

最新 IBM DevOps ソリューションのご紹介

2014年10月3日

日本アイ・ビー・エム株式会社
ソフトウェア事業本部 Rational 事業部

IT テクニカル・セールス
熱海 英樹

Agenda

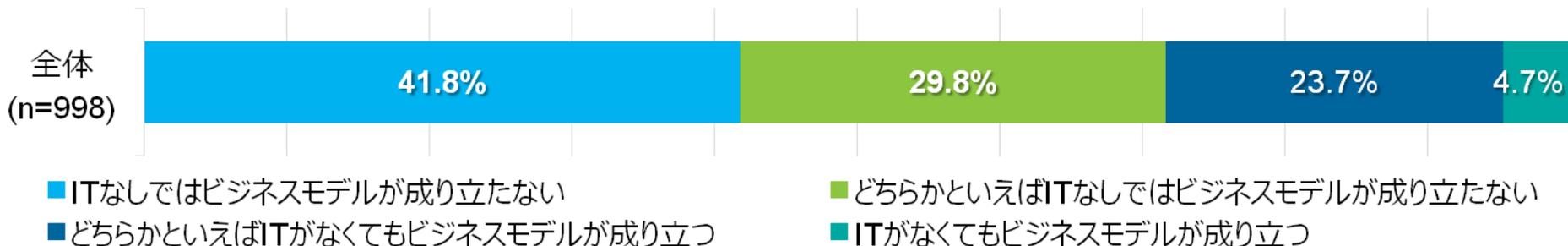
- DevOps とは何か？
- IBM DevOps の視点
- 特徴的なソリューションのご紹介
 - デプロイ自動化 — IBM UrbanCode Deploy
 - テスト仮想化 — Rational Test Virtualization Server

DevOps が求められる背景

■ 転換期に求められるIT部門の役割とは (調査テーマより抜粋)

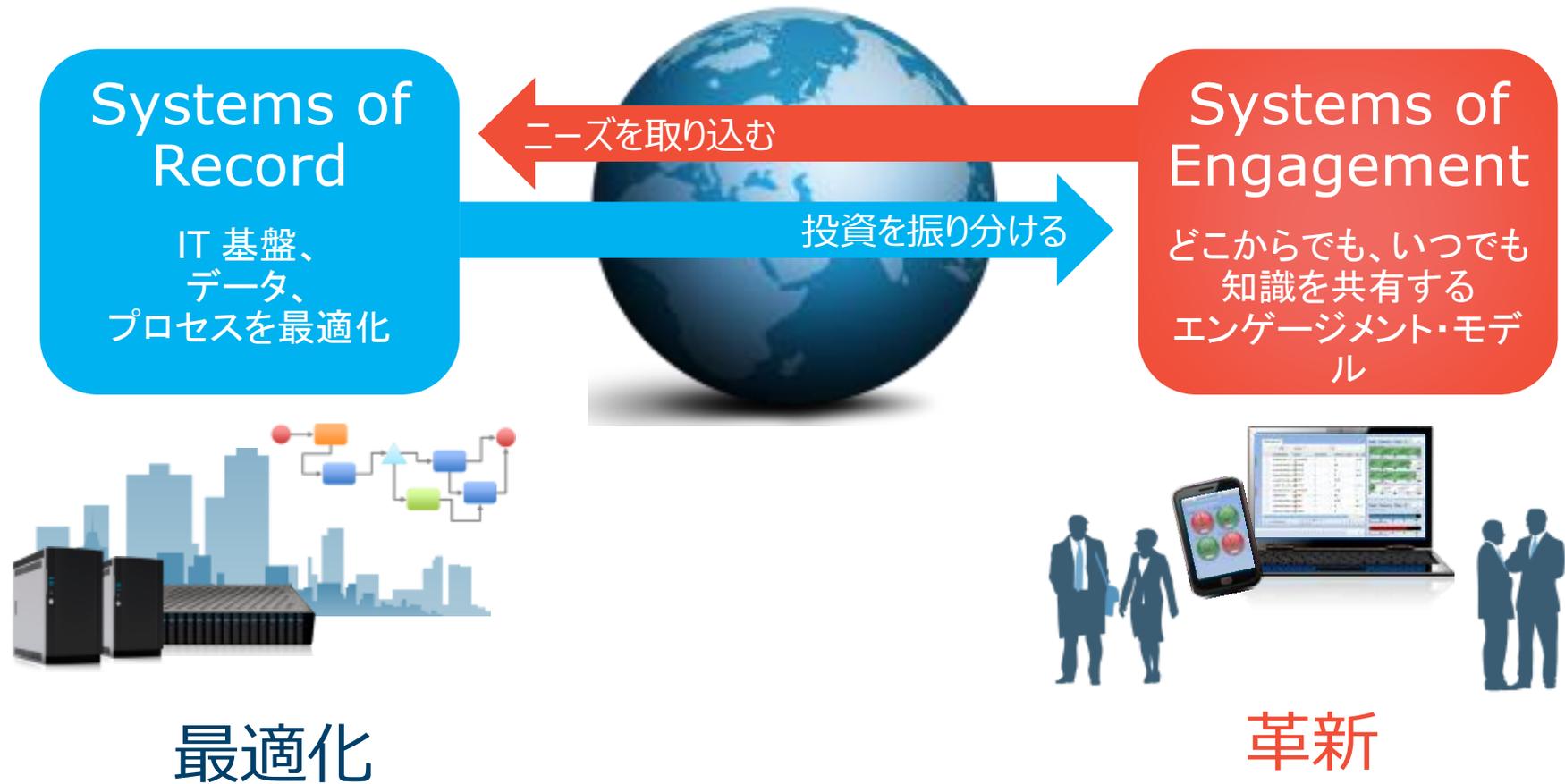
企業を取り巻く環境が激変しています。このような時代では、ユーザー企業の IT 部門は、従来に増して全方位的な役割が求められています。クラウドやビッグデータなど多様化するIT環境への対応、巧妙化するセキュリティ脅威への対策、さらには、ビジネスイノベーションを誘発するIT活用の提案といった経営に直結する役割も求められるようになってきます。当然のことながら、既存のシステムの保守・安定運用なども手を抜くことはできません。

■ 主たるビジネスモデルとITとの関係



出典：日本情報システム・ユーザー協会「企業IT動向調査 2014」より抜粋
<http://www.juas.or.jp/servey/it14/index.html>

革新と既存システムの最適なバランス



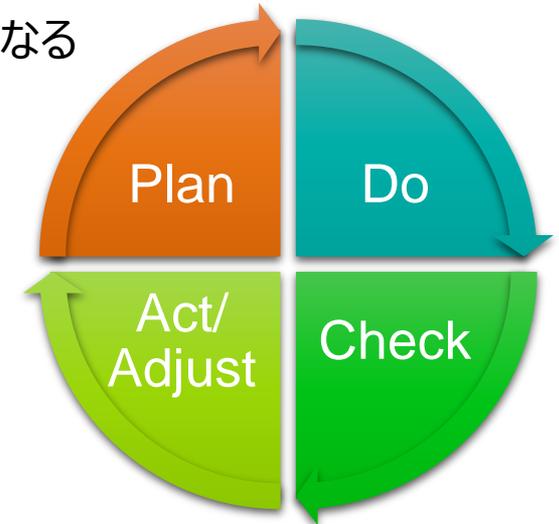
DevOps とは何か？

■ ソフトウェア・デリバリーの手法

- リーン開発およびアジャイル開発の原則に基づき、事業部門から、開発、品質保証、および運用に至るまでのすべての利害関係者が協力して、顧客（ユーザー）から実際に寄せられたフィードバックに基づいてソフトウェアを提供することを目的としています。

