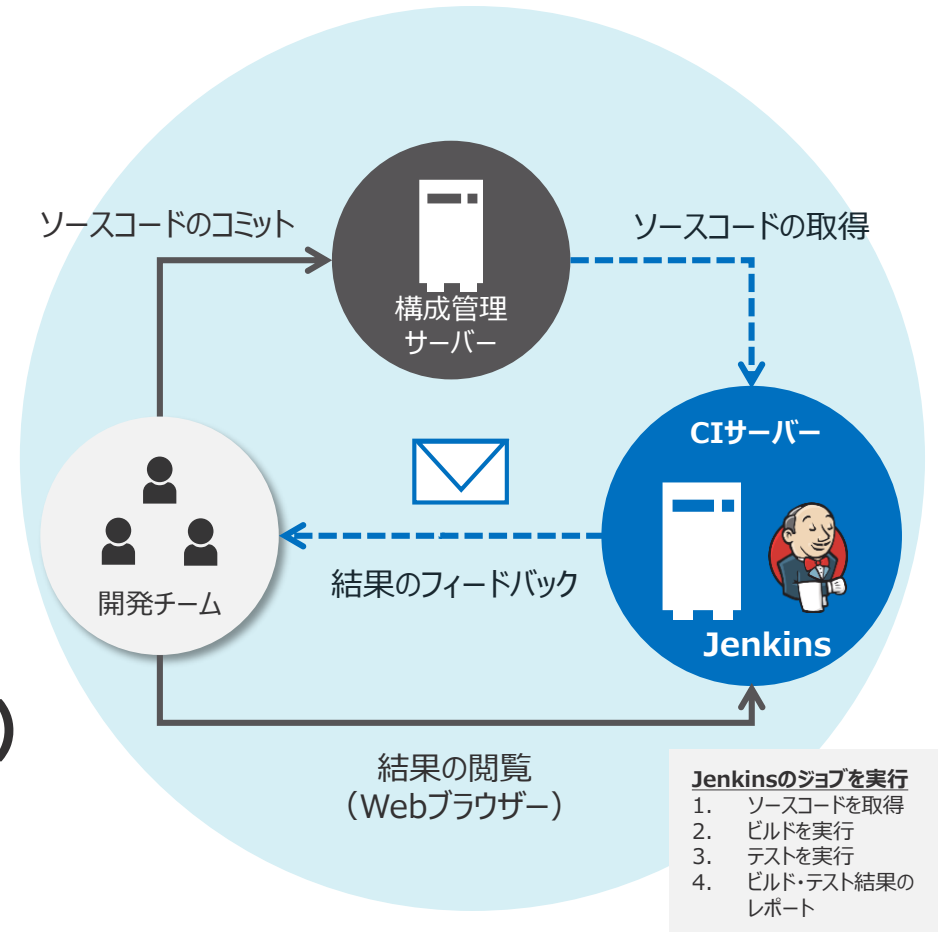
ジョブの作成

- プロジェクトにて実施しているビルドおよびテストを自動的に実行するジョブを作成します。

3

構築したCI環境の保守サポート(3ヶ月間)

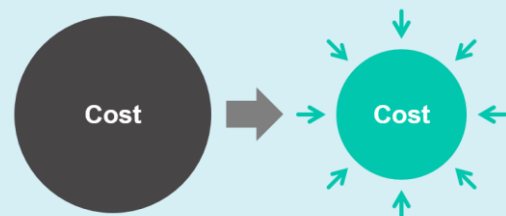
- 環境構築後、3ヶ月間の保守サービスを提供します。
- 構築した環境に関するお問い合わせをメールにて対応します。



「CI環境構築サービス」ご利用のメリット

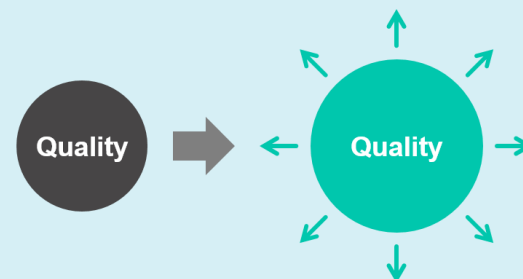
自動化の基盤を手に入れた後、現在の開発現場に大きな改善が期待できます。

- ・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)」
- ・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



