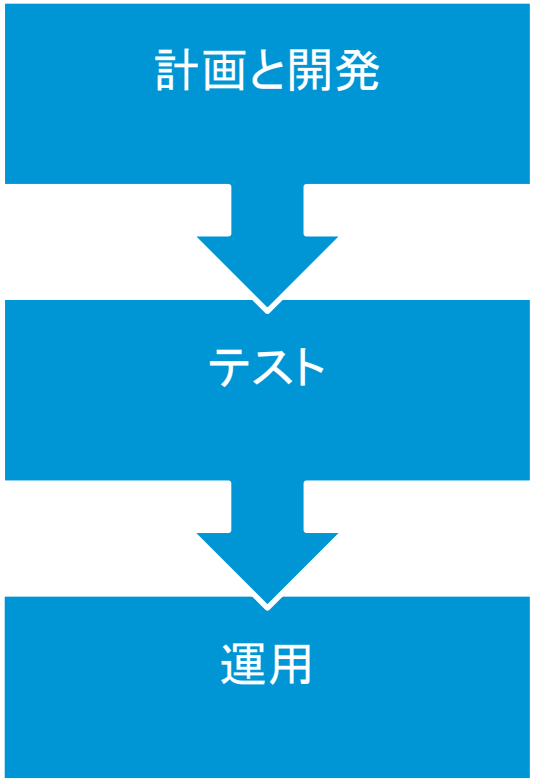
DevOps時代の
モバイルアプリケーション
継続的テストフレームワーク

日本ヒューレット・パッカード株式会社
HPソフトウェア事業統括 ITマネジメントプリセールス本部 テクニカルコンサルティング部
シニアコンサルタント　小宮山 晃

モバイルアプリのデリバリ課題



- 正しい要件を如何に定義するか
- 要件は常に変化してい
- 要件と開発の間でトレーサビリティが低い

アジャイル方法論

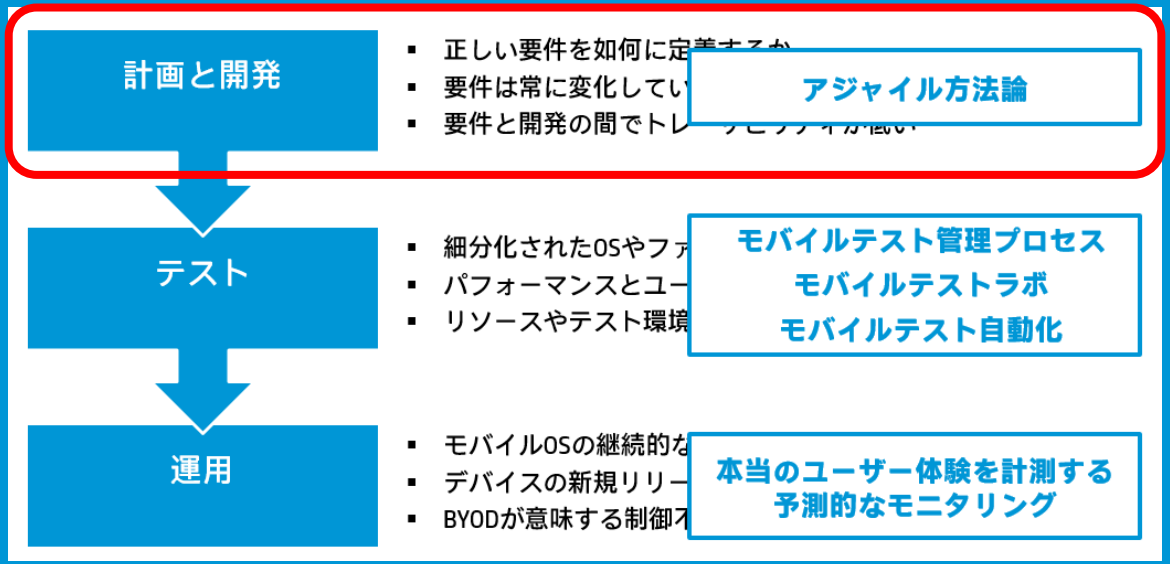
- 細分化されたOSやファー
- パフォーマンスとユーザ
- リソースやテスト環境の

モバイルテスト管理プロセス
モバイルテストラボ
モバイルテスト自動化

- モバイルOSの継続的な
- デバイスの新規リリース
- BYODが意味する制御不

本当のユーザー体験を計測する予測的なモニタリング

計画と開発



エンプラ向けアジャイルプロセスを考える

スケール(大規模システム、地理的分散チーム、コンプライアンス等)をもったプロセス

最もアジャイル開発手法として浸透しているSCRUM(注)ではスケールをもったプロセス実行が難しくなる。

そのため、企業が求めるスケールに答えるためのプロセスが登場している (SAFe, DAD, LeSS)。

ウォータースクラム &
スクラムフォールに陥らない

なぜなら

要件定義フェーズが長く開発が遅れる
本番環境へのデリバリーは従来どおり

市場投入機会の見逃し

(注) VERSIONONE「9th Annual State of Agile Survey」調査結果によるとSCRUM 56%, SCRUM/XP Hybrid 10%であわせると66%で採用されている

経営層の意思決定からチームレベルへ

SAFeでは Portfolio, Program, Team の3層でアジャイルのスケールに対応

Portfolio

ポートフォリオ層では、戦略テーマ(Thema)から企業や事業部門で取り組むべき要望(Epic)を管理

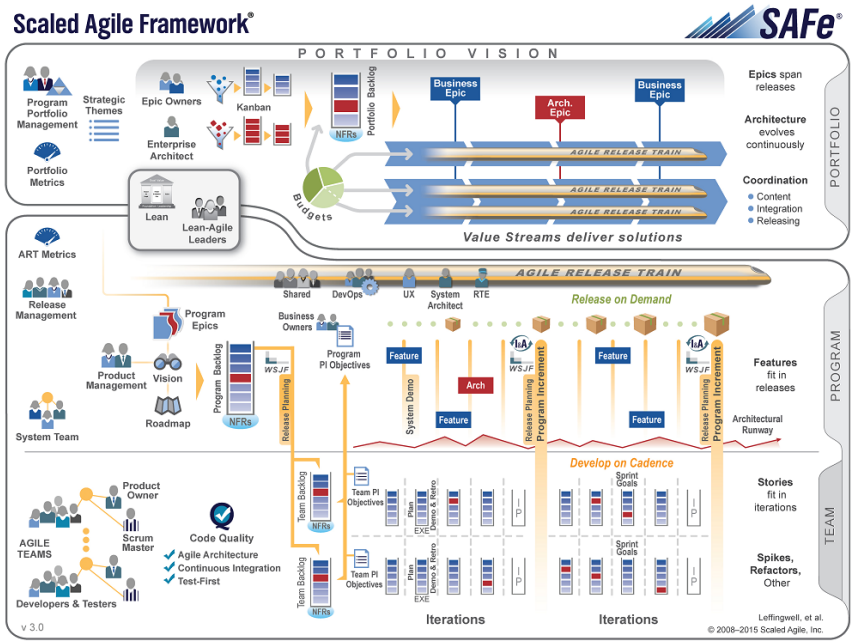
Program

プログラムレベルではエピックを実現する機能(Feature)の管理と各チーム割り当てを管理

Team

チームレベルでは、機能(Feature)からユーザーストーリ(US)に落とし開発

チームレベルのコアはSCRUM



(出典) http://www.scaledagileframework.com/

HP Agile Manager

お客様の抱える課題

❑ オフショア等の分散開発でアジャイル開発を試みているが状況確認(品質状況も含めた)が出来ていない
❑既存の資産(OSS環境)間の情報連携に苦労している
❑初期段階でコストをかけたくない

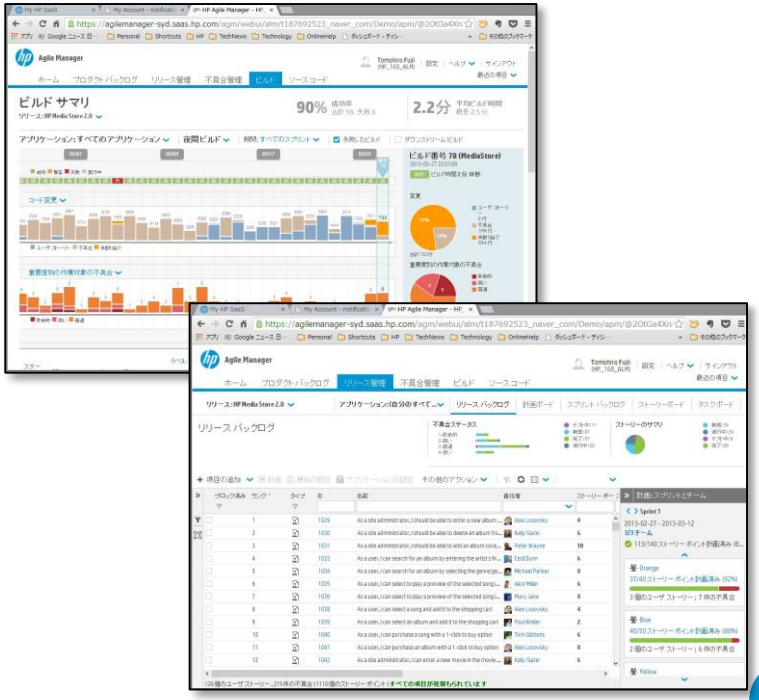
HPソフトウエアによる解決

❑SAFeに準拠したアジャイル・ポートフォリオ管理を提供
❑SaaSおよびオンプレミス両形態で提供
❑弊社のライフサイクル管理製品(ALM等)と連携することで、テストまで含めた全体の管理が可能

お客様の価値

❑専用クライアント不要の直感的なWebベースUIで、学習時間・コストを低く直ちに開始が可能
❑提供形態はSaaSまたはオンプレミスを条件にあわせて選択が可能
❑既存資産(OSS環境)連携し進捗と品質状況管理が向上

バックログ、不具合、テスト結果まで含めたアジャイルプロジェクトの進捗管理を可視化



テスト管理はスプレッドシートのまま？

モバイルアプリケーションのテストはWebシステム等と比べてテストケース数が肥大化しスプレッドシートでのテスト管理は限界に来ていませんか

- プラットフォームの多様性への対応（マルチプラットフォーム対応）
  - iOS, Android OS, Windows Mobile ….
- 複数バージョン対応（市場では複数バージョンが混在）
- 複数デバイス対応
- 各デバイスベンダ（Manufacturer）より一定周期で出荷される新デバイス対応

HP ALM/QualityCenter

お客様の抱える課題

❑オフショア等の分散開発体制で、プロジェクトや品質に関する状況が把握できず、品質向上が改善できていない
❑品質に対して不安。定量的な指標や資産の再利用で、組織力を高めたい