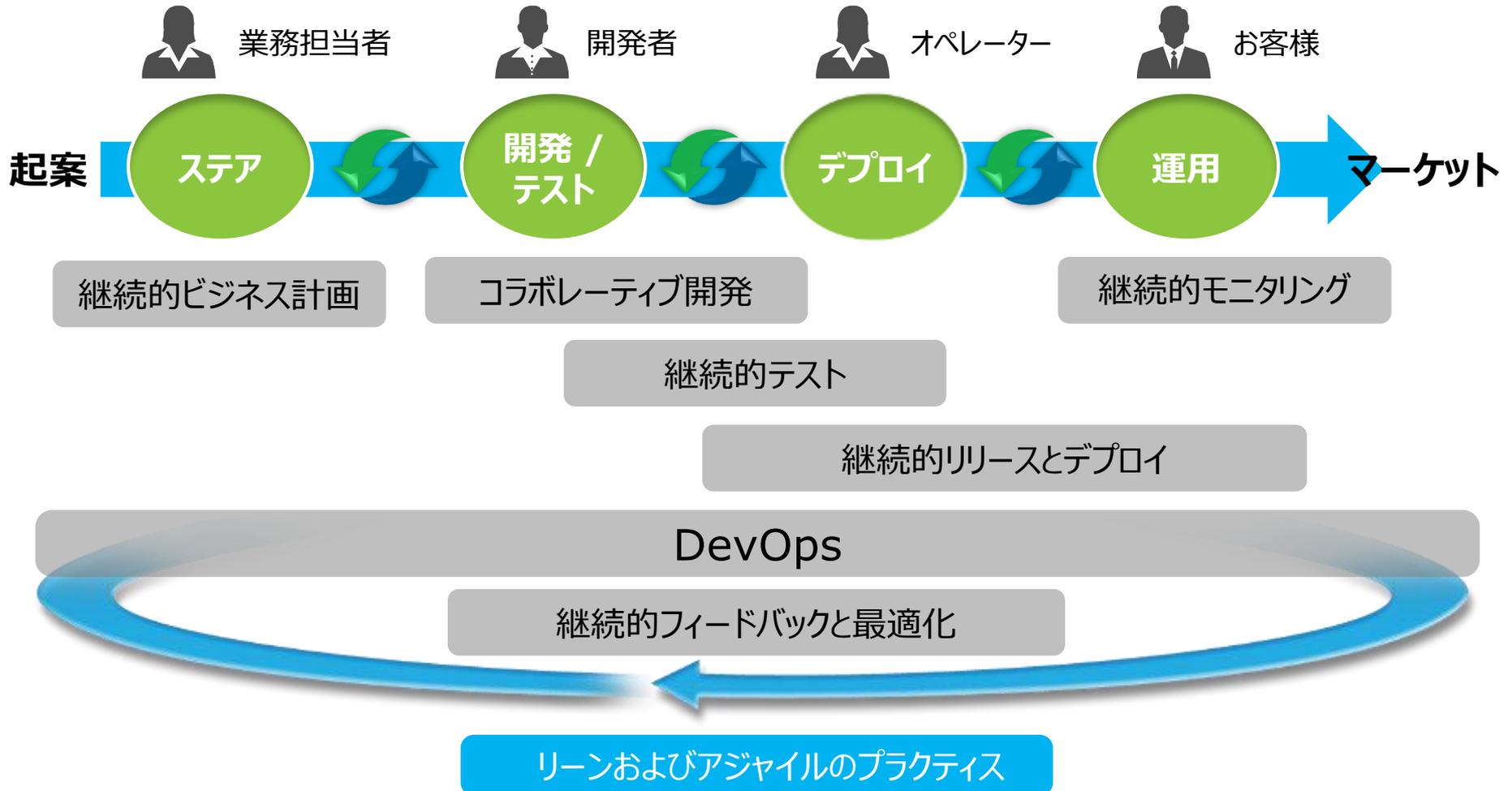
■ ビジネスを継続的に改善するサイクル

- DevOps を支える原則は、継続的な製造品質の改善を目的とした、計画 (Plan)、実行 (Do)、評価 (Check)、改善 (Act) または調整 (Adjust) からなる PDCA サイクルです。
- リーン生産方式のムーブメントは、このコアとなる手法を基に、生産する製品を継続的に改善し、その生産プロセスにおける無駄を常に減らしていくことを目指しています。



