



ALM 手法のご紹介 第八回

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 小宮山晃

前回に引き続き、「HP の考える ALM アプローチ(コアライフサイクルと完全ライフサイクル)」の冊子である「アプリケーションハンドブック～最新のアプリケーションライフサイクルを理解するためのガイド」の概要紹介をしていきます。

第 5 回～7 回にかけて従来の SDLC によって培われたお客様のプロセス管理手法を HP がヒアリングし把握した、優れた開発チームに共通する 4 つの特性のうちの「変更対応力」、「予測能力」、「反復性」について説明しました。今回は、最後の 4 つ目の「高品質」について優れた開発チームにみられる取り組みを説明します。

高品質

アプリケーションの近代化(アプリケーションモダナイゼーション)は単にレガシーシステムからマイグレーションとしてだけでなく最適かつ疎結合などの柔軟性をもった強いシステムを実現する手法であり、多くの IT ベンダーはそのための支援ツールやコンサルティングサービスを提供しています。

しかしながら、アプリケーションの近代化にもテスト方法には改善すべき課題があると考え、「テストの近代化」の必要性を説明してきました。また、現在の開発プロジェクトでよく聞かれるのはプロジェクト期間の短縮化の傾向です。新しいテクノロジーやフレームワークは絶え間なく変化しており複雑さを増している中、期間の短縮化も重なり開発チームのストレスは増加し続ける一方ではないでしょうか。

以上のような環境下において、どう高い品質を維持するのかはアプリケーション開発における課題であり、HP では解決案として HPQM(HP Quality Methodology)という CMMI や PMBOK をベースにした品質管理方法で段階的に改善していくマチュリティモデル(CMMI 同様 Level0 から Level4 までの 5 段階成熟度が定義されています)を提唱しています。HPQM の一部を紹介させていただくと、テスト計画、設計を前倒することによる、テストベースに潜む欠陥を早期に検出する効果やリスクベースドテストの効果、トレーサビリティによる派生開発時の影響範囲を可視化する効果などがあり、非常に多くの有益なプラクティスが詰まっています。HPQM については機会があれば別途ご紹介させていただくとして、ここでは、特にアプリケーションの近代化と共に増す複雑さ(「テストのしづらさ」とも言いかえる事ができます)に対して、以下の 3 つの問いを考えてみたいと思います。

1. 機能は正しく動作するか？
2. 性能目標は達成しているか？
3. セキュアか？

機能は正しく動作するか

アプリケーションの近代化によりオープン系システムへ進むと、いわゆる疎結合のアーキテクチャーが採用されていきます。それにより、いままで機能確認としてユーザー側の GUI を機能テストしていただけではなく、シェアードサービスやサブコンポーネントに対する機能テストの重要性も増してきます。そのため、シェアードサービスやサブコンポーネントの開発チームはそれらをテストするためのドライバとなる部品(テストドライバ)を機能テストのために作成しなければなりません。

システムが大きくなるにつれてこれらのサービスやコンポーネントは大量になり、テストドライバ作成もかなりの工数的な負荷になっていきます。また、システム規模が大きい場合は分散開発体制がとられる事が多いため、自分達のチームがつくったサービスやコンポーネントが「正しく機能する」をテストする事は他のチームとの問題切り分けのために非常に重要になっていきます。ここで以下のようなコンポジットアプリケーションのある例を考えてみます。

