Hewlett Packard
Enterprise

HPE Software
プロセス連携を踏まえた
モバイルテスト自動化

日本ヒューレット・パッカード株式会社
ITﾏﾈｼﾞﾒﾝﾄﾌﾟﾘｾｰﾙｽ本部 ﾃｸﾆｶﾙｺﾝｻﾙﾃｨﾝｸﾞ部　小宮山　晃

2016年2月25日　15:30 ~ 16:30

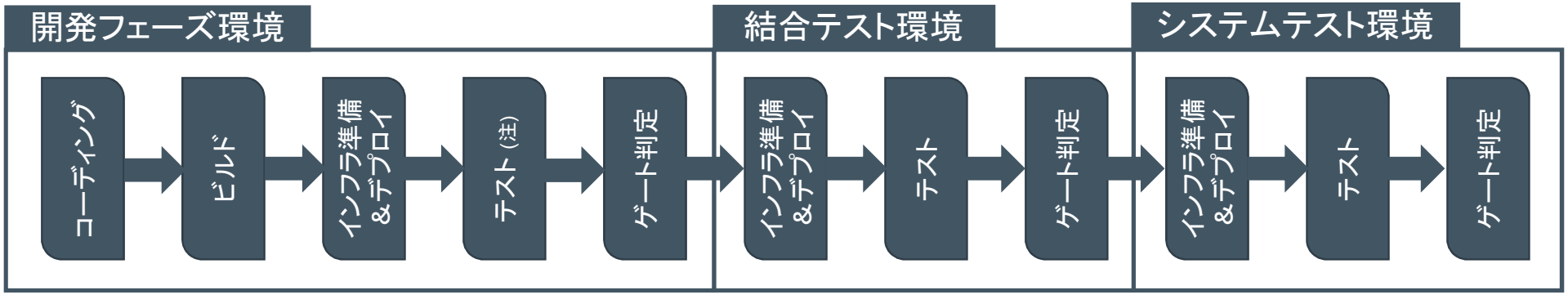
はじめに

- 本日の内容
  - プロセス連携の視点でモバイルテスト自動化について考慮するポイントを説明させていただき、どう工夫するかについてHPEソフトウェアでのソリューションをベースに説明します。

- 背景
  - テスト数の肥大化やさらなるコスト意識の強まりから、モバイルアプリケーションのテスト自動化についてここ数年お客様からの問い合わせが増えてきています。
  - モバイルアプリケーションのテスト自動化は、「どう自動化スクリプトを作成する事」も重要ではあるが、そこにフォーカスがされがちではないでしょうか？
  - 実際のテスティングフレームワークやDevOpsフレームワークにどう当てはめていくかも初期段階から考えておかないと部分最適で終わってしまう事になり、あてはめて連携させるための追加の工数やそのメンテナンス工数も増えてしまいます。

本日の資料で前提としているプロセスと主活動

テストに関わる「開発チーム」と「QA（テスト）チーム」の活動

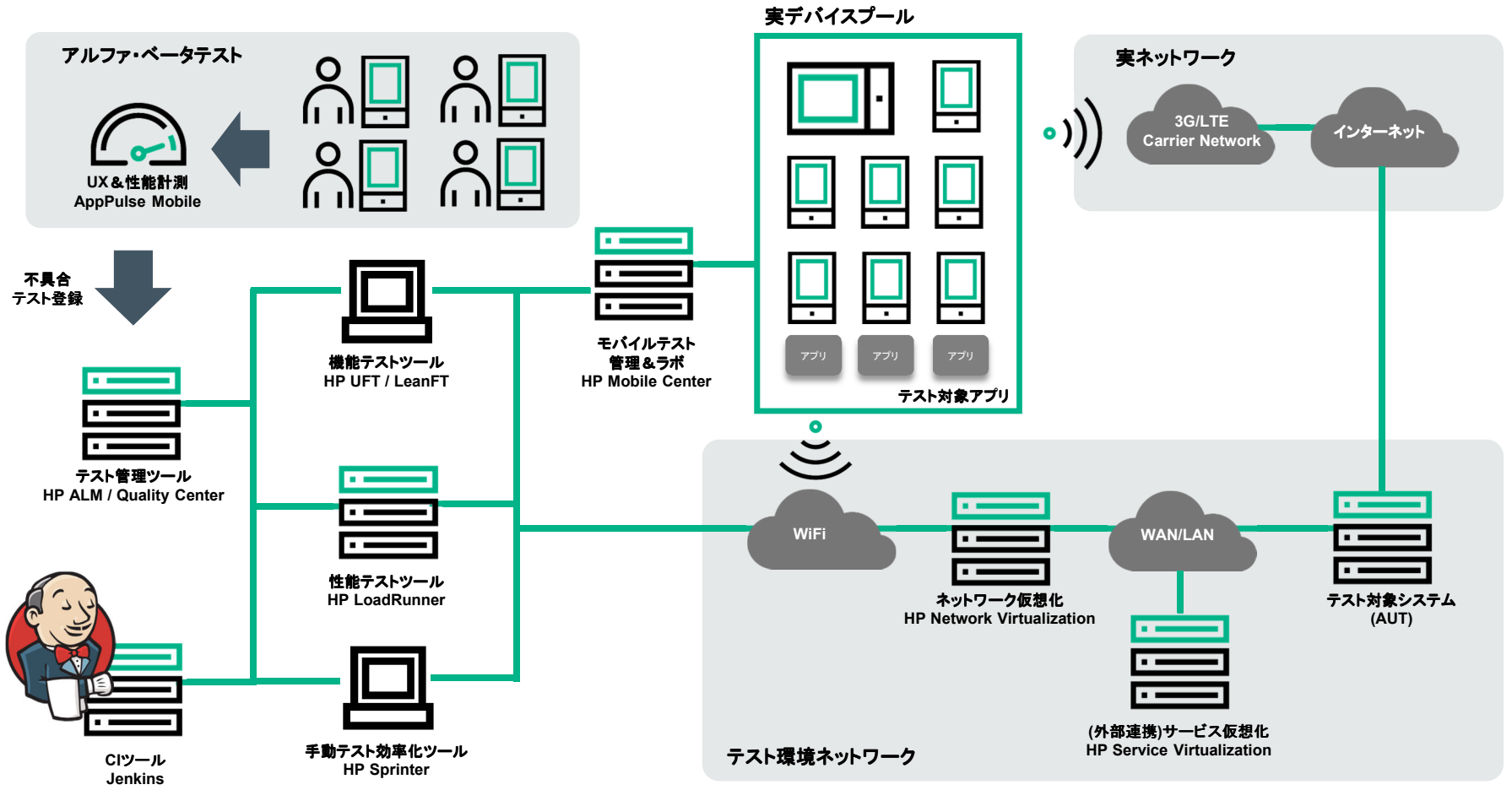


開発チーム	・単体テスト実装（コンポーネント含む） ・UI機能テストの自動化実装(注) ・ゲート判定		
QAチーム	・テスト設計 ・影響度分析＞テストケース選定 ・UI機能テストの自動化実装	・手動テスト ・ゲート判定	・ゲート判定