アプリケーション・デリバリーを反復的に成功させる

DevOps はリーンとアジャイルのプラクティスを拡張する



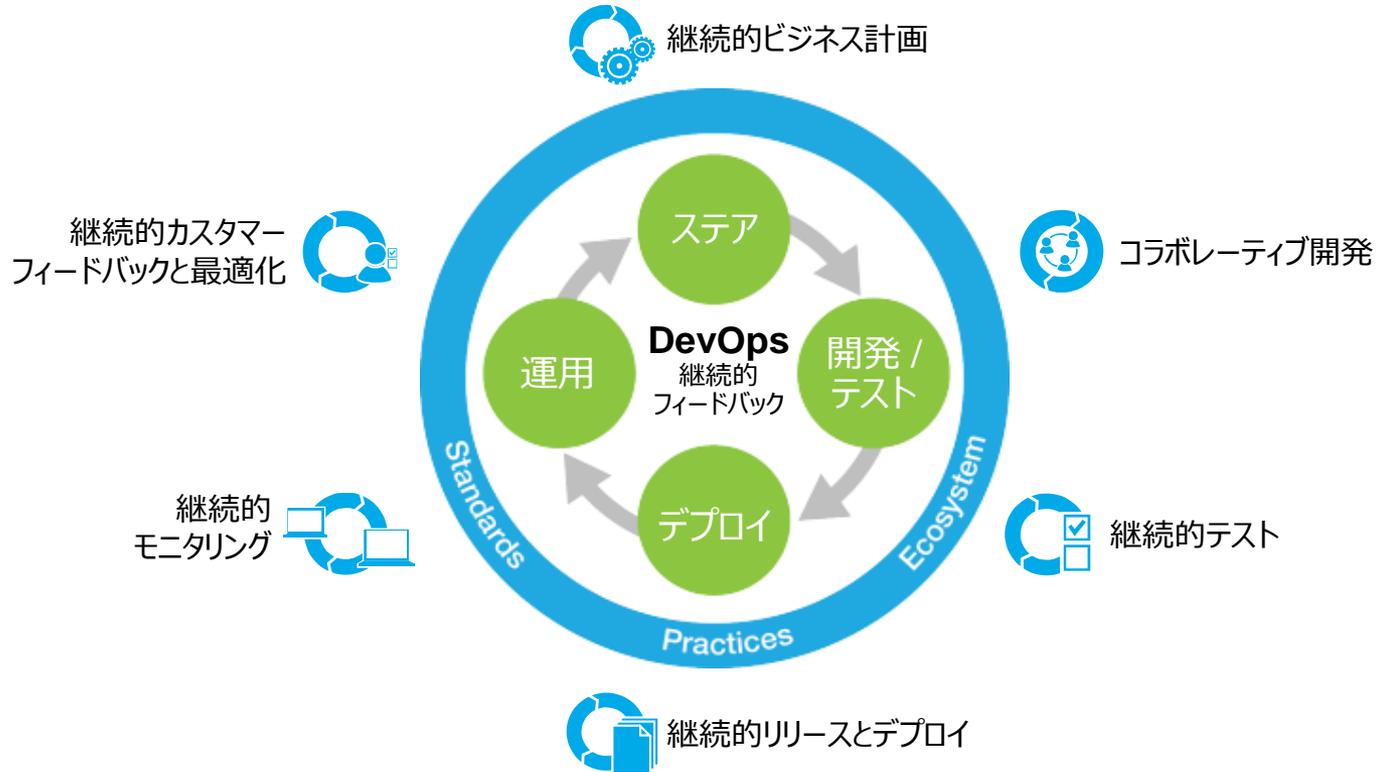
IBM DevOps の視点

“攻めの IT” を実現するソフトウェアを継続的にデリバリーするための企業の組織的能力

迅速に成果を得るために、ソフトウェアデリバリーを高速化する

革新する許容を増やすために、スピード/コスト/品質/リスクのバランスを調整する

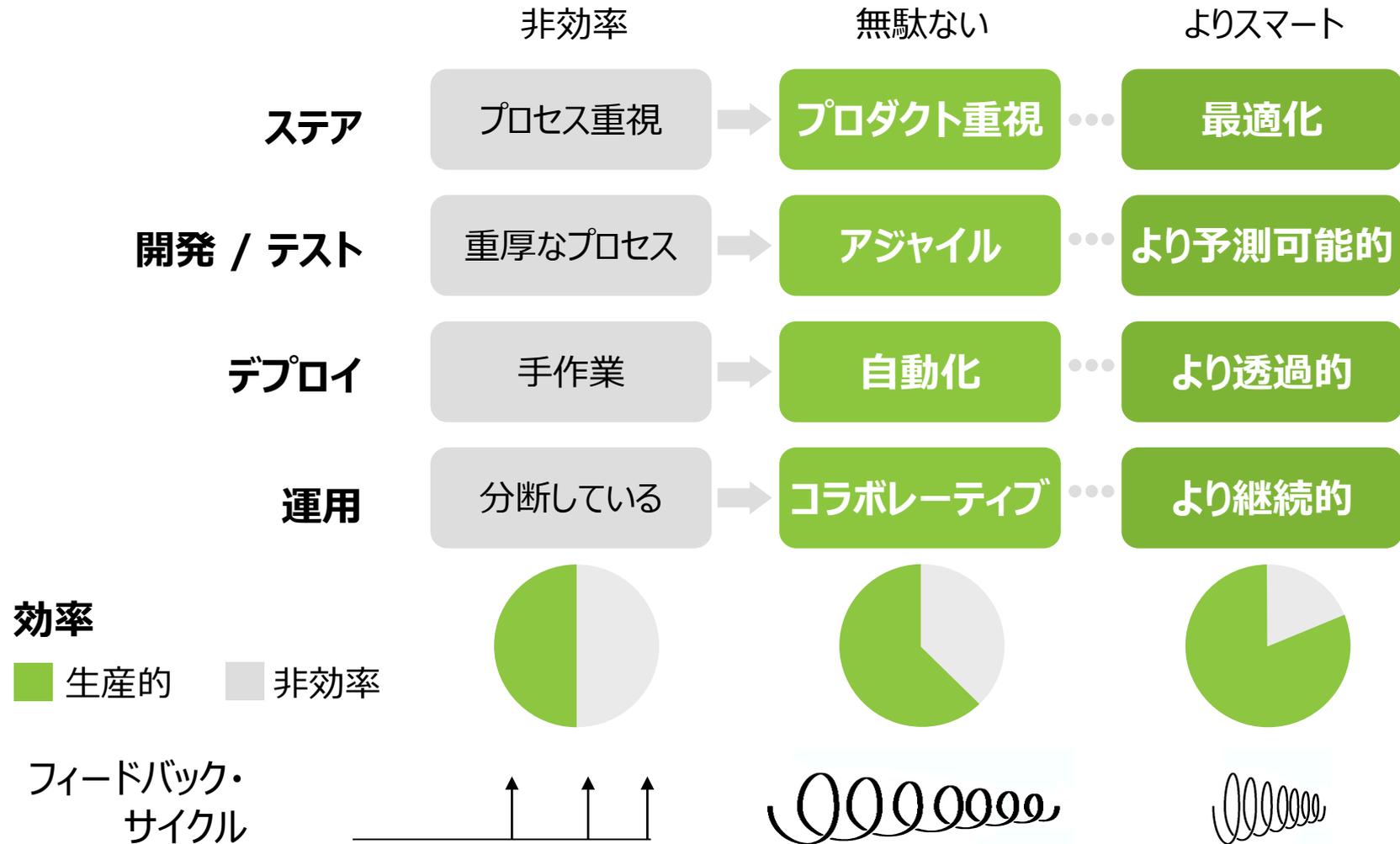
顧客体験を向上するために、フィードバック反映の時間を短縮する



多くのお客様が IBM DevOps の実践によりビジネスの成果を享受されています

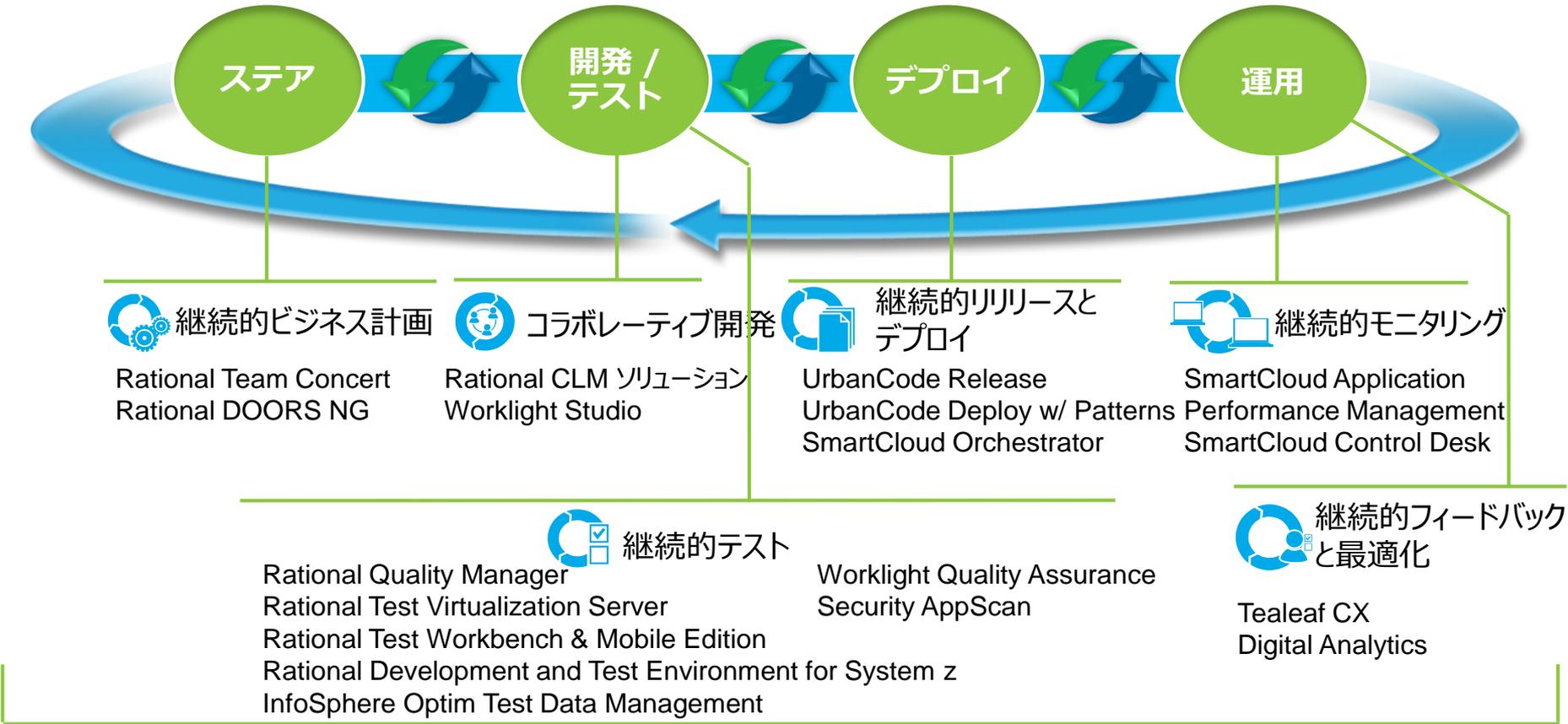
IBM DevOps 適用モデル

DevOps のプラクティス、ツール、サービスを段階的に適用する計画と実施



IBM DevOps – 幅広い領域をカバー

アプリケーション・デリバリー全般のボトルネックを解消



Bluemix DevOps services

| | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> Agile planning and tracking Application auto-scaling AppScan mobile analyzer Continuous delivery pipeline Git hosting | <ul style="list-style-type: none"> Mobile application security Mobile data Mobile quality assurance Monitor & analytics Push | <ul style="list-style-type: none"> RapidApps (beta) Server-side code Web IDE |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|

