ALM 手法のご紹介　第五回

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

第四回の記事では「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）」にて、ALM 導入で求められる全体最適化を実現する HP のアプローチを解説した冊子「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」から概要を紹介しました。

また、紹介の中で完全ライフサイクルの「IT 戦略（PLAN）」にて要求を要件として採用するためのポリシー遵守のためのワークフロー自動化、指標の見える化を支援する「HP　Project and Portfolio Management Center（HP PPM）」を紹介し、「HP Quality Center（HP QC）」、「HP Service Manager（HP SM）」 との連携についても説明させていただきました。

コアライフサイクル

開発を中心に考えると、ライフサイクルとは一般に SDLC と呼ばれる開発プロセス（ウォーターフォール型開発や反復型開発など）を指してきましたが、前回説明したように、HP では全体最適化を実現するためには、SDLC と呼んできたライフサイクルを中核に、「IT 戦略（PLAN）」から「廃棄（RETIRE）」までを含めた完全ライフサイクルを理解する事を重要視しています。完全ライフサイクルの中核という意味で SDLC はコアライフサイクルと表現し、「IT 戦略（PLAN）」の次の「開発（DELIVERY）」にあたる部分になります。

この関係はちょうど SDLC をオーケストラの演奏家であるとしたら、ALM は指揮者にあたると言えます。指揮者は演奏家が演奏中に作り出す音（要件、テストケース、不具合）の把握（トレーサビリティ）、演奏家へのテンポや拍子（ワークフロー）を指示、パートごとの音高（KPI）をリアルタイムに把握しコントロール（管理）するなど、様々な情報を楽曲演奏中（リアルタイム）に必要としています。

前回でツールが支援するポイントは「自動化と管理機能の統合」だと説明しましたが、「IT 戦略（PLAN）」で言えば、ポートフォリオの管理とワークフロー自動化がまさにそれを象徴しています、これは「開発（DELIVER）」でも重要な考えです。開発において自動化というと「テストの自動化」を思い浮かべるでしょうが、それ以外の数多くの複雑な作業にも自動化を適用する機会があります。小規模の開発では、チーム全員とリーダー（プロジェクトマネージャ、テストマネージャ）は細かい調整を直接やりとりして行く事も可能でしょうが、大規模になると複数のチームを調整するためにどれだけのチーム間の情報のやりとりや受け渡しが必要になるか想像してみてください。オフショア、ニアショアに関わらず、社内でも大規模開発であれば複数のチームに分かれてそれぞれ担当するプログラムを開発する事が多いでしょう、その際、ビジネスアナリスト（日本でもここ数年で今までプロジェクトマネージャ等が兼任していた業務分析・要件分析を海外のように専門職として確立させようという動きが活発になっています）が利用部門の要求を開発要件に組み入れた事を開発マネージャはどのように知り、各チームに伝えて開発指示を出すのでしょうか。また、中国のオフショアチームが修正した不具合をテスターは再テスト可能になっている事をどのように知るのでしょうか。規模が大きくなる

ほどこれらのやりとりが同時並行的に複雑になっていきますが、このやりとりの回答が「一日一回のチーム会議」や「電話」であるならば、ワークフローの自動化を検討すべきです。

ワークフロー自動化として、HPQC ではトリガーメカニズムを採用しています。具体的にはトリガーとして「不具合のステータス変更」や「新しい要求の発生」が発生した際に、影響を受けるチームメンバーへ、新たに発生したタスクが原因でなんらかのアクション（調査、コード修正、再テスト）が必要になることを自動通知します。また、作業項目の遅延など、何らかの問題が生じると設定しておいたエスカレーション基準にもとづいてアラートを発行すると言う事も可能です。また、標準のワークフローではない場合でもワークフローのカスタマイズ機能（図1）を利用して採用する事も可能です。