開発工数の削減

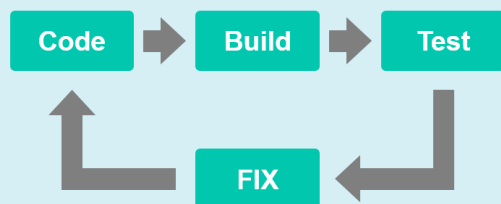
ソースコードの早期チェックにより、
バグ解決数の向上や、
コードレビューの工数が削減。



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。

これらのメリットが**1日**で手に入ります



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。



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、短時間で、開発現場に自動化
基盤を導入。



Jenkins のためのBest of Breed なCI/CD環境を

TechMatrix Jenkins Platform Package

Start

構成管理ツール

コミット

ビルド

静的解析

単体テスト

メトリクス計測



TechMatrix Jenkins Platform Packageとは？

TechMatrix Jenkins Platform Package は、お客様の現在のCI/CDへの取り組み状況、理想とされるCI/CDに合わせて、Jenkinsと各種ツールを組み合わせたパッケージ型のソリューションです。

構造分析

- アーキテクチャ分析ツール

Lattix

- ソースコード解析ツール

Understand

コード解析 ・単体テスト

- C/C++/Java/.NET対応 自動テストツール

Parasoft C++test

Parasoft Jtest