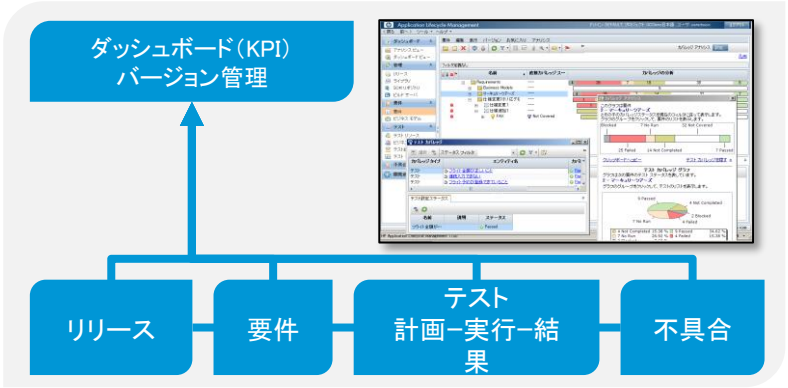
HPソフトウエアによる解決

❑要件、テストケース＆結果、不具合情報を構造的に管理でき、品質状況にかかる管理コストを削減
❑各種レポーティングおよびKPIにより、品質状況をリアルタイムに把握、プロアクティブな対応が可能
❑OSSも含む目的に応じた様々なサードパーティツールと連携することで品質管理のフレームワークを構築可能

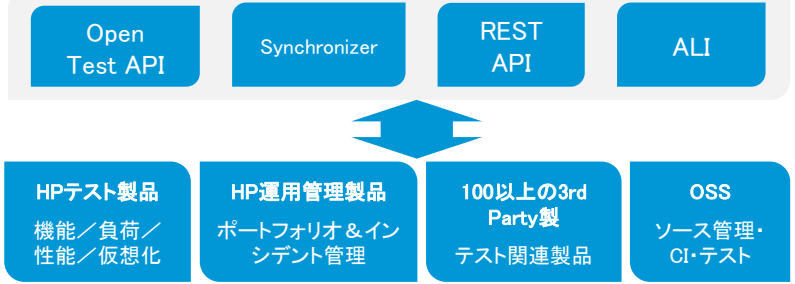
お客様の価値

❑OSS等既存資産と連携し短期間でライフサイクル管理環境を構築可能、即座にプロジェクトの可視化が実現可能
❑成功したプロジェクトの資産を再利用することで組織力の向上に寄与

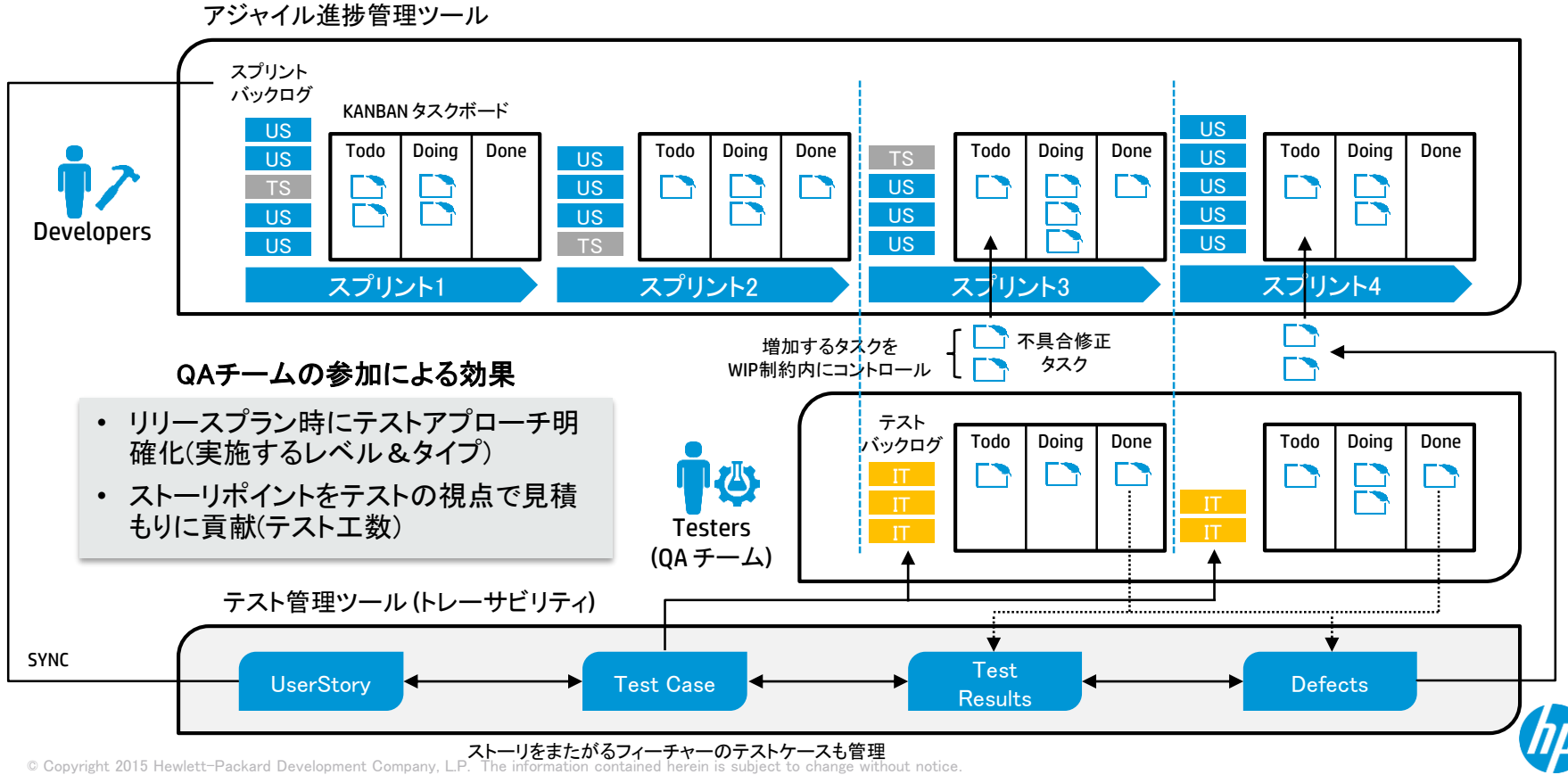
HP ALM/QC 品質情報管理コンポーネント構成



他ツール連携用HP ALM/QC API



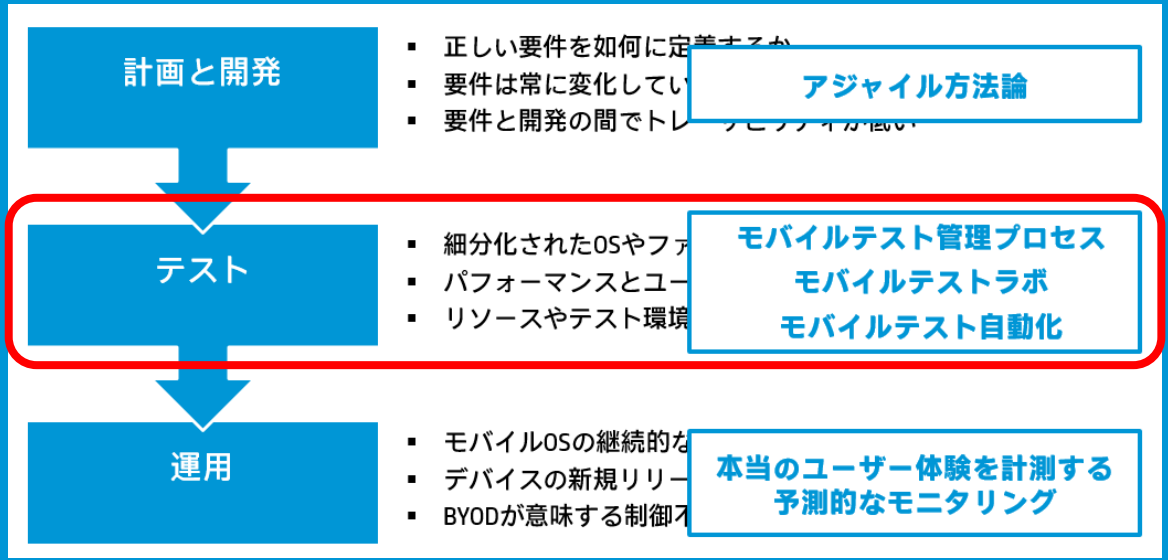
QAチームの積極的なSCRUMチームへの参加



QAチームの参加による効果

- リリースプラン時にテストアプローチ明確化(実施するレベル＆タイプ)
- ストーリポイントをテストの視点で見積もりに貢献(テスト工数)