Hewlett Packard Enterprise

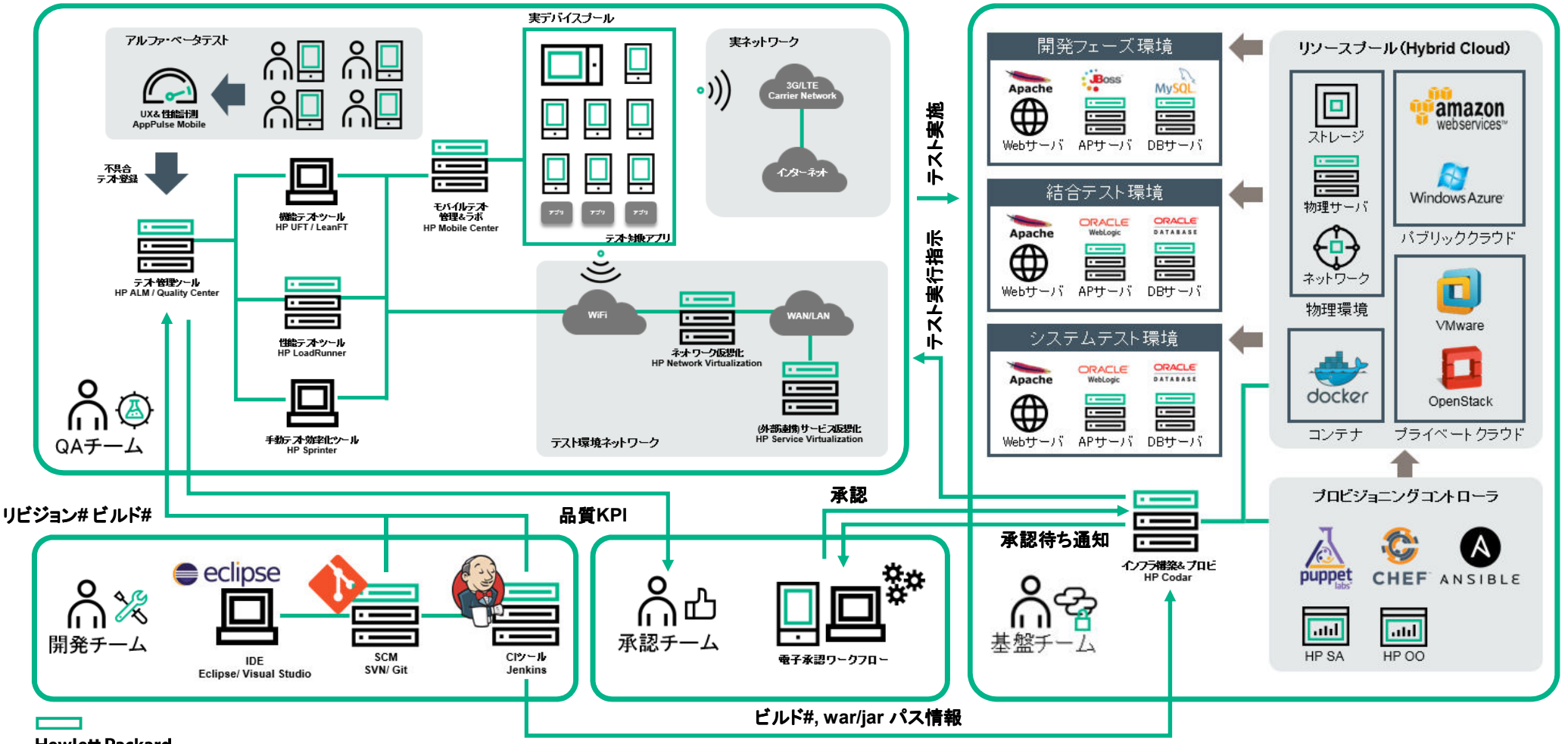
（注）主に開発チーム主導による UI 機能テストの自動化を実施する場合

HPE ツールのモバイルアプリケーションテスト向け構成例



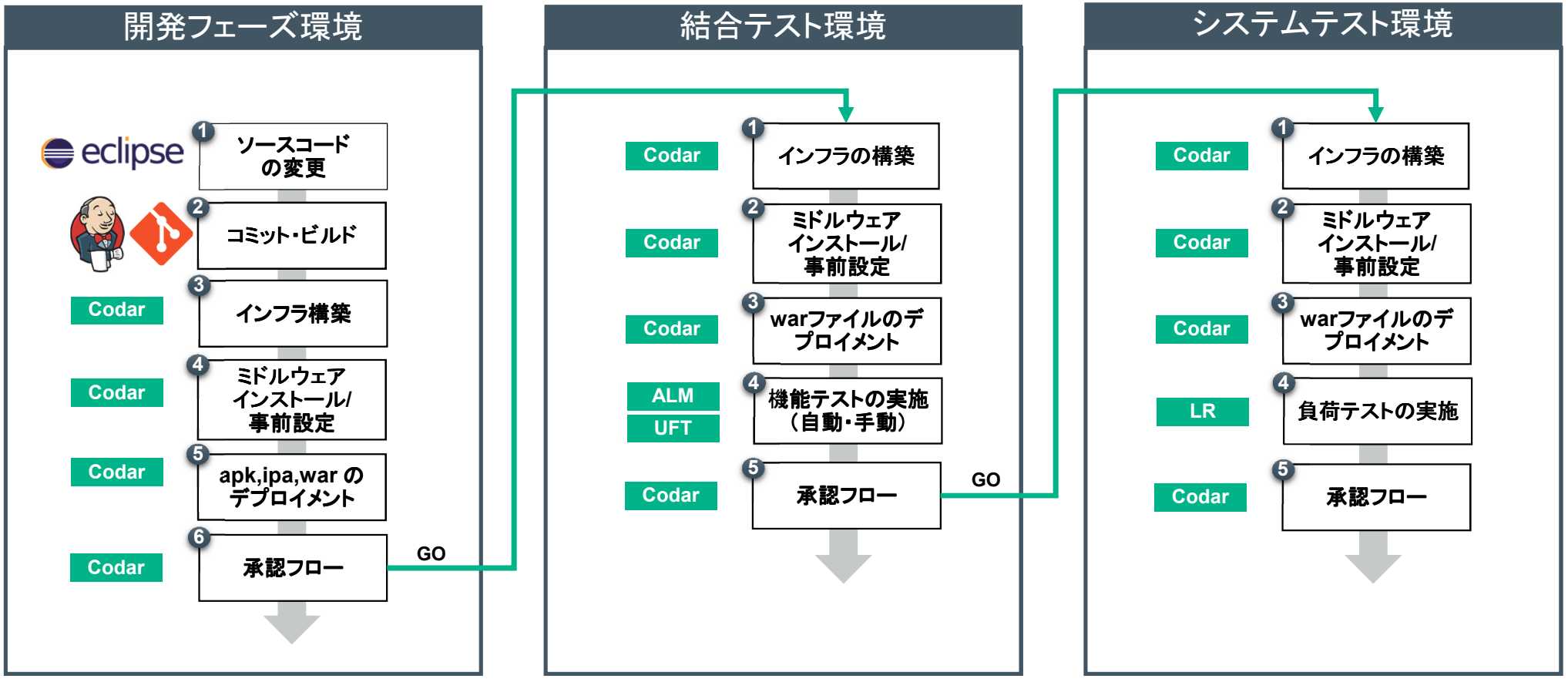
Hewlett Packard
Enterprise

HPE モバイルDevOpsフレームワーク構成例（チーム単位）



Hewlett Packard Enterprise

前提プロセスを支援する HPE ツール 適用例



プロセス連携を踏まえた モバイルテスト自動化

1. 自動化取り組みの前に知っておく事と注意点
2. 自動化率について知るべきポイント
3. HPE UFT & HPE MC を使った自動化スクリプト作成手順
4. テスト用モバイルデバイス＆アプリ管理（ラボ管理）
5. CI/CD自動化プロセスへの統合方法とTIPS
6. 複数デバイス＆アプリケーションのコントロール方法とTIPS
7. インフラ構築＆デプロイ自動化ツールとテストツールの連携TIPS

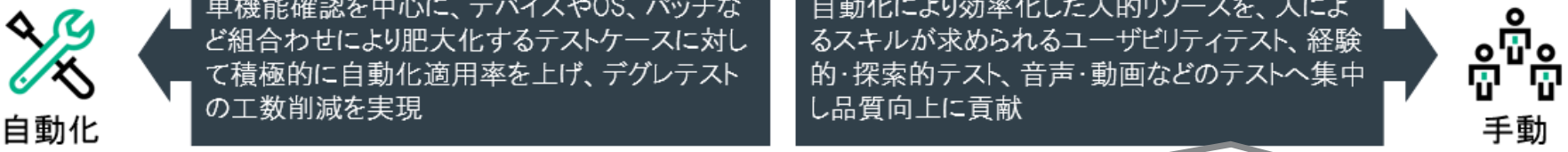
自動化に取り込む前に

- 全てのテストケースが自動化できるわけではない
  - 自動化すべきテストケース、向かないテストケースを分別する

- 自動化に**向いている**もの
  - 同一のシナリオに対して複数データでテストをする
  - ビルド変更やパッチ適用の度にテストを繰り返す
  - ビジネスに影響の大きい重要なアプリケーション
  - 単純な操作
  - 機能的に安定したアプリケーション

- 自動化を**避けた方がよい**もの
  - 複雑な検証やシナリオから始める(処理毎に複数のテストに分けた方がよい)
  - エラー処理や例外処理を1つのテストに含める(複雑な条件分岐が必要になる)
  - 安定していない機能の自動テスト(仕様変更が頻発する箇所)
  - すべてを自動化する
  - 1回しか行わないテストの自動化

  - 手動テストにすべきテスト活動に人的リソースを注力する



タイミングに依存する不具合(省電力モード切替瞬間での操作や複数機能の同時押下)など経験的テストは重要

モバイルアプリにおいて自動化が難しい操作（弊社経験上）

- 地図やチャットのように標示される内容が画像でしか判断できないもの（オブジェクトでのチェックができない、画像チェックも毎回異なる（地図座標やコンパスによるずれ））
- 通話機能操作（メッセージを話す、メッセージを聞き取る）
- カメラ（特定の画像、タイミングで撮影したものを確認する）
- ICカード機能（オサイフ機能など）
- センサー機能（振ってアプリが変わるとか、AR機能で標示されるとか、地図がコンパスで動くという操作）
- 複雑なジェスチャー操作（ジェスチャーを組み合わせたり軌跡が複雑な操作）

予めツールの不得意な操作を理解し、事前に該当テストケースについては手動テストで実施するようにリソースを調整する

手動

自動化範囲の向上にはプロセス改善も併せて実施

- 自動化範囲対象
  - テストレベル　単体・コンポーネント、結合、システム、受入、運用
  - テストタイプ　機能、非機能（性能、セキュリティ、ユーザビリティ）
  - テストプロセス　計画、分析、設計、実装、実行、判定

赤字部分が主にツールによる
自動化が実践されている領域

取り組みのベストプラクティス

自動化	テストプロセス改善
短期間で効果が 可視化しやすい	効果が分かるまで 時間がかかりやすい

同時に取り組んで行く方がよい

テストプロセス改善の必要性はほとんどの
チームで課題となっているが、時間がかかる
事と可視化が難しいため投資がされにくい。
自動化への取り組みと同時に進める事で社
内の投資判断も得られやすい

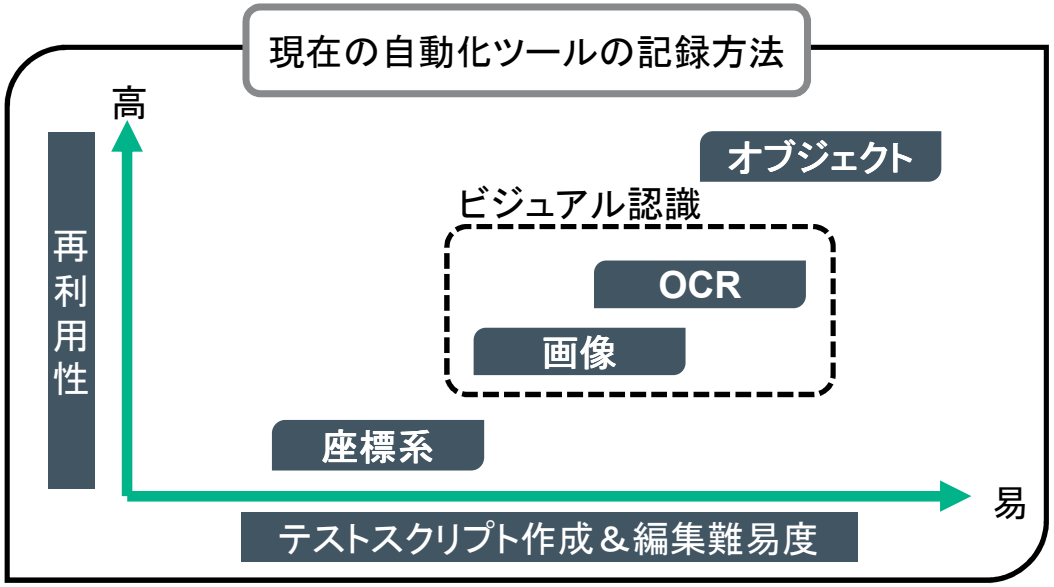
プロセス連携を踏まえた モバイルテスト自動化

1. 自動化取り組みの前に知っておく事と注意点
2. 自動化率について知るべきポイント
3. HPE UFT & HPE MC を使った自動化スクリプト作成手順
4. テスト用モバイルデバイス＆アプリ管理（ラボ管理）
5. CI/CD自動化プロセスへの統合方法とTIPS
6. 複数デバイス＆アプリケーションのコントロール方法
7. インフラ構築＆デプロイプロセスとテストツールの連携方法

モバイル機能テストの自動化率を支える自動化記録方法

- 自動化率①「自動化できたテストケース」/「自動化を予定していた全テストケース」を計測しておく
  - 「自動化に向かないテストケース」を除いた自動化対象テストケースのうち、自動化できたテストケース率
  - 自動化率が低い場合の要因として自動化ツールの技術的な制約が大きく、中でもモバイルアプリケーション記録方式（スクリプト化）は選定時の考慮ポイント

1. 座標系
   座標軸（x、y）をベースに操作を記録します
2. ビジュアル
   表示画面の情報からテキストをOCR、画像を
   対象オブジェクトとして操作を記録します
3. オブジェクト
   ソースコードに専用ライブラリを利用しリコンパイル、
   または専用エージェント経由で対象アプリ操作する
   仕組みを利用し正確に記録します
   ただしプリインストールアプリは対象外