特長的なソリューションのご紹介

IBM UrbanCode Deploy

アプリケーション実行環境の構築自動化

デプロイの課題

開発から本番環境での運用までの距離

34% のプロジェクトでは本番運用に到達するまでの時間が掛かり過ぎている

41%

アプリケーションのテスト
環境構築に時間が
掛かり過ぎている

51%

十分なテストができず
本番で障害が入る

45%

本番環境での問題で
展開が遅れている

4-6 週

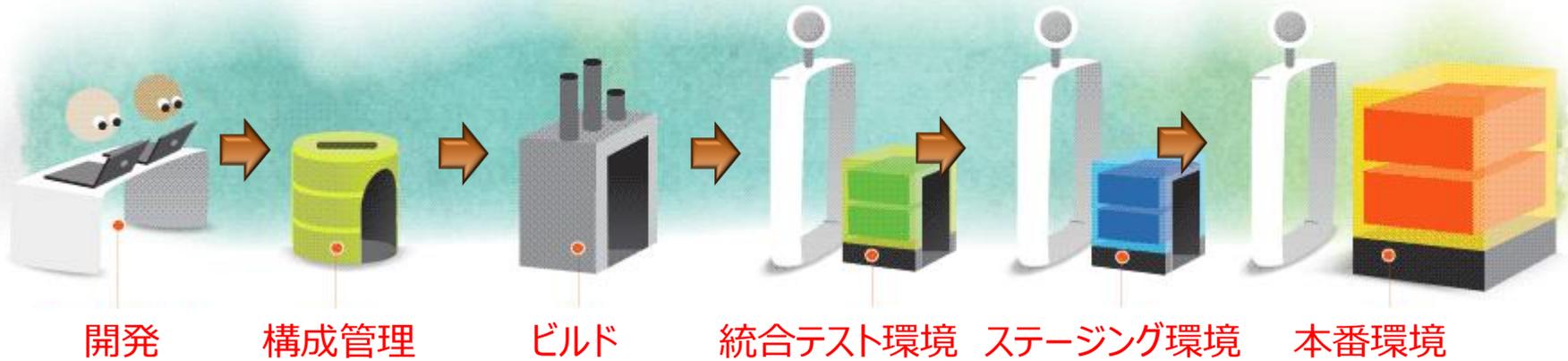
簡単な変更をした時に
本番展開するまでの
平均的な時間



出展：2011年にIBMが大手の調査会社に委託して行ったサーベイ結果

リリース自動化による短縮

アプリケーションのリリース・プロセス



環境の複雑性

人手による手作業

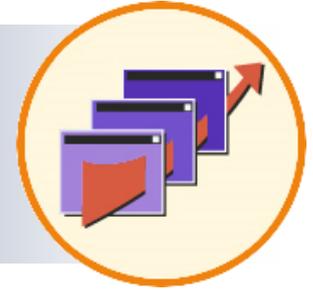
運用担当者の
作業負荷

UrbanCode : アプリケーションのリリース自動化による課題解決

UrbanCode の価値

1. 容易な定義

ノンプログラミングによる容易なプロセス定義



2. 品質の担保

品質ゲートによるプロセス制御

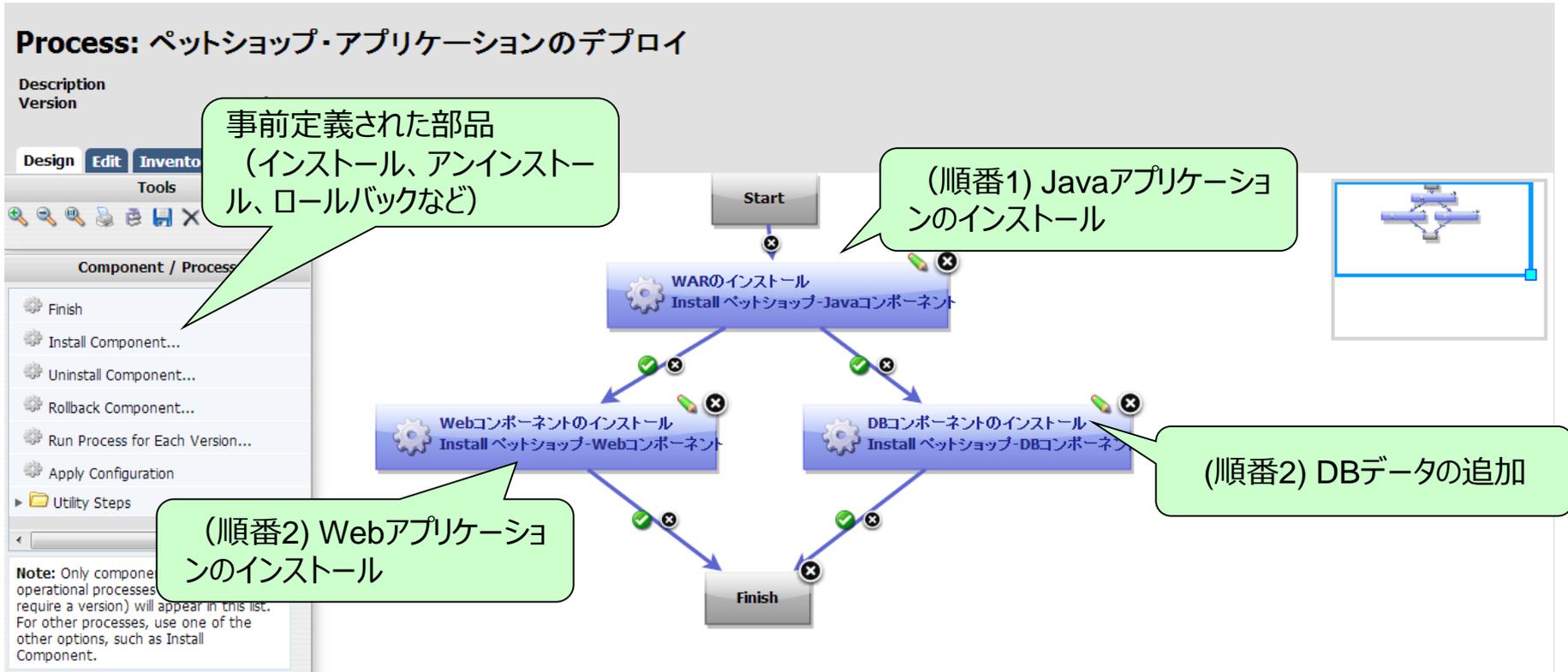


3. リリース管理負荷の軽減

計画・実行・実績の可視化



1. 容易な定義ーノンプログラミングによる容易なプロセス定義



- 事前定義されたテンプレートを利用することで、人手によるプロセス定義ミスを削減
- リリースプロセス構築、変更の容易性

2. 品質の担保 — 品質ゲートによるプロセス制御

自動化されたリリース処理が、“勝手に実行されない”ようにコントロール



- 基準を満たしていないアプリケーションのリリースを防止することで品質が担保されたアプリを常に本番環境にデプロイ可能とする

品質ゲートの設定

Application: ペットショップ・アプリケーション

Description

Environments Gates History Edit Properties Components Snapshots Processes Calendar Tasks Changes Security

Environments

| | |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 統合テスト環境 | コンパイルエラー無し <input checked="" type="checkbox"/> ビルドエラー無し <input checked="" type="checkbox"/> 単体テスト完了 <input checked="" type="checkbox"/> And... <input type="button" value="▼"/> - or - Add a new condition... <input type="button" value="▼"/> |
| ステージング環境 | 統合テスト完了 <input checked="" type="checkbox"/> And... <input type="button" value="▼"/> - or - Add a new condition... <input type="button" value="▼"/> |
| 本番環境 | 疎通テスト完了 <input checked="" type="checkbox"/> And... <input type="button" value="▼"/> - or - Add a new condition... <input type="button" value="▼"/> |

Save Conditions

各環境ごとに設定された品質ゲート

各環境のアプリケーションのリリース状況の確認

Application: ペットショップ・アプリケーション

Description

Environments Gates History Edit Properties Components Snapshots Processes

Environments

Drag environments by their label boxes to order them.

各環境に、どのアプリのどのバージョンがリリースされているか

統合テスト環境

Actions

Request Process
Compare Copy
Inactivate Delete

Latest Desired Inventory

| Component | Version | Snapshot | Properties | Status | Complancy | Actions |
|------------------------------|---------|----------|------------|--------|-----------------|------------------------------|
| Show Filters | | | | | | |
| ペットショップ-Webコンポーネント | 1.1 | リリース1.1 | Version 1 | Active | Compliant (1/1) | View Request |
| ペットショップ-Javaコンポーネント | 1.0 | リリース1.1 | Version 1 | Active | Compliant (1/1) | View Request |
| ペットショップ-DBコンポーネント | 1.1 | リリース1.1 | Version 1 | Active | Compliant (1/1) | View Request |

10 per page 3 records - [Refresh](#) [Print](#) << 1 / 1 >>

ステージング環境

Actions

Request Process
Compare Copy
Inactivate Delete

Latest Desired Inventory

| Component | Version | Snapshot | Properties | Status | Complancy | Actions |
|------------------------------|---------|----------|------------|--------|-----------------|------------------------------|
| Show Filters | | | | | | |
| ペットショップ-Webコンポーネント | 1.1 | リリース1.1 | Version 1 | Active | Compliant (1/1) | View Request |
| ペットショップ-Javaコンポーネント | 1.0 | リリース1.1 | Version 1 | Active | Compliant (1/1) | View Request |
| ペットショップ-DBコンポーネント | 1.1 | リリース1.1 | Version 1 | Active | Compliant (1/1) | View Request |

10 per page 3 records - [Refresh](#) [Print](#) << 1 / 1 >>

本番環境

Actions

Request Process
Compare Copy
Inactivate Delete

Latest Desired Inventory

| Component | Version | Snapshot | Properties | Status | Complancy | Actions |
|----------------------------------------------------------------------------------|---------|----------|------------|--------|-----------|---------|
| Show Filters | | | | | | |
| No components have been installed to this environment. - Refresh | | | | | | |

3. 管理負荷の軽減—計画・実行・実績の可視化

The screenshots illustrate the uRelease and uDeploy tool's capabilities in managing and visualizing deployment processes. The uRelease interface shows a deployment plan that is complete, with a timer and progress indicators. The uDeploy interface provides detailed logs for a specific component deployment, listing steps such as artifact download, WAR expansion, and application deployment. The Reports section includes a line chart showing deployment counts across different environments over time, allowing for trend analysis and performance monitoring.

- 各環境にデプロイされた証跡をExcel等で残しているケースが多いが、情報が自動的に収集されるため、管理負荷を軽減可能
- リアルタイムで正確な情報が常に得られるため、問題分析時に非常に有効

リリース状況のリアルタイムな確認

～ リリース・ステップの詳細 ～

uDeploy Hello admin | Preferences | REST API Doc | Help | Logout

Dashboard Components Applications Configuration Processes Resources Calendar Work Items Reports Settings

Home > Components > ペットショップ-Javaコンポーネント > Process Request on udpot-sit

Deployment of Component: ペットショップ-Javaコンポーネント

Process: [Javaコンポーネントのデプロイ \(Version 3\)](#)
 Version: [1.0](#)
 Resource: [udpot-sit](#)
 Agent: [udpot-sit](#)
 Date: 07/10/2013 05:13 午前
 Requested By: admin [View Appli...](#)

Log **Properties**

Sort By: [Graph Order](#) [Start Time](#)

自動化された一連の
リリースプロセス

リリース結果
(リアルタイム更新)

| Step | Type | Start | Duration | Status | Actions |
|------------------------|---------------------------------------------------------------|----------------|----------------|----------------|---------|
| Workディレクトリのクリーン | Shell v. 3.260933 | 5:13:50 | 0:00:04 | Success | |
| Artifactsのダウンロード | UrbanCode Versioned File Storage v. 16.369060 | 5:13:54 | 0:00:04 | Success | |
| WARの展開 | File Utils v. 27.368911 | 5:13:58 | 0:00:04 | Success | |
| プロパティファイルの書き換え | File Utils v. 27.368911 | 5:14:02 | 0:00:04 | Success | |
| WARのアップデート | File Utils v. 27.368911 | 5:14:06 | 0:00:04 | Success | |
| アプリケーションのアンデプロイ | Tomcat v. 3.324493 | 5:14:15 | 0:00:04 | Success | |
| アプリケーションのデプロイ | Tomcat v. 3.324493 | 5:14:19 | 0:00:04 | Success | |
| サーバーの起動 | Tomcat v. 3.324493 | 5:14:11 | 0:00:03 | Success | |
| Total Execution | | 5:13:50 | 0:00:34 | Success | |

リリース実績のレポート

Reports ?