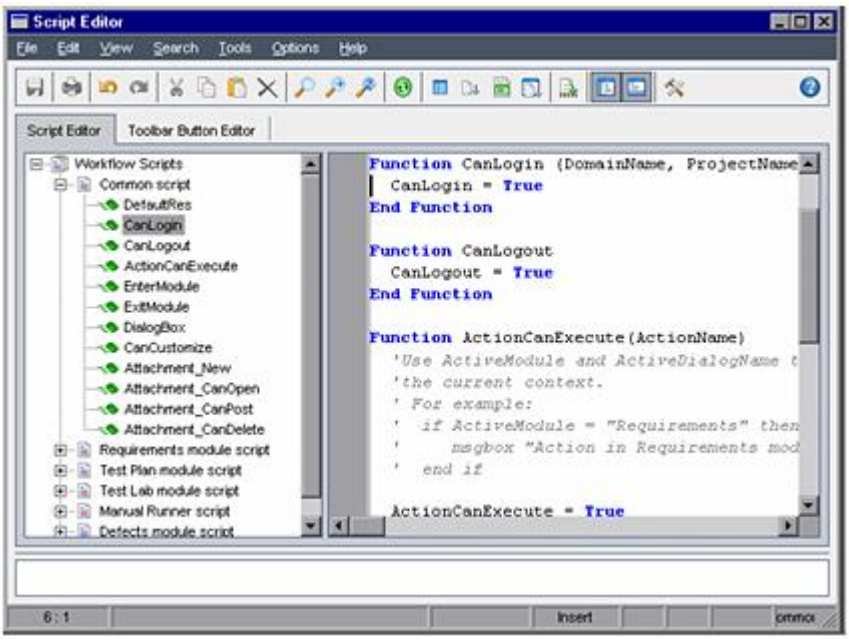

図 1

このように「自動化と管理機能の統合」は完全ライフサイクルの全体で有効に適用できるツールのメリットになります。また、コアライフサイクルについて従来の SDLC によって培われたお客様のプロセス管理手法から HP が把握した優れた開発チームに共通する特性として次の 4 つがあります。

1. 変更対応力
2. 予測能力
3. 反復性
4. 高品質

変更対応力

現在のビジネス業務のほとんどは、アプリケーションによって支えられています。いいかえれば、ビジネス業務のわずかな変化でもアプリケーションに対する機能変更、機能追加を伴う事になります。そのため、開発チームにとっては、ビジネス変化への対応は重要な課題です。しかし、現在のビジネス業務、組織の変化スピードはますます早くなって

おり、開発チームにおける悩みの種です。前回、要求におけるポリシー、ワークフローの自動化の導入効果についてお話しましたが、ここでは、変更要求に対してタイムツーマーケットでの開発対応力について話したいと思います。

変更要求は一度に 1 件ということはほとんどないでしょう、複数の変更要求についてまず頭を悩ませるのが影響分析にかかる時間です。複数のチーム間に点在する過去の資産、成果物をチェックするにしても、管理方法やフォーマットがばらばらであったり、時には現在のビルドをアップデートしたドキュメントがなかったりします。様々な成果物が散在していることは、成果物同士の関係や依存性を一元的に把握できないという事です。

要件を効率的に補足し、監査とロールバックのためにそれらの要件の変更管理を行う必要があります。具体的には既存の成果物をステークホルダー全員で共有できる一元的な保管場所、保管方法が必要になります。さらに、既存の成果物（テストケース、不具合、ソースコード）は、要件とリンクされているべきです。それによって、変更要求が来た場合に、関係する要件が影響を及ぼすソースコード、テストケースが把握できます。また、ソースコードや、テストケース、不具合に対する変更はどの要件にリンクされているかも把握されるべきです。要件の変更影響に対して追跡が容易であることを、「トレーサビリティ」があると言います。HP QC では、要件、テストケース、不具合のモジュールからそれぞれリンクを辿り影響範囲を把握する事、つまりトレーサビリティが可能です。また、トレーサビリティマトリックス機能で、特定の要件同士の関係（例：あるビジネス要件に関係する機能要件）や、要件に影響するテストケースを分析し、表形式で表示させる事ができます（図 2）。

