

DevOpsは、なぜ“バズワード”扱いされちゃうか？

～あるいは「講演タイトルはよく考えてからつけよう」の件～

日本ヒューレット・パッカー株式会社
ソフトウェア事業統括
シニア・コンサルタント
藤井 智弘(tomohiro.fujii@hpe.com)

まずは、お詫びから…

mOm 印刷物提出期限に間に合いませんでした。
資料は後日オンラインにて公開します。
お手元の資料は、“メモ用紙”としてご活用ください。

mOm 時間の関係で製品の詳細は省いています。
来週前半に、以下のフォローアップページを
立ち上げますので、ご興味のある方はそちらも
参照いただければ。

<http://www.hp.com/jp/alm>
or <http://bit.ly/1W05Pes>



きっかけ

…インタビューが記事になったら、いつのまにか『DevOpsをパスワード扱いしてる人』みたいに見られて困る昨今、いえいえそんなことはありません…だって、始まってないだけですから。

で、そもそもなんでパスワードみたいに見えちゃうのかというあたりから、取り組みのヒントをあぶりだしてみたいな、と、若干チャレンジングな35分です。…

～セッションご案内文より

The screenshot shows the ATMARKIT website interface. At the top, there's a navigation bar with categories like TCP, @IT Special, @ITセミナー, eBook一覧, Q&A@IT, ITトレメ, ブログ, 転職, and 派遣. Below that is a banner for the 15th anniversary of @IT, with the text 'そのギモンは、あなただけのもの?' and 'ITエンジニアのためのQAサイトはコチラ'. The main content area features an article titled 'DevOpsが浸透しない「本当の理由」(1/3)' under the 'System Design' category. The article text discusses the reasons why DevOps hasn't fully permeated in Japan, mentioning that while it's being adopted, it's often treated as a password. A photo of a man, presumably the author, is shown. The right sidebar contains various news items and advertisements, including one for 'ITmedia エンタープライズ 始動会' and another for '外部・内部の脅威に備えて 情報漏洩を防ぐ'.

<http://www.atmarkit.co.jp/ait/articles/1503/19/news019.html>

本セッションの流れ

- ☑なぜ、“バズワード”と見られるのか？
- 解決へのアプローチ
- 参考情報 ～ちょっと会社を離れて

“パスワード”化の要因

✓“集客の奴隷”

✓“エバンジェリズムの
ライフサイクル”

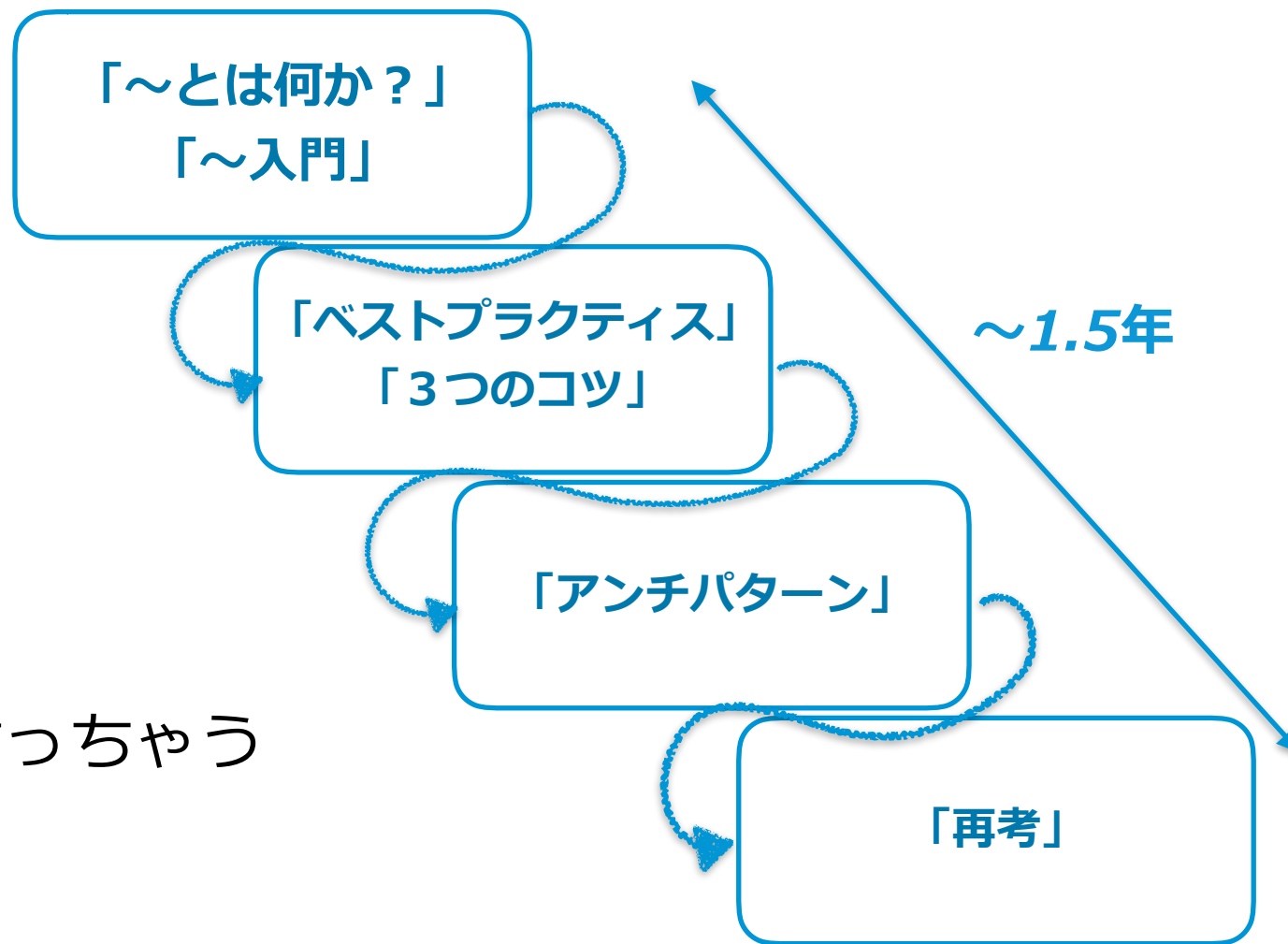
✓なんでもかんでも
“エンタープライズ”って言っちゃう

“パスワード”化の要因

✓“集客の奴隷”

✓“エバンジェリズムの
ライフサイクル”

✓なんでもかんでも
“エンタープライズ”って言っちゃう



“パスワード”化の要因

✓“集客の奴隷”

✓“エバンジェリズムの
ライフサイクル”

