SPI Japan 2018

HITACHI
Inspire the Next

新しいITサービス開発への取り組み、DevOps実践を目指して！

2018/10/10

株式会社日立ソリューションズ
技術革新本部　戦略技術部

細美 彰宏

HITACHI
Inspire the Next

Contents

HITACHI
Inspire the Next

1. デジタルソリューションの波

HITACHI
Inspire the Next

デジタルソリューション市場の拡大

- ITとOTの融合によるビジネスのデジタル化の加速。デジタル技術で、データを価値に変換することで、お客様の課題解決を図ったり、新しい価値を生み出すことへのニーズ増加。

- 取り扱うシステム対象領域・性質が変化。
  - ✓ SoEから、新事業を生み出したいというお客様の増加
  - ✓ 定型業務・問題解決型から価値創造型へ変化
  - ✓ ニーズ・技術の多様化で、PoCでトライ&エラーを繰り返す開発

SoR: 企業内データが中心、業務の効率化が目的
SoE: IoT・ソーシャルデータが中心、新しい価値・新事業の創出が目的

継続的に、新しいサービスを価値として創出することが必要
従来型の開発プロセスでは対応できない

1-2 デジタルソリューションへの課題と対策

HITACHI
Inspire the Next

① 新事業創出のためのニーズ発見・仮説検証

- 新しい事業創出、価値創造は非常に難しい。試行錯誤、再検討など、手戻りが多く、非効率な進め方。

⇨お客様の潜在的なニーズ発見、アイデア創出を促す**顧客協創手法**。

② ソフトウェア開発の期間短縮・生産性向上

- 創出したアイデアの早期実現と、ソフトウェア開発の生産性向上・品質確保は必須。「継続的に」「迅速に」価値を提供。

⇨PaaSによる環境構築の省力化と、CI(継続的インテグレーション)によるテスト環境構築～テスト実施の自動化を実現する**高速開発技法**。

③ 本番運用と改良(運用・保守)

- お客様の要望を製品・システムにいち早く取り込み、より迅速に市場へ投入することが重要。

⇨迅速な要望への対応、開発のリードタイム短縮に**“DevOps”が最適**。

HITACHI
Inspire the Next

開発(Dev)と運用(Ops)の一体化

- お客様への提供価値を最大化するために、ITサービスをニーズに応じて、タイムリーに提供する考え方。
- 品質も確保しつつ、ニーズを迅速にリリースする、繰り返し提供するような、予測が困難なITサービスへの開発へDevOps導入のメリットは多い。

DevOpsの起源
Velocity 2009 - O'Reilly Conferences
「10+Deploys per Day: Dev and Ops Cooperation at Flicker」
John Allspaw & Paul Hammond