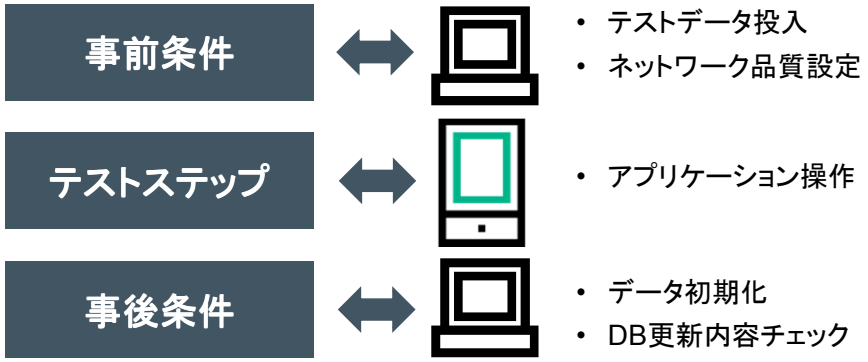


HPツールでは「オブジェクト」、「画像」、「座標系」による記録方式を採用

モバイルアプリとはいえ、モバイルだけの自動化かどうか

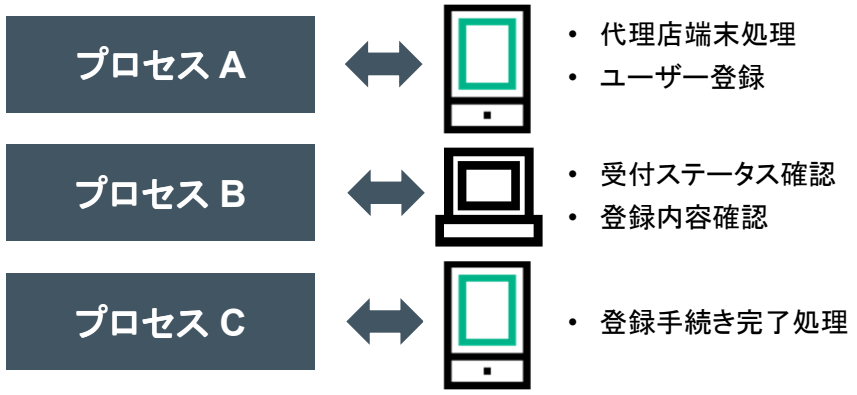
- 自動化率②「自動化できた手順」/「テストケース内で自動化を予定していた手順」を計測しておく
  - 自動化率が低い場合の主な要因として自動化ツールの技術的な制約が考えられる

結合テストケース例



事前条件、事後条件で他システム操作が発生するテストケース

受入テストケース例



業務としてモバイル以外のシステム確認などが発生するテストケース

Hewlett Packard Enterprise

HPツール はテストスクリプトの「コンポーネント化」と広い「技術サポート範囲」で高率化

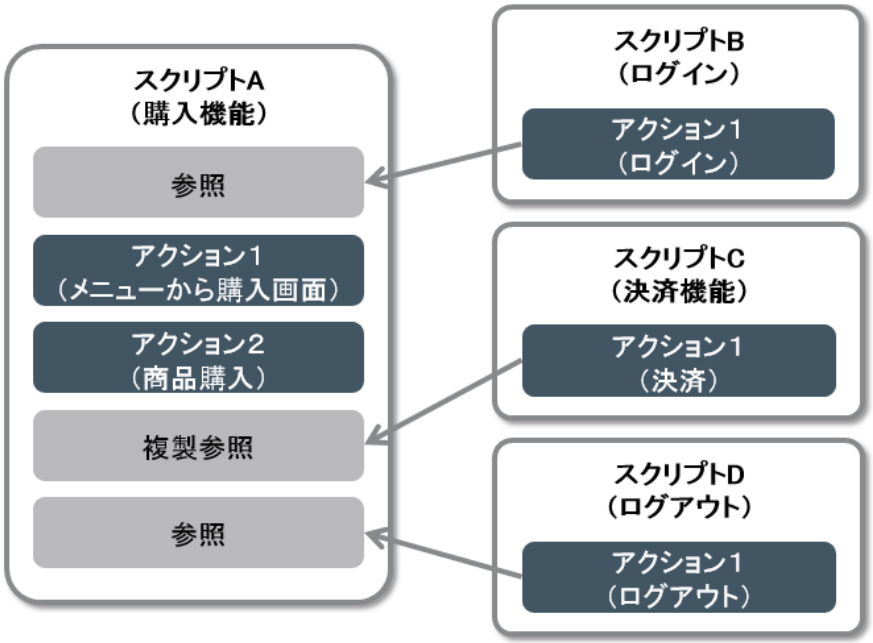
機能テストツールのコンポーネント活用

テストスクリプトの保守性を高め、モバイル以外の対応により自動化率を高率化

- 自動化に取り組んだものの保守性が低く回帰テスト工数が減らない経験はありませんか？

画面Aに修正がはいり、デグレ確認のため、回帰テストとして影響範囲の全スクリプトを実行したい。ここで、
もしシナリオごとにスクリプトを作っていたら100シナリオを修正しなければならなくなってしまう？

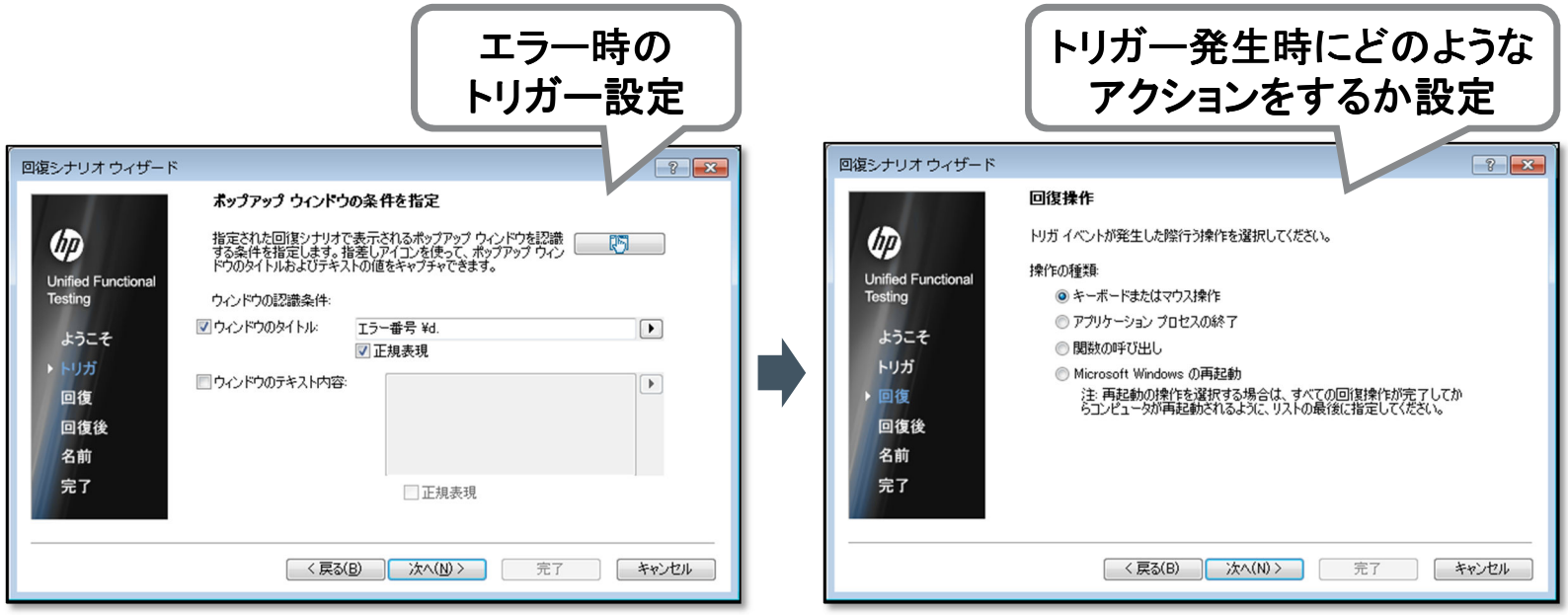
1. UFTでは、スクリプトを最小単位としてアクションとして管理、再利用する事が可能
2. 変更があった場合でもスクリプト修正を影響があるアクションのみに抑える事が可能
3. UFTではモバイルアプリだけではなく、Web、ERP（SAPやOracleEBS）、.NETアプリなども対応しておりテストの事前事後条件部分の処理や受入テスト時の複数端末を通したテストケースも可能



自動実行中に発生するエラー対処（夜間・週末実行時のエラーに対応）

発生するエラーから自動回復を支援

- 自動テストにおけるエラー対応処理は、テストスクリプト中にエラー処理をプログラミングしなければならないのが一般的な方法です
- UFTでは、「回復シナリオウィザード」に従ってエラーパターンを登録することで、プログラミングなしでエラー時の回復処理を追加できます

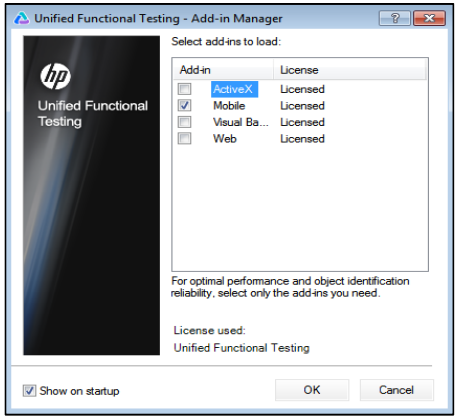


プロセス連携を踏まえた
モバイルテスト自動化

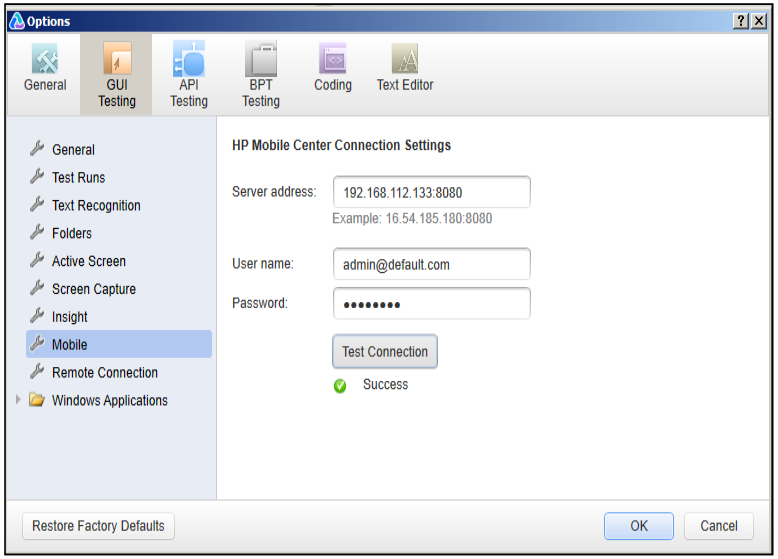
1. 自動化取り組みの前に知っておく事と注意点
2. 自動化率について知るべきポイント
3. HPE UFT & HPE MC を使った自動化スクリプト作成手順
4. テスト用モバイルデバイス＆アプリ管理（ラボ管理）
5. CI/CD自動化プロセスへの統合方法とTIPS
6. 複数デバイス＆アプリケーションのコントロール方法
7. インフラ構築＆デプロイプロセスとテストツールの連携方法

機能テスト操作 (1/3)