✓なんでもかんでも
“エンタープライズ”って言っちゃう

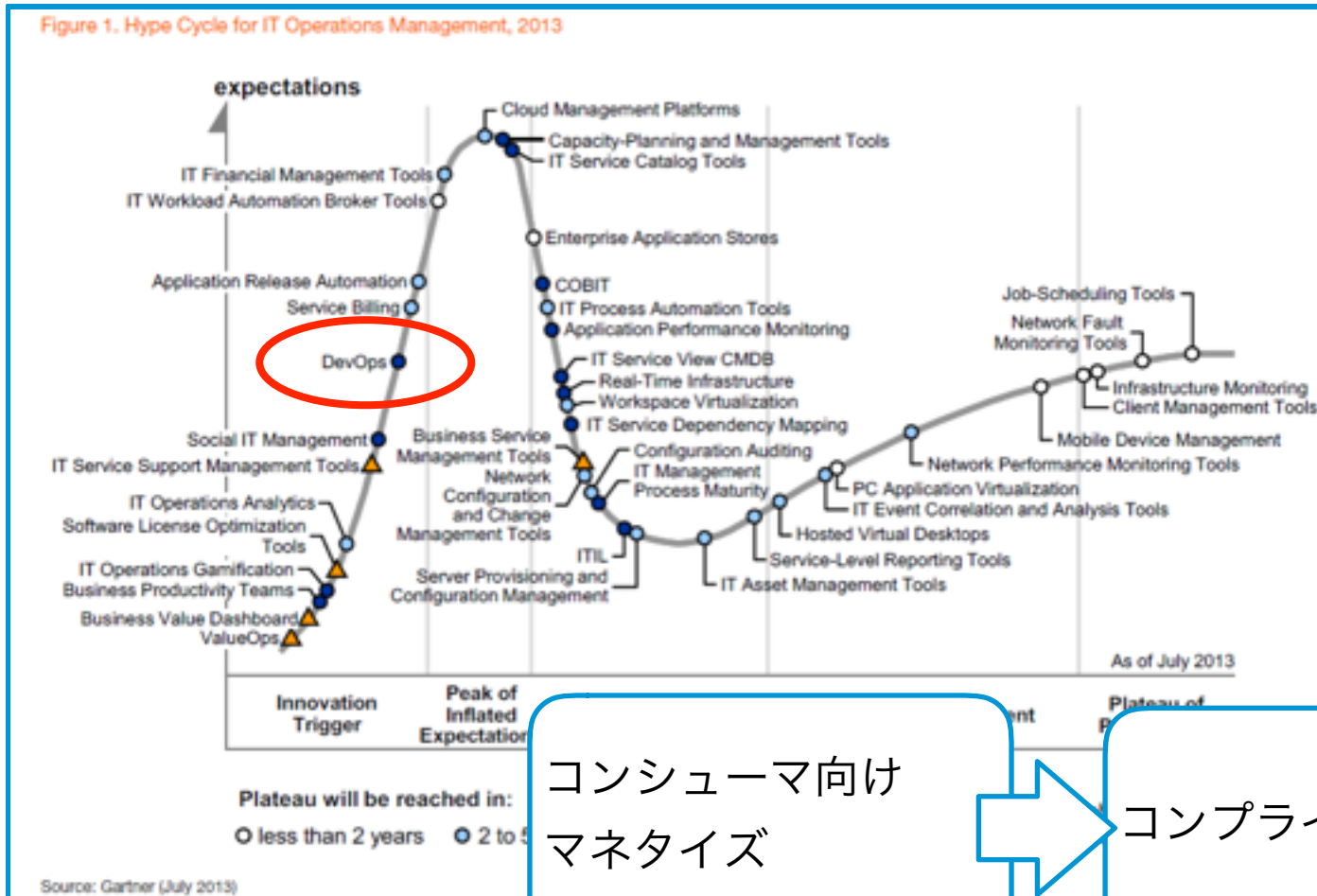
“エンタープライズ”
→“みんなで使う”
→“たくさん売れる”

自粛w

何がギャップを産むか

- 📍 言うほど“エンジニアリング”じゃない（ユーザ）
 - 📍 「自動化したら社内失業がおこっちゃう」
 - 📍 「削減したいのはコストではなく、“キャッシュ・アウト”」
- 📍 効率化≠ビジネスリスク？（ベンダー）

エンタープライズという文脈～その1



“2016年までに、DevOpsは現在の
の大規模なクラウド事業者による
ニッチな戦略から、**グローバル
2000企業の25%**で利用される主
流の戦略へと発展するだろう”

“2019年までに、ITインフラストラ
クチャとその運用サービスでの
**規制対応について、80%の課題で
プラクティスを利用することにな**

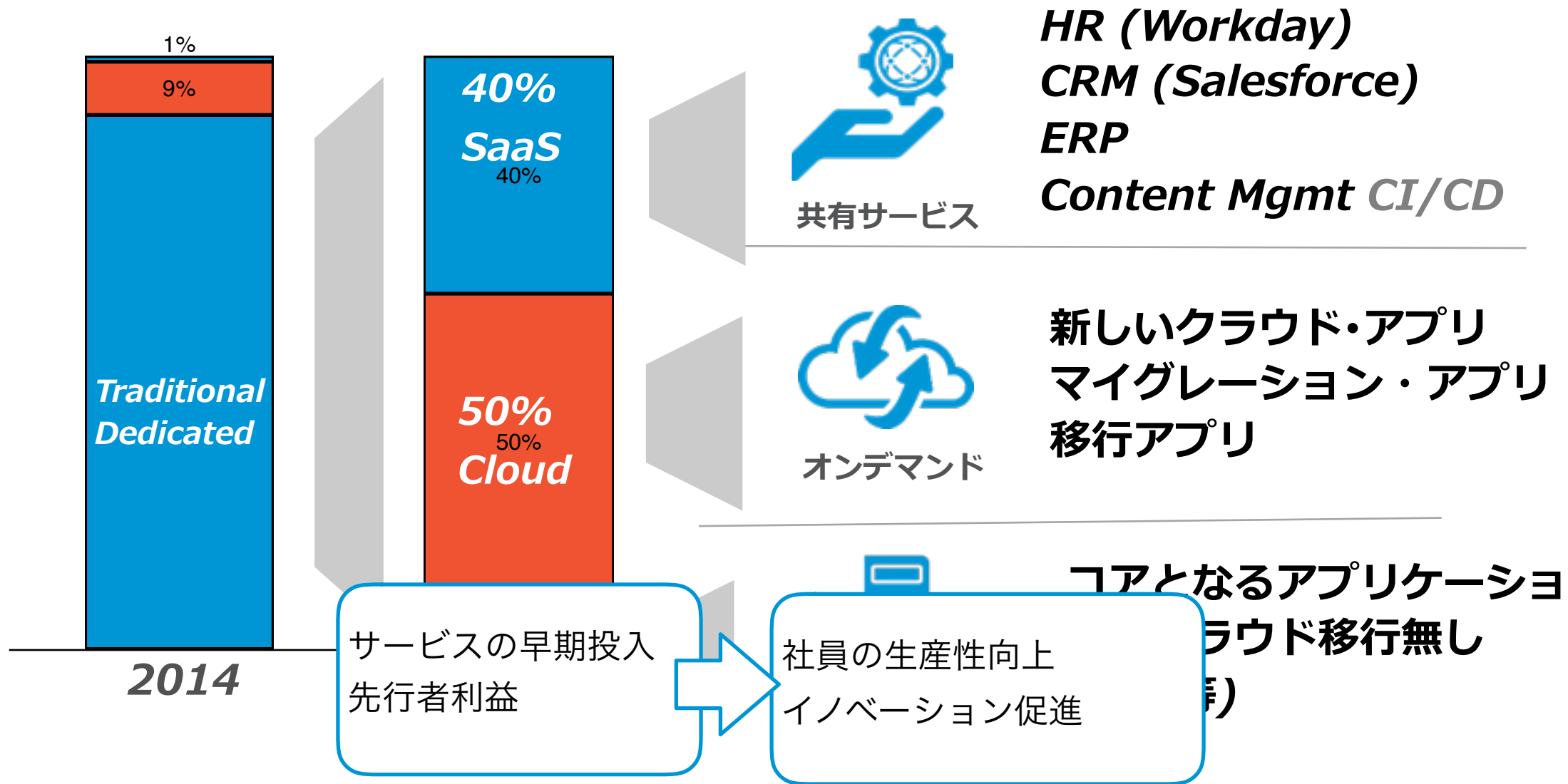
コンシューマ向け
マネタイズ

コンプライアンスエンジン

2014年においては、
DevOpsのコミュニティ
しか利用されていな

い。”