DevOps適用による高速開発の実現、デジタルソリューション事業の加速をめざし、高速開発技法・環境の整備に取り組んだ。

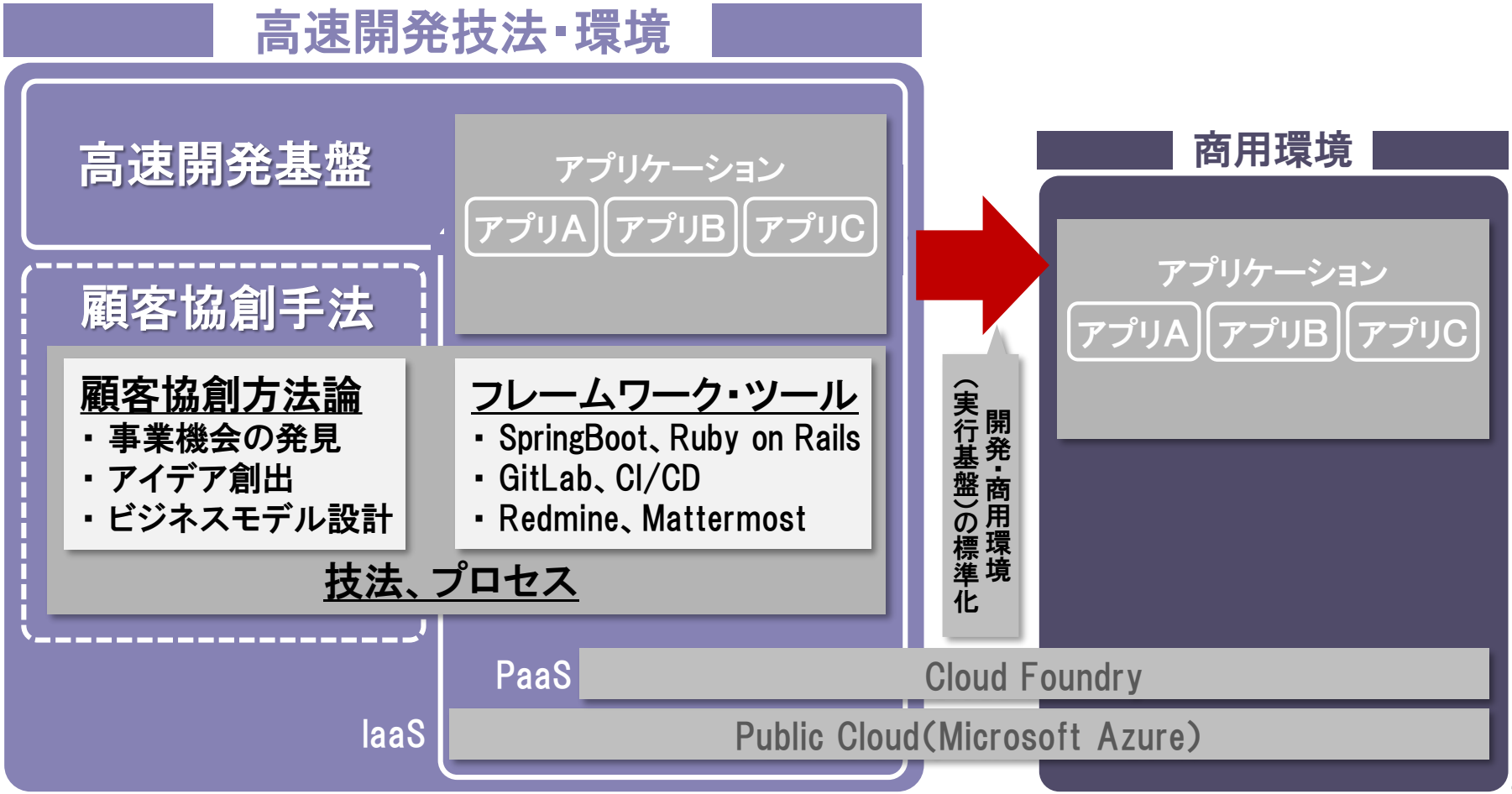
HITACHI
Inspire the Next

2. 高速開発技法・環境

2-1 高速開発技法・環境

HITACHI
Inspire the Next

- 新事業創出、アイデア創出の効率化 ➡ 顧客協創手法
- アイデアの早期実現と開発生産性向上 ➡ 高速開発基盤
- 早期市場投入と迅速なニーズ反映 ➡ 開発・商用環境の標準化



2-2 顧客協創手法と高速開発基盤

HITACHI
Inspire the Next

顧客協創手法

- ニーズの発見・仮説立案を実践する手法を提供することで、新事業創出を効率化。日立の**顧客協創方法論**をベースにプロセス・手法・ノウハウを整理、体系化。
- デザイン思考をベースに、**“ユーザー視点”**の発想で、お客様と協創し、イノベーティブな新コンセプト・アイデアを創出する手法。

高速開発基盤

- クラウド上のWebアプリケーションを効率よく開発するためのプラットフォーム。主に、**PaaSとCI/CD環境**を提供。開発環境構築と維持運用のコスト削減、開発リードタイムの短縮を目的とする。