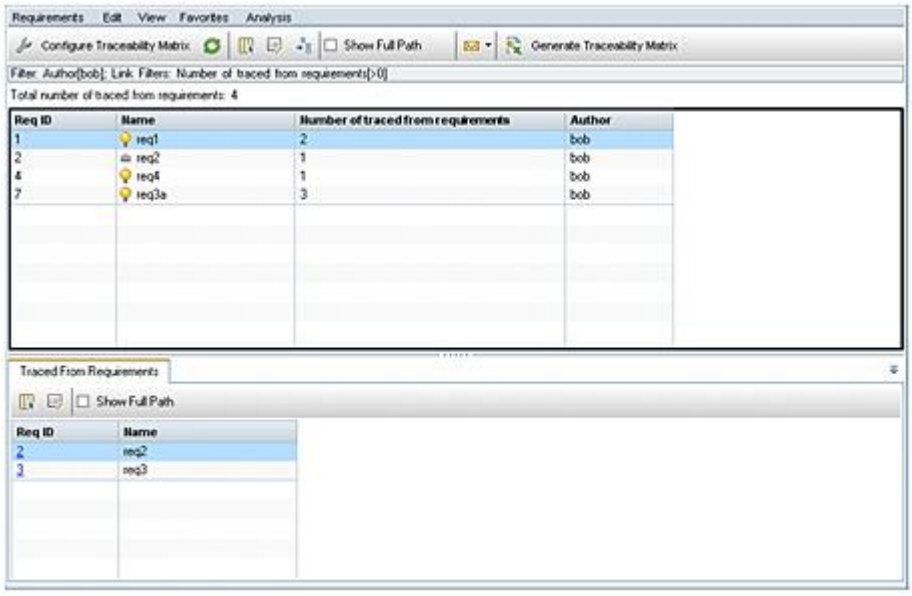

図 2

変更を徹底的に行う事は、時間がかかる困難なプロセスであり、見落としや人為的なミスが起きやすいものです。したがって、トレーサビリティは現在のビジネス業務の変化スピードへタイムツーマーケットで対応するためには不可欠であると言えます（つまり、ビジネスのアジリティとアプリケーションのアジリティとは同じものだと考えられます）。HP QC は上述のようにツールとしてこのトレーサビリティを実現し、コアライフサイクルにおける変更対応力を支援することが可能なのです。

まとめ

今回は、前回に引き続き HP の考える ALM アプローチ概要について前回紹介した「完全ライフサイクル」の中核にあたる「コアライフサイクル」についてお話させていただきました。「IT 戦略（PLAN）」と同様に「開発（DELIVERY）」においても HP ではツールで「自動化と管理機能の統合」を支援し、開発規模に依存しない完全ライフサイクル視点でのメリットを提供できる事をご説明しました。

コアライフサイクルには従来の SDLC で培われたプロセスのノウハウの中から HP が優れた開発チームに共通している 4 つの特性についてご紹介し、そのうちの「変更対応力」についてどのような特性なのかお話しました。次回は、残りの特性について HPQC を用いた例などを入れながらお話させていただきたいと思います。

「自動化と管理機能の統合」の自動化という所で、今回は触れなかった「テストの自動化」があります。HP では主に回帰テストを支援するテスト自動化ツール（キャプチャ・リプレイツール）として HP QuickTest Professional（HP QTP）があり、第三回で HPQTP と HPQC の連携方法や連携による効果についてご紹介しておりますので、併せてお読みいただければと思います。　テスト自動化についても、有効活用するための成熟度フェーズがあると考えられます。詳しくはここでは触れませんが、大まかな成熟度フェーズとして

- 「画面ができあがった段階からツールで画面遷移を記録し、回帰テスト用スクリプトを作成するフェーズ」
- 「ツールを意識した開発を行い、ツールでカバーできるテストケースを広げるフェーズ（HP QTP でいう“オブジェクトの認識”を考慮した開発）」
- 「画面ができあがる以前からスクリプト作成準備を開始しテスト準備を早めるフェーズ（テストケースとスクリプト作成を並行作業または統一作業できるフェーズ）」

があります。