エンタープライズという文脈～その2



Pets vs Cattle ～2つのスタイル～

従来型IT = *Pets*



By Stefan Tell

- 名前をつけてかわいがる
- 病気になれば治療する

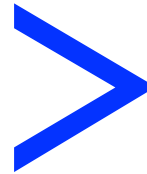
アプリに最適化した
インフラを構築・維持する
(優先するのは“安定・持続性”)

攻めのIT・クラウド = *Cattle*



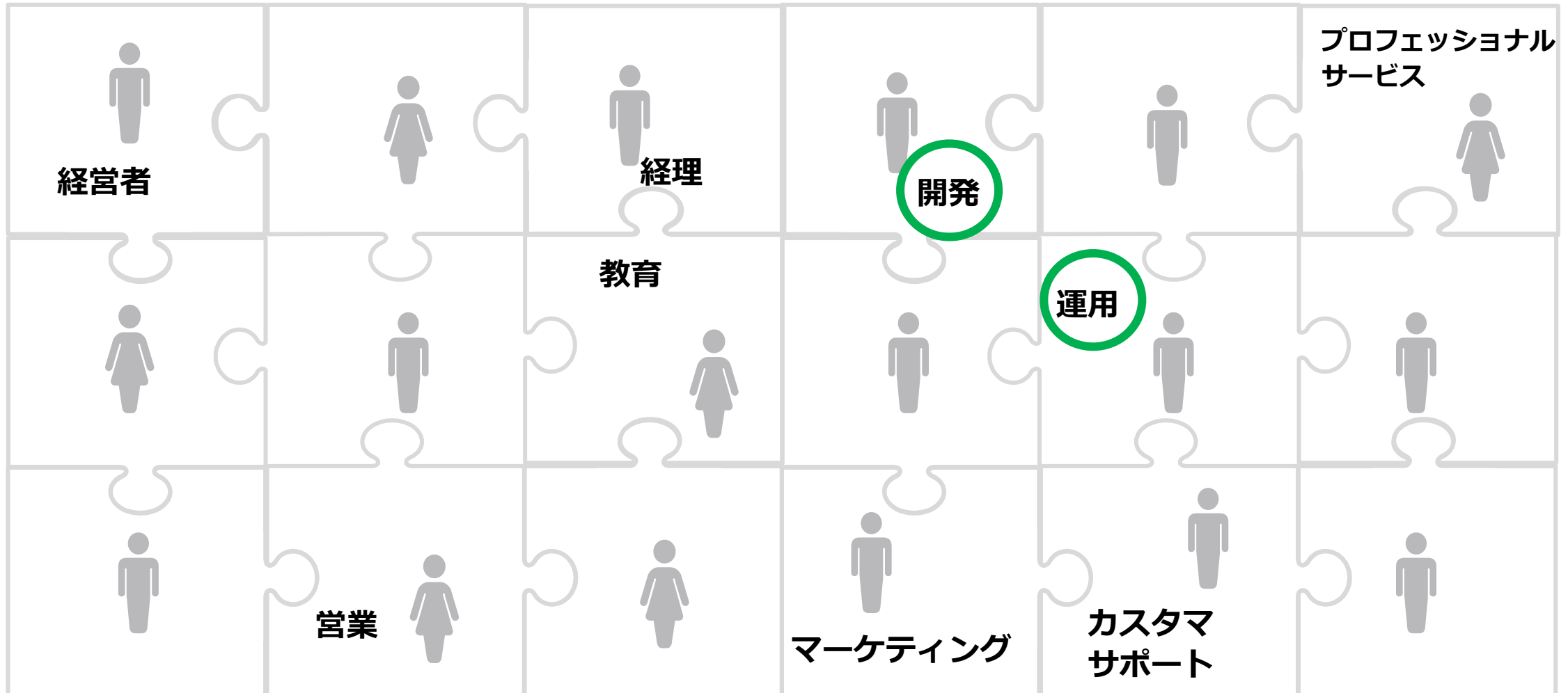
By misteraitch

- 識別子で管理
 - 病気になれば隔離、補充
- シンプルなインフラを
必要に応じ、作っては消す
(優先するのは“迅速・柔軟性”)



エンタープライズという文脈～その3

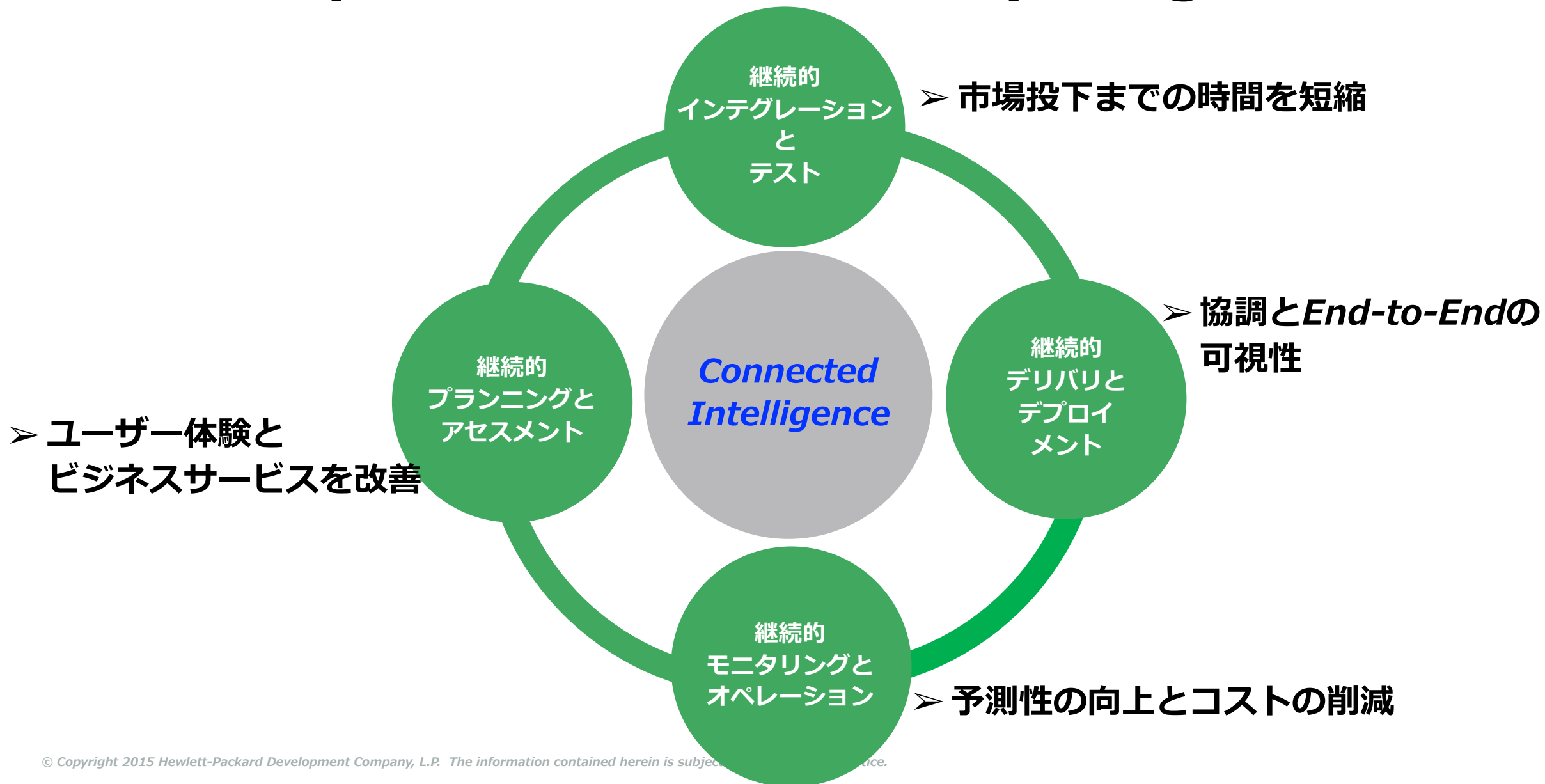
‘開発’と‘運用’に限らない



本セッションの流れ

- なぜ、“バズワード”と見られるのか？
- ☑解決へのアプローチ
- 参考情報 ～ちょっと会社を離れて

HP DevOps ; Continuous Everything



HP DevOps ; Continuous Everything

"コア"なイニシアティブ

継続的
インテグレーション
と
テスト

➤ 市場投下までの時間を短縮

継続的
デリバリと
デプロイ
メント

➤ 協調とEnd-to-Endの
可視性

**そもそものDevOpsの
イメージ**

"開発と運用の連携"