テスト



モバイルテストにおけるテストアプローチ

- SDK内にフレームワークも存在
- デバイス依存問題検出の遅れ
- 条件はソフトシュミレーションであるため実デバイスでの動作見逃しの恐れ

単体テスト
フェーズで有効

- 実デバイスによる動作検証
- 実デバイスの準備とセットアップ、アプリのインストールが必要

結合テスト以降フェーズ
で限定的(注)に有効

- 多種多様な実デバイスによる動作検証
- 希望セグメントにあわせたユーザーに近いテストが可能
- 手動テスト

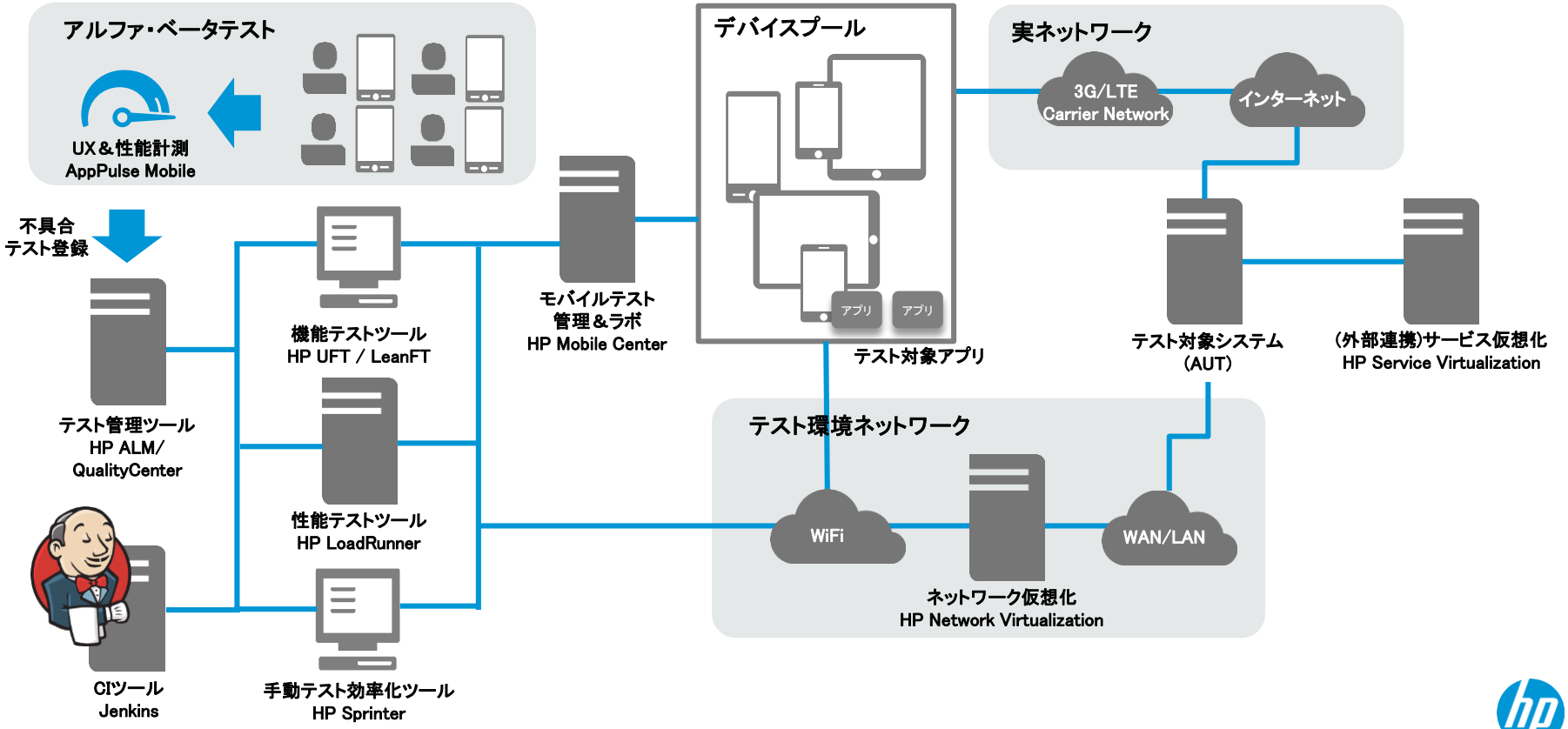
UXテスト、
ベータテスト
として有効

- 多種多様な実デバイスによる動作検証
- 自動テスト適用
- 既存CD、ALMプロセスへの適用

テストカバレッジ拡張、
統合フレームワークに有効

(注)通常、想定している機種すべてを準備することは難しい

モバイルアプリケーションテスト環境構成例



HP Mobile Center

お客様の抱える課題

❑複数デバイス、複数バージョンが存在するモバイルアプリ動作チェックにかかるテスト工数の削減と、品質の向上
❑リリース後に発生するモバイルアプリのサービス動作確認にかかる監視工数の削減

HPソフトウエアによる解決

❑マニュアルテストの自動エビデンス取得＆テスト自動化（機能テスト、性能テスト）によるテスト工数の削減
❑複数端末を一元集中管理する事でバージョン、デバイスと不具合、テスト、要件へ横断トレーサビリティを提供
❑モバイルアプリのサービス監視を提供し、サービス障害発生の検出＆通知と障害発生操作の特定を迅速化

お客様の価値

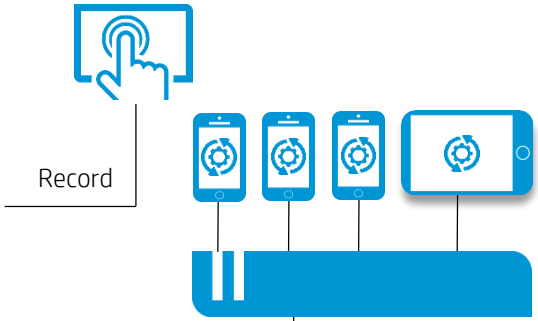
❑これまで手動、または様々ツールで対処していた作業を、一元的に管理・実行し、テスト＆運用工数の圧縮を実現

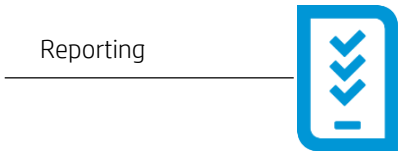
ライフサイクルを通して必要とされる機能を
センター化して提供

これまでのHP ALM（ライフサイクル管理）手法を
モバイル領域まで拡張し、実績ある管理手法に統合

機能テスト(テスト自動化)

機能テストツール連携(HP UFT)

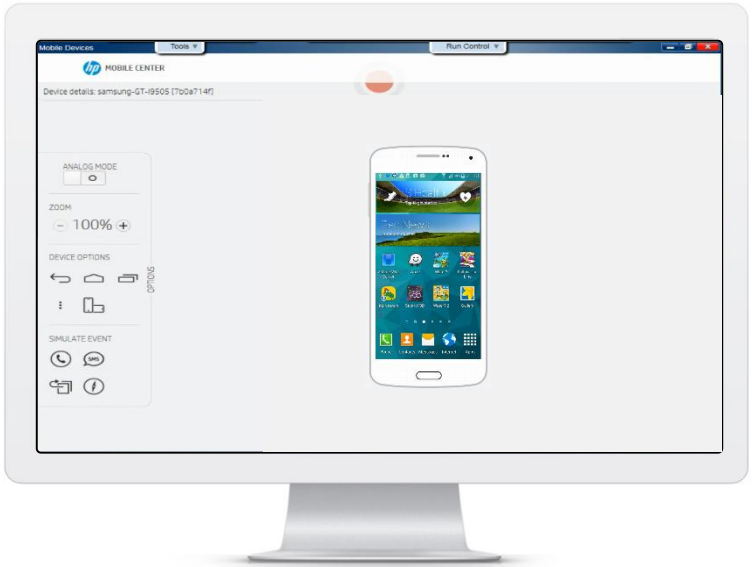




- オブジェクト認識
- マルチデバイステスト
- チェックポイント
- 継続的テスト
- デバイスログ
- 結果レポート
- 不具合自動登録

機能テスト(手動テスト、探索的テスト)

手動・探索的テストツール連携(HP Sprinter)

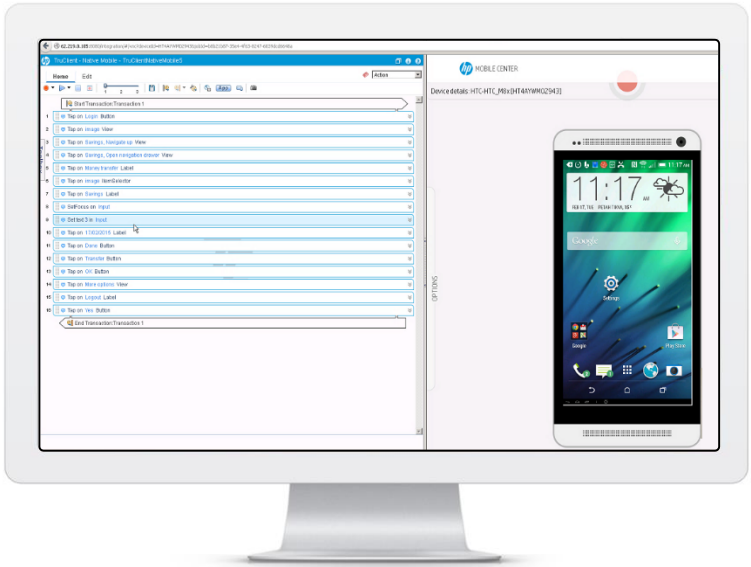


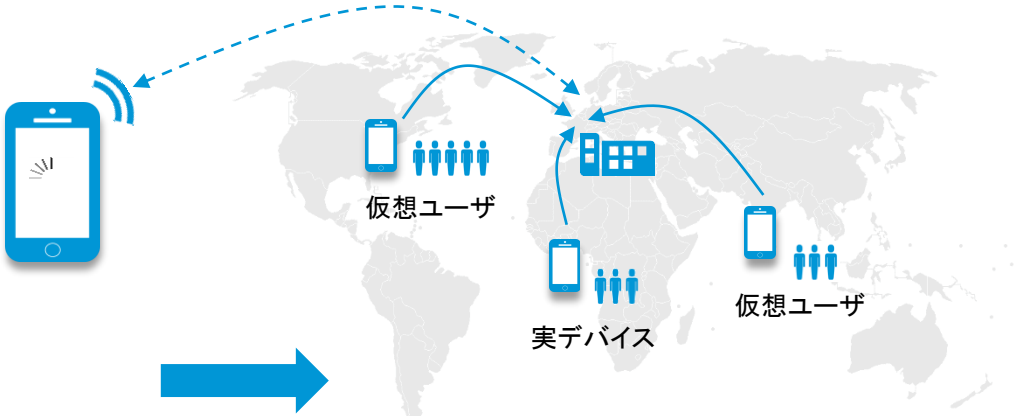
リモート
スクリーン同期

- 操作ログ収集
- エビデンス画像収集
- テスト管理ツール連携
  結果レポート
- 不具合登録

性能テスト(負荷テスト、限界値テスト)

性能テストツール連携(HP LoadRunner)



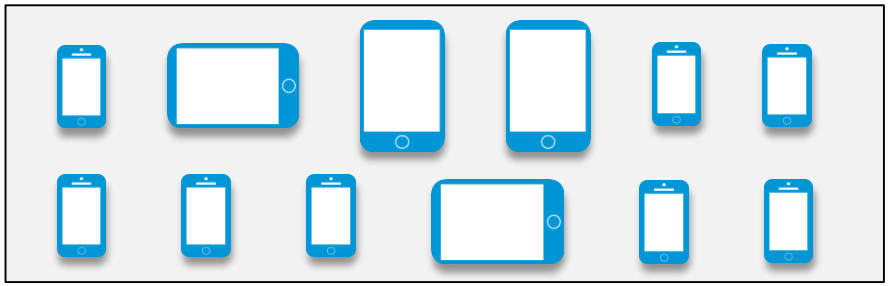


高負荷時の実デバイス機能テスト
モバイル通信のダンプデータからスクリプト生成
テスト環境におけるネットワーク品質仮想化

クラウド型環境を提供するテストラボ

ラボ管理機能(HP Mobile Center)