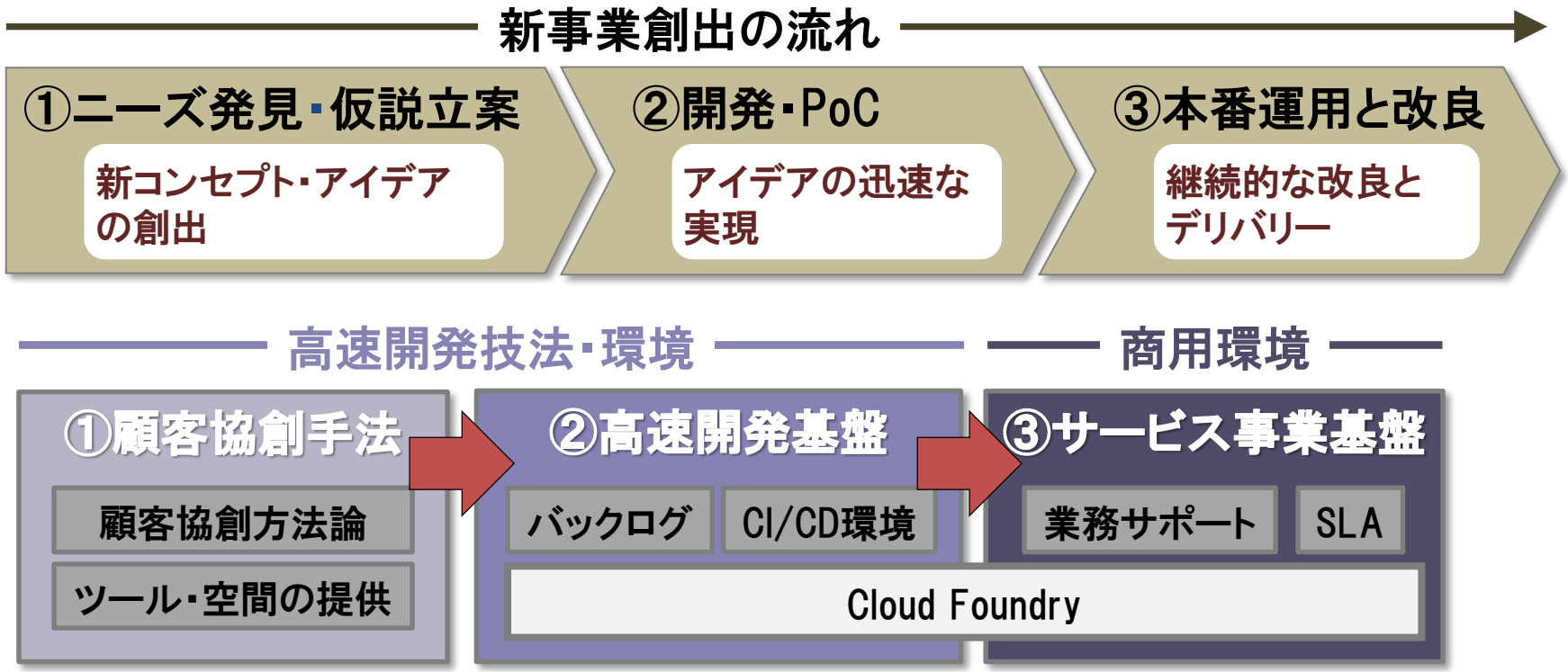
- PaaS：高速開発に寄与する要素技術を定義した開発環境を提供。開発者は、アプリ部分のみの開発に専念（ハードやネットワーク、OS・ミドルから解放される）。
- 継続的インテグレーション（CI）、継続的デリバリー（CD）：動作確認やテスト作業を自動化し、システム環境構築・デプロイも予め、プログラムされた手続きとして、自動化・管理。プログラム変更リクエスト/レビュー/差分マージ操作をオンライン化。