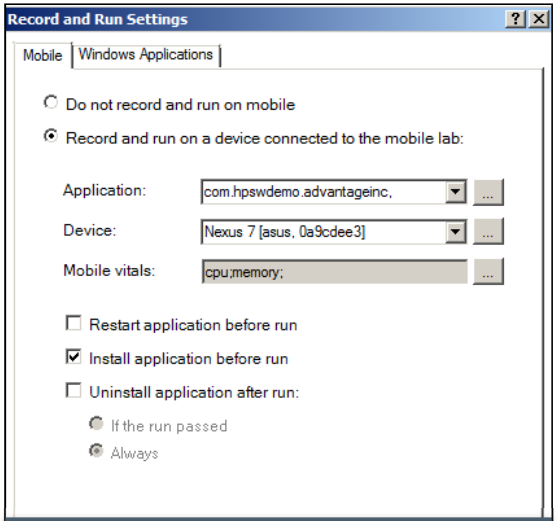
モバイルアプリケーション操作記録前準備



①UFTのアドインマネージャでMobileアドインを有効化



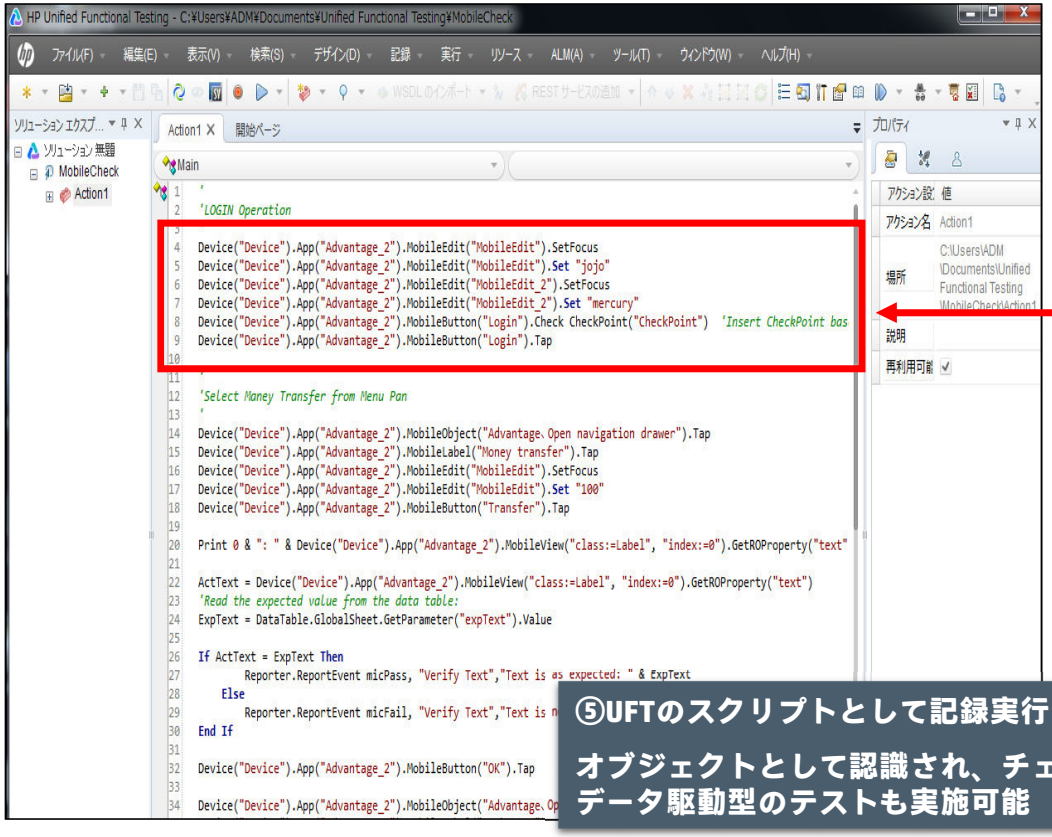
②オプションでMobile Center との接続情報やログインユーザ情報を入力



③実行環境設定でテスト対象アプリケーションや利用するデバイスを選択。CPUやMemory情報の取得設定やアプリケーションのデプロイ設定等も可能

機能テスト操作 (2/3)

モバイルアプリケーションの操作をRDPもしくは実デバイス操作でスクリプト記録



④アプリケーションのログイン操作をオブジェクトして記録

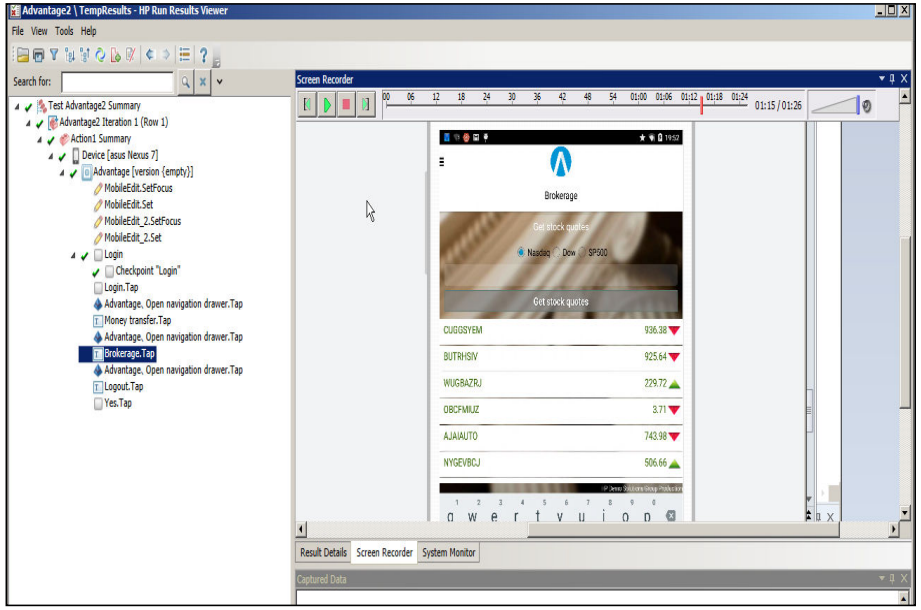
どのボタンをタップしたか
フィールドに何を入力したか
等を個別オブジェクトとして記録

⑤UFTのスクリプトとして記録実行が可能

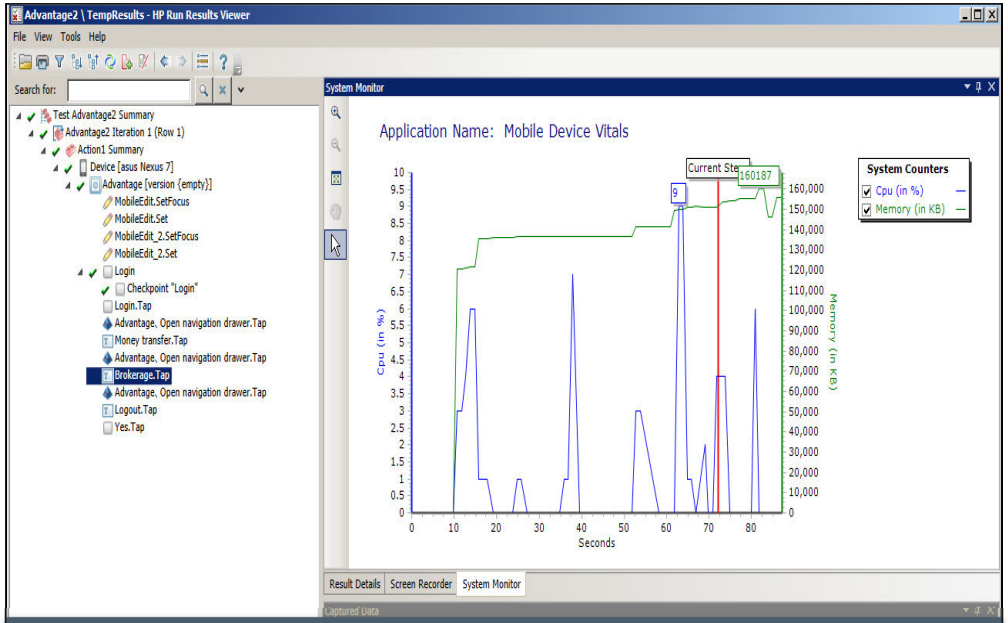
オブジェクトとして認識され、チェックポイントやデータ駆動型のテストも実施可能

機能テスト操作 (3/3)

テスト結果レポート（チェックポイント結果、エビデンス画面＆動画、デバイスバイタル＆ログ）



⑥各操作ステップの画像（エビデンス）や実行動画を取得しテスト実行時のデバイス動作を確認することも可能



⑦デバイス側のCPUやメモリ状況などを実行ステップと相関しながら確認することが可能(注)

どのステップでCPU消費量が高くなるのか等を分析することが可能

(注) 消費電力量については将来予定

プロセス連携を踏まえた
モバイルテスト自動化

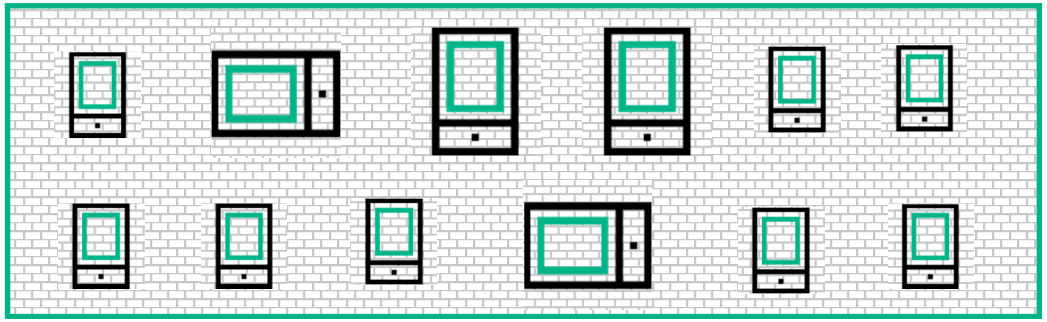
1. 自動化取り組みの前に知っておく事と注意点
2. 自動化率について知るべきポイント
3. HPE UFT & HPE MC を使った自動化スクリプト作成手順
4. テスト用モバイルデバイス＆アプリ管理（ラボ管理）
5. CI/CD自動化プロセスへの統合方法とTIPS
6. 複数デバイス＆アプリケーションのコントロール方法
7. インフラ構築＆デプロイプロセスとテストツールの連携方法

「管理＆ラボ機能」により大規模で煩雑になりがちな管理に対応

テスターはUFTからデバイスとアプリをクラウドとして利用する事で管理から開放されます

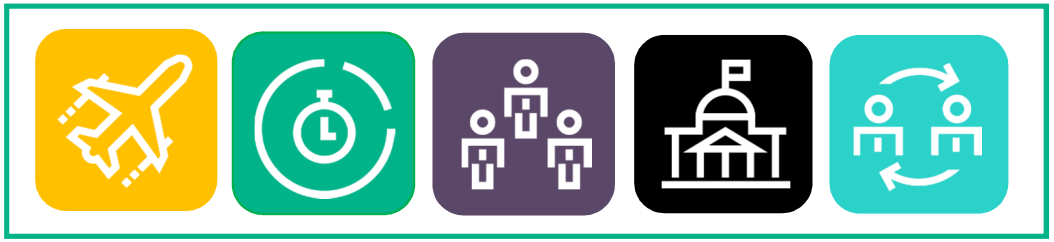
ラボ管理機能(HP Mobile Center)