*Connected
Intelligence*

継続的
プランニングと
アセスメント

サービスとしての継続性を
実現する補助的役割

➤ ユーザー体験と
ビジネスサービスを改善

継続的
モニタリングと
オペレーション

➤ 予測性の向上とコストの削減

"サポート"的なイニシアティブ

継続的インテグレーションとテスト

計画

開発

QA

ステージング

本番

ユーザー経験

継続的インテグレーション；可視化と協業

継続的テスト；品質とスピード



協調

- KPIとメトリクス
- 分析
- ガバナンス
- ライフサイクルを横断した統合



自動化

- テスト用ビルドの自動化 (ST, SIT, UAT)
- 機能とパフォーマンス
 - ・ テストの自動化
- レポーティングと通知



時間短縮

- サービスの仮想化
- ネットワークの仮想化
- 機能とパフォーマンス
 - ・ テストの統合

“テストツールCI強化”でのトピックス

✓HP ALMでのJenkins+JUnit/Nunit/TestNG連携

テストサイクル全体の情報一元管理で品質の可視性を向上

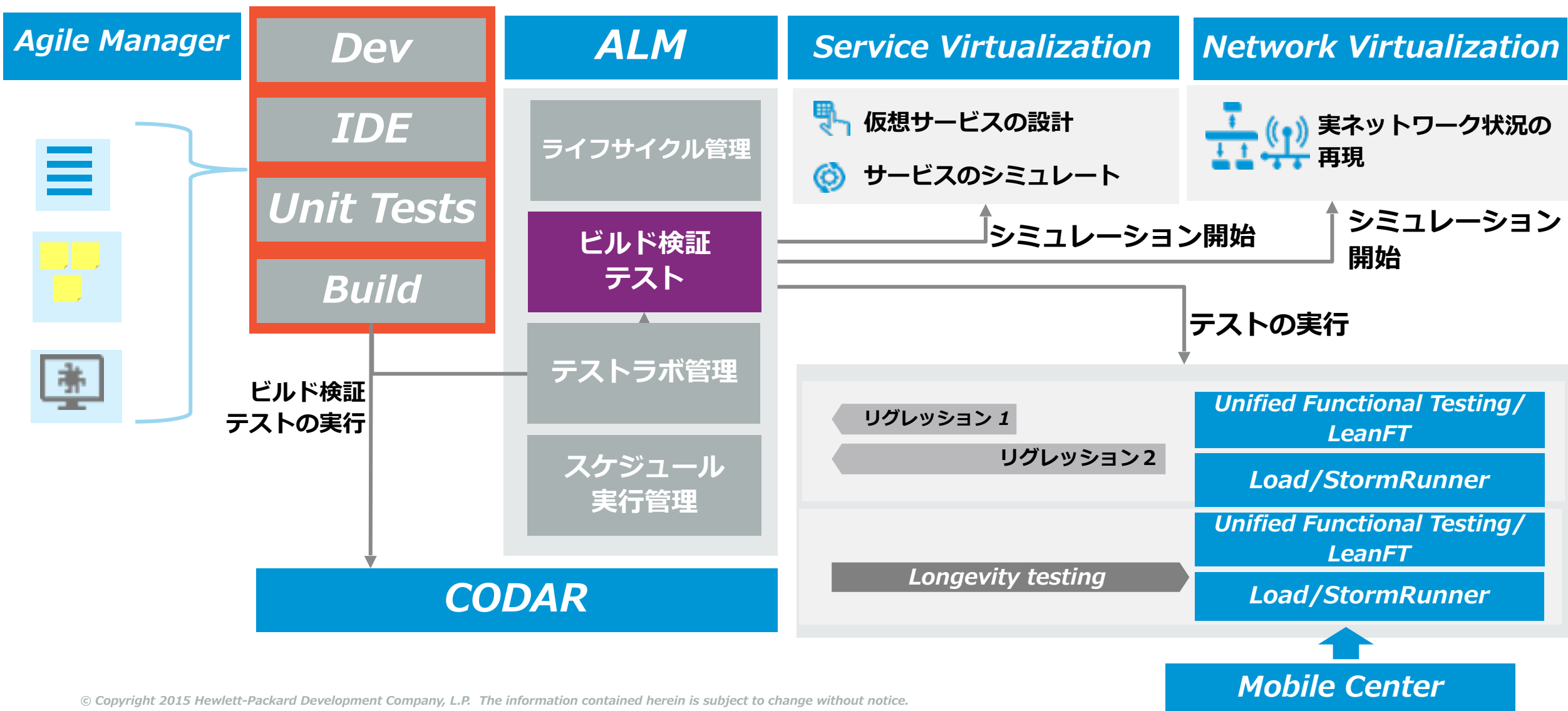
✓Mobile Centerによるモバイルアプリテストサポート

UFT(機能テスト自動化、旧QTP),LR(負荷テスト),Sprinter(探索テスト)で、モバイルデバイス上のアプリをそのままテストCIの一環としてモバイルテストを実行可能に