2-3 事業創出のための高速開発の流れ

HITACHI
Inspire the Next

高速開発プロセス

- 事業創出の仮説段階からトータルで、サービスのライフサイクルを通して活用できる手順を提供。

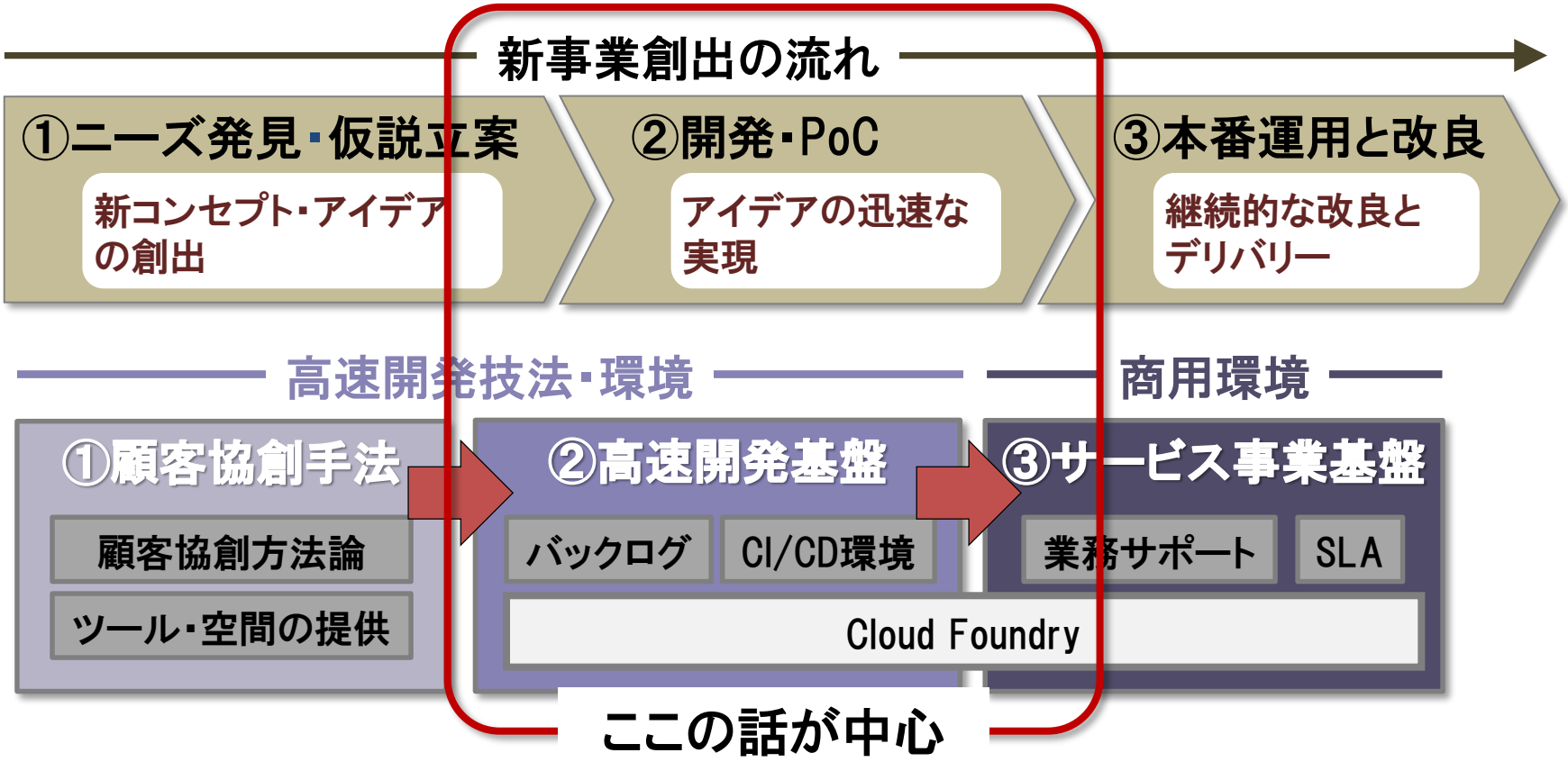


2-3 事業創出のための高速開発の流れ

HITACHI
Inspire the Next

高速開発プロセス

- 事業創出の仮説段階からトータルで、サービスのライフサイクルを通して活用できる手順を提供。



2-4 継続的インテグレーション(CI)/継続デリバリー(CD)

HITACHI
Inspire the Next

品質を確保するためには石橋を叩いて渡りたい

- 品質を確保しつつ、リリース速度も上げたい、という課題に対し、「超高速に石橋を叩いて渡れば良い」というアプローチ。
- 継続的デリバリーは、クラウドやコンテナ技術の進展によって実現。

■ 継続的インテグレーション(Continuous Integration)
ビルド、テスト、コードの品質検査など、定型的で反復可能な作業を自動化し、繰り返し、細目に実行することで、省力化を図る。
⇒メリット:リリース準備時間の短縮、バグの早期発見

■ 継続的デリバリー(Continuous Delivery)
CIを通じて、でき上がったモジュールをテスト環境、本番環境へ自動でリリースできるようにし、反復可能なプロセスを構築する。問題が発生しても、すぐに、リリース前の状態に戻せる。
⇒メリット:成果物の素早い共有と、フィードバックの取得