デバイスとOSの管理



オンプレミス, iOS, Android, プラグ＆プレイ

アプリの管理



配布, インストール, アンインストール

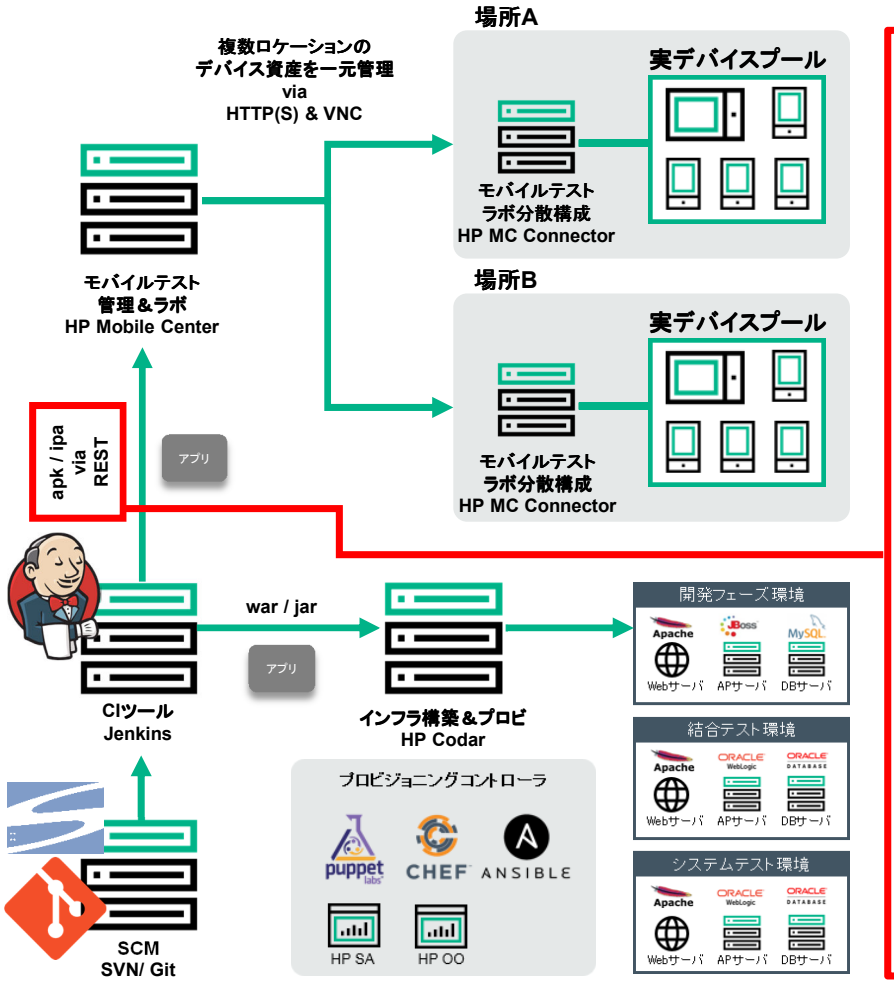
CIツール連携し apk/ipa ファイルをデプロイ、インストール可能

Hewlett Packard Enterprise

プロセス連携を踏まえた
モバイルテスト自動化

1. 自動化取り組みの前に知っておく事と注意点
2. 自動化率について知るべきポイント
3. HPE UFT & HPE MC を使った自動化スクリプト作成手順
4. テスト用モバイルデバイス＆アプリ管理（ラボ管理）
5. CI/CD自動化プロセスへの統合方法とTIPS
6. 複数デバイス＆アプリケーションのコントロール方法
7. インフラ構築＆デプロイプロセスとテストツールの連携方法

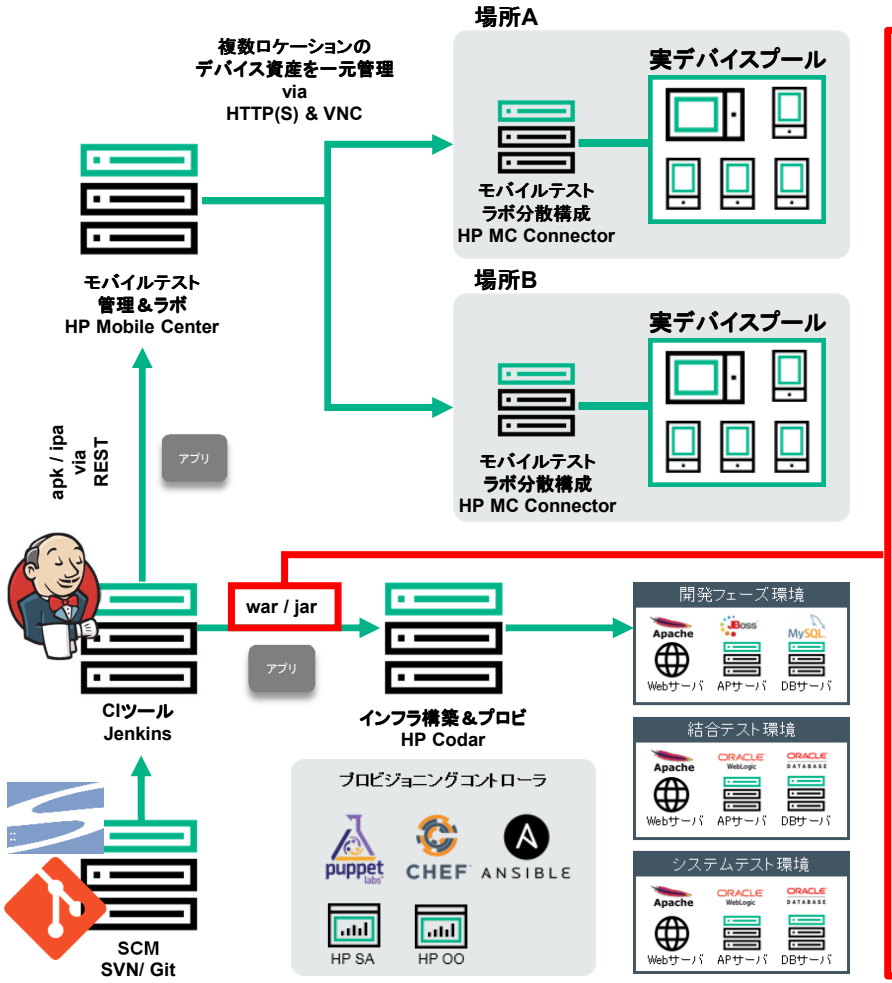
テスト時のアプリデプロイ自動化 (1/2)



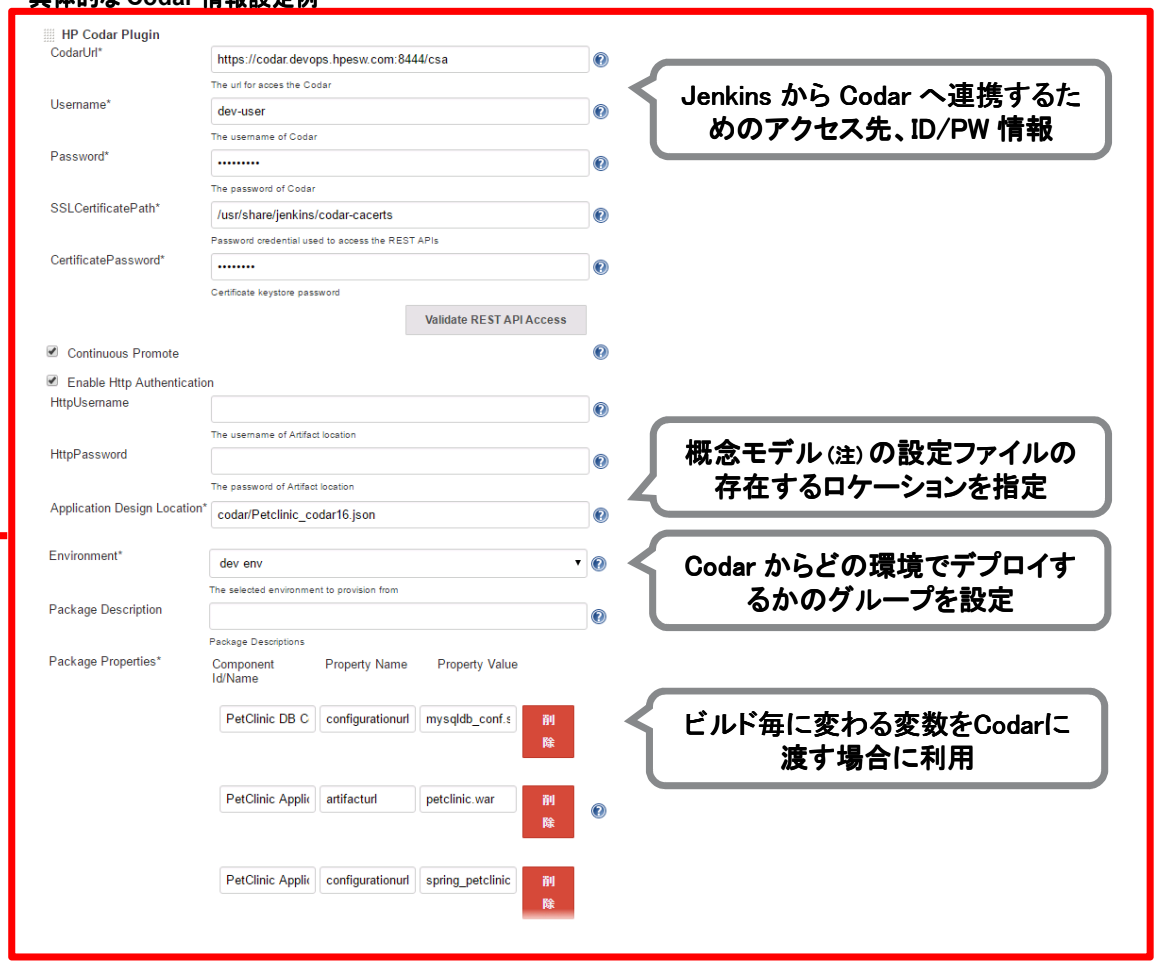
REST サンプル（Ruby ケース）：このスクリプトを CI ツールから実行

```
import sys,json, requests
PROTOCOL = 'http'
PORT = '8080'

SERVER = sys.argv[1]
USER = sys.argv[2]
PASS = sys.argv[3]
FILE = sys.argv[4]

BASE_URL = PROTOCOL + '://' + SERVER + ':' + PORT + '/rest'
s = requests.Session()
# LOGIN
url = BASE_URL + '/client/login'
headers = {'Accept': 'application/json' , 'Content-Type': 'application/json; charset=UTF-8'}
payload = {'name': USER , 'password': PASS , 'accountName': 'default'}
response = s.post(url, headers=headers, data=json.dumps(payload))
print ("Connect: ", response.status_code)

hp4msecret = response.headers['x-hp4msecret']
jsession = s.cookies['JSESSIONID']


if response.status_code == requests.codes.ok:
# UPLOAD APP
    url = BASE_URL + '/apps'
    file = {'fileUpload': open(FILE, 'rb')}
    headers = {'x-hp4msecret': hp4msecret, 'JSESSIONID': jsession}
    response = s.post(url, headers=headers, files=file)
    print ("Upload: ", response.status_code)

if response.status_code == requests.codes.ok:
    print ("App Name:", response.json()["name"])
    print ("App Id:", response.json()["id"])
    print ("App version:", response.json()["version"])
    print ("App Count:", response.json()["count"])
```

テスト時のアプリデプロイ自動化 (2/2)