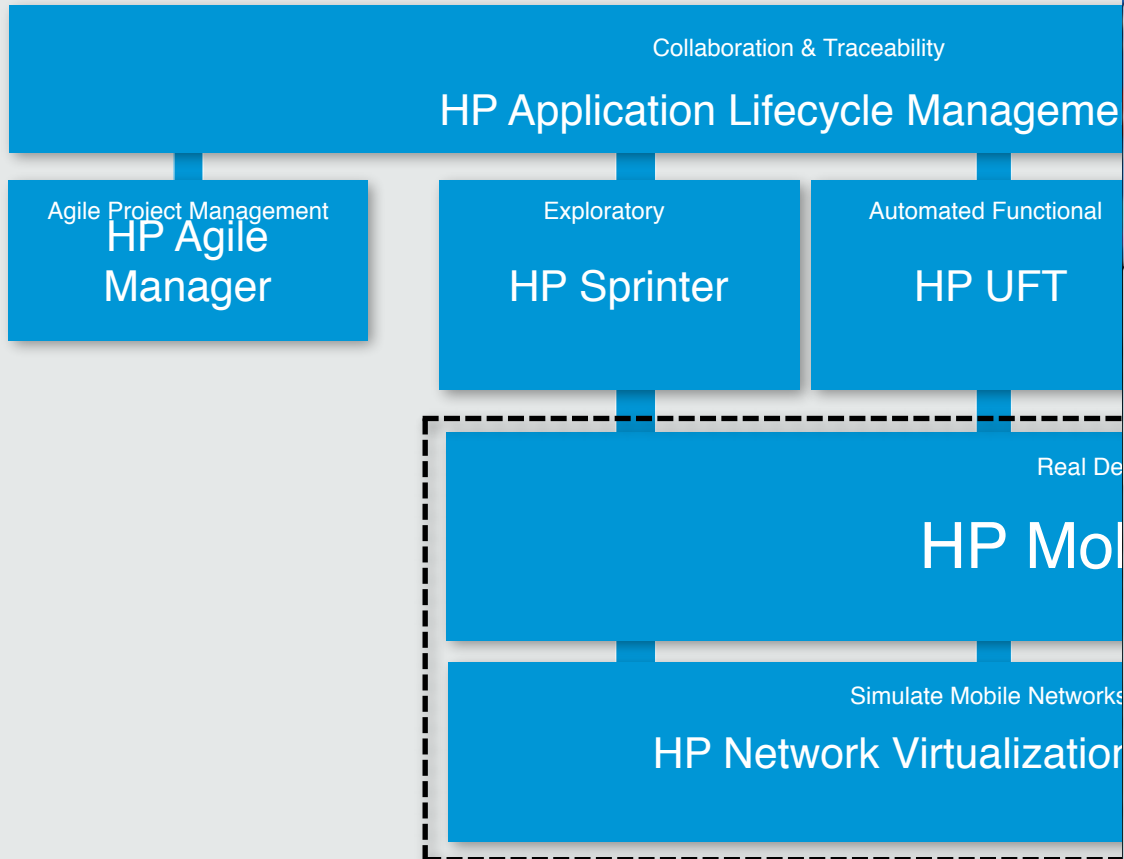
✓Cucumber/Junitと一緒にGUIテスト (LeanFTによる)

BDDの一環としてGUIテストの組込みが可能に

継続的インテグレーションとテスト



たとえばモバイルテストも…



HP Unified Functional Testing - C:\Users\ADM\Documents\Unified Functional Testing\20150122_Mobile_01

```
1 Device("Device").App("Advantage Online").MobileEdit("MobileEdit").SetFocus
2 Device("Device").App("Advantage Online").MobileEdit("MobileEdit_3").LongPress
3 Device("Device").App("Advantage Online").MobileEdit("MobileEdit").Set "魚"
4 Device("Device").App("Advantage Online").MobileEdit("MobileEdit_2").SetFocus
5 Device("Device").App("Advantage Online").MobileEdit("MobileEdit_2").Set "*****"
```

hp MOBILE CENTER 20150122_Mobile_01 の記録
Action1

デバイスの詳細: asus-Nexus 7 [0a38ac2a]

実機の画面を投影して、Webアプリと同様の記録

オープン

ホリクス

This block shows a screenshot of the HP UFT software interface. The top part displays a test script with several lines of code for interacting with a mobile application. Below the code is a browser window showing the 'hp MOBILE CENTER' interface, which includes a recording button and device details for 'asus-Nexus 7 [0a38ac2a]'. A central image shows a mobile device screen with various app icons, overlaid with a network diagram. A blue callout box on the right contains the text '実機の画面を投影して、Webアプリと同様の記録' (Project the real device screen to record like a web application). The diagram is flanked by 'オープン' (Open) on the left and 'ホリクス' (Horix) on the right.

参考) 継続的デリバリとデプロイメント

計画

開発

QA

ステージング

本番

ユーザー経験

協調;
可視化の欠如

時間短縮;
デプロイの長期化

自動化;
デプロイメントの
不整合

協調;
サイロ化された
ツール

パイプライン管理

宣言型
サービスデザイン

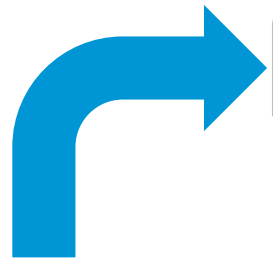
Infra as Code

オープンなAPI
他社ツール、OSSとの連
携・統合



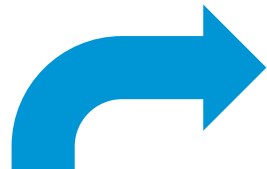
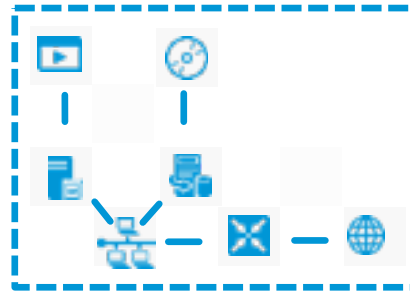
参考) Application パイプライン管理

APPLICATION : Pet Clinic



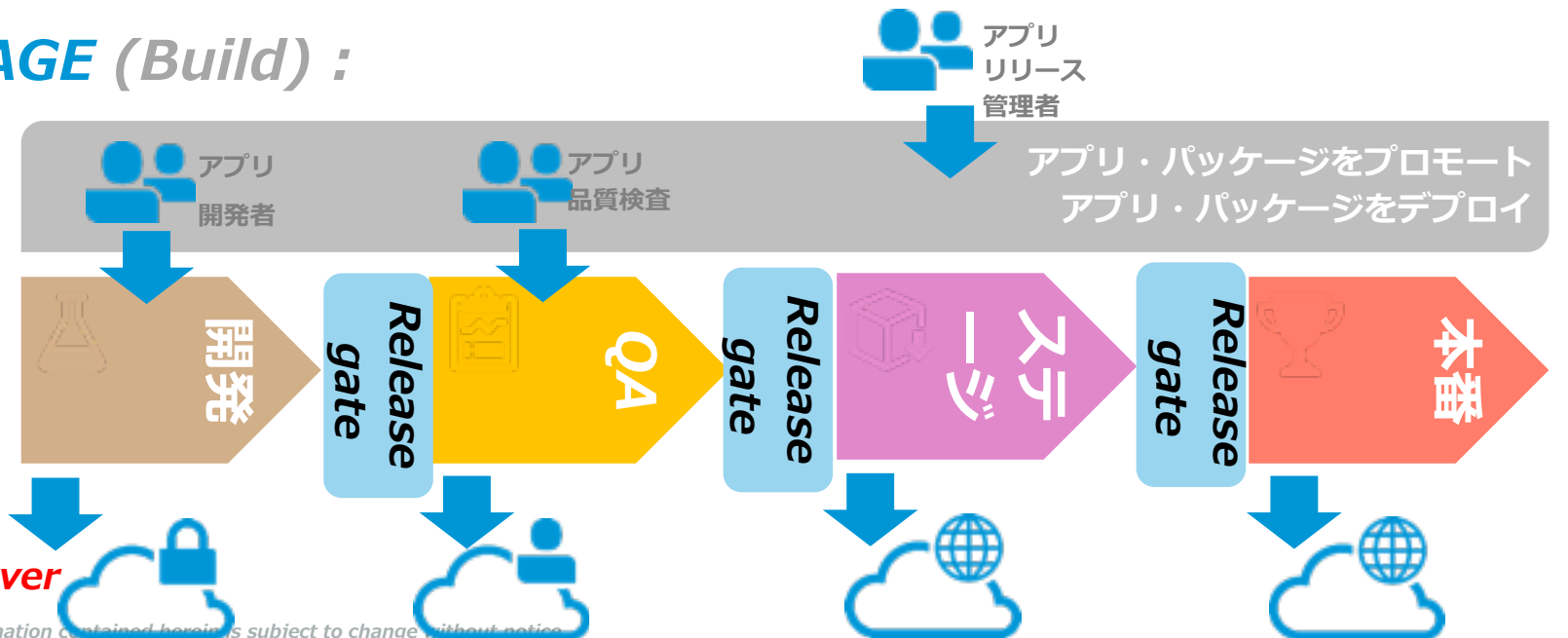
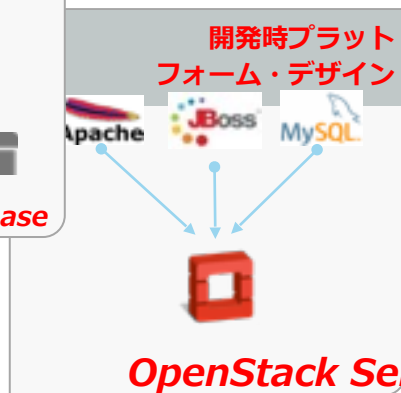
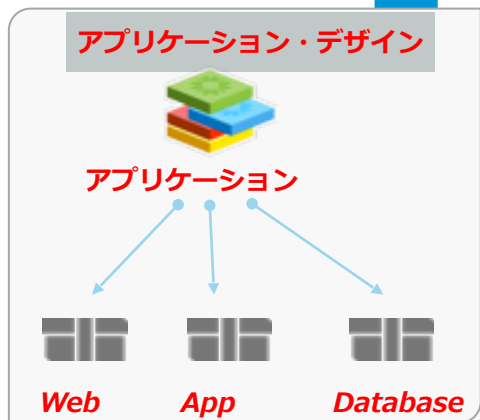
アプリ・リリース管理者 アプリ作成