Deployment

Deployment Count

Deployment Total Duration

Deployment Average Duration

Deployment Detail

Security

My Reports

Date Range

Current Month ▼

2013/7/1

2013/7/31

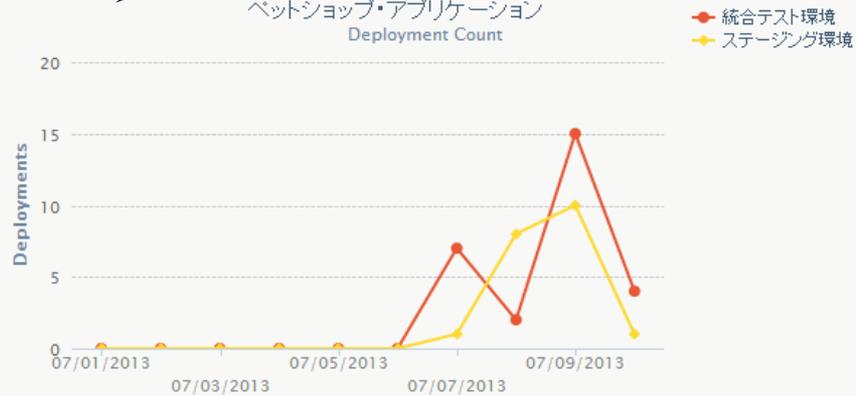
Save

過去のリリース実績の
統計レポート

Deployment ▶ Deployment Count



ペットショップ・アプリケーション
Deployment Count



| Application | Environment | 07/10/2013 | 07/09/2013 | 07/08/2013 | 07/07/2013 | 07/06/2013 | 07/05/2013 | 07/04/2013 | 07/03/2013 | 07/02/2013 | 07/01/2013 |
|------------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| ペットショップ・アプリケーション | 統合テスト環境 | 4 | 15 | 2 | 7 | 0 | 0 | 0 | 0 | 0 | 0 |
| ペットショップ・アプリケーション | ステージング環境 | 1 | 10 | 8 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

UrbanCodeの事例

International Investment Firm Driving Down Costs



リリースプロセスは相当の努力が必要であり、**手作業によるエラーでリリースが遅延**していた。

- ソリューション：**リリースプロセスの自動化**
- 結果：年間\$2.3Mのコストを削減でき、**2-3日**かかっていたリリース時間は、**1-2時間**に短縮でき、事実上、テストチームの“停止時間”を無くすことができた。

Online Retailer Speeding Time-to-Market



アプリケーションの変更を本番環境に適用する際に、**致命的な遅延**が発生していた。

- ソリューション：**継続的デプロイメントの拡大**
- 結果：容易にデプロイメントを拡大できることで、**デプロイメント時間を95%以上削減**できた。また、2ヶ月の実装の中で、250サーバーへのデプロイメントを実現した。

Higher Education Speeding Time-to-Market



開発、およびテスト環境への**デプロイメントが遅い**ことに、アジャイル開発チームは苦悩していた。

- ソリューション：**自動化されたデプロイ作業**を開発チームが、**自身でデプロイできるように**したことで、デプロイメント作業を効率化
- 結果：**数時間**から、**数分**へデプロイメント時間を短縮でき、これまで以上のサーバー数に対し、より少ないリソースでデプロイを実現した。

SaaS Software Provider Reducing Risk



複数のお客様構成や、**ソフトウェアのバージョンの管理が困難**であった。

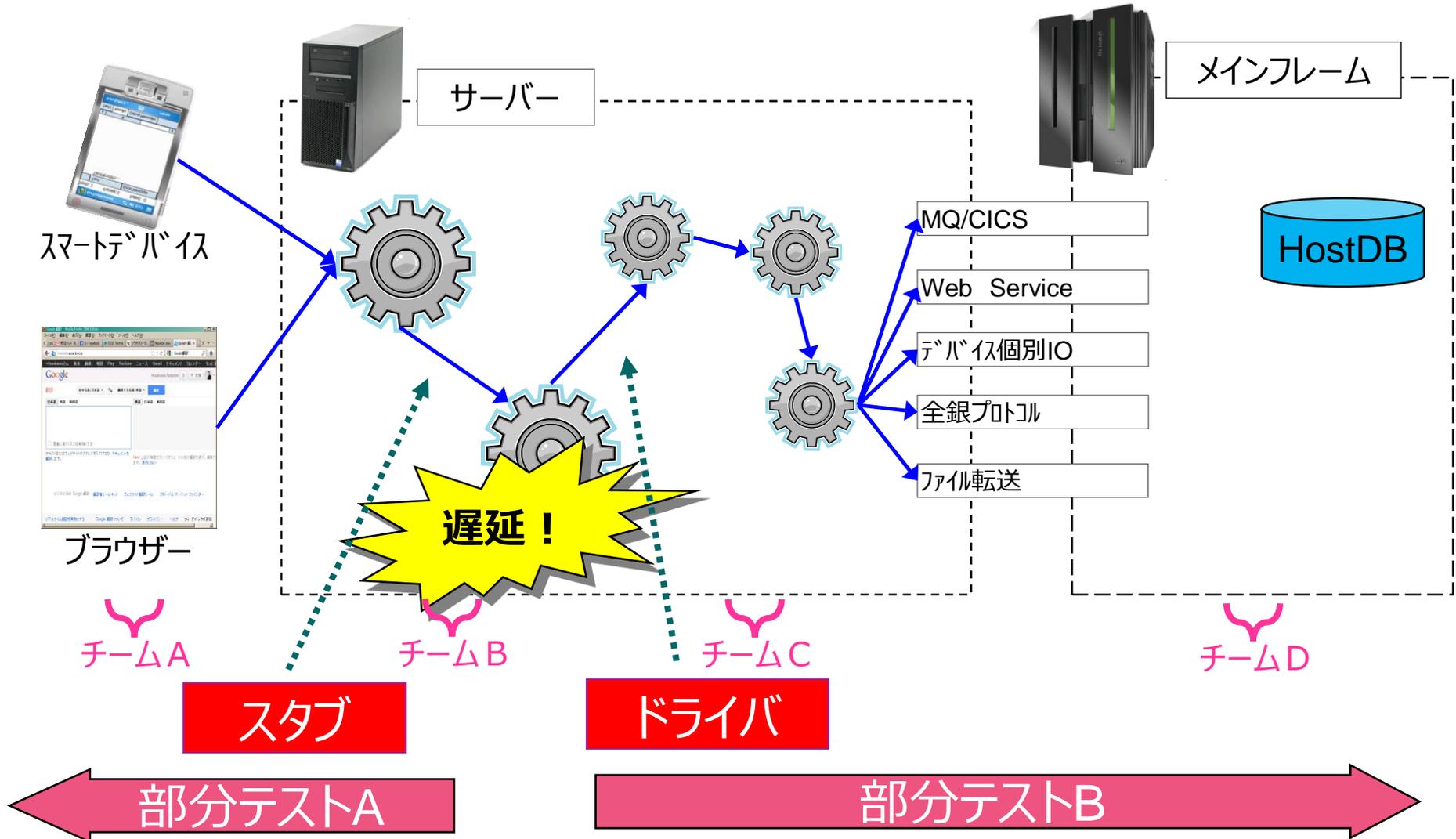
- ソリューション：**デプロイメントの構成とバージョンを自動的に管理**
- 結果：お客様の**特定のリリースを実行できる**ようになり、デプロイメントによる**停止時間を90%以上、削減**できた。