具体的な Codar 情報設定例



Jenkins から Codar へ連携するためのアクセス先、ID/PW 情報

概念モデル(注)の設定ファイルの存在するロケーションを指定

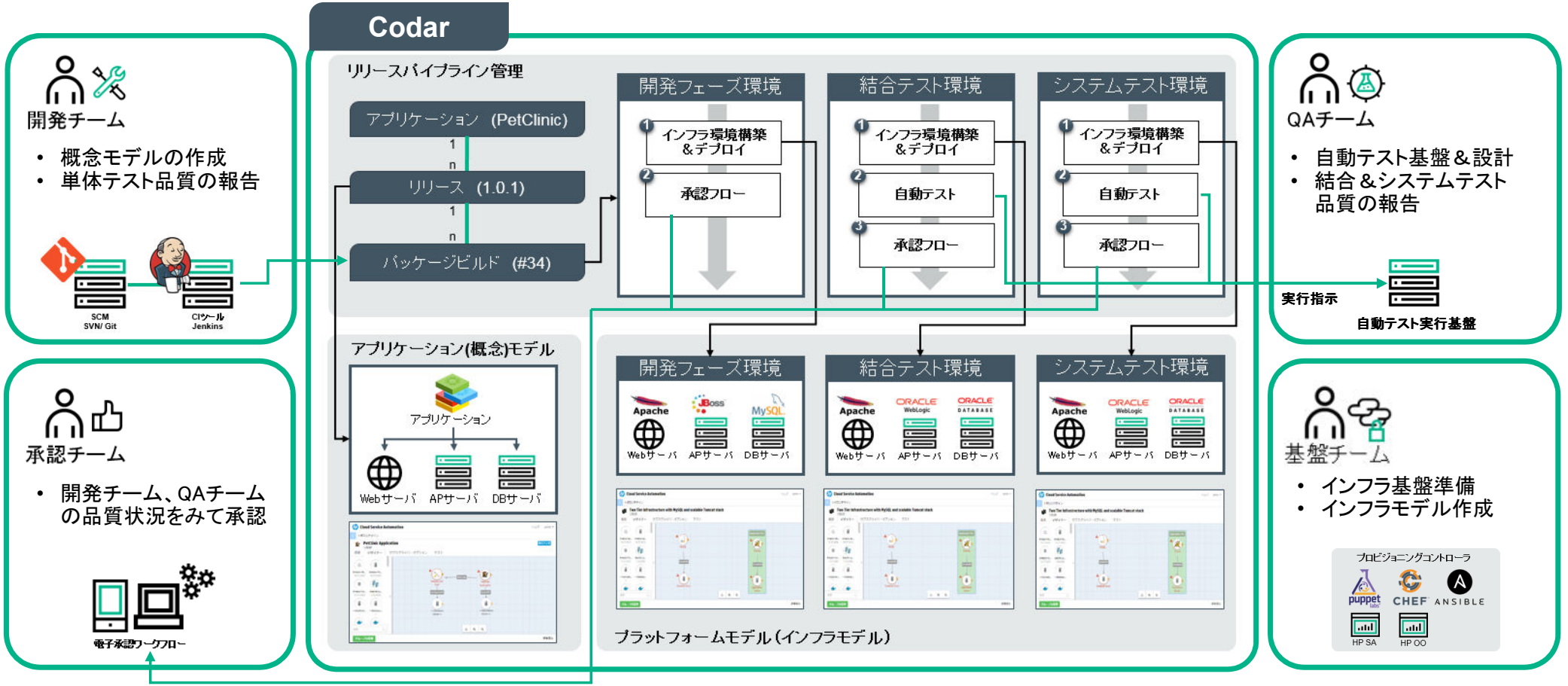
Codar からどの環境でデプロイするかのグループを設定

ビルド毎に変わる変数をCodarに渡す場合に利用

(注) 概念モデルについては次スライドで説明

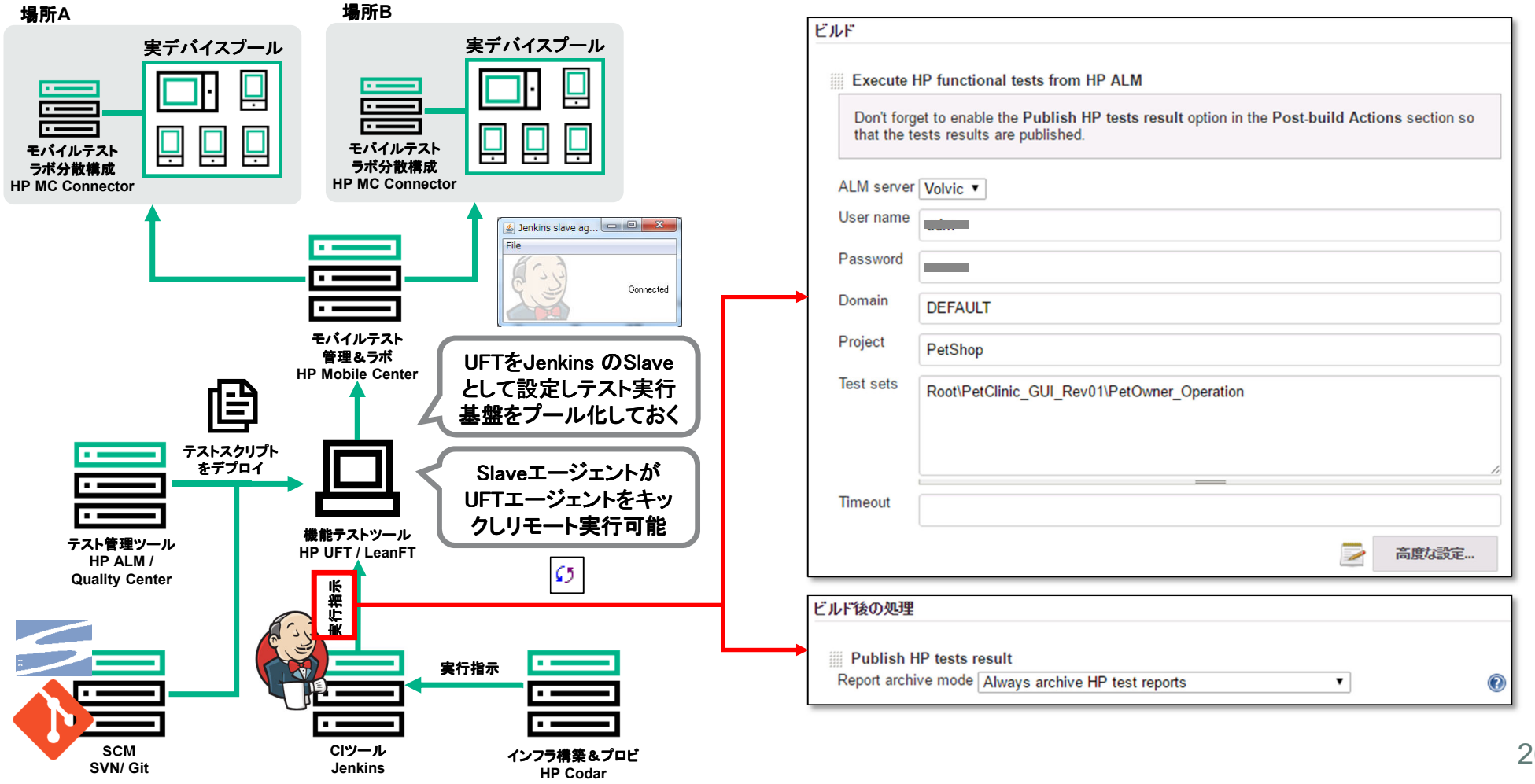
Codar をチーム全体で共有しプロセス標準化を実現

過去にデプロイした環境の開発からテストまで含めて保存し、短時間での環境再現が可能



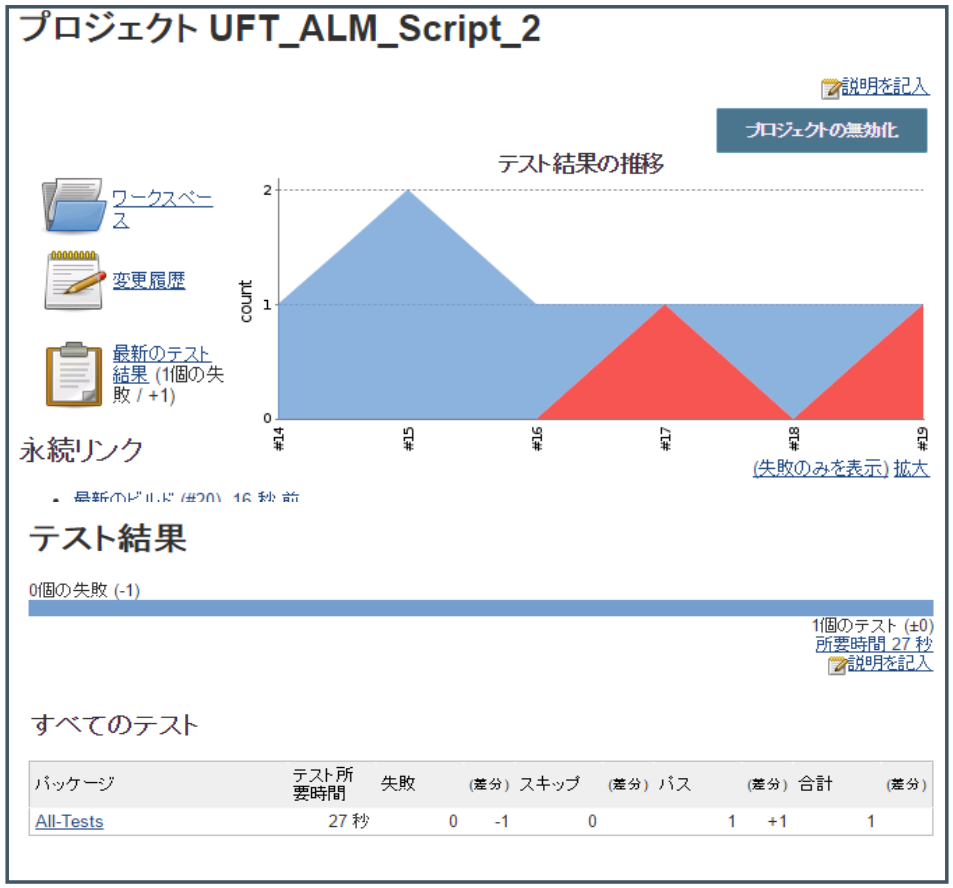
CI/CDプロセスにおけるUFT実行 (1/3)

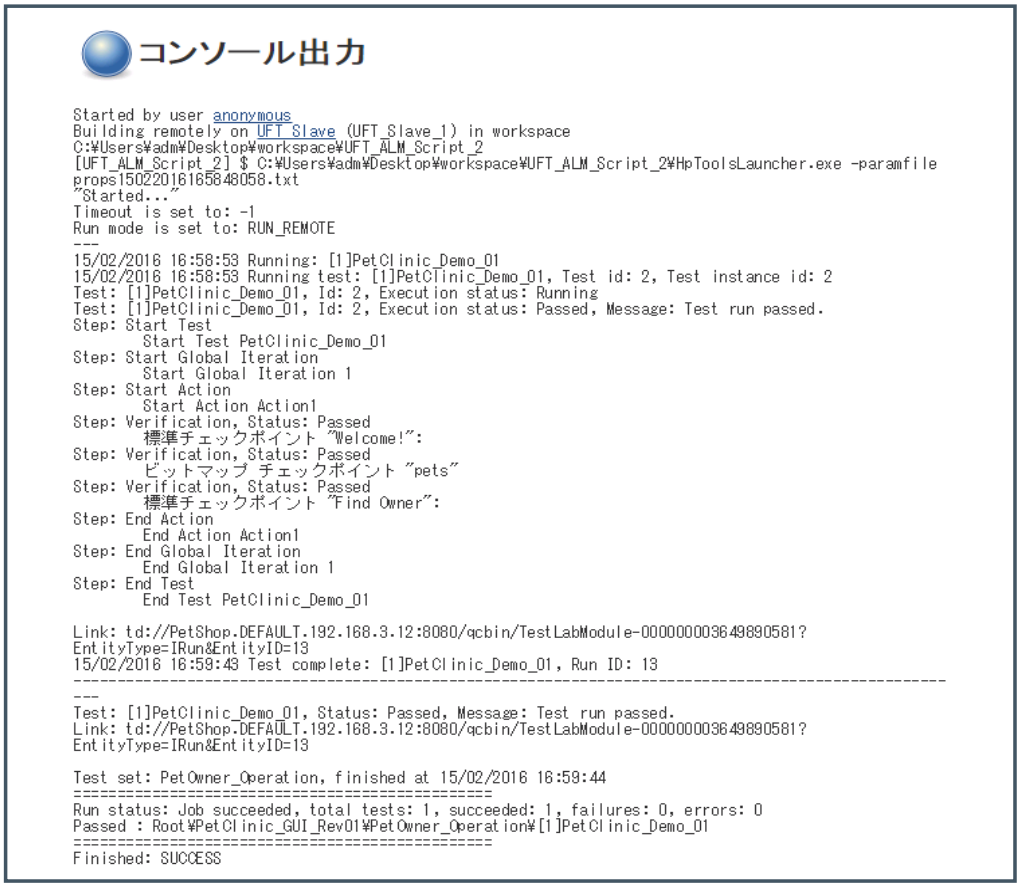
Jenkins専用アドインを利用して設定可能



CI/CDプロセスにおけるUFT実行 (2/3)