APPLICATION VERSION :
1.0.0



アプリ設計者 アプリのクラウド・デザイン

PACKAGE (Build) :
333



参考) 継続的モニタリングとオペレーション

計画

開発

QA

ステージング

本番

ユーザー経験

継続的モニタリングとオペレーション; 短時間と高頻度に向けたシフト・レフト

不具合を認識・分析するためのデータ収集

- アプリケーションのライフサイクルの全てのステージを監視
- テスト中の不具合を検知し記録
- 不具合改修のために、開発部門が本番環境のデータへアクセス

実ユーザーのデータを活用

- テストを実ユーザーのアプリケーションフローに基づき実施
- 本番環境での負荷プロファイルをテストで利用
- 本番環境のパフォーマンスとベースライン、ステージング状況を監視し、テスト改善に利用
- *Blue Green Deployment*への対応

ユーザー経験を最適化

- 開発部門において、ユーザーの共通の操作パターンを分析し利便性を向上
- 予測的なオペレーションに拡大し、ユーザーの満足度向上に繋がるアップデートを図る

参考) 継続的プランニングとアセスメント

計画

開発

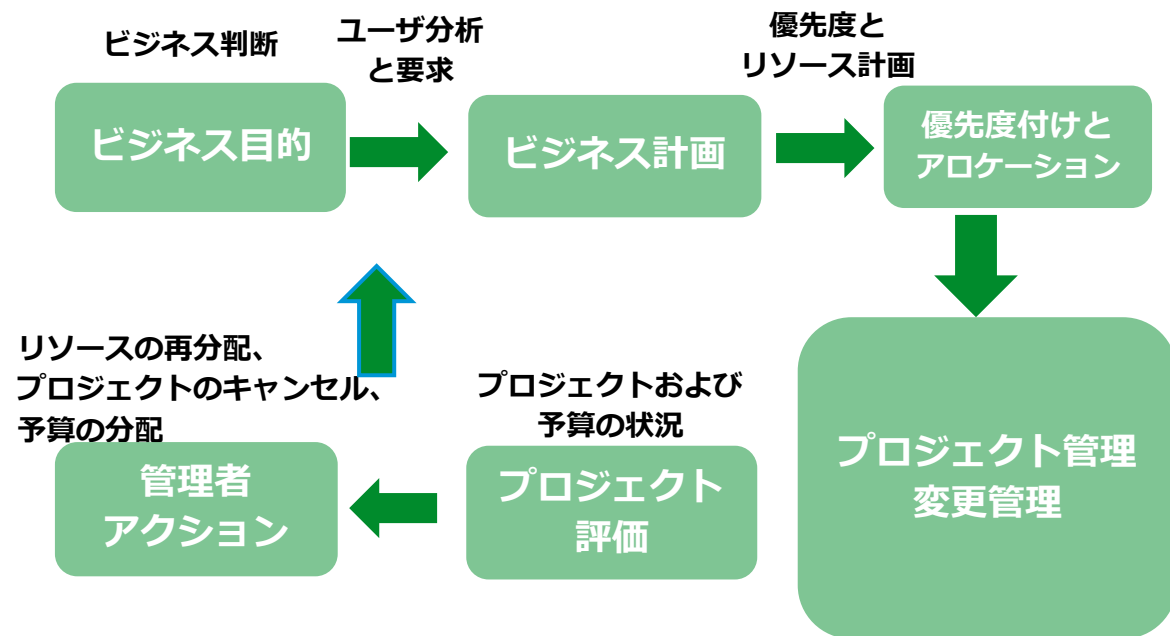
QA

ステージング

本番

ユーザー
経験

継続的プランニングとアセスメント; フィードバック、変更管理、分析の継続



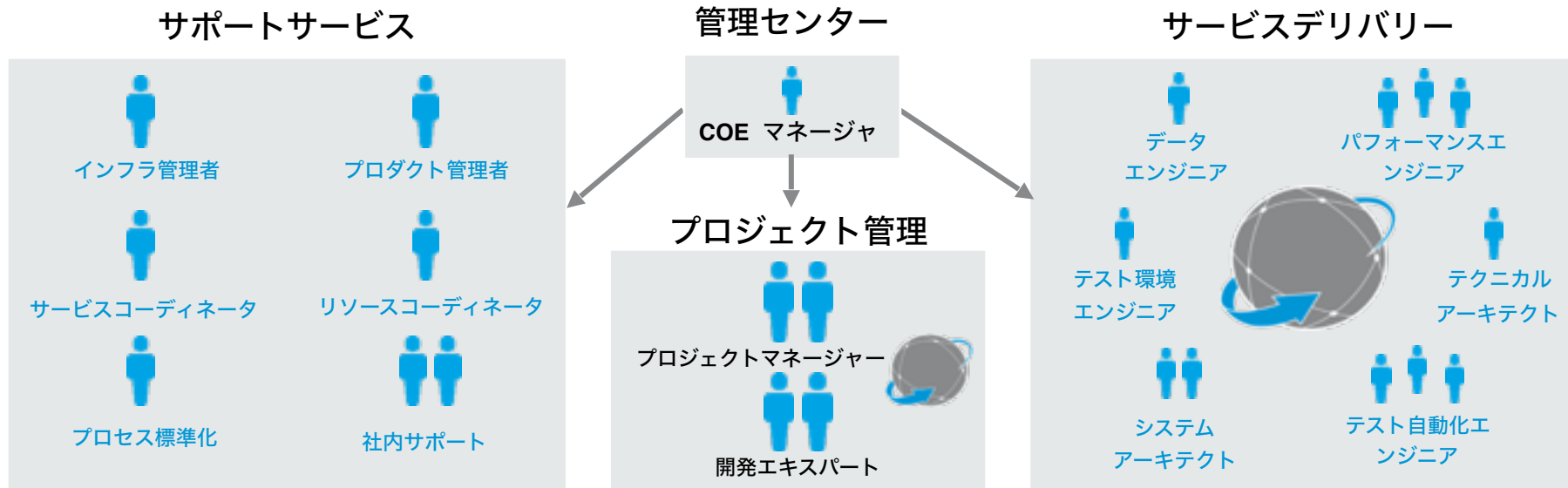
継続的アセスメント

- *Big Data*を活用した **Fundex**測定
- アプリケーションのユーザー経験を監視
- ピークの負荷時にスケールインとスケールアウト対応
- 自動化された修正機能