IBM Rational Test Virtualization Server

接続先を仮想サービス化して通信を伴う統合テストを前倒し実施

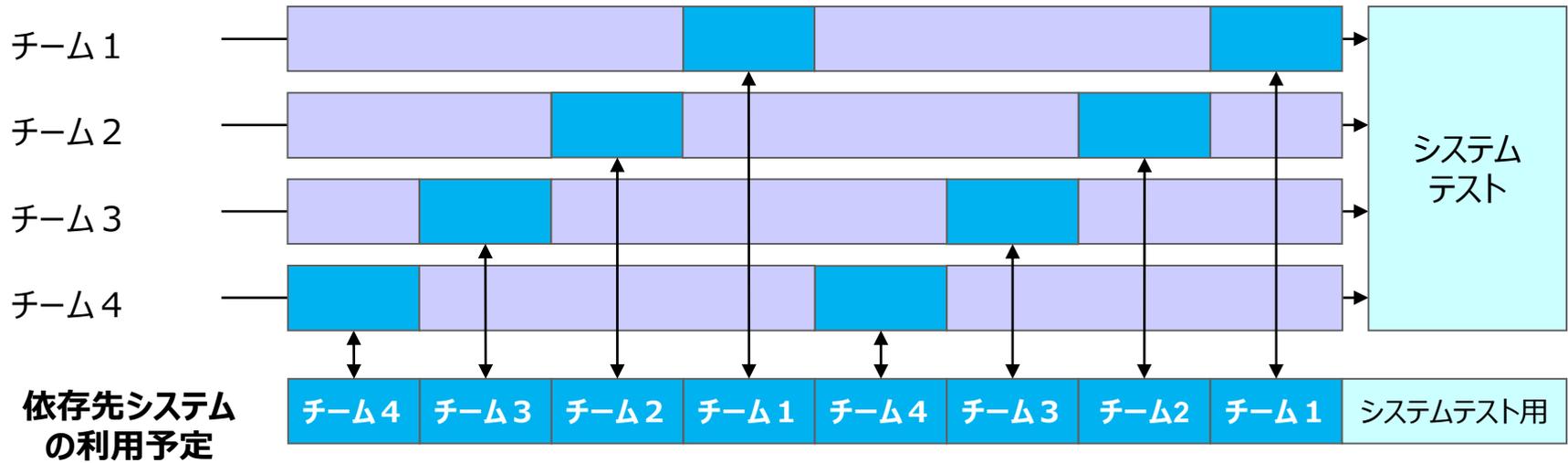
新たなテストツールの必要性： インターフェーステストの存在

他チームとの依存性を減少させ、他チームの進捗に引きずられないテストを実現



テストの効率化に関する外部接続システムの課題

複数のプロジェクトが一つのシステムをシェアしてテスト

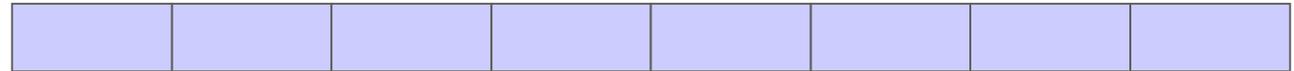


テスト時のシステム利用調整
にとっても苦労している。

テストできた時間



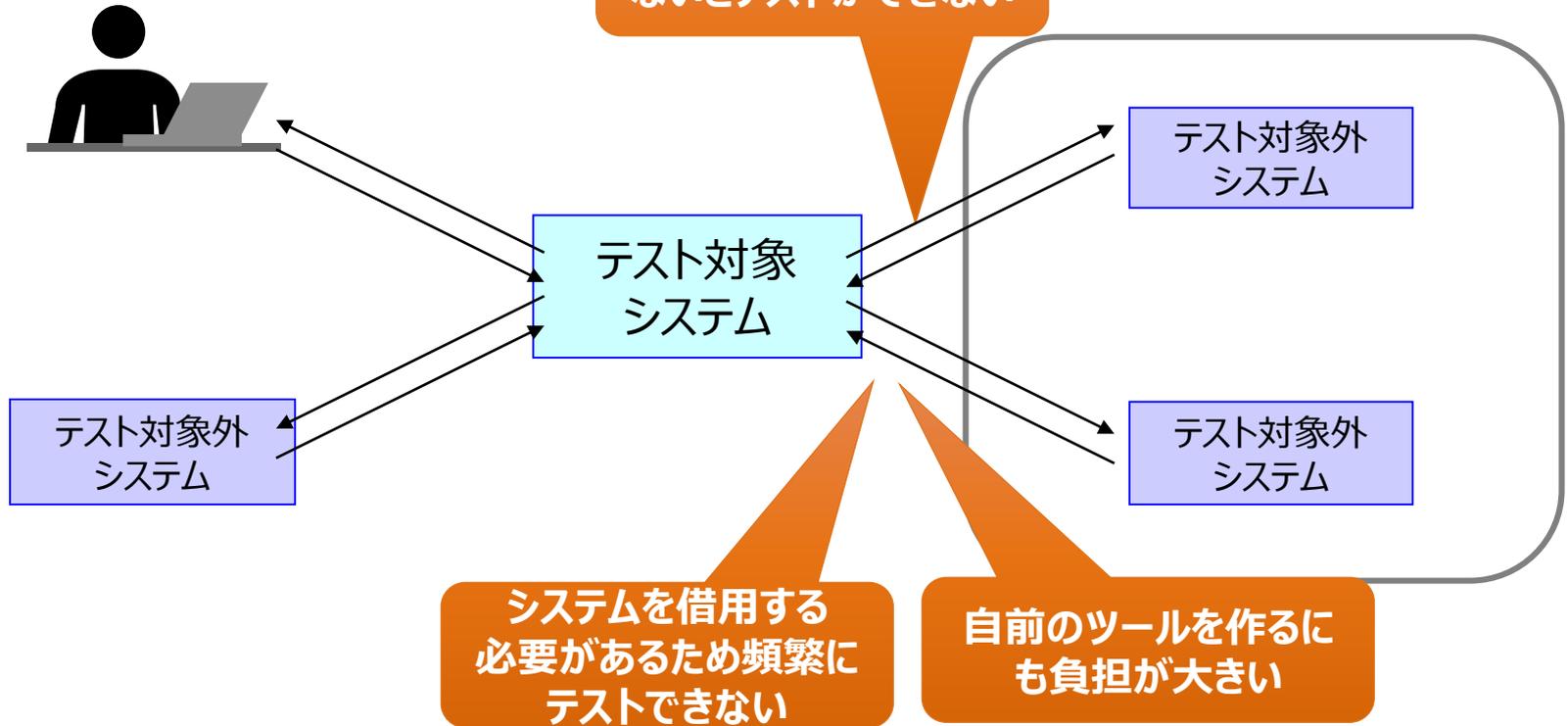
テストを待っていた時間



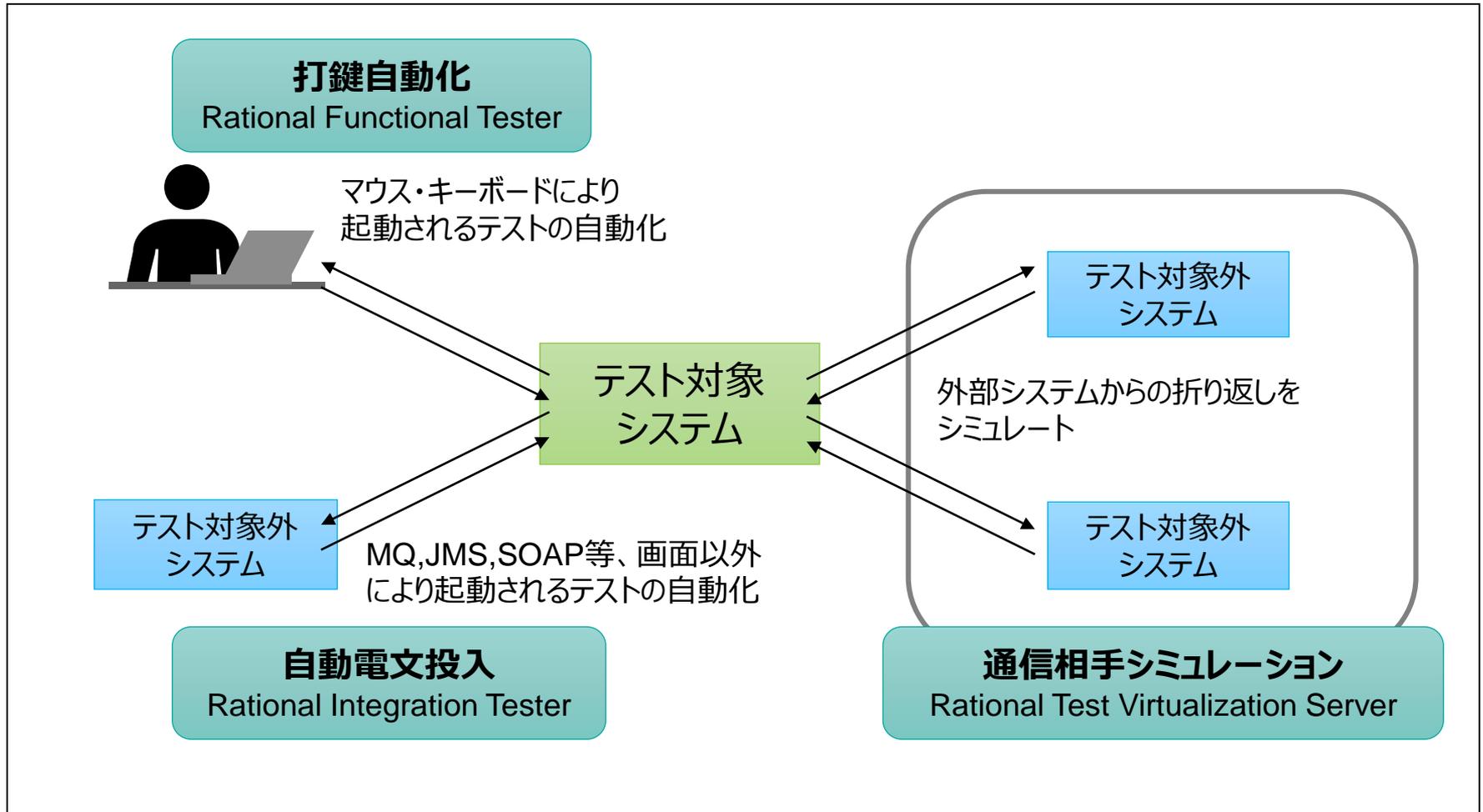