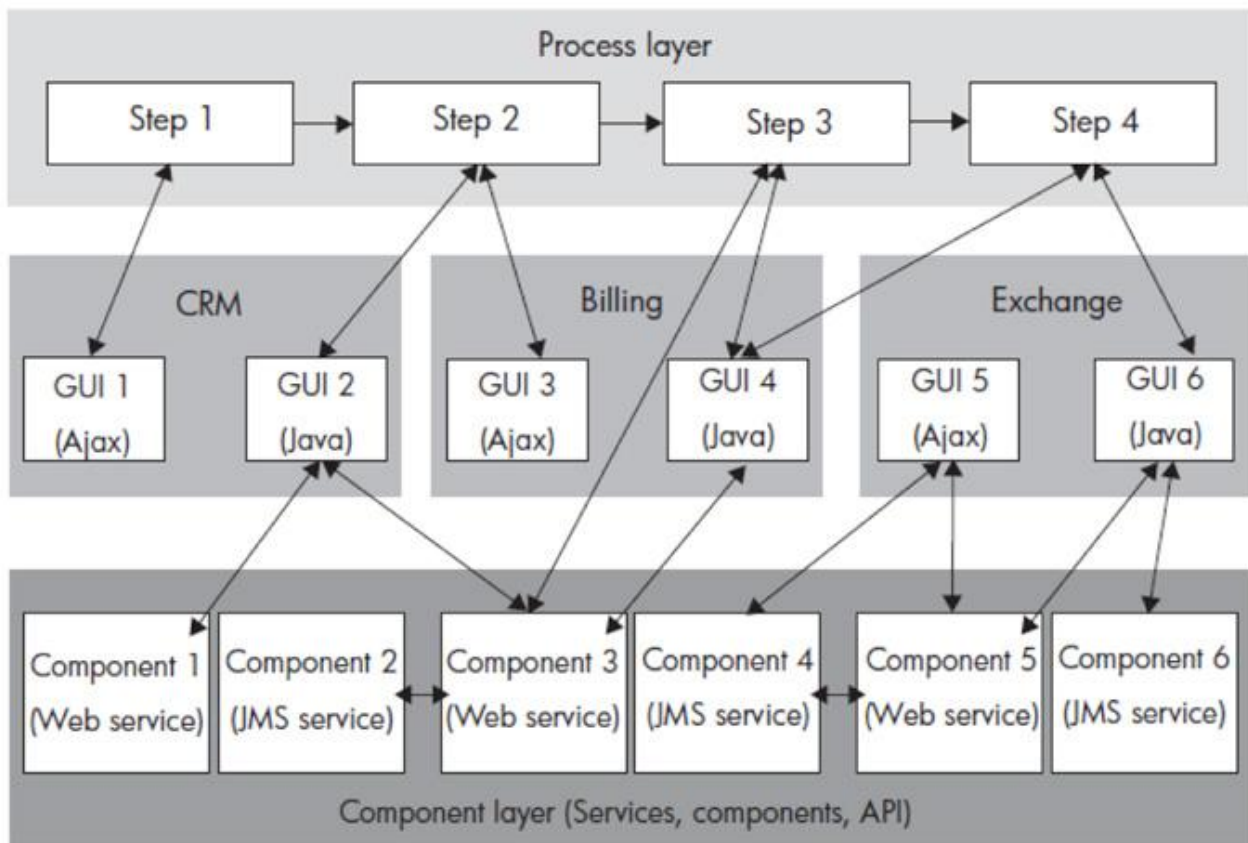


図 1

ユースケースシナリオなどから機能テスト用テストケースを作成する際に、様々なレイヤー(この例では大きく3つのレイヤーで表現)内の複数コンポーネントやサービス呼び出す場合、個々のサービスは予めテストを済ませておくことが必要です。実際には、テストを済ました後でも機能追加や不具合修正のたびに影響範囲の確認やスモークテストの観点で回帰テストを実施する事になります。このように、一度のテストで終わる事はないため、テスターの工数は計り知れないものになります。