デバイスとOSの管理



USB(今後WiFi対応予定), プラットフォーム, デバイス

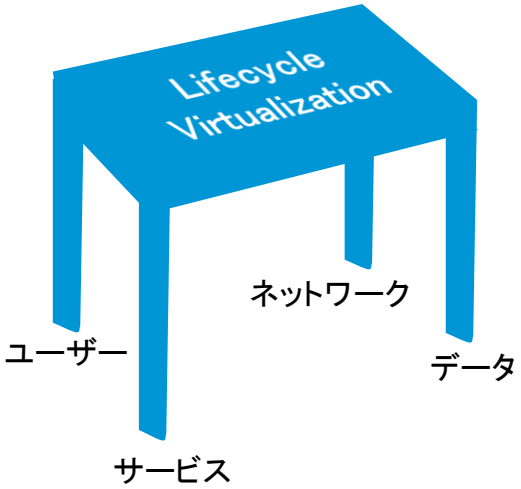
アプリの管理

配布, インストール, アンインストール

性能テストのトレンド 1

Lifecycle Virtualization の説明と必要性

クラウドやモバイルの実環境を検証環境で再現するための仮想化を性能テスト時に利用するようになってきました。HPでは仮想技術「Lifecycle Virtualization」と称し、次の4要素で構成しています。



ユーザー（クライアントアプリケーション）

実ユーザーの操作(画面遷移)、ペーシング、同時接続数、TPSを実現すること

操作対象クライアントアプリケーションのテクノロジー、プロトコルに対応していること
(SPDY/HTTP2.0, SPA, WebSocket)

サービス

実外部サービスの応答時間をシュミレートし、かつ性能テスト時の負荷に耐えられること

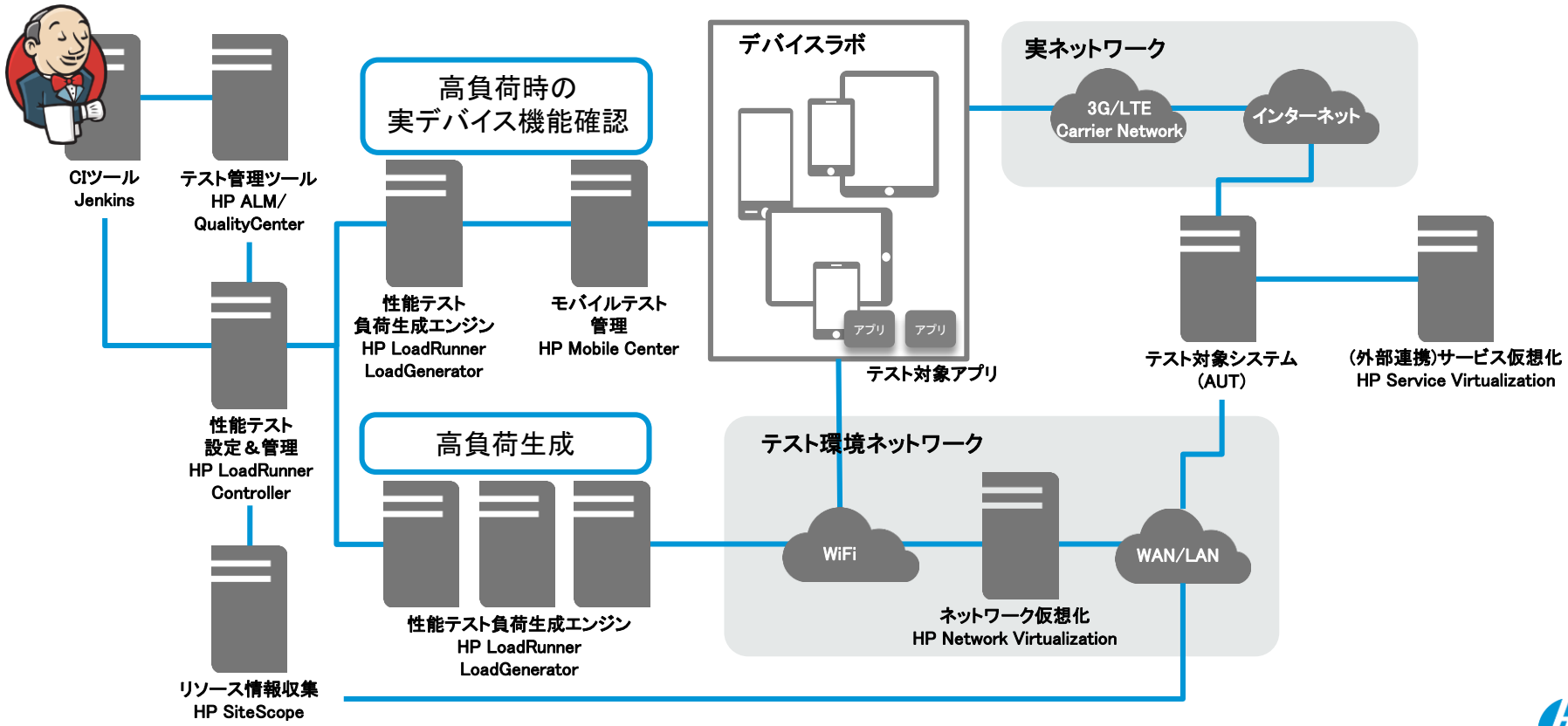
ネットワーク

想定される、モバイルアクセス、クラウドアクセス時のネットワーク品質(パケットロス、回線帯域)をシュミレーションできること

データ

サービス仮想化、ユーザー仮想化時に送信、応答データを実ユーザーを想定したデータをシュミレーションできること

Performance Lifecycle Virtualization ポートフォリオ

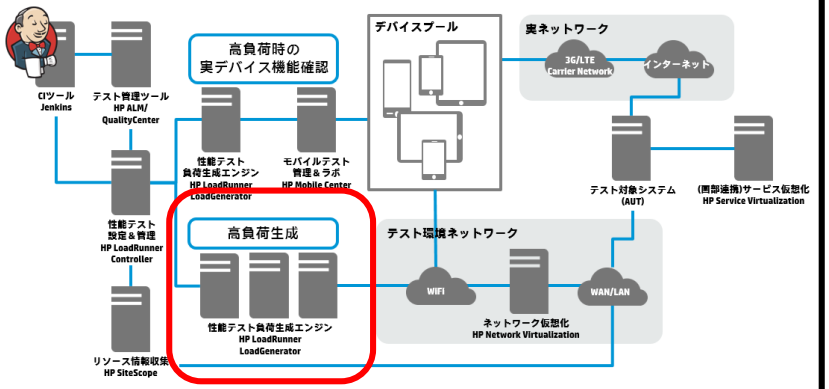


性能テストのトレンド 2

システムテスト環境構築の準備期間やリソース(人や費用)は変化

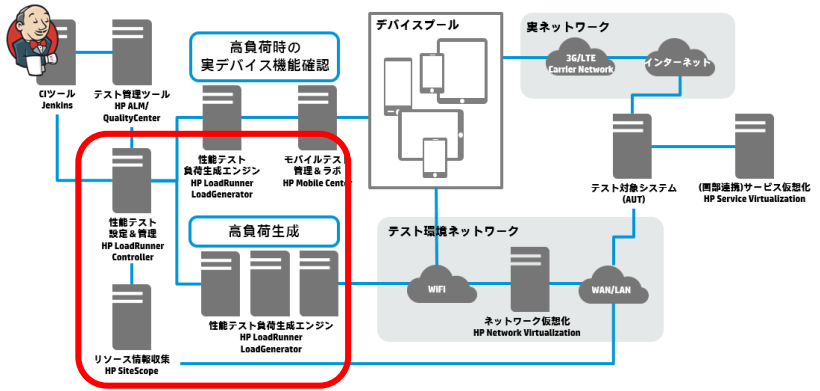
テスト対象システムはクラウド環境上で柔軟に拡張が可能になってきています。それに伴い負荷生成エンジンマシンも生成負荷量にあわせて柔軟に拡張できなければなりません。

解決インフラパターン1



HP LoadRunnerでは設定＆管理マシンから負荷生成エンジンをAWS、Azure上に構築(10分程度)

解決インフラパターン2



設定＆管理、リソース収集、負荷生成エンジン全てをクラウド上のサービスとして利用

HP StormRunner Load (解決インフラパターン2)

お客様の抱える課題

❑負荷テストの準備にかかる工数や機材コストを削減したい
❑大規模な負荷テストを迅速に行い、早急にボトルネックを発見したい

HPソフトウエアによる解決

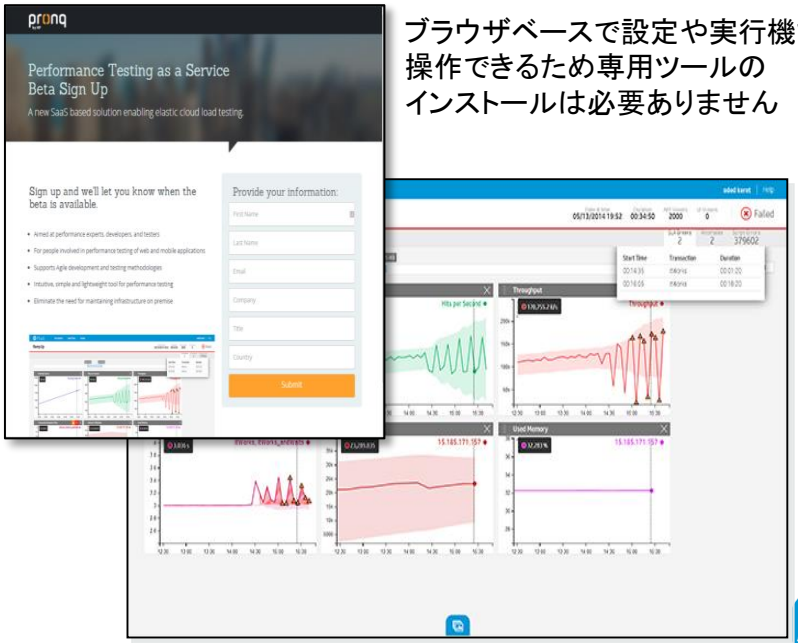
❑完全SaaS型の負荷テストソリューションで、ブラウザ越しにスクリプト作成が行え、負荷生成端末等の機材準備等を意識する必要なく、負荷テストの実行が可能
❑1ユーザから百万ユーザまで、世界中のクラウドを利用し大規模な負荷を迅速にかけることが可能