HITACHI
Inspire the Next

OSSを中心とした構成

PaaS(Cloud Foundry)と、継続的インテグレーションツール(GitLab、GitLab CI、Mattermost)で構成。

サービス
インスタンス
実行基盤

サービス
ブローカー

開発者
ポータル

ログ収集基盤

CI/CD基盤

- Git リポジトリ
- CI ランナー
- ChatOps

- 構成管理
- 死活監視

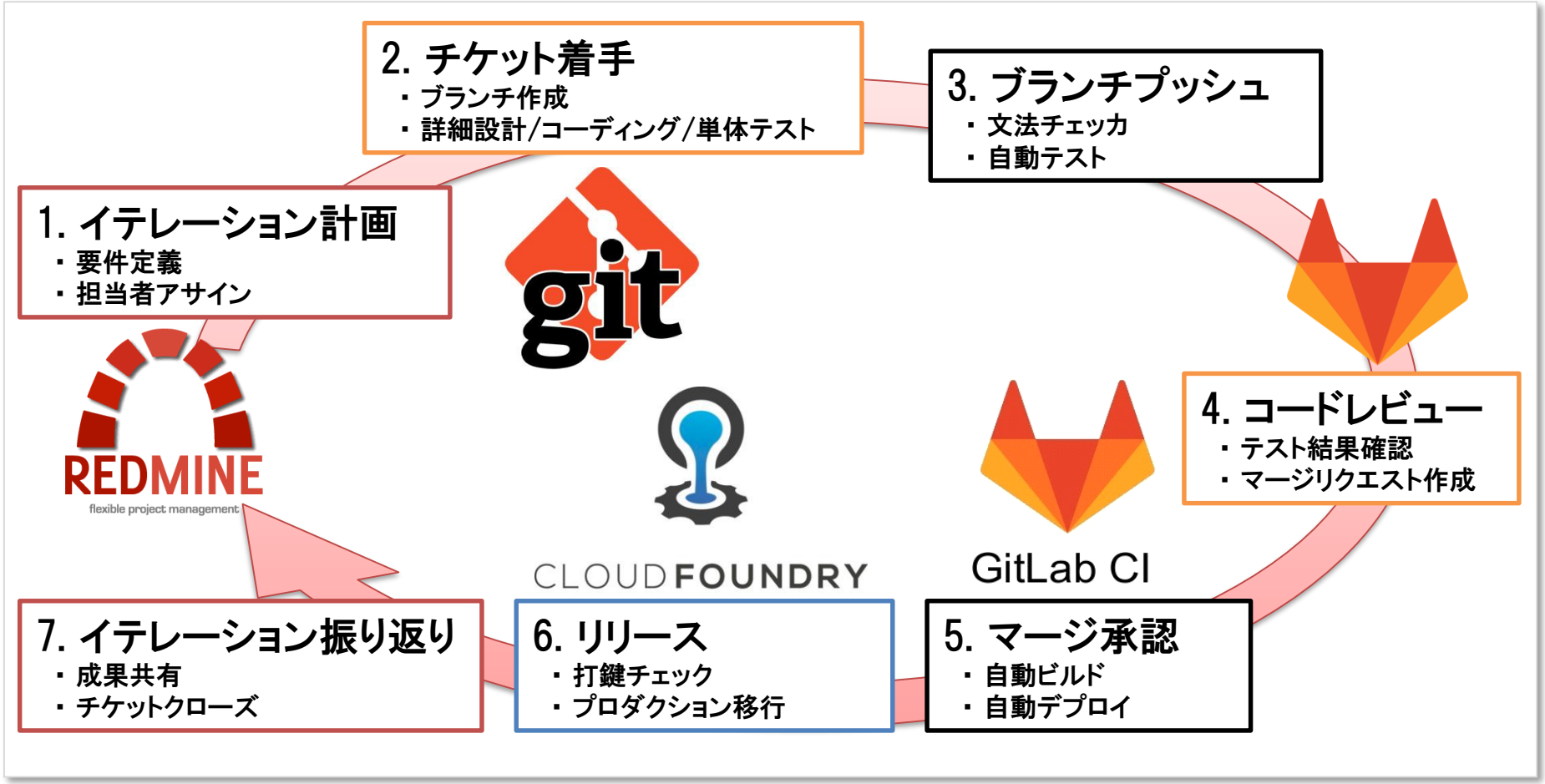
- 仮想マシン
- 仮想ネットワーク
- ストレージアカウント
- ファイアウォール
- ロードバランサ
- DNSゾーン、等