Parasoft dotTEST

- C/C++対応 自動エラー検出ツール

Parasoft Insure++

機能テスト

- UIテスト自動化ツール

Ranorex

- APIテスト自動化ツール

Parasoft SOAtest

- サービス仮想化ツール

Parasoft Virtualize

- Web・モバイル向け負荷テストプラットフォーム

NeoLoad

ソフトウェア開発基盤

- テスト管理プラットフォーム

Parasoft DTP

- CI/CDのための組織的な利用に最適化したJenkins

CloudBees Jenkins

- RedmineのPM/PL業務向け機能拡張プラグイン

Lychee Redmine

- 構成管理・変更管理ツール

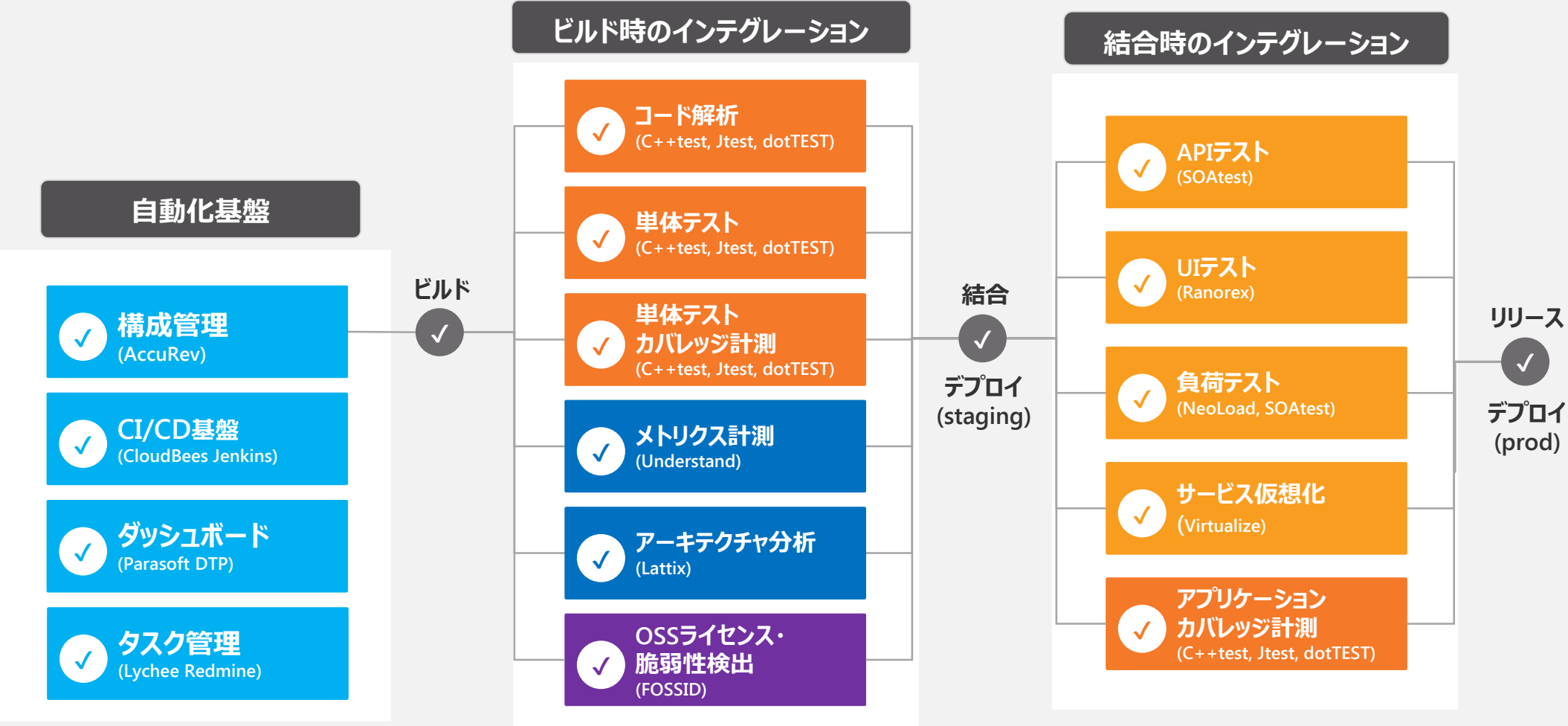
AccuRev

オープンソース ソフトウェア管理

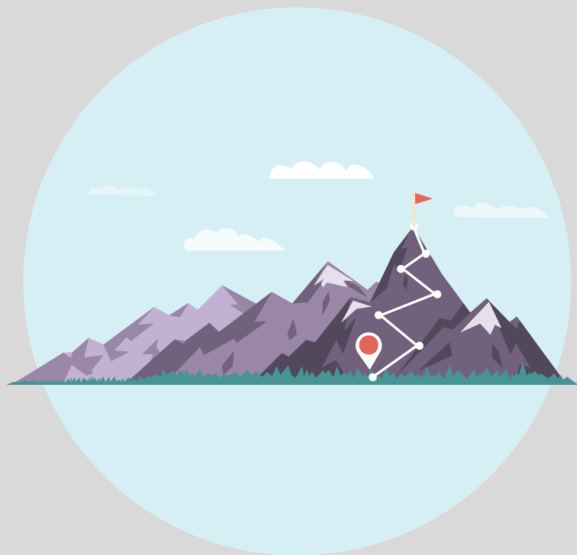
- 利用ライセンスの把握、脆弱性、OSSのリスクを可視化

FOSSID

ソリューションの適用箇所



※ 括弧内は製品名
※ ビルド時のソリューションは結合時にも適用が可能



パッケージご提案例

Start

構成管理ツール

コミット

ビルド

静的解析

単体テスト

結合テスト



パッケージその1:静的解析運用徹底パッケージ

こんな課題に有効

- Jenkinsをビルドの自動化くらいにしか使えていない
- 静的解析ツールのメリットはわかるが、解析を自動化しても違反の対応を開発者がちゃんとしてくれない
- 静的解析ツールの運用状況が見えない

