



ALM 手法のご紹介 第三回

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

前回の第二回目は、Application Lifecycle Management(以後 ALM)手法を支援するツールとして ISTQB では「テストマネジメント支援ツール」に位置づけられる HP Quality Center(以後 HPQC)を紹介させていただき、ツールがもつ 2 つの特徴をあげさせていただき、そのうちの「過剰作業の抑止による作業量の最適化」についてお話をさせていただきました。

三回目では、前回テストマネジメントツールが特徴として書いたもう一つの「情報連携・自動実行による生産性向上」を実際に利用していく際のフローを想定しながら説明していきたいと思えます。

「情報連携による生産性向上」効果

テストに関する情報のあるべき姿とは、そんな事はよく分かっていると思われるでしょうが、「情報があるべき場所に整理されており、必要な時に取り出せる」という事です。

当り前のことですが、この事はアプリケーションのライフサイクル観点的に言っても重要で、一度リリースした後も廃棄されるまで、派生開発(機能追加、バグ修正)時にテスト資産を再利用する事ができますし、運用中の不具合も同じ方法で登録する事で不具合に関する機能はテストされたのか、そもそも有効なテストケースがあったのかと言った情報をトレースすることができます(実際にトレースするために実際には関連する要件、テストケース、不具合の間で正しくリンクが張られている必要があります)。

このように情報を一元化しておく事(具体的にはテストマネジメントレポジトリというデータベースに情報を集約する事)と同時に重要なのは、どのように情報を整理していくかと言う点です。さらにいえば、整理の仕方についてはプロジェクト単位で実施するよりも全社的に整理の仕方を標準化する事が「テスト資産の再利用」の効果を最大化する観点からも重要と言えます。

今までの内容を実際のテストプロセスのフローを横軸として整理したものが下の図 1 になります。テスト計画からテスト報告までにマネジメントすべき情報の入力と出力が一元化されたテストマネジメントレポジトリに対して行われているのが把握できるかと思えます。HP のテストマネジメントツールである HPQC は一言でいえば、このテストマネジメントレポジトリに対するユーザーインターフェースと言えます。

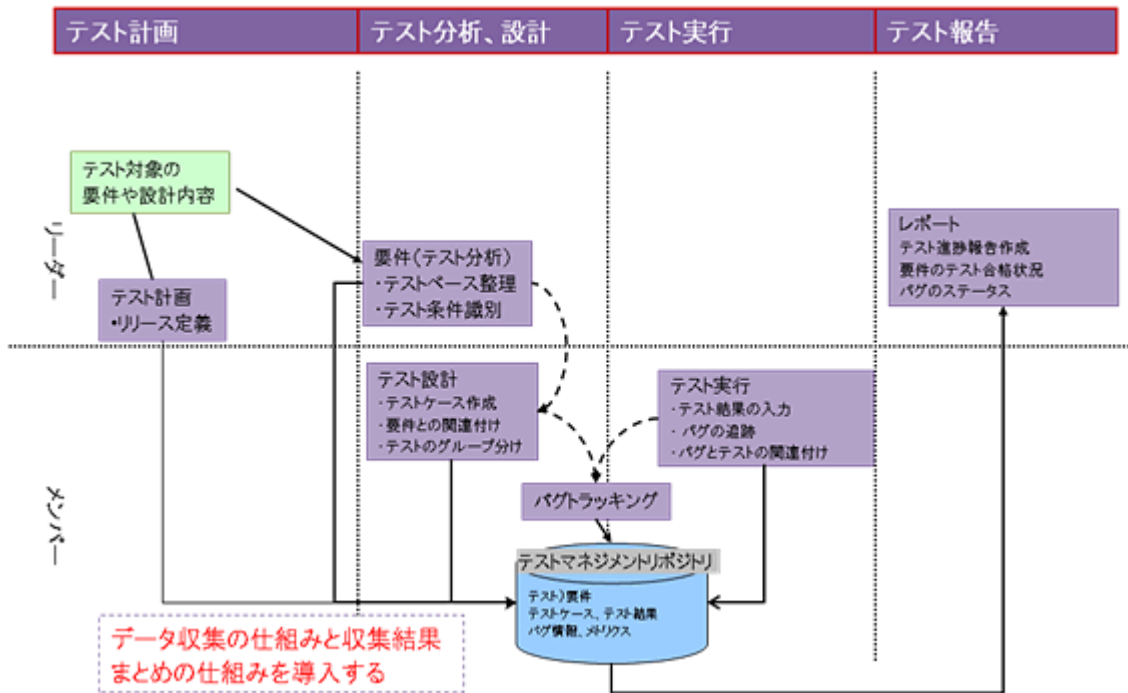


図 1

情報を一元化したら、集めた情報を有効に使うためにテストマネジメントツールの特徴である「情報連携」ができます。HPQC であれば、代表的なものとして次のような情報連携機能を持っています。