お客様の価値

❑これまでローカルで機材準備を行い、各設定を必要としていたコストを削減し、クイックに大規模な負荷テストを実行することでボトルネックを発見することが迅速に行えます

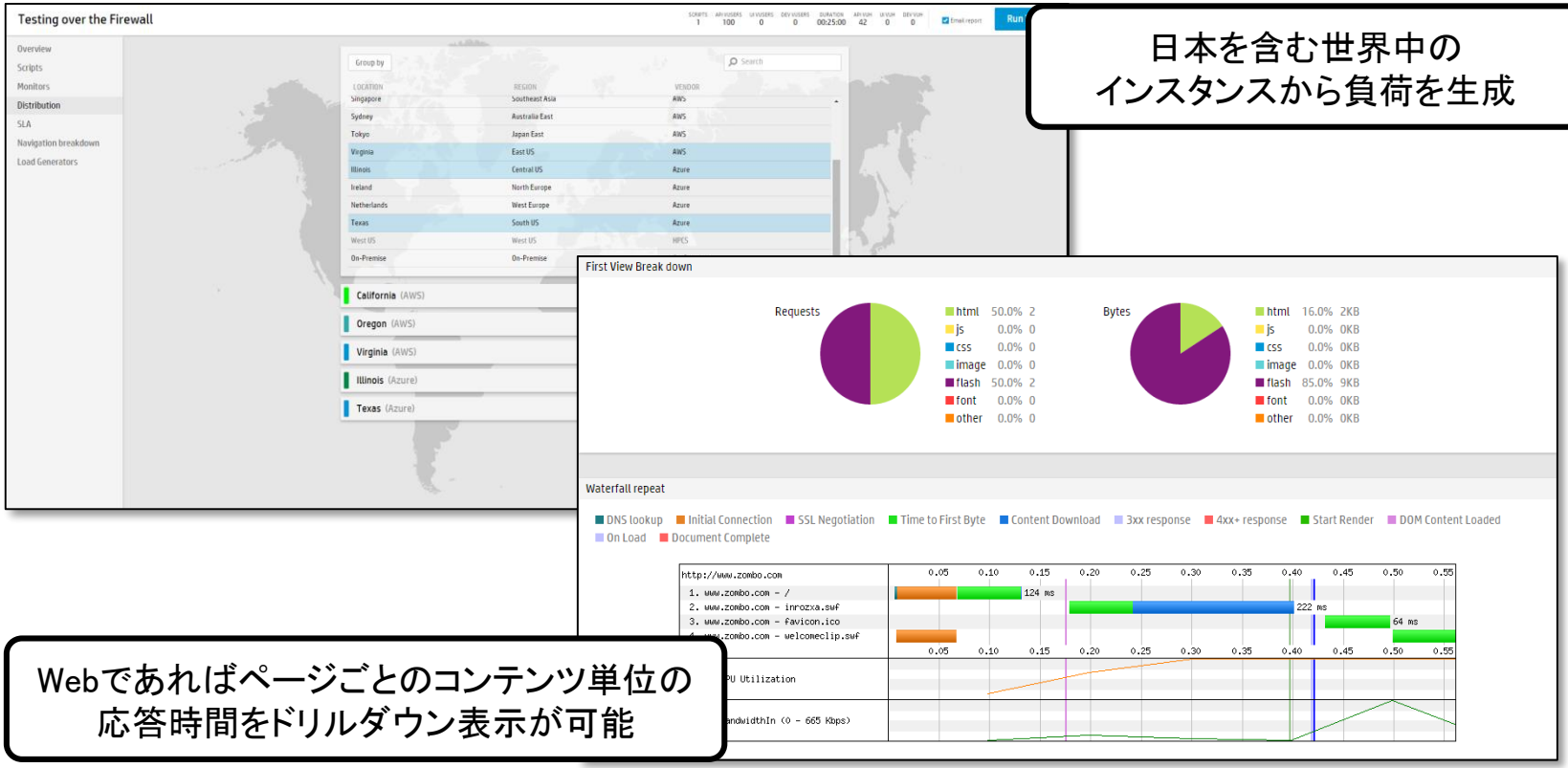
SaaS型の負荷テストソリューション

Webポータルで契約し、スクリプト作成、実行、分析すべてをクラウドサービスとして利用でき、即座に大規模負荷テストを実行することが可能になります。



ブラウザベースで設定や実行機能を操作できるため専用ツールのインストールは必要ありません

SaaS 提供で短時間の性能テスト準備＆実行可能



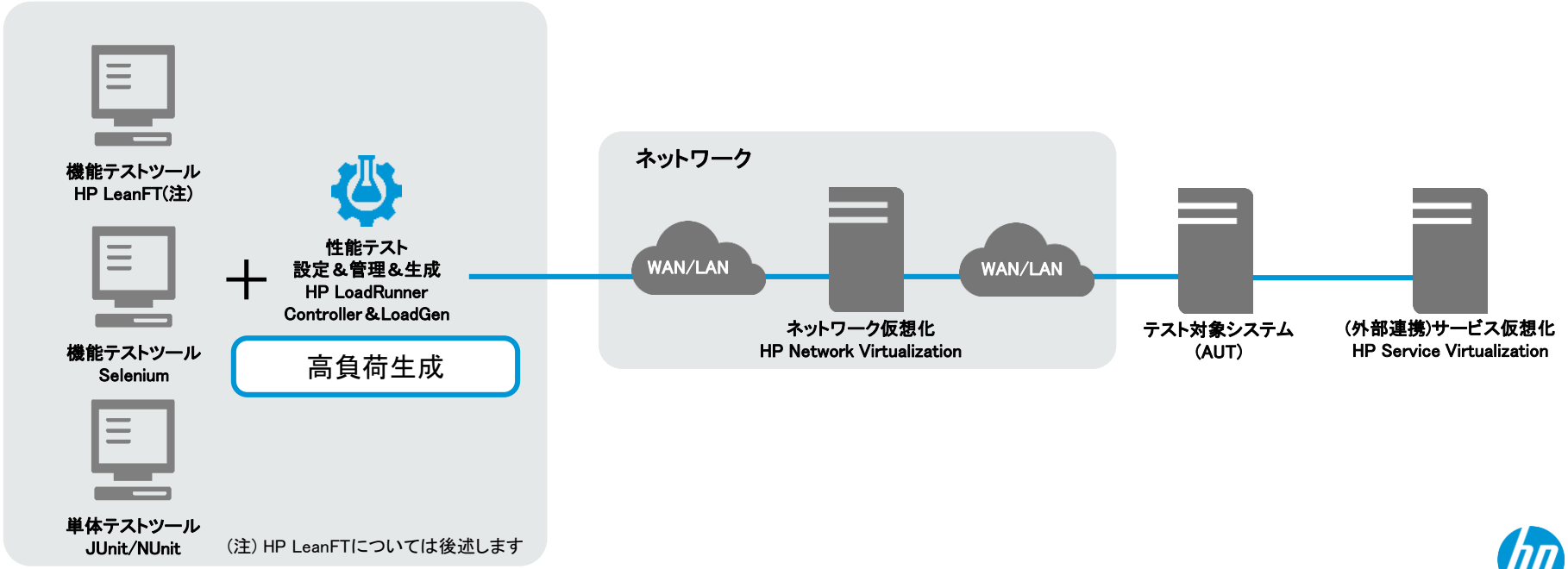
日本を含む世界中の
インスタンスから負荷を生成

Webであればページごとのコンテンツ単位の
応答時間をドリルダウン表示が可能

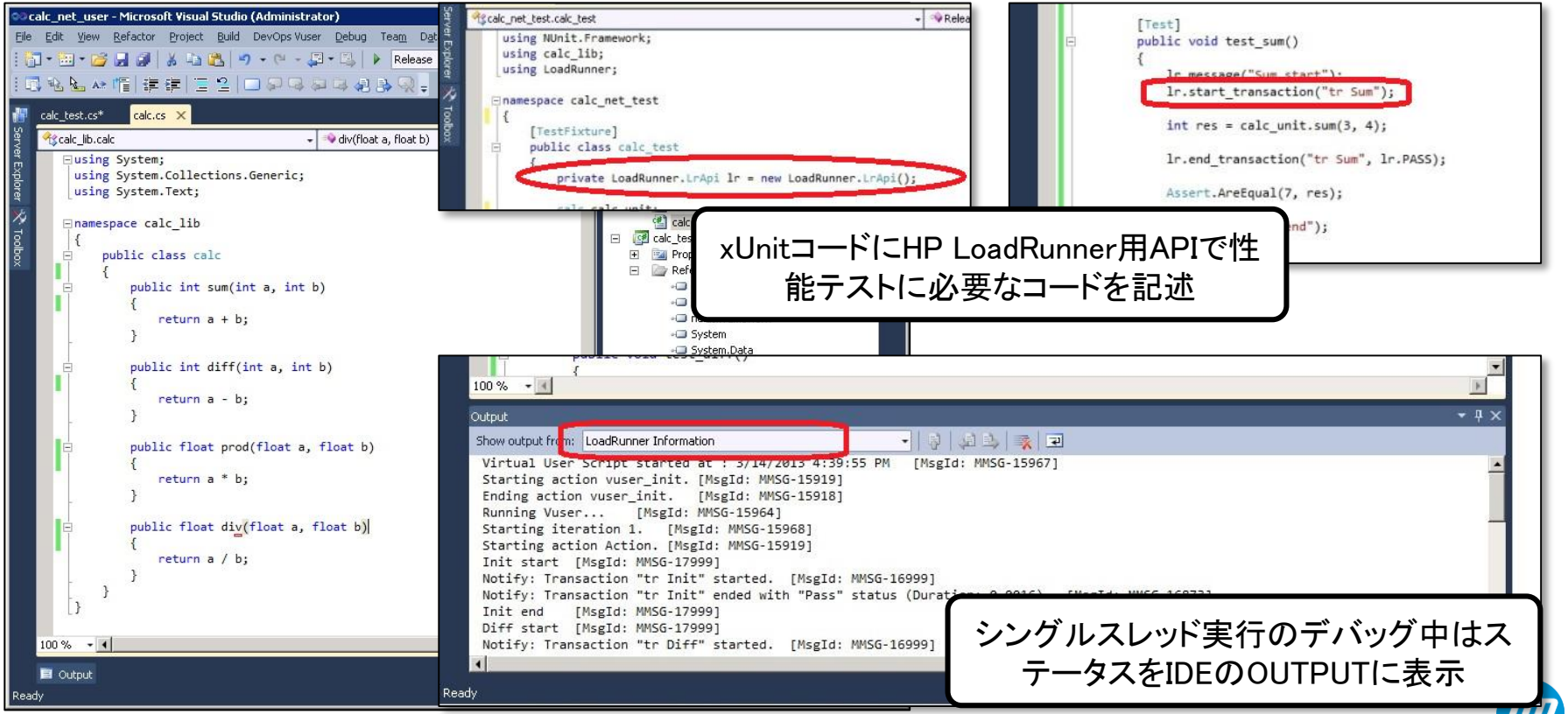
性能テストのトレンド 3

開発テストアセットの活用

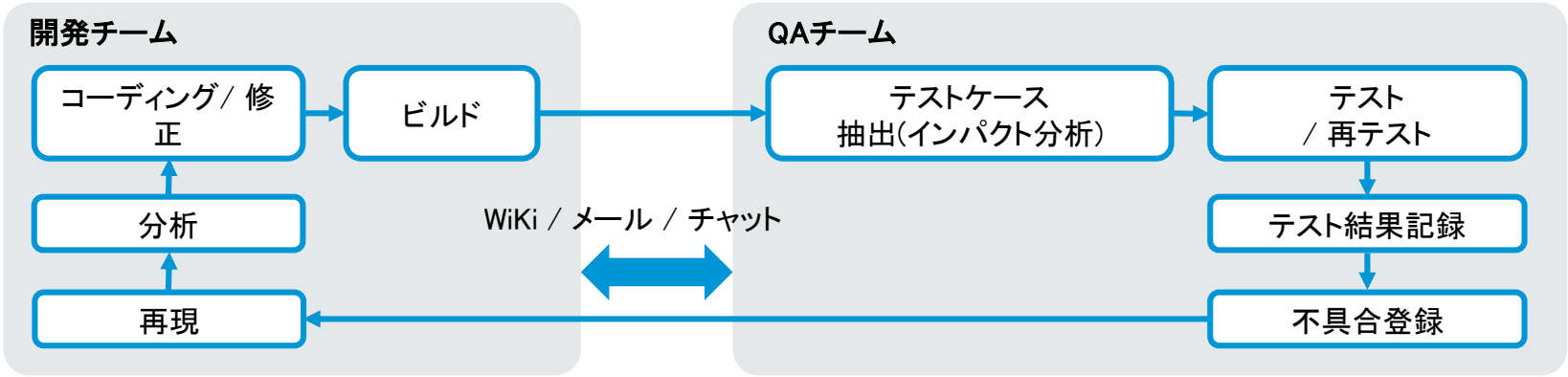
早期から性能要件確認のため開発資産を活用しPLVと併せてShift-Leftを実現