2-6 高速開発基盤を使った開発プロセス

HITACHI
Inspire the Next

アジャイル・スクラムをベースとしたプロセス

基盤開発・運用チーム自身がプロセスを実践し、ケースとして発信。



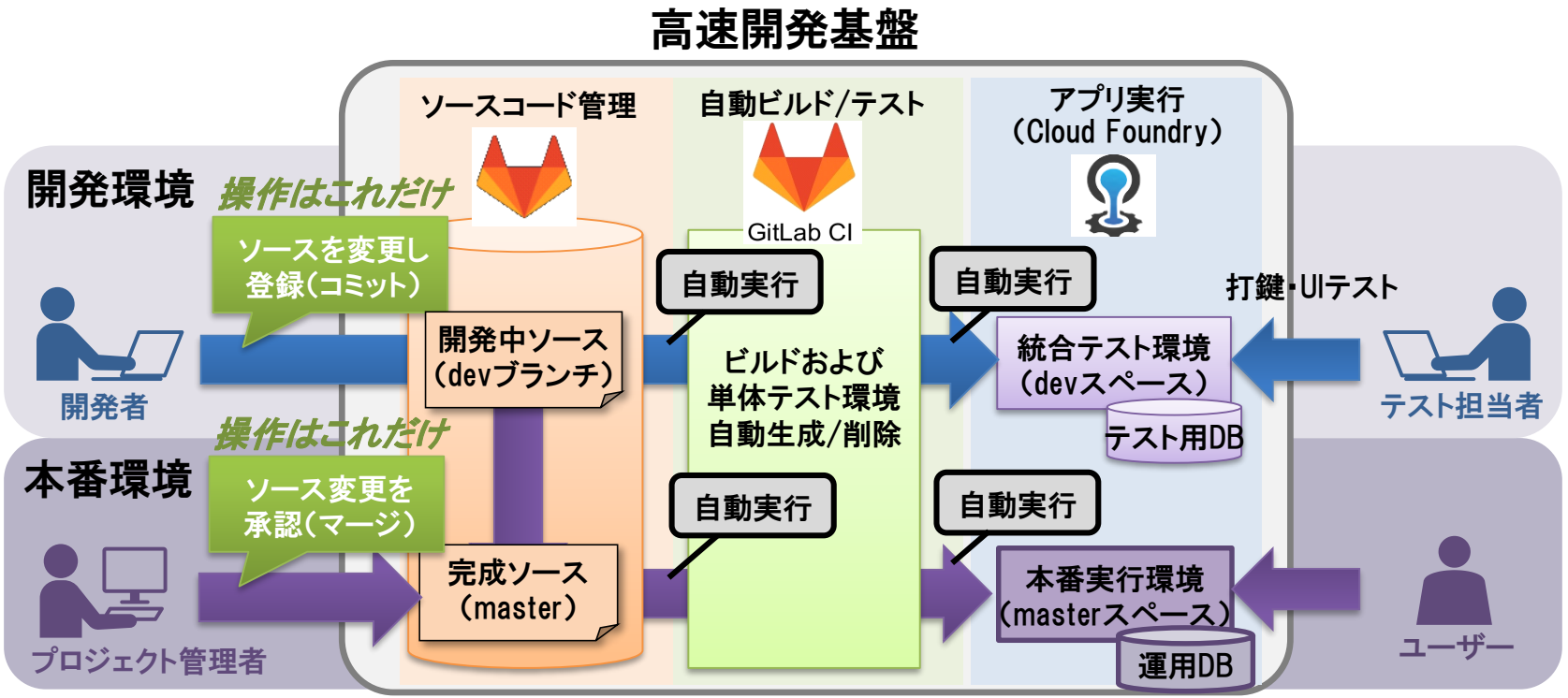
2-7 特徴①:アプリ開発に集中できる仕組み

HITACHI
Inspire the Next

■ コミットやマージリクエストに、単体テスト・デプロイを実行するパイプラインを定義。品質確認作業を自動化、都度実施することにより、高頻度で機能更新(CI)を実現。