- ・ テスト状況のリアルタイム表示
- ・ 要件、テスト、不具合のトレーサビリティ
- ・ 基準となるテスト資産のベースライン化

「テスト状況のリアルタイム表示」であれば、次のようなグラフ(図 2)を作成して、現状を把握する事ができます。テストケース、不具合を別々のスプレッドシートで管理してそれらをマクロ機能で情報をグラフ化する場合、開発規模が大きくなるほど大量のスプレッドシートを管理し、その情報が最新のものかどうか管理するという追加の管理作業が発生します。

HPQC を利用した情報連携による管理工数の生産性向上という視点はツール導入を検討する際の大きなポイントになります。

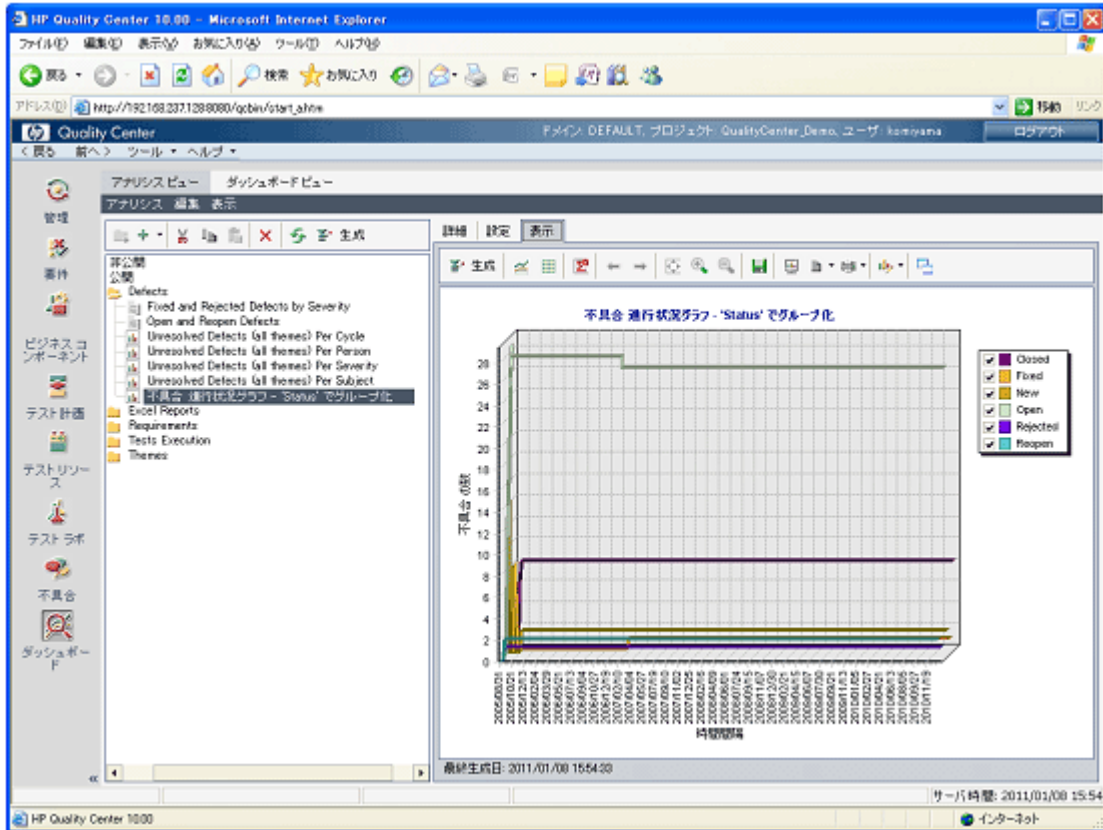


図 2

図 3 は、「基準となるテスト資産のベースライン化」の例ですが、テスト要件やテストケースを再利用する場合（派生開発など）、基準となる要件、テストケースに対してスナップショットをとっておくことで、再利用時の差分を比較することができるようになります。機能追加をした際にはベースラインで取得したテストケースと比較してどのようなテストケースが追加され、また既存のテストケースから今回は実行必要性がないと判断し削除されたのかといった考察を行う事ができます。