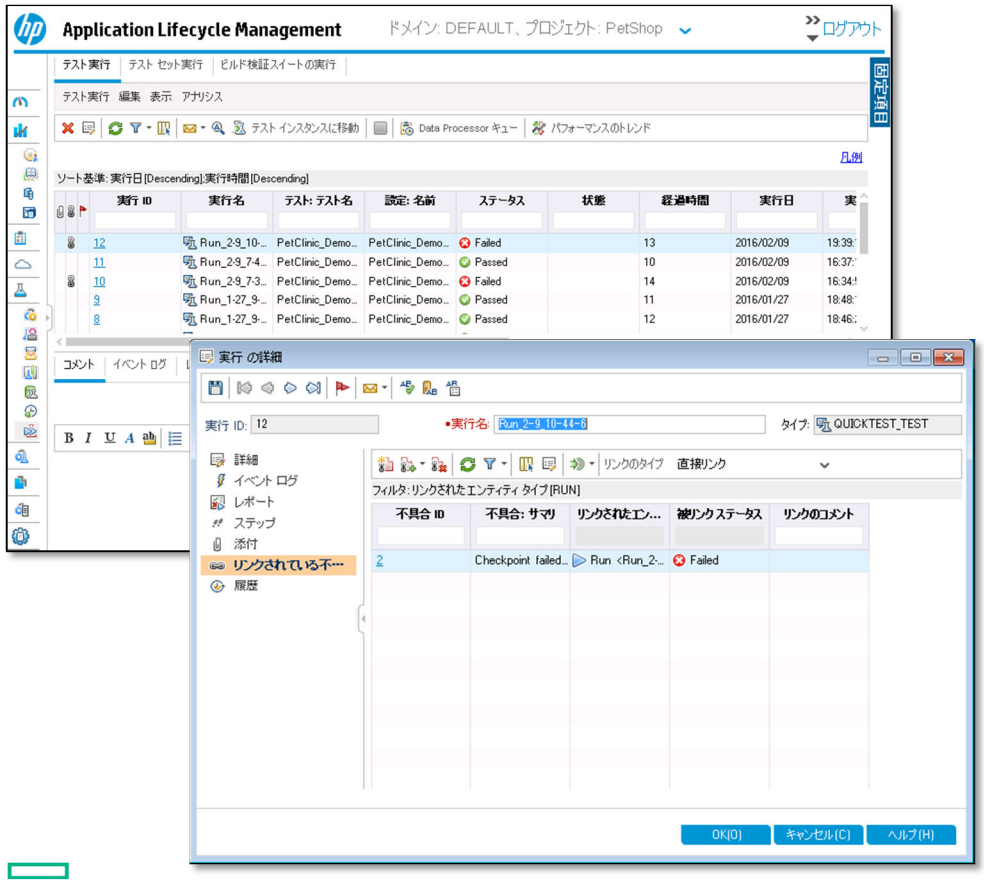
テスト結果はUFTやALMクライアントを利用せずにJenkinsでも確認することが可能

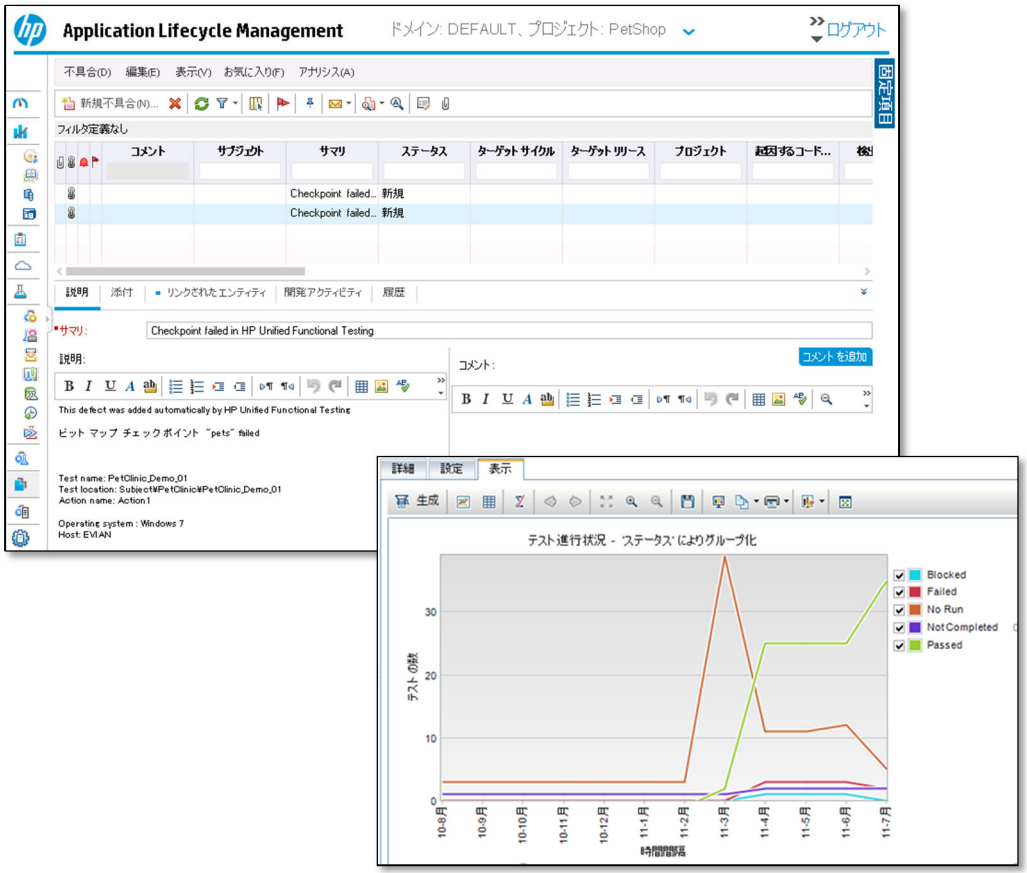




CI/CDプロセスにおけるUFT実行 (3/3)