単体テストアセットで多重セッション生成

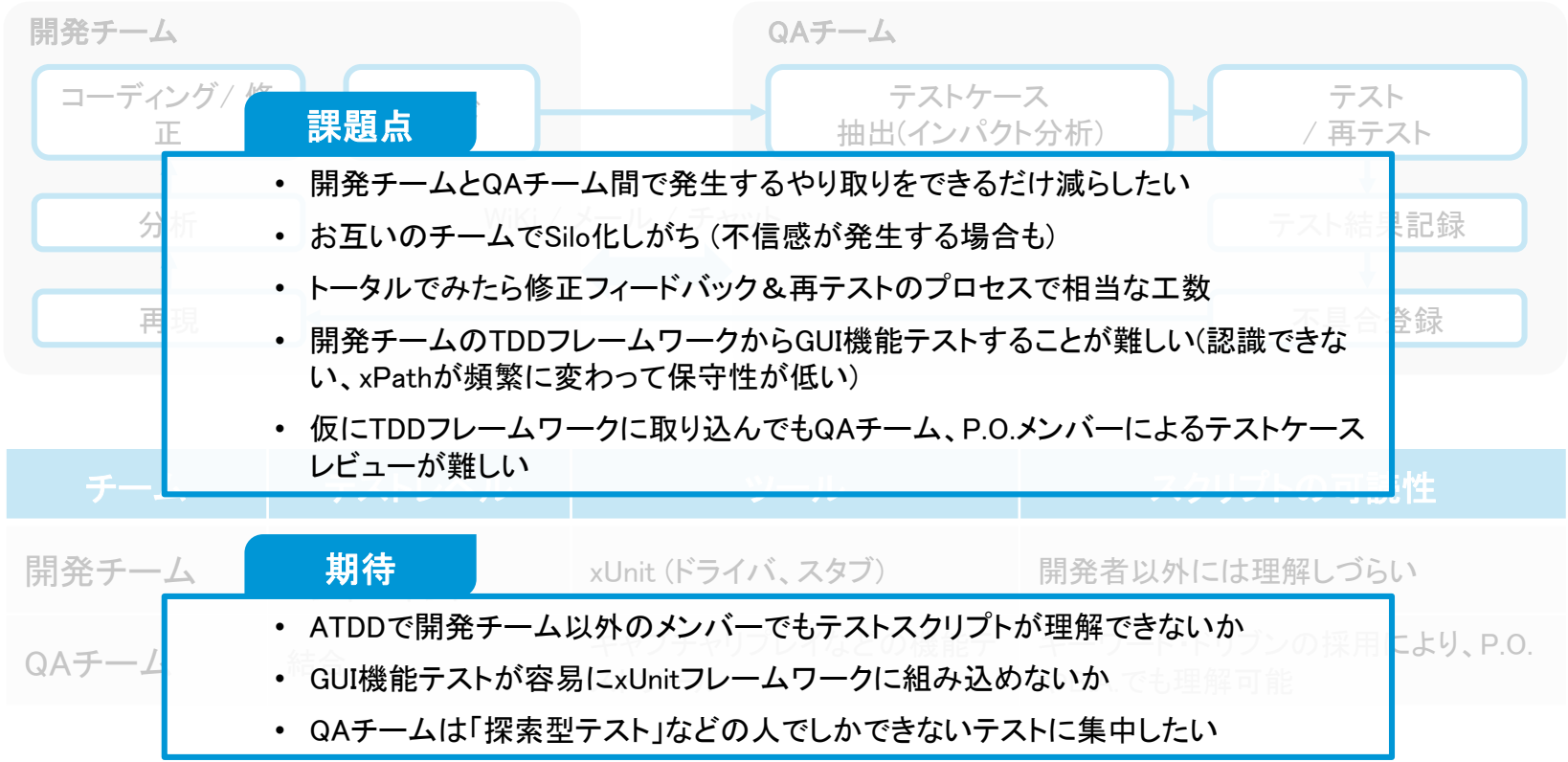


一般的な機能テスト作業プロセスと環境



チーム	テストレベル	ツール	スクリプトの可読性
開発チーム	単体、 コンポーネント	xUnit (ドライバ、スタブ)	開発者以外には理解しづらい
QAチーム	結合	キャプチャリプレイなどの機能テストツール	キーワード・ドリブンの採用により、P.O.やB.A.でも理解可能

アジャイル開発時の機能テスト作業課題



HP LeanFT

お客様の抱える課題

❑QAチームによる機能テスト実施で発生する不具合＆再テスト(影響範囲の他テストケースも実施)で発生する開発チームとのやり取りがアジャイル開発で無視できない

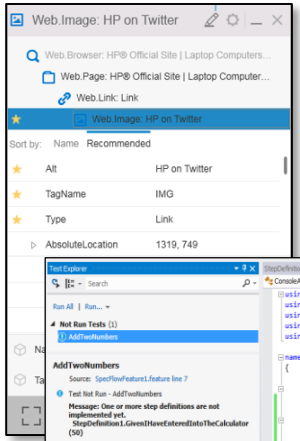
HPソフトウエアによる解決

❑開発チームの開発IDE(Visual Studio, Eclipse) のUnitテストフレームワークにGUI機能テストを可能にし、TDDやCucumberを利用したBDD、ATDDフレームワーク上で機能テストを実現します
❑機能テスト、性能テスト(多重負荷)を実現(注1)

お客様の価値

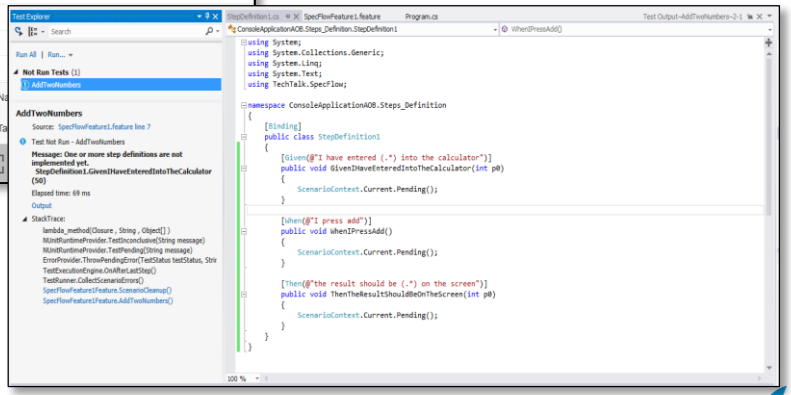
❑開発チームによるUIレベルのAcceptance Testの実現により、QAチームへ一定の品質以上のビルドで探索的テストを実施できるようになり、Shift-Leftを実現します