ここで、シェアードサービスやサブコンポーネントを「GUI がない」という意味で「ヘッドレス」サービスと呼ばせてもらいます。ヘッドレスサービスを機能テストするには、先ほど説明したドライバに相当するものをヘッドレスサービスの開発チームから共有してもらいテストを実施する事になりますが、不具合修正などで呼び出し方法やパラメータの定義が変わった場合にはドライバ自体を修正する事も必要になってきます。これがヘッドレスサービスの数が多いほど、そして分散開発体制であるほどテストの工数負担を増やす原因にもなります。例えば、上のコンポジットアプリケーションサンプルの例で請求書の顧客住所を変更するプロセスを考えてみると、課金(Billing)アプリケーションは CRM システム内の住所を相互参照しています。CRM で住所が更新されていないと、請求トランザクションは失敗します。この失敗は課金システムのエラーに見えますが、実際には呼び出し先である CRM 側に起因しています。

これらが示しているのは、ヘッドレスサービスを効率よくテストするためには、次の課題を解決する必要があるということです。

- ・ 機能テストケースにおいてヘッドレスサービスについてもテスト計画段階からどのように検証するかをテストアプローチとして体系化、確立すること
- ・ テストドライバやテストスタブなどをより効果的に作成しヘッドレスサービスも GUI の機能テストと同様にテストが検証できるようにすること

ヘッドレスサービスを考慮したテスト計画を立てるためのテストベースとして、ビジネスプロセスモデル(BPM)は有効でしょう。単純に機能単位でテストを実行していくアプローチではなく、全体としての業務、ビジネスプロセスを把握して呼び出されるシステムを通した有効な検証ができるようになります。ミドルウェアベンダーの多くはビジネスプロセスモデル作成を支援する BPM ツールを提供しておりますが、UML のアクティビティ図や業務フロー図で表現したビジネスプロセスでも有効なテストベースとなるでしょう。BPM ツールを利用した場合、作成モデルを XPDL 形式で出力できる機能が実装されている事が多いのですが、HP ALM11 では XPDL 形式のビジネスプロセスモデルを取り込む事が可能です。ビジネスプロセスモデルの各アクティビティに要件、テストを関連付けする事ができるため、ビジネスプロセスモデルから作成されたテストを把握しやすくなります。

さらに、PDCA サイクルにより業務の俊敏な改善を実現している組織であればモデルを頻繁にアップデートされますが、その場合も既存のアクティビティについてはそのまま要件、テストの関連付けを引き継ぐ事ができるため、テスト設計の短縮化に有効です。

ヘッドレスサービスをテストするために投入するデータとその戻り値、期待値の検討資料としてインターフェース仕様書はテストにとって重要であり、早めに確認することでテストケースのパターンをどう作成するかを前倒して検討できるようになります。さらに漏れを発見した場合にも早めに確認することで手戻り工数を圧縮する事にもつながります。

技術的な点でいえば、テストドライバやテストスタブをより効率的に作成するツールもあります。たとえば、HP Service Test(HP ST)はテストドライバとして利用する事ができるツールであり、WSDL, JMS, RESTといったヘッドレスサービスに対して少ない工数で検証ができるようにスクリプトを作成、再生できます(図2に画面例)。スクリプトはデータ駆動型で再生する事ができるため、インターフェース仕様書などから設計した投入、出力値のパリエーションのテストを効率化する事ができます。また GUI の機能テストツールである HP Functional Testing(HP FT(旧 QuickTest Professional))のスクリプトも連携できるため HP FT で実行した結果の画面出力値を HP ST の投入パラメータとして

適用すると言った事が可能になります。HP の機能テストツールは、HP FT と HP ST を交互に連携できることから、現在は HP Unified Functional Testing 11 という名称になっています。