ビルド



課題への解決策

Jenkinsをビルドの自動化くらいにしか使えていない

⇒静的解析、単体テストツールをCIパイプラインに加えることで、ソースコードの変更ごとに解析を自動実行
問題発生から修正までの時間を最短にします

静的解析ツールのメリットはわかるが、解析を自動化しても違反の対応を開発者がちゃんとしてくれない

⇒静的解析の違反をRedmineのチケットとして自動的に登録。違反やバグへの対応を厳密に管理

静的解析ツールの運用状況が見えない

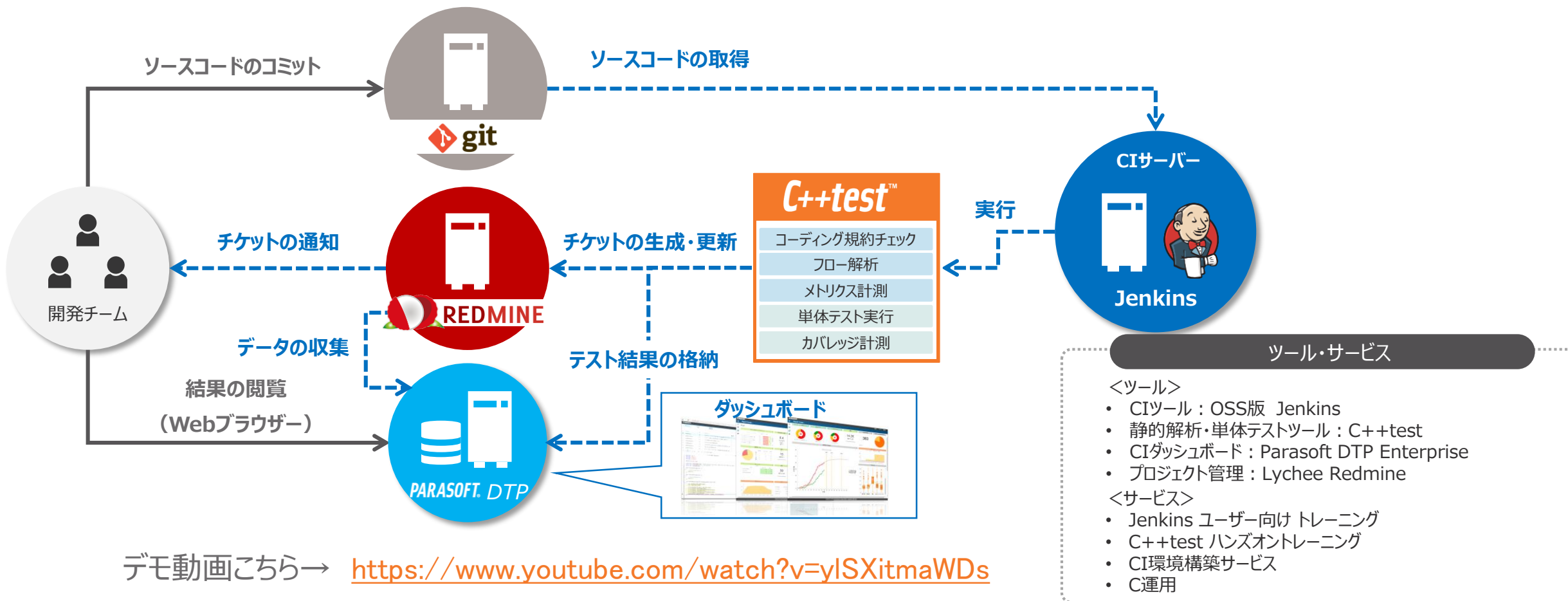
⇒ダッシュボード上でビルドや静的解析の状況をグラフィカルに可視化
チームメンバー、リーダ、マネージャー、品質保証担当者、経営層、など
それぞれの立場に合わせた粒度の情報をダッシュボード上に表示

開発状況が
把握できるよう
になった



パッケージその1: 静的解析運用徹底パッケージ

Git, Redmine, Jenkins, C++test(C/C++向け静的解析ツール), Parasoft DTP(ダッシュボード)を連携し、問題に対するワークフローの一部を自動化



パッケージその2:UIテスト・負荷テスト自動化パッケージ

こんな課題に有効

- OSSの利用における、意図しないライセンスの使用や、セキュリティ脆弱性をチェックしたい
- ステージング環境にデプロイしてからの機能テスト、負荷テストもCI/CDのパipelineに組み込みたい
- 品質に対するチェックの仕組みをまるごと導入したい