外部接続システムとの依存関係における課題

テストに外部システムとのデータ連携が必要

外部システムと連携しないとテストができない



自動化と仮想化によるトータルなテスト効率化

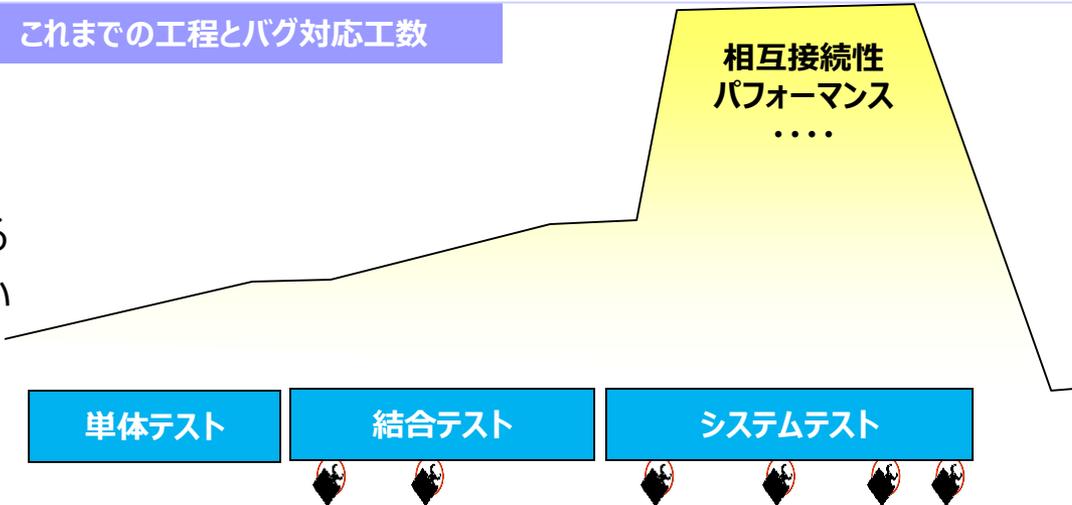


期待効果：仮想テスト環境により重大バグを早期検出

課題

- システムテスト実施の遅延
- テスト環境構築に時間が掛かる
- リリース直前に重大バグが検出される
- テスト環境の構築・保守コストが高い
- リリースサイクルが長期化してしまう

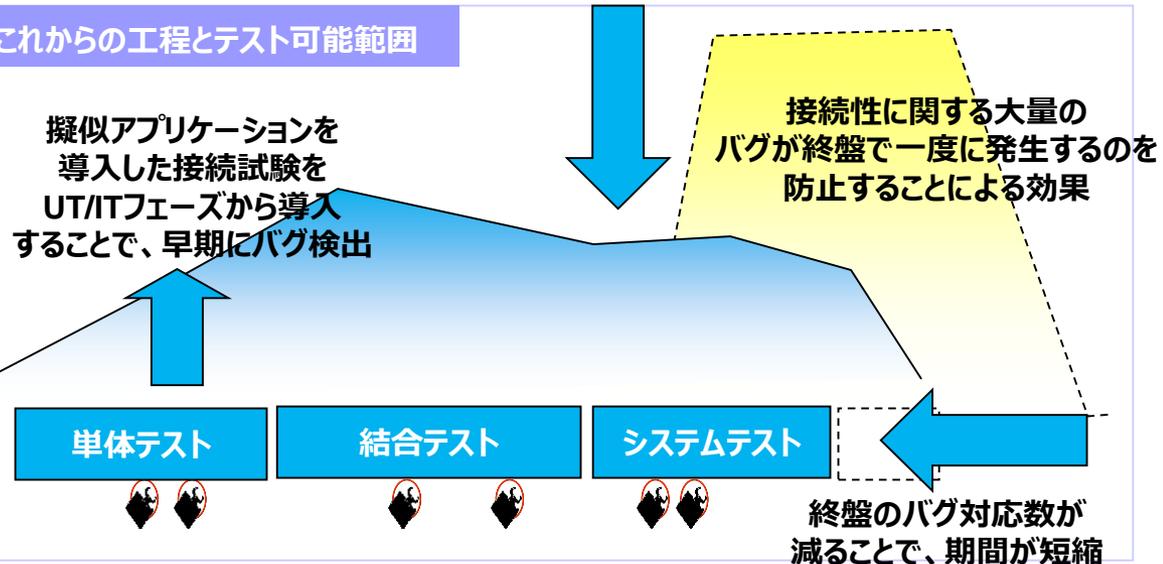
これまでの工程とバグ対応工数



効果

- 【コスト低減】テスト環境を容易に構築でき、コスト削減
- 「リスク低減」早期に結合テスト環境が準備できることで、バグを早期に検出
- 【期間短縮】リリースサイクルを短縮