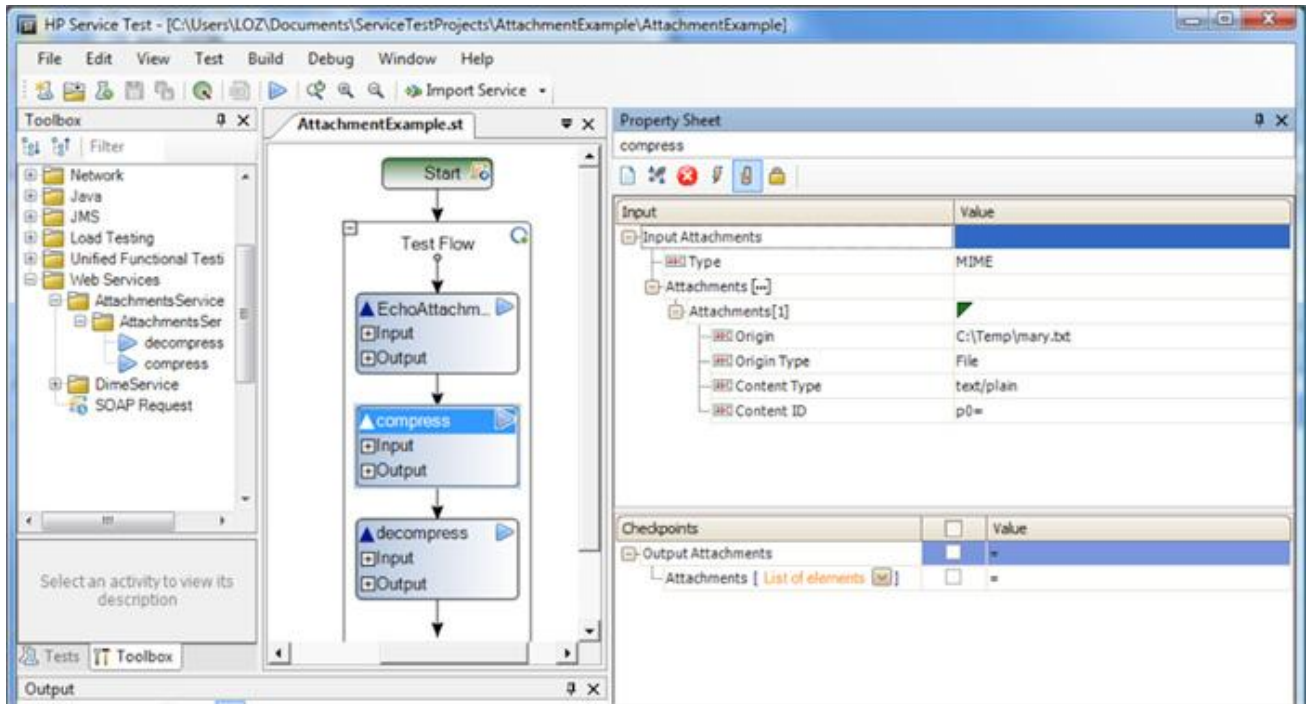


図 2

HP ST ではビジュアル化されたヘッドレスサービスの呼び出しフロー作成が可能であるため、テスターがどのパラメータを投入するか、どのパラメータをデータ駆動にするか、期待値と実際の戻り値の判定といった作業を統一された方法で実施する事ができます。

さらに、ツール連携の視点でいえば、HP QC 側へ HP ServiceTest Management (HP STM) をインストールする事で、HP ST で作成したスクリプトを本連載の第 3 回で紹介した HP FT と HP QC の連携と同様に HP QC にて一元集中管理する事が可能です。

テストスタブについては現在 Web サービスについては HP ST に同胞されている Web Service emulator を利用する事で可能です。他のサービスについてもテストスタブとして利用できるツールを計画しており、ご提供できる時期には是非ご紹介させていただきたいと思っております。

まとめ

前回からの続きとして優れた開発チームに共通する 4 つの特性の 4 つ目である「高品質」についてお話させていただきました。品質管理方法として HP 独自の HPQM というメソドロジーについて触れましたが、今回はアプリケーション近代化により疎結合化がもたらすヘッドレスサービスに対してどのように機能テストすべきかにフォーカスして説明しました。HPQM については、また別の機会にご説明できればと思います。

次回は、引き続き「高品質」の「性能目標は達成しているか？」についてお話させていただきます。