ビルド



課題への解決策

OSSの利用における、意図しないライセンスの使用や、セキュリティ脆弱性をチェックしたい

⇒いつの間にか混入してしまうこともあるOSSをチェックする体制をCIに組み込むことで、プロジェクトのポリシーを守り、セキュリティ対策も実現

ステージング環境にデプロイしてからの機能テスト、負荷テストもCI/CDのパipelineに組み込みたい

⇒機能テストや負荷テストを回帰的に自動実行することで、バグやパフォーマンスの劣化をすぐにキャッチ

品質に対するチェックの仕組みをまるごと導入したい

⇒お客様のご状況に合わせて、必要なソリューションを組み合わせでご提供可能
静的解析、単体テスト、OSSチェック、UIテスト、負荷テストなど、必要な技術をまとめてCIで実現

継続的な
品質保証体制が
すぐに手に入った！



パッケージその2:UIテスト・負荷テスト自動化パッケージ

静的解析、単体テストだけではなく、OSSのライセンス・セキュリティチェック、結合後のステージング環境へのデプロイから、UIテスト、負荷テストまでも自動化することができます。



ツール・サービス

<ツール>

- CIツール : OSS版 Jenkins
- 静的解析・単体テストツール : Jtest
- UIテストツール : Ranorex
- 負荷テスト・パフォーマンステストツール : NeoLoad
- OSSライセンス&セキュリティ管理ツール : FOSSID

<サービス>

- Jenkins ユーザー向け トレーニング
- CI環境構築サービス

Jenkins運用における悩み

～Jenkins Day Japan2018ご参加者様のアンケートより～



- 運用方法がメンバー全員に共有されていないために問題が発生している。
メンバーに共有しようにも複雑になっており簡単に共有ができない。
- 運用コストを減らしたくて、Masterの数を少なくしているが、最近ハングすることが多くなってきた。
- 細かなセキュリティの設定ができず、だれでも設定を編集できてしまう。
- ジョブの構成がプロジェクト間で統一できない。次のプロジェクトで改善すると、旧プロジェクトはそのまま、バラツキが解消できない。



有償版のJenkinsが問題を解決します！



組織的なCI/CD運用を実現する

CloudBees Core

CloudBees Coreの主な機能

- Jenkinsを拡張したCI/CDソリューション
- Jenkinsの管理のしやすさとセキュリティを拡張するためのエンタープライズ機能を提供
- 増え続けるJenkinsマスター／パイプラインの管理をサポート
- 安定版を毎月リリース
- Jenkinsのサポート窓口を提供