⇨ 品質を確保しつつ、スピーディな試行錯誤ができる。

(1) 開発者はソースを書いてコミットするだけ。単体テスト実施と統合テスト環境の公開ができる。
(2) プロジェクト管理者は変更をマージするだけ。これで本番環境を公開できる。

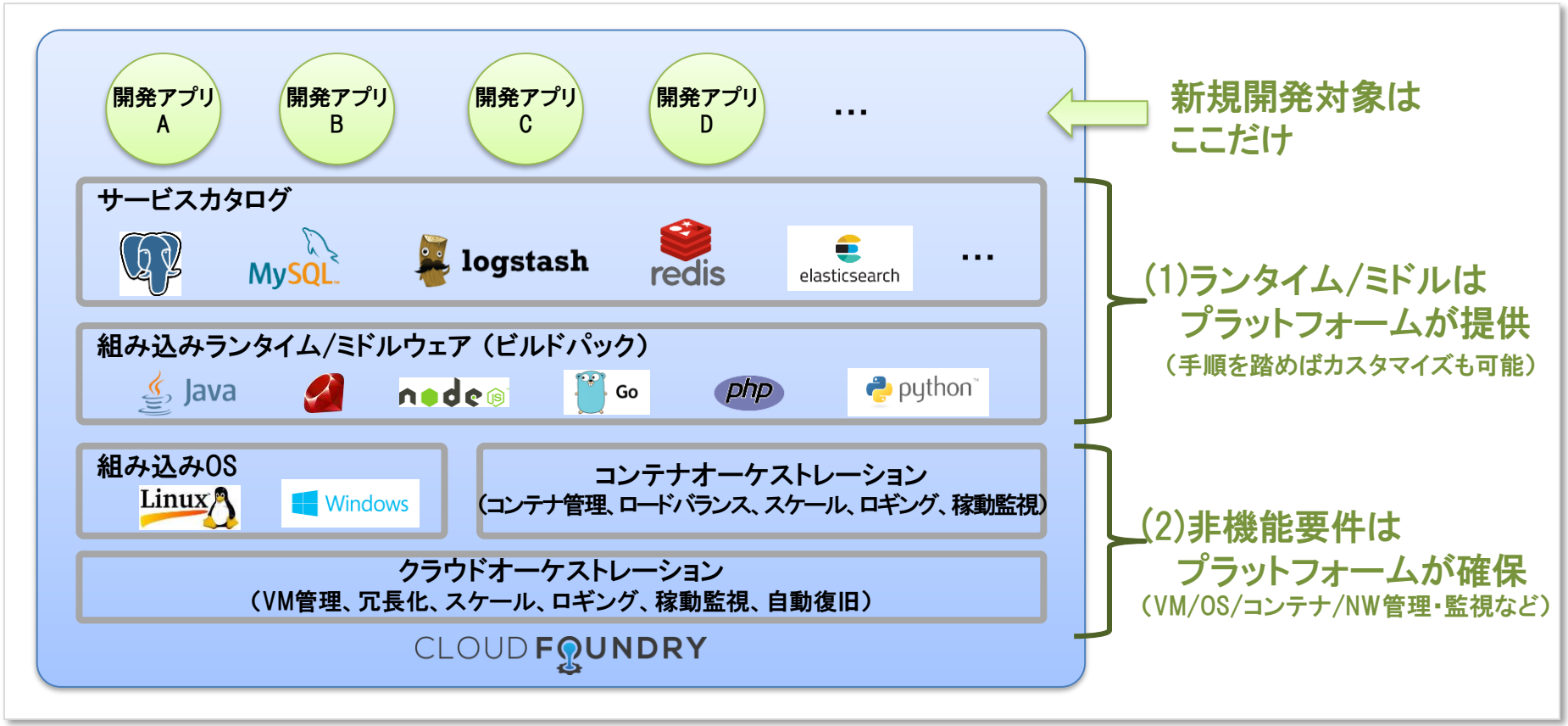


2-8 特徴②：開発して、すぐにお客様と共有する仕組み

HITACHI
Inspire the Next

■ 実行基盤にPaaS(Cloud Foundry)を活用し、非機能要件を確保。コード変更とpushを繰り返すだけで、いつでも運用できる状態(CD)を実現。

⇨ 品質を確保しつつ、成果物をすぐに触ってもらうことができる。



HITACHI
Inspire the Next

3. DevOpsを広めるための取り組み

3-1 DevOpsを広めるための取り組み

HITACHI
Inspire the Next

高速開発技法・環境の適用拡大の取り組み

○:展開済み、△:着手または一部展開、ー:未着手

分類	取り組み・状況
ツール 開発基盤	○:標準的な高速開発技法・環境の整備、CIツール群の提供
	△:プロジェクトダッシュボードの整備
開発技術 プロセス	○:高速開発基盤の導入ガイド(機能・手順解説、構成例、等)
	○:CI/CDによるパイプライン構築、設定ガイド、等
	△:アプリ設計ガイド(△:クラウドネイティブ、ー:マイクロサービス)
	ー:品質管理プロセス、テスト駆動開発ガイド
技術者育成	○:社内外の最新技術・活用事例の紹介セミナー、メルマガ
	○:開発者・運用者を集めたコミュニティ活動(社内Meetup)
	△:アプリ開発の一連の流れを体験できるハンズオン
サポート	○:DevOps技術者をプロジェクトに参画させた立ち上げ支援
	○:チャットチャンネルサポート(チャットでQ&A)

3-2 現場からの意見①

HITACHI
Inspire the Next

適用プロジェクトからの意見

- 既存商品のサービス化(SaaS)を実施
  - 最小限の改修で、開発対象を基盤に載せ、CIの導入までを完了。
  - 開発ツールが一通り揃っており、有効。これから、CDを含め、開発コスト圧縮やポータビリティ等のメリットを最大限にできるよう継続したい。
- オンプレの既存サービスをストレートコンバージョン
  - リフト&シフト(そのまま既存を載せて(リフト)、その後、最適化の移行(シフト))を実施。外部連携等の技術的な課題を解決しながら進め、リフトまでを完了。シフトでは、改修の洗い出し、設計見直しで苦戦。