Web, .NET, WPF, SAP アプリケーション、Android, iOS(注2)のネイティブ＆ハイブリッドアプリケーションをサポート



SPY機能により画面上のオブジェクトのパラメータを取得可能

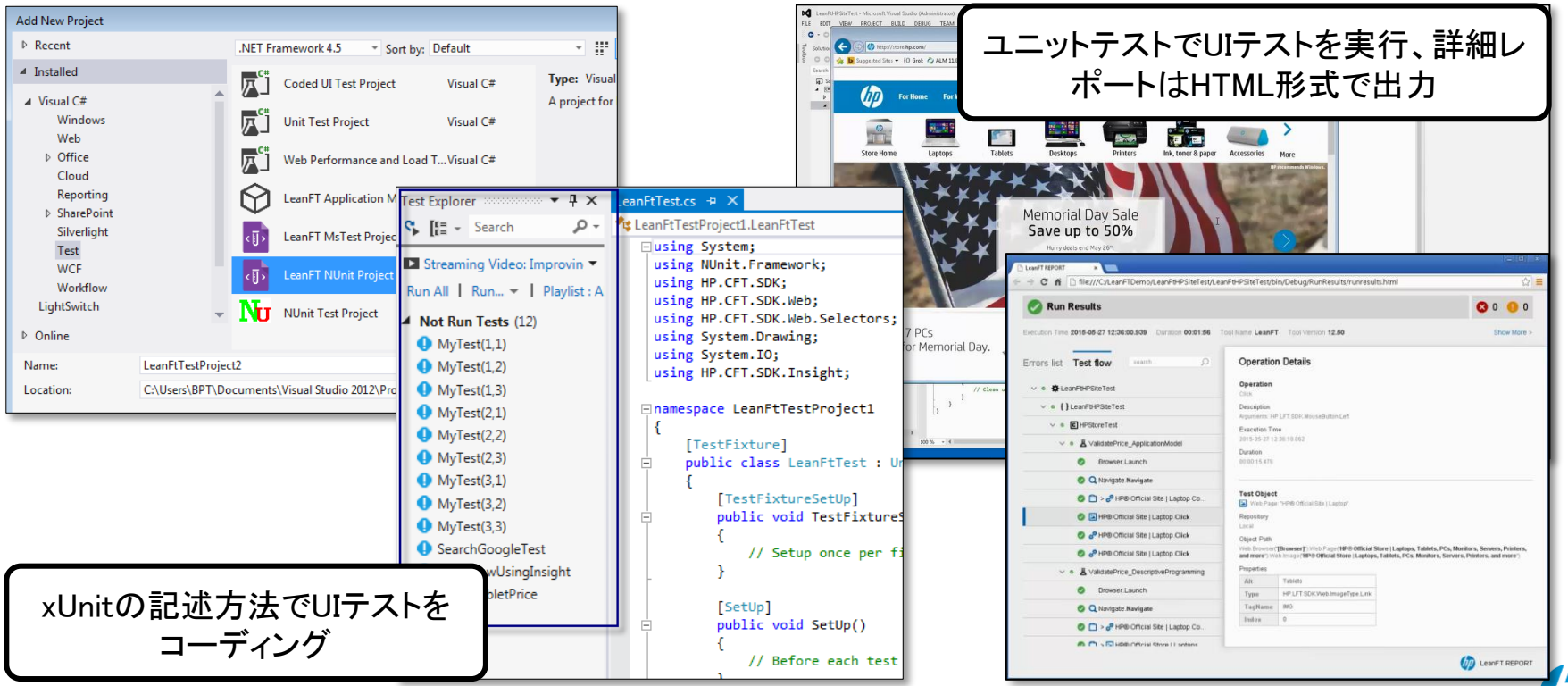
TDD、BDDフレームワークでGUI機能テストを実現



(注1) 別途 HP LoadRunner が必要

(注2) 次期パッチ対応予定

単体テストフレームワークでUIテストを実行



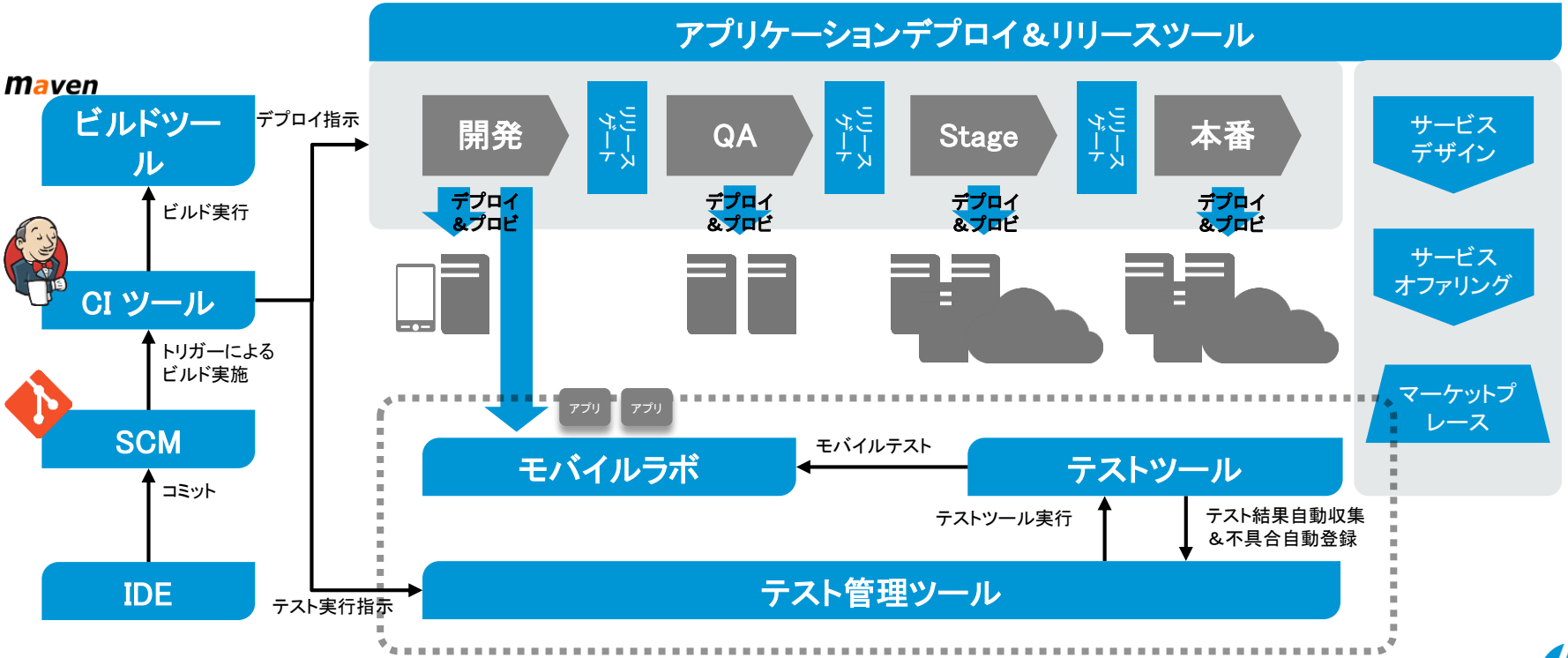
ユニットテストでUIテストを実行、詳細レポートはHTML形式で出力

xUnitの記述方法でUIテストをコーディング

SPY機能でオブジェクト情報を取得可能

DevOpsフレームワークへの統合

点線範囲がここまでで
説明したテスト部分

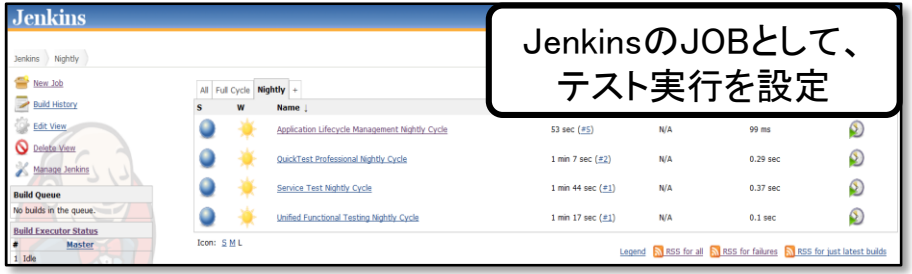


ジョブとしてテスト実行指示(開発フェーズ)

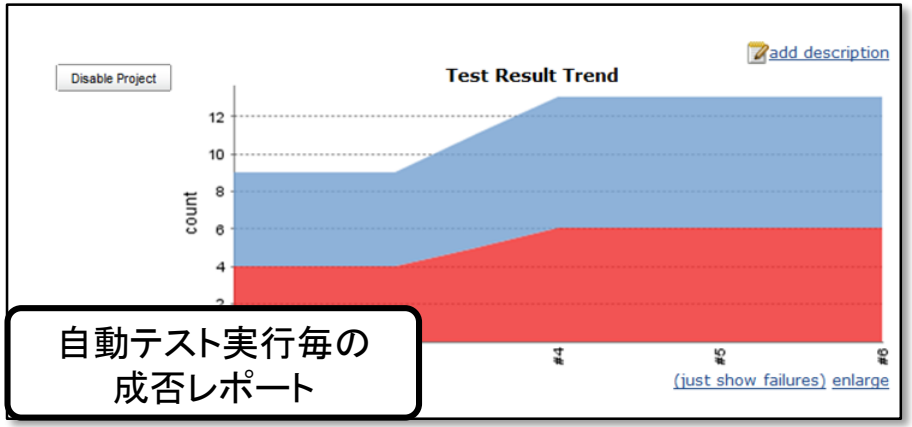
ビッグバンテストのリスク軽減のために継続的テストを実施

Jenkinsのジョブとして実施

- CIツール(Jenkins)向けのプラグインを提供
- テスト管理ツール上のテストアセット機能テスト・負荷テストスクリプトを指定(テスト管理ツールを利用しない場合は直接ツールのスクリプトを指定)
- テスト結果をビルドレポートに統合
- 不具合発生時はテストツールからテストマネジメントツールへ自動で起票