オペレーションセンター



共有エージェント



セキュリティ



ユーザー管理



高可用性



分析・解析



サポート

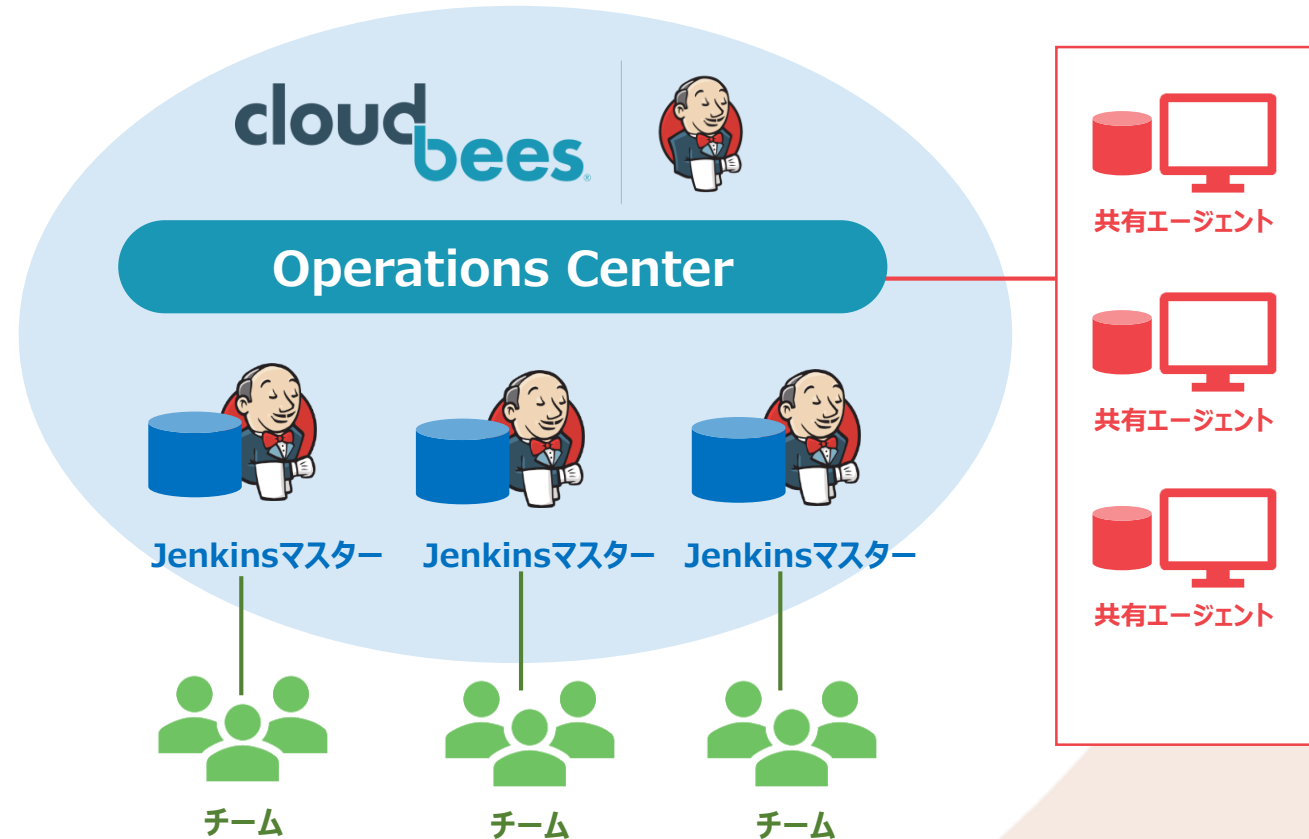


Operations Centerで組織的にJenkinsを運用

CloudBees Coreで組織的なJenkinsの運用を支援いたします。※ライセンスの形態は、別途ご説明いたします。

クラウド環境・オンプレミス環境で自動的にスケールアウトし、チームが増えても変わらない性能のプラットフォームとして組織的な継続的デリバリーを支援します。エラーでJenkinsがダウンしても自動的に復帰する機能もあり、安心してソフトウェア開発のビルドパイプラインとしてご利用になれます。

- Operations Centerで複数のJenkinsを管理
- ユーザーや権限を一括で管理
- 共有エージェントで資源を有効活用





Operations Centerで複数のJenkinsを管理

Operations Centerというマスターを管理するためのJenkinsがあり、煩雑になりがちな複数のJenkinsの管理を簡単に行うことができます。複数チームで1つのJenkinsを利用する必要がなく、チームごとに個別のJenkinsを提供できます。また、全てのJenkinsからエージェント(スレーブ)として利用できる共有エージェントの設定やマスターごとのプラグインアップデートセンターを独自に作成するなど、組織的にJenkinsを利用する際の管理を支援する機能が多くあります。