ALM上からテスト結果と不具合のKPIやレポートをチェック可能（QA担当者や承認者が利用）

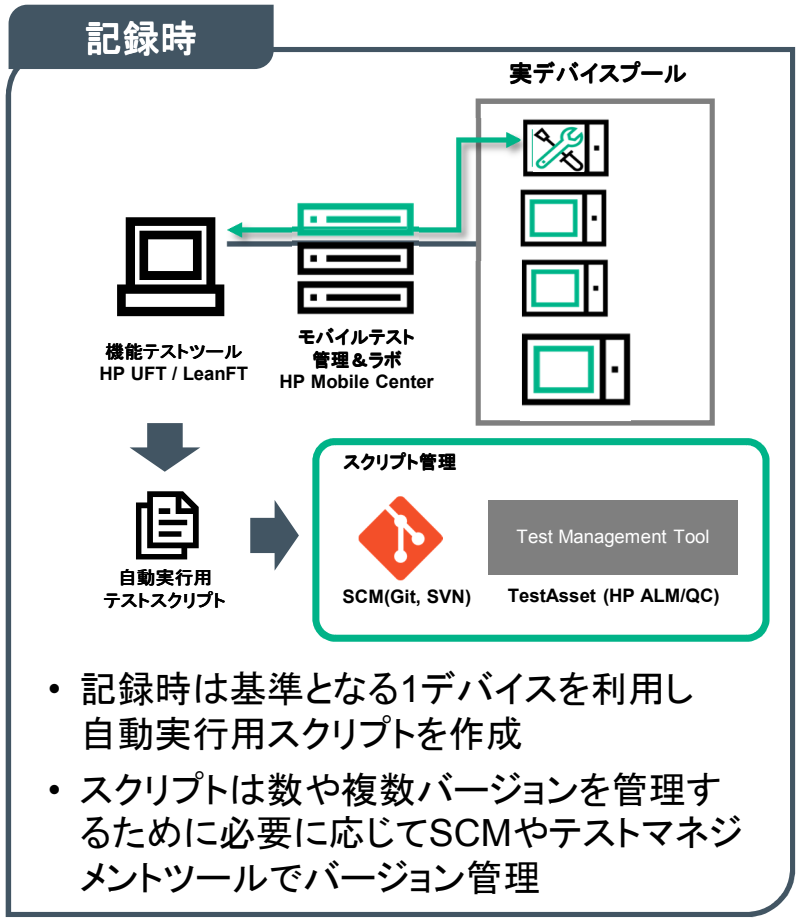




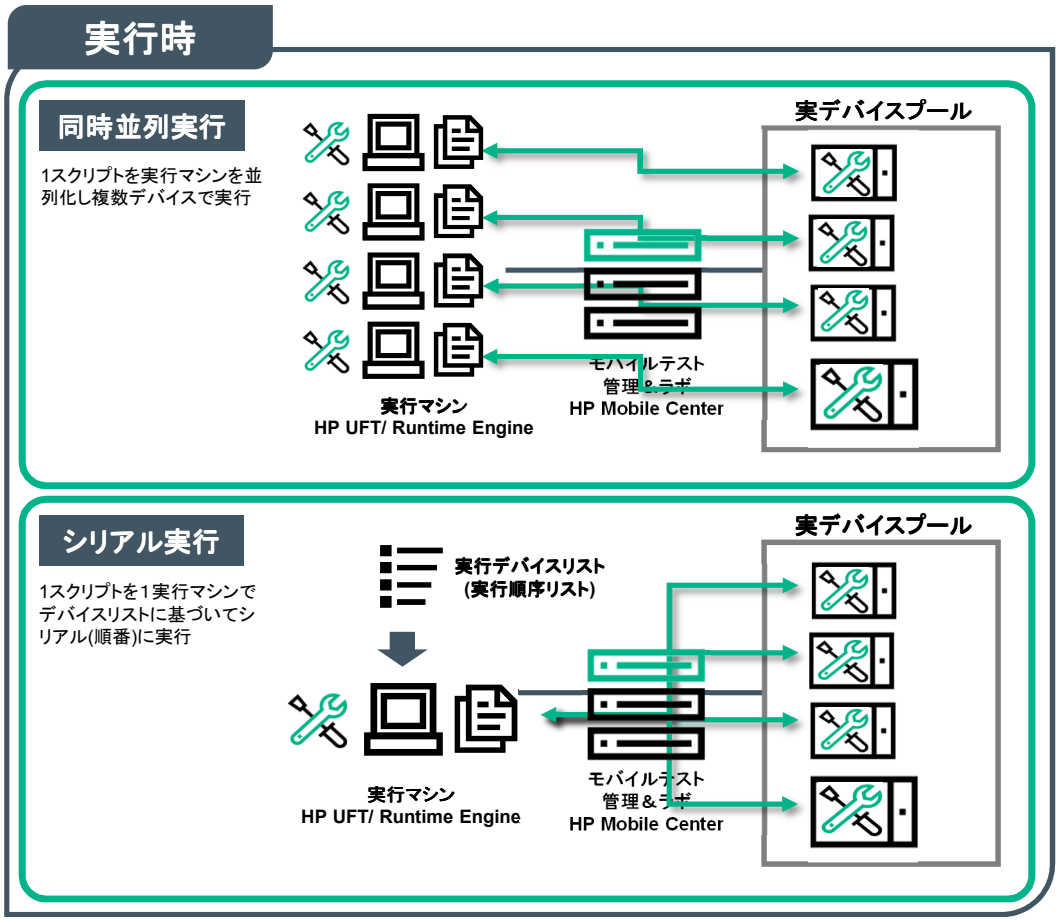
プロセス連携を踏まえた
モバイルテスト自動化

1. 自動化取り組みの前に知っておく事と注意点
2. 自動化率について知るべきポイント
3. HPE UFT & HPE MC を使った自動化スクリプト作成手順
4. テスト用モバイルデバイス＆アプリ管理（ラボ管理）
5. CI/CD自動化プロセスへの統合方法とTIPS
6. 複数デバイス＆アプリケーションのコントロール方法
7. インフラ構築＆デプロイプロセスとテストツールの連携方法

1つのスクリプトを複数デバイスで実行し効率化(注)



- 記録時は基準となる1デバイスを利用し自動実行用スクリプトを作成
- スクリプトは数や複数バージョンを管理するために必要に応じてSCMやテストマネジメントツールでバージョン管理



Hewlett Packard Enterprise

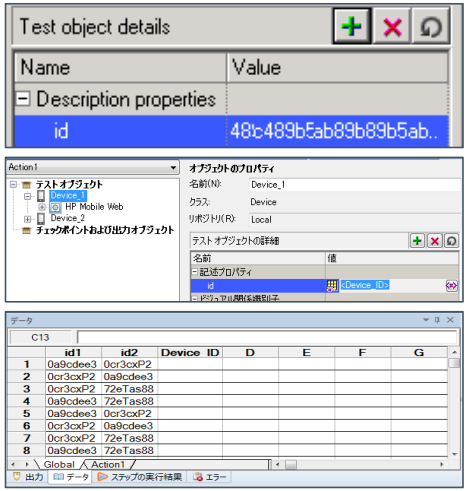
(注)現在はWeb以外はプラットフォーム毎

自動スクリプトによるマルチデバイス、アプリコントロール記述方法

サンプルスクリプト

自動実行用
UFTテストスクリプト

```
Device("Device_1").App("MyApp").Launch DefaultInstallBehavior, DefaultRestartBehavior
Device("Device_1").App("MyApp").MobileList("tableMainFirst").Select 5,0
Device("Device_1").App("MyApp").MobileDropDown("RegularPicker").Select "Arman",0
...
...
Device("Device_2").App("AnotherApp").Launch DefaultInstallBehavior, DefaultRestartBehavior
```



オブジェクトプロパティのデバイスIDをパラメータ化

パラメータをデータテーブルでデータ駆動適用

Launch関数引数（object.Launch [InstallApp], [RestartApp]）

- InstallApp
  - DefaultInstallBehavior:「Record and Run Settings」設定に基づきアプリをインストール
  - DoNotInstall: インストールしない
  - Install:「Record and Run Settings」設定を無視してインストール
- RestartApp
  - DefaultRestartBehavior:「Record and Run Settings」設定に基づきアプリを再起動
  - DoNotRestart:再起動しない
  - Restart:「Record and Run Settings」設定を無視して再起動

プロセス連携を踏まえた
モバイルテスト自動化

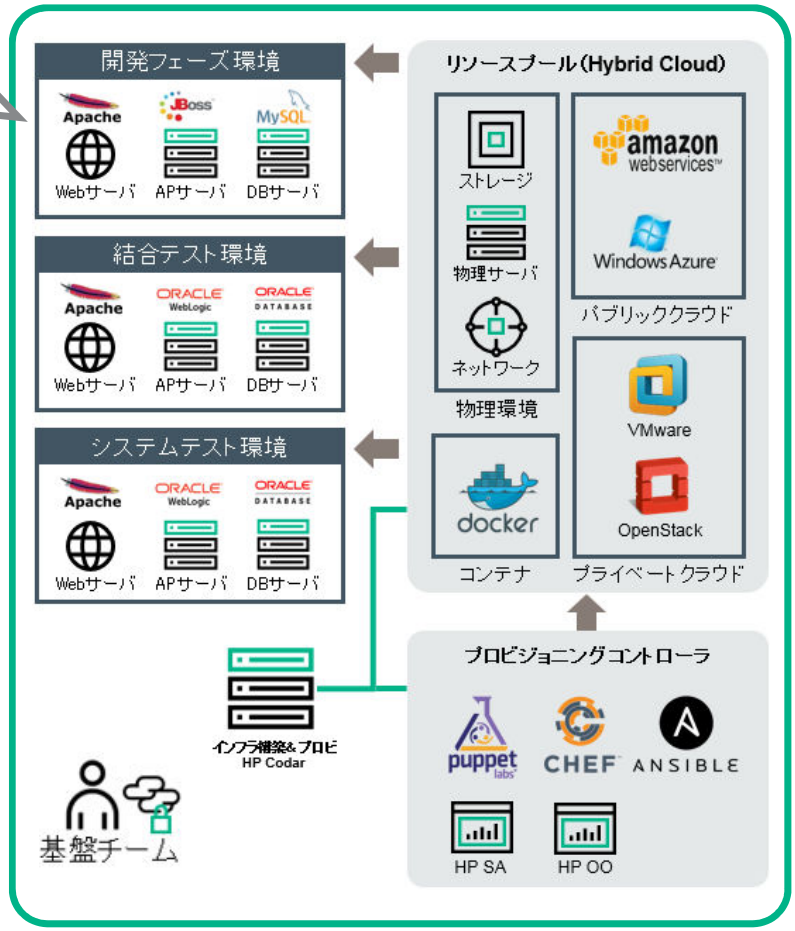
1. 自動化取り組みの前に知っておく事と注意点
2. 自動化率について知るべきポイント
3. HPE UFT & HPE MC を使った自動化スクリプト作成手順
4. テスト用モバイルデバイス＆アプリ管理（ラボ管理）
5. CI/CD自動化プロセスへの統合方法とTIPS
6. 複数デバイス＆アプリケーションのコントロール方法
7. インフラ構築＆デプロイプロセスとテストツールの連携方法

インフラ環境で動的に変わる環境情報がある場合の課題と対応

- クラウド環境でオーケストレーションツールで構築された環境においてIPアドレスなど環境情報が動的に変化する場合、テストスクリプトが動作しない

- 回避策としては、次が考えられる（IPアドレスの場合）
  1. 割り当てられたIPアドレスを取得し DNS を修正するスクリプトを実行する
  2. 予めテストスクリプトのアドレス部分をパラメータ化しておき、割り当てられたIPアドレスを取得しスクリプトを実行する

次スライドで2の方法を説明します



テストツールのアクセス先を動的に変更する設定例（UFTパラメータ利用）

Codar

プロビジョニングコントローラ

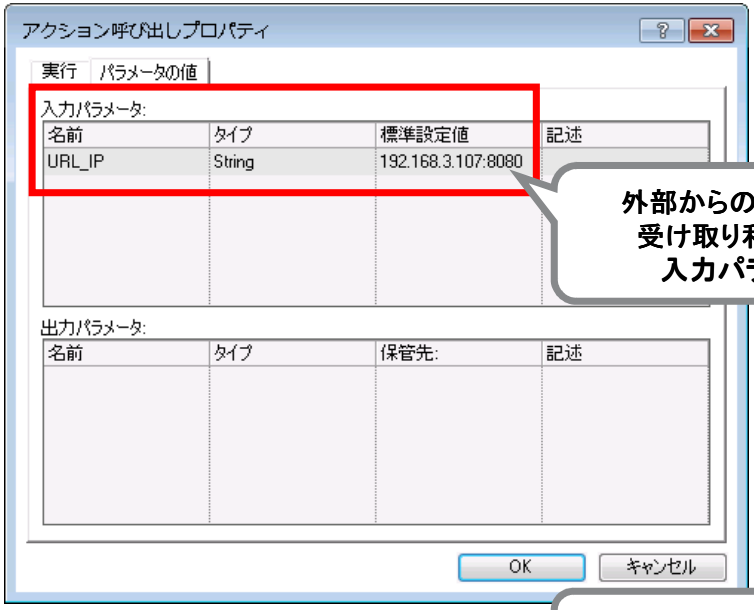
puppet labs　CHEF　ANSIBLE

HP SA　HP OO

プラットフォームモデルで定義された自動デプロイに基づき Codar で動的環境情報（IPアドレス等）を取得可能

HPE OO などを利用し HPE UFT の実行時の動的パラメータとして渡す処理を定義しておく

UFT



外部からの動的パラメータを受け取り利用できるように入力パラメータを設定

入力パラメータを利用してURLを変更できるようにスクリプトを編集

まとめ

HPE の優位性

Capgemini 社のWQRに7年連続で協力、企業の品質への取り組み状況をリサーチ

- 最新版（2015年10月版）では、世界の1500人以上の開発者、テスターを対象にリサーチを実施
- 2014年度に比べ、テストケースに対する自動化率が28%から45%まで大きく伸びており、世界的にもテスト自動化にフォーカスがあたっている
- キーベネフィットとして「早い不具合検出」、「テスト内容の透明性」、「テストサイクル時間の短縮」、「テスト費用の削減」、「再利用性」が認識されており、テスト自動化への投資を後押ししている

対象企業内テスト自動化率

Hewlett Packard Enterprise

まとめ

本セッションのまとめ

- モバイルアプリケーションの自動化は、デバイスで動作するアプリの機能テストを自動化するスクリプト作成をどうするかにフォーカスされがちだが、自動化の範囲としてどこまで本来はカバーすべきか、期待する自動化範囲に対する自動化率を押さえて取り組むのがよい。
- テスト自動化ツールをCI/CDプロセスの中で活用するためにはビルドしたアプリパッケージのテスト環境デバイスへのデプロイ、CIツールからのテスト実行連携、品質計測のためのテスト管理ツールとの連携が出来るかを確認しておくとよい。
- DevOpsを実現する中でインフラ構築＆プロビツールによる動的環境情報がテストスクリプト実行に影響するのであれば、どのように動的環境情報を取得し利用できるかを事前に確認しておくとよい。

「どこから手をつけるべきか」や「進め方のベストプラクティス」などお気軽にご相談ください

Hewlett Packard
Enterprise

Thank you

Contact information