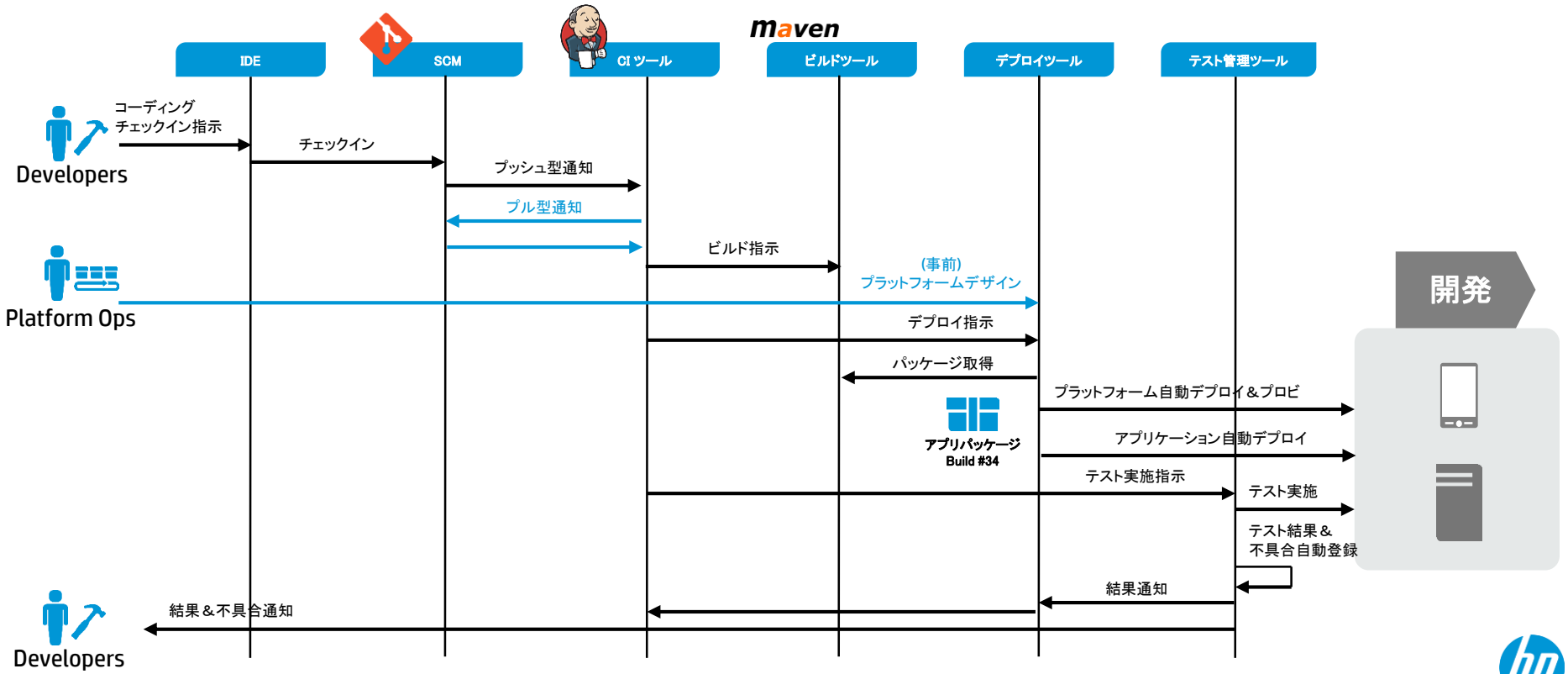


JenkinsのJOBとして、テスト実行を設定

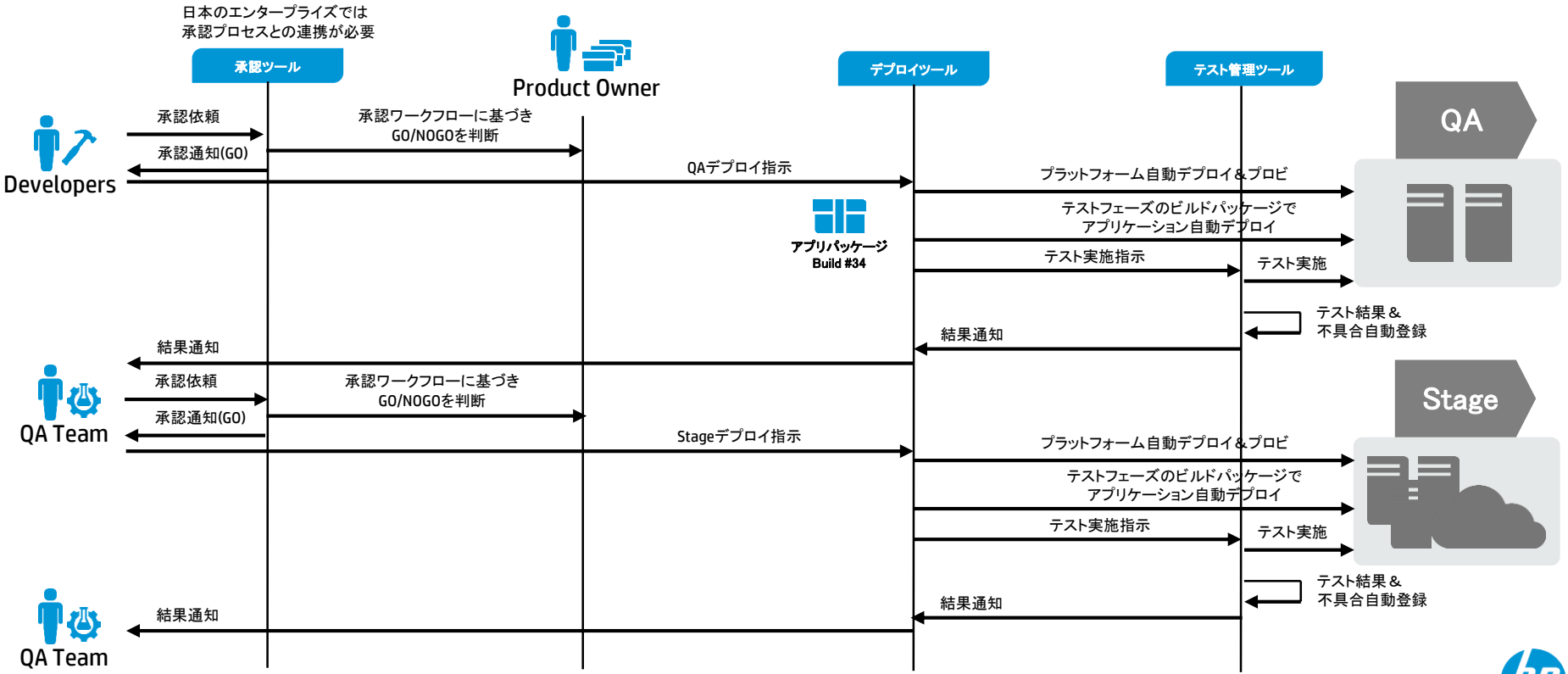


自動テスト実行毎の成否レポート

開発フェーズ：Jenkinsのジョブと連携例



QA以降フェーズ：デプロイツールでのコントロール例



HP Codar

お客様の抱える課題

❑デプロイ環境デザインが非効率的で自動化が進まない
❑開発からQA、Stage、本番まで一貫性のあるアプリケーションデプロイができていない
❑チームごとにばらばらのデプロイ＆プロビツールを利用して独自連携していてガバナンスが効かない

HPソフトウエアによる解決

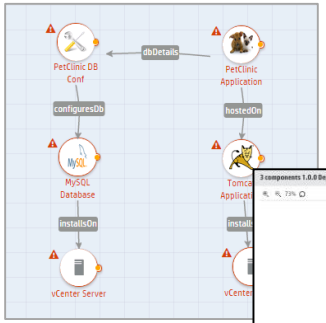
❑GUI上でトポロジーデザインをモデル化する事でデプロイ手順と構成関係の理解が容易で、開発から本番まで同じー貫したサービスデザインが可能
❑既存デプロイツールと連携する事で導入が容易
❑本番環境で実績のあるHP運用自動化ツールで本番環境におけるデプロイ自動化を実現

お客様の価値

❑企業の開発、運用チーム間のデプロイ標準化を実現しDevOps導入によるTTMをお客様へ提供

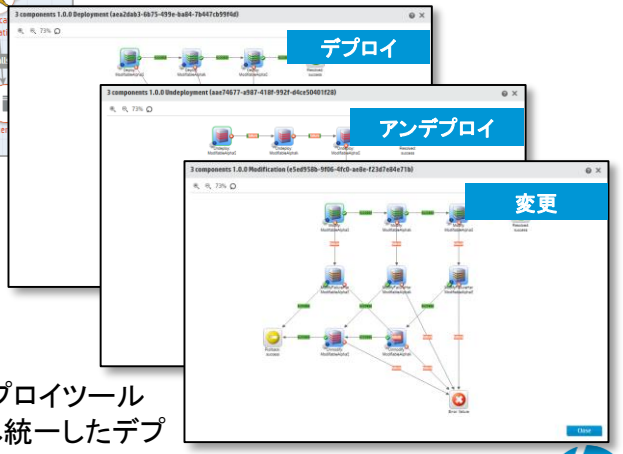
既存デプロイツールや本番環境の運用自動化で実績あるHP OO, SAと連携し標準したトポロジーモデルを提供

サービスデザイン



デザインしたトポロジーに従って自動的にデプロイ手順を生成

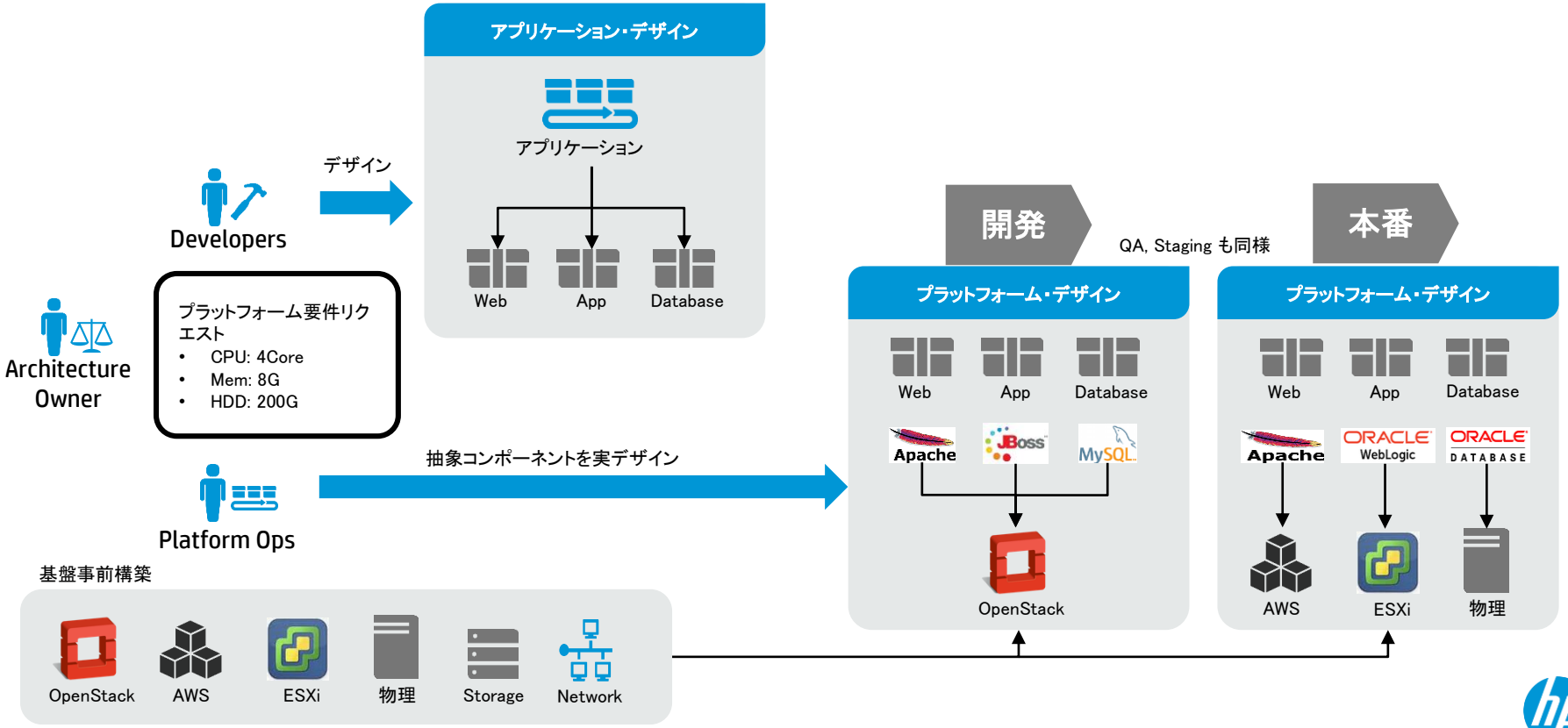
トポロジーを構成するコンポーネント間の関係により実施順次決定



既存デプロイツールと連携し統一したデプロイコントロール

（注）QA以降のテスト実行プラグインは次期バージョンで予定

アプリとプラットフォームのデザインを分離



まとめ

計画と開発

- ビジネスサイドの要望をTTMで実現するための開発手法としてアジャイル手法に取り組む傾向にあるがエンタープライズレベルで求められる「スケール」がなければならない
- エンタープライズで求められる品質をアジャイルに取り組むにはDevチームとQAチームの強い連携とテストマネジメントのKPIがアジャイル進捗管理へフィードバックされる仕組みが重要

テスト

- テストケース数や結果が膨れがちなモバイルテストにおいては積極的な自動化適用を単体・機能テストから、システムテストの領域まで追求、手動テストについては実行効率化と探索的テストを実践
- テスト環境におけるLifecycle Virtualizationと開発チームによるBDD, ATDDの取り組みで「Shift Left」を実現
- 様々なデプロイ＆プロビツールをコンポーネントとして扱う「トポロジーモデル」化を採用し、開発環境から本番環境まで標準プロセス化を実現とエンタープライズレベルのITガバナンスに貢献

Thank you

本日のセッション資料と関連ホワイトペーパーは後日、インプレス社より本日お越しいただいた皆様へDL用URLをメールにてご案内させていただきます。