Operations Center



開発チーム

管理者

簡単に設定

Jenkinsの知識がなくても利用できるUI



Project Team



Project Team

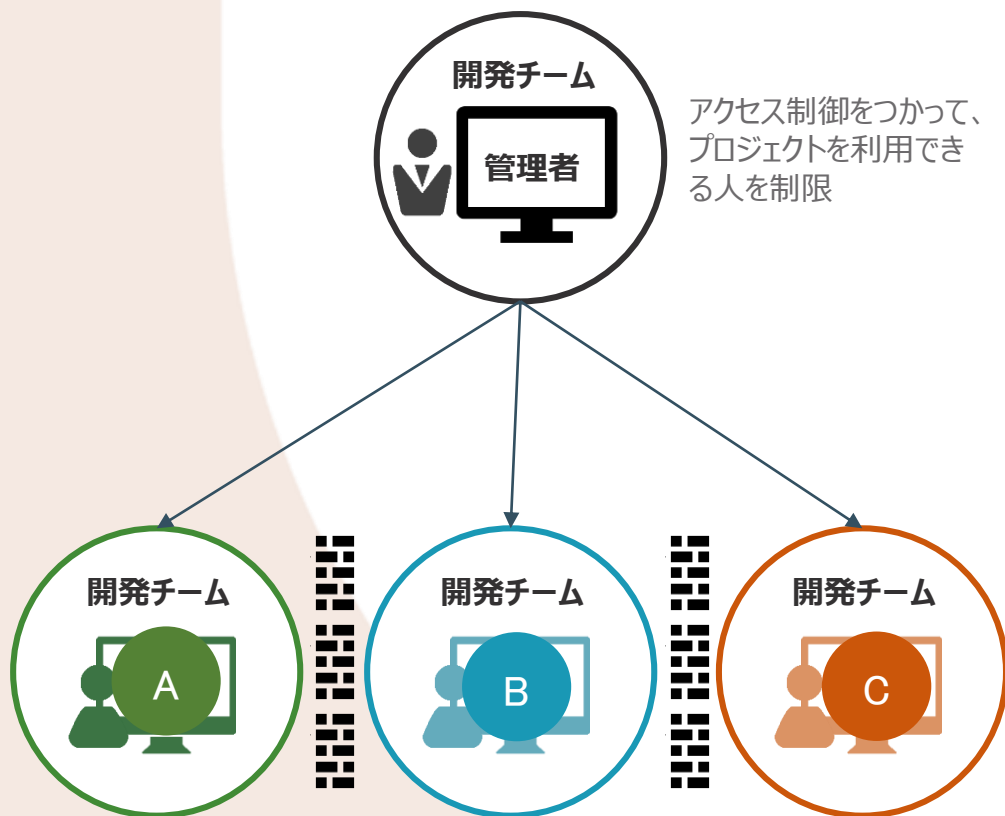


Project Team

ユーザーや権限を一括で管理：アクセス制御（RBAC）



ユーザーと権限の管理をOperations Centerで一括で行うことができます。それぞれのJenkinsでは一括で管理されているユーザーの権限でログインを行うことができ、Jenkinsを新しく起動するたびユーザーの設定などを行う必要がなくなります。マスターごとにもアクセス権限が設定できるので、プロジェクト単位で権限のあるユーザーしかアクセスできないJenkinsを立てることも可能です。





可用性の高いJenkins環境を提供

Jenkinsがダウンしたら自動で復帰する機能があり、ミッションクリティカルな処理でも安心してJenkinsを利用することができます。ロードバランスの仕組みもあり、多くのチームがCloudBees Coreを利用してもストレスのないCI/CDプロセスを提供します。

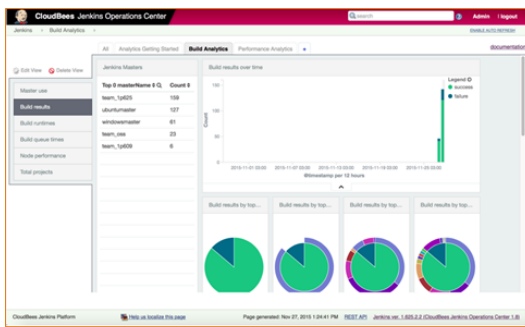
また、各Jenkinsのシステムやビルドのメトリクスを自動で収集してグラフ化して表示します。成功しているプロジェクトの確認やうまくJenkinsが使えていないプロジェクトへの横展開のために使うこともできます。また、システムの負荷も確認でき、リソースが逼迫しているようであれば必要なVMを追加するだけで問題を解消できます。