継続的プランニング

- ビジネス計画
- ポートフォリオ優先度付けと近代化
- 予算とリソースの配分
- ビジネスとオペレーションへの変更要求
- プロジェクト管理、コスト最適化

テストニングCoE(Center of Excellence)



事例

社内コンサルティングチーム

複数プロジェクト横断のサポートロール

標準化活動

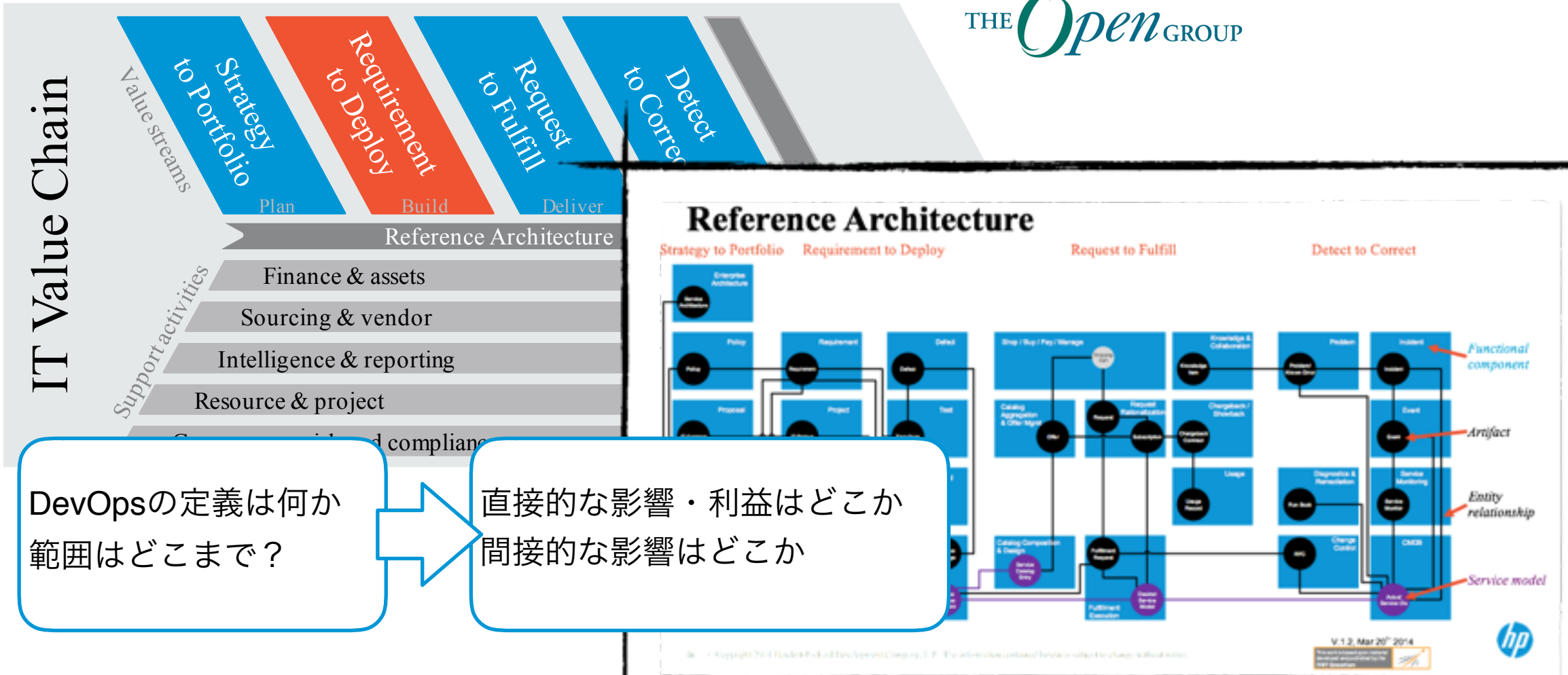
新規サポート対象/プロジェクトを立ち上げる専門家とリソースプール

- Capital One: テストコスト40% 削減
- Fedex: テストコスト50% 削減; テストサイクル時間70% 削減
- Citibank: 3年で \$26M 削減
- GMAC: 4年で \$21M 削減
- Vodafone: QAコストが25% 削減
- DirecTV: テスト工数70% 削減



見方を変えてみるはどうでしょうか？

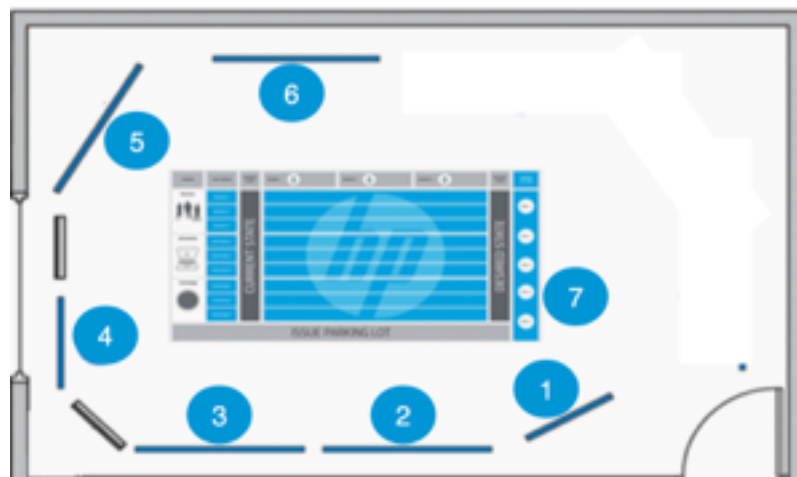
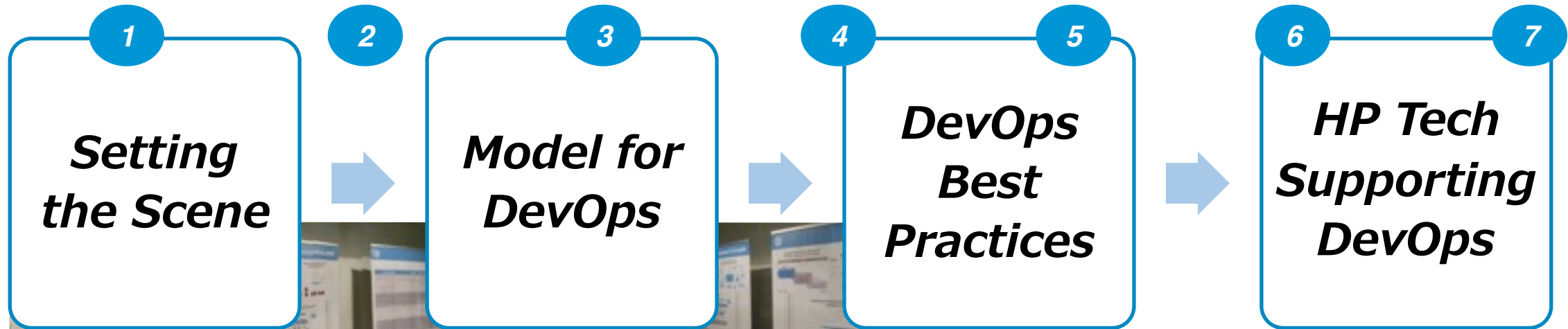
THE *Open* GROUP



DevOpsの定義は何か
範囲はどこまで？

直接的な影響・利益はどこか
間接的な影響はどこか

“DevOps Transformation Workshop”