  - 膨大な確認作業の自動化を検討中で、CI/CDの導入・自動化による効率化を期待している。ただし、自動化するためのテストコードの作成工数、技術者の確保が悩み。
- 製品開発で活用
  - テスト・カバレッジ・静的コード解析・ドキュメント生成の自動化で活用し、効果を感じている。以前から自動化を推進しており、CI/CD環境を利用。
  - 国内分散拠点との共同開発で活用している。

3-3 現場からの意見②

HITACHI
Inspire the Next

非適用プロジェクトからの意見

- DevOps、高頻度のデリバリーは不要
- 品質管理方法が不明
  - CI/CD、テスト自動化を導入した場合、どう品質管理を行えば良いのか分からない。自動化するためのテストコードの作成工数、技術者の確保も困難。

⇨ 開発・運用・品質管理プロセスを明確化。標準的な開発計画書、運用計画書、品質管理計画書のひな形を準備する予定。

- 従来型開発から変更困難
  - 既存システムを移行する工数が出ない。既存システムの保守で適用するメリットが少なく、新規案件でしか検討できない。
  - 従来型の開発から変わることへの抵抗。開発・運用に関する蓄積したノウハウを基に効率化しており、新たに導入するメリットが薄い。

⇨ リフト&シフト実施プロジェクトのノウハウを整理し、手順化する予定。

HITACHI
Inspire the Next

ツールと組織文化の両面からの改善

Velocity 2009,「10+Deploys per Day: Dev and Cooperation at Flicker」
John Allspaw & Paul Hammond

■ ツール
1. 自動化されたインフラストラクチャ(Automated infrastructure)
2. バージョン管理の共有(Shared version control)
3. ワンステップによるビルド&デプロイ(One step build and deploy)
4. フィーチャーフラグ(Feature flag)
5. メトリクスの共有(Shared metrics)
6. チャットとボット(IRC and IM robots)

■ 組織文化
1. お互いを尊重する(Respect)
2. お互いを信頼する(Trust)
3. 失敗に健全な態度をとる(Healthy attitude about failure)
4. お互いに避難しない(Avoiding blame)

ツール面は、ほぼ整備済み。次は、組織文化も含めた改善。

HITACHI
Inspire the Next

END

新しいITサービス開発への取り組み、
DevOps実践を目指して！