高い可用性



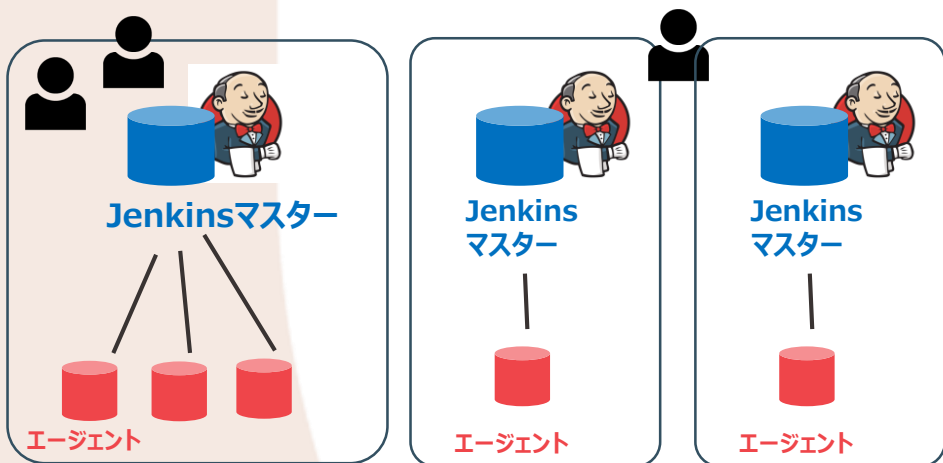
障害を検知
自動で復帰

モニタリングとアラート通知



CloudBees Coreで解決できること

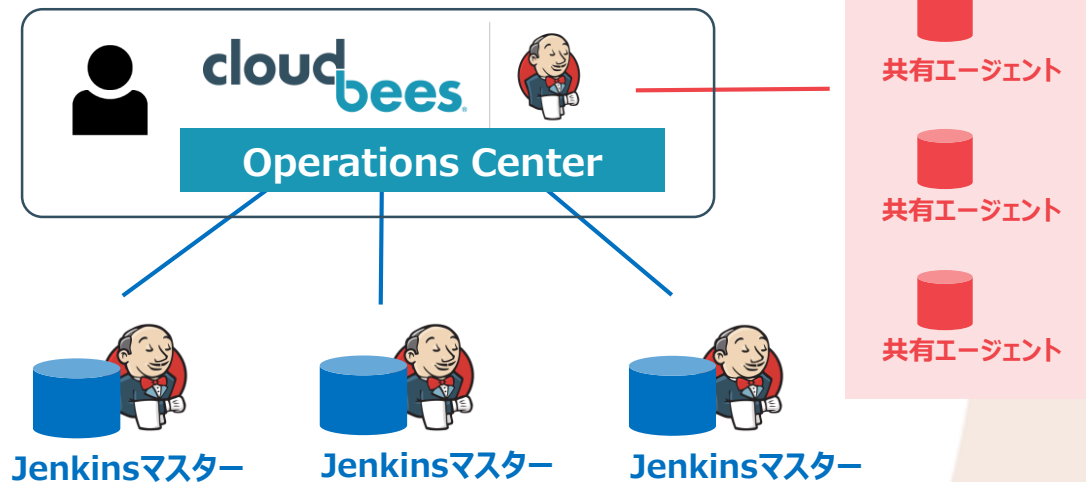
Jenkins OSS版での運用



管理者

- ◆ 各マスターを個別に管理しなければならない
- ◆ 各エージェントを個別に管理しなければならない

Cloudbees Coreでの運用



管理者

- ◆ ライセンス数の範囲内で、クラウド上に環境の構築が可能
- ◆ 「Operations Center」を経由し、ユーザー、アクセス権限、プラグイン、各マスターを一括管理可能
- ◆ バックアップの取得や、バージョンアップなど、様々な設定も一括で処理することができ、統制の取れた運用が可能
- ◆ 共有エージェント機能により、エージェントも一括で管理可能
- ◆ 「Operations Center」を介し、運用状況を細かくモニタリング可能

まとめ

世界と戦うための武器を持ちましょう！

- お客様のご状況に応じた、テストの自動化、Jenkins活用のためのご提案をさせていただきます。
- Jenkinsをこれから利用したい、Jenkinsをより活用したい、CI/CDを加速させたい、そんなお悩みがございましたら、ぜひお声がけください。



お問合せ



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
橘 祐史

TEL : 03-4405-7853
FAX : 03-6436-3553
Mail : ytachibana@techmatrix.co.jp
URL : <http://www.techmatrix.co.jp/solution/quality/index.html>