DevOps ; 成熟度モデル

Draft

人、プロセス、ツール

Stage 1

個別最適、受動的

- ビジネスの要件変更への遅い対応
- 暫定的なコラボレーション、統合、コミュニケーション
- 予測不可能、制御不可能、受動的な方法
- 個別ソリューション
- 低い可視性
- 低いアプリの品質
- 高コスト
- 本番の切り戻しは高負荷

Stage 2

領域限定の専門性

- 自動化されたアプリテスト
- 模擬トランザクション監視
- 統合された運用管理
- 構成、変更、リリース管理
- 要件とポートフォリオ管理
- プロジェクト横断の品質標準 - *Centre of Excellence*
- コミュニケーション管理

Stage 3

機能横断の専門性

- 継続的なテストと統合
- 意思決定と説明責任の共有
- データ駆動型要件とポートフォリオ管理
- 実際のユーザー体験の理解
- 影響と意思決定の支援
- *KPI* ベースの共通目標、プロセス、マネジメントシステム
- プロセス標準と統合された自動化
- テスト環境管理

Stage 4

継続的コラボレーション

- 継続的なデリバリの自動化
- バランススコアカード - ビジネスゴールに対する評価
- 首尾一貫した可視性と予測性
- プラットフォームのセルフサービス化
- プラットフォームへの自動化されたリリース
- モデルベースのデプロイ (*Infrastructure as a code*)
- 継続的な欠陥解析
- 継続的なパフォーマンス管理
- 自動化された改善
- 自動化された再利用

Stage 5

卓越したDevOps

- 継続的なアセスメントを備えた結合された知性
- DevOps ベースのアプリ分析
- ユーザーパフォーマンス課題の予測
- 高いリソース利用率
- プロセスリスクとコストの最適化
- 効果的なナレッジの共有

ビッグデータによって推進される *Connected Intelligence*

本セッションの流れ

- なぜ、“バズワード”と見られるのか？
- 解決へのアプローチ
- ☑参考情報 ~ちょっと会社を離れて

DevOps戦略のカatalog

<http://disciplinedagiledelivery.jp/>

「ディシプリンド・アジャイル・デリバリー – エンタープライズアジャイル開発への実践ガイド」日本語版公式サイト

ホーム About

DevOps戦略：組織的リリース管理の戦略 – DDevOps#8

2015/06/21 に 藤井 智弘 が投稿

今回のテーマはリリース管理戦略をまとめ上げるにあたっての2つの課題について取り上げる。その2つとは、リリース管理の範囲をどうやって決めるかとチーム構成だ。

リリース管理作業の範囲を考慮する際に、2つの基本的な課題がある。

- 1. パラダイムのサポート** あなたの組織のリリース管理プロセスがフォーカスをあてているのは単一のパラダイムのチーム…例えばアジャイル/リーンチーム…だけだろうか？あるいはもっと広範囲な…アジャイル/リーンチーム、従来型の手法を採用するチーム、反復型を採用するチーム、そしてアドホックでプロセスに則らないチームまで…サポートを検討しているだろうか？リリース管理については現在著述している人々の多くが単一のパラダイムにフォーカスをあてる傾向がある（そのように明確に記してはいないが）。しかし、実際には複数のパラダイムが必要とされているのが、エンタープライズの現実だ。
- 2. ドメインのサポート** あなたのリリース管理プロセスは、ITに関連する課題だけにフォーカスをあてているだろうか？それともビジネスに関わるリリースの課題すべてを扱おうとしているのだろうか？ITに関連する課題には、新しいソフトウェアとハードウェアを運用環境にデプロイすることが含まれる。ビジネス面でのリリースの問題には、2, 3例を挙げると、マーケティングキャンペーンや販売組織へのトレーニング、エンドユーザーのための外部支援の枠組みの構築等がある。これは、エンドユーザーに向けて構築された商業的なソリューションであれば、とりわけ重要な要素だ。

日本語版書籍（発売中）



Disciplined Agile Delivery 日本語版書籍「ディシプリンド・アジャイル・デリバリー エンタープライズ・アジャイル実践ガイド（翔泳社）」好評発売中

最近の投稿

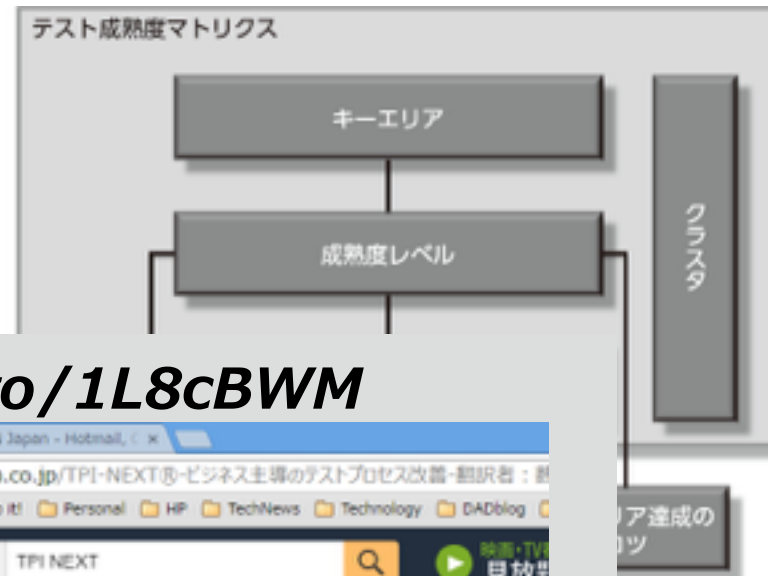
- DevOps戦略：組織的リリース管理の戦略 – DDevOps#8
- DevOps戦略：サポート – DDevOps#7
- DevOps戦略：運用 – DDevOps#6
- DevOps戦略：災害対策 – DDevOps#5
- DevOps戦略：チーム構成戦略 – DDevOps#4

カテゴリー

- DAD解説 (11)
- Uncategorized (6)
- セミナー (4)
- メディア (1)

BDTPI (TPI NEXT)

- テストプロセスの改善に特化した成熟度モデル
 - プロセス参照型モデル（フレームワークとロードマップの提示）
 - 連続的モデル
 - 実行順序の自由度が高く、組織はもっとも優先度の高い問題に対応



キーエリア

	初期レベル	コントロールレベル				文
1 利害関係者のコミットメント		1	2	3	4	1
2 関与の度合い		1	2	3	4	1
3 テスト戦略		1	2	3	4	1
4 テスト組織		1	2	3	4	1
5 コミュニケーション		1	2	3	4	1
6 報告		1	2	3	4	1
7 テストプロセス管理		1	2	3	4	1
8 見積もりと計画		1	2	3	4	1
9 メトリクス		1	2	3	4	1
10 欠陥管理		1	2	3	4	1
11 テストウェア管理		1	2	3	4	1
12 手法の実践		1	2	3	4	1
13 テスト担当者のプロ意識		1	2	3	4	1
14 テストケース設計		1	2	3	4	1
15 テストツール		1	2	3	4	1
16 テスト環境		1	2	3	4	1

<http://amzn.to/1L8cBWM>



まとめ

- ✓なぜ、バズワードと感じているのか？に目を向けよう
- ✓自分達の置かれた**“文脈”**を明確にしよう
 - ✓安直なコスト削減メッセージ？
 - ✓夢みたいなビジネス貢献メッセージ？
- ✓「開発と運用をつなぐ」というなら、**閉じこもるのはやめよう**

ツールはもちろんですが（よろしく申し上げますm0m）、それ以外にもいろいろお手伝い出来ます！



ご清聴ありがとうございました