これからの工程とテスト可能範囲



仮想環境の構築および活用の仕掛け

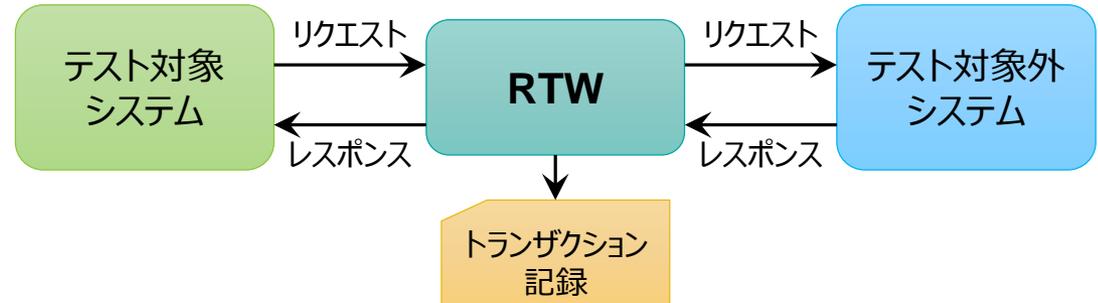
Step1

– 実環境の接続



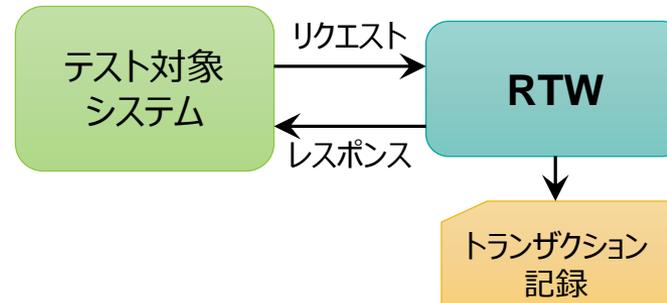
Step2

– トランザクションデータの記録
– トランザクションデータの可変
対応修正



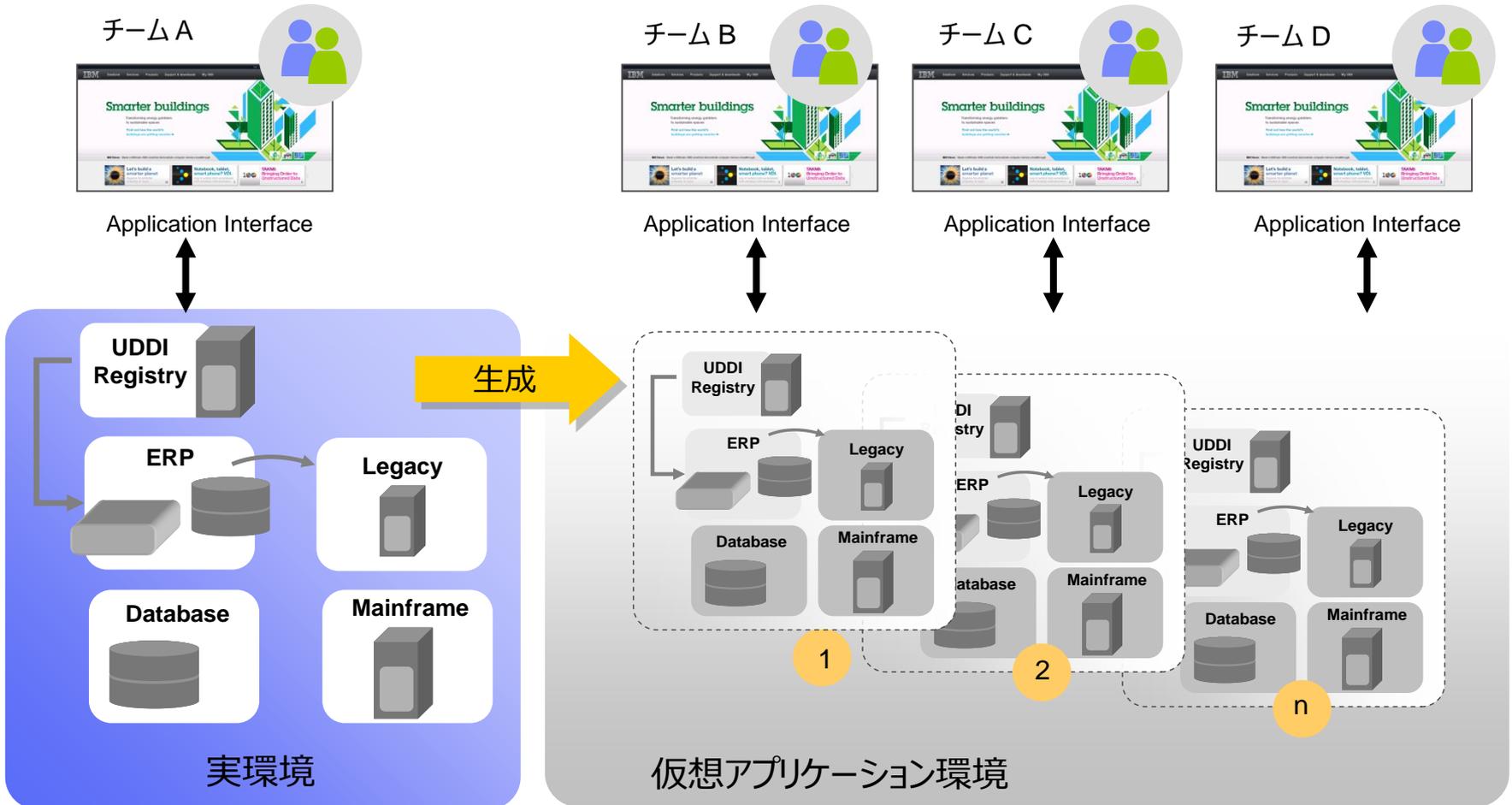
Step3

– 仮想環境を使用してテストの実施



仮想化されたテスト環境の利用イメージ①

依存先システムの振る舞いをキャプチャによりモデル化
仮想的な外部システム環境を複数のプロジェクトで共用

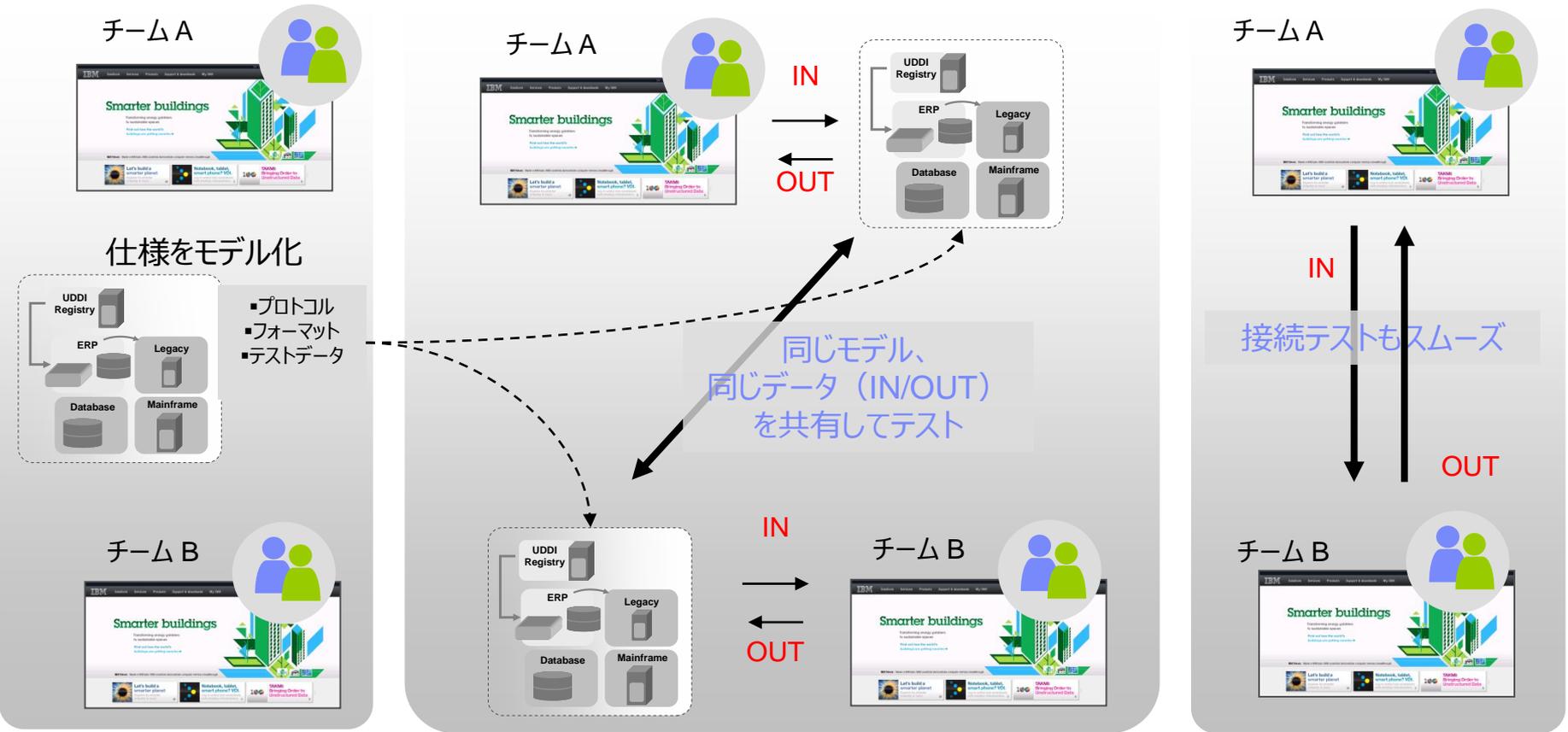


利用時間、利用条件、テストデータに制限がある

いつでも利用可能な、自由度の高いテスト環境、限られた資源でも複数作成可能

仮想化されたテスト環境のイメージ②

チーム双方の合意仕様をモデル化し仮想アプリケーションを作成
 テストの早期段階から共有することで、接続テストもスムーズに完了



アプリケーション仮想化の適用事例（海外）

大手通信事業者



同時進行するテストによる追加のテストリソース要求により増大するテスト費用が課題でした。

- **ソリューション:** サービスを仮想化し複数のテスト環境で再利用可能なテストケースを定義
- **効果:** テスト時間を50%短縮、3年で80万ドルの費用を節約

資産2兆ドル超、グローバル金融企業



多くの異なる電文フォーマットを有するレガシーシステムを次世代システムとして統合する必要がありました。

- **ソリューション:** テスト中の利用が困難な第三者提供システムを仮想化
- **効果:** 手動テストの要する時間を10日から10分に短縮。累積7億ドルの費用を節約。

米国保険大手



複雑化するSOA及びWebサービスのテストは手動では十分に行えませんでした。

- **ソリューション:** レガシーシステムの機能を仮想化
- **効果:** テストの95%を2時間まで短縮。また、料率計算のテストの94%を320時間まで短縮。

企業統合を行った大手製造業



従来レンタルしていたシステム基盤を社内標準ミドルウェアプラットフォームに以降する必要がありました。

- **ソリューション:** 移行期間中に利用できなくなる重要システムを仮想化。
- **効果:** 当初予定より2ヶ月短い6ヶ月で完全移行を完了。システム基盤のレンタル及び関連の第三者システム費用を削減。

Thank You

www.ibm.com/software/rational

© Copyright IBM Corporation 2014. All rights reserved.

本資料の情報は情報提供のみを目的として現状のまま提供され、明示的または暗黙的であるかを問わず、いかなる保証もいたしません。IBM は本資料の使用に起因するいかなる損害についても責任を負いません。本資料は、IBM (または IBM のサプライヤーまたはライセンサー) にいかなる保証責任を負わせるものではなく、また、IBM ソフトウェアの使用に際し適用される、プログラムのご使用条件の内容も変更するものではありません。本資料に記載の製品、プログラム、またはサービスが日本においては提供されていない場合があります。日本で利用可能な製品、プログラム、またはサービスについては、日本 IBM の営業担当員にお問い合わせください。本資料で言及されている製品のリリース日および性能は、市場機会やその他の要因に基づいて、IBM の判断により随時変更される場合があります。また、それらによって、今後その製品または機能が利用できるようになることを IBM が確約するものではありません。IBM、IBM ロゴおよびその他の IBM 製品は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。



免責事項

当資料は、お客様の問題解決のためのヒントとしてご利用ください。当資料における記載内容は、お客様固有の問題に対し適切であるかどうか、また正確であるかどうかは十分検証されていません。結果についていかなる保証も責任も負いかねますので、あらかじめご了承ください。