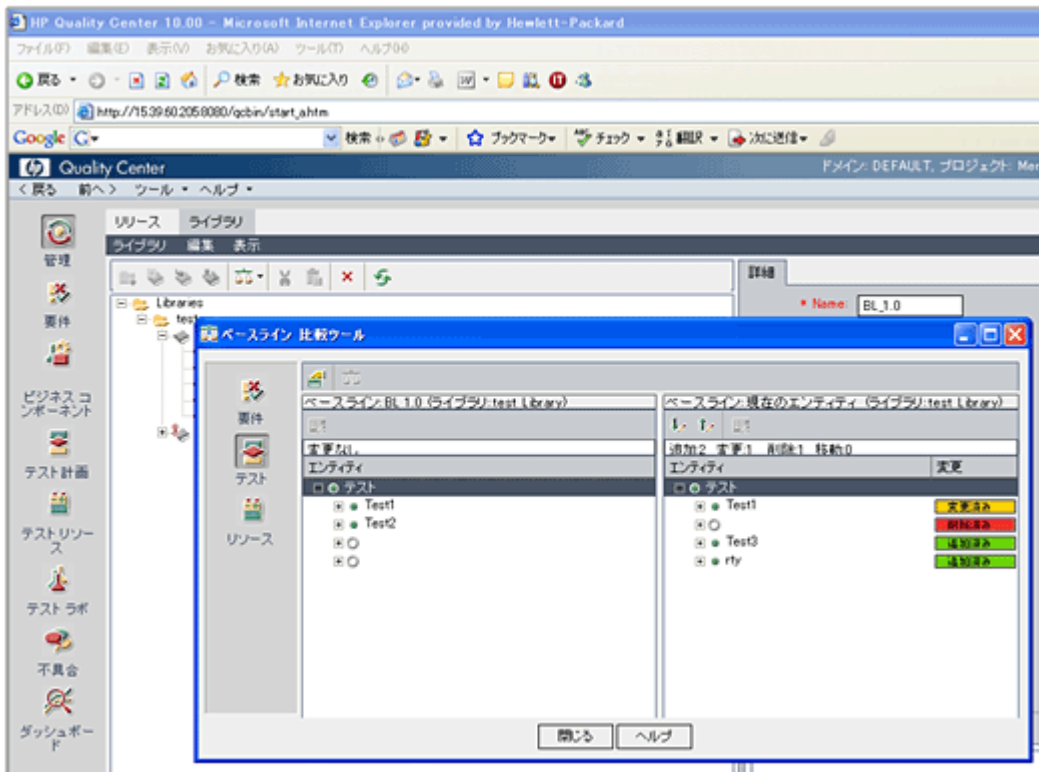


図 3

「自動実行による生産性向上」効果

また、自動実行による生産性の向上についても HPQC であれば、代表的なものとして次のような機能が実装されています。

- ・ テスト実行の簡便化
- ・ テスト結果収集自動化
- ・ 変更履歴自動保存
- ・ 報告書の自動作成
- ・ バグ修正用の情報収集自動化

「テスト実行の簡便化」、「テスト結果収集自動化」として、手動テストを実行する場合を考えてみます。対象となるアプリケーションと同時に図 4 で表示されている Manual Runner を起動しておく、テストステップ毎にどんな操作をすればいいの、次は何をすればいいかを指示してくれますし、かつ不具合登録もこの Manual Runner 上から行う事ができます (Manual Runner と HPQC が連携)。

これにより、テストケースが既述されたスプレッドシートをみながら操作し、結果を別のスプレッドシートに記述したり、不具合がでたらまた別のスプレッドシートへ登録するといった作業を簡便化することができます。(次期バージョン 11 では HPQC の Enterprise Edition と Premier Edition に相当する ALM11.0 では、手動テストを支援するツールとして HP Sprinter が新しく登場します。)

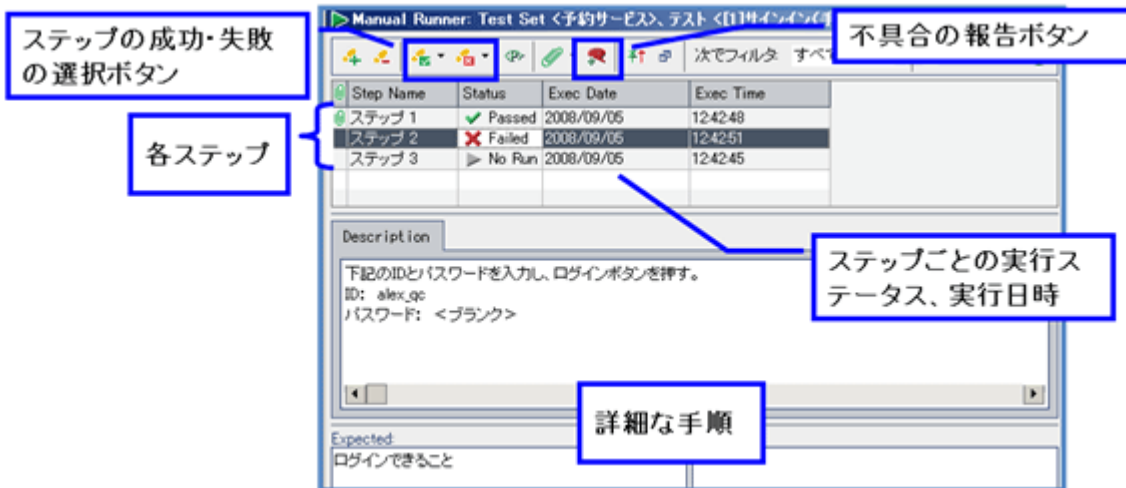


図 4

さらに、複数のテスト(自動テスト、手動テスト)をテストセットとしてグループ管理したり、複数のテストの実行順序(フロー)や実行スケジュールを管理することも可能です。これにより、どの順番でテストを行うべきかといった管理も簡便化することができます(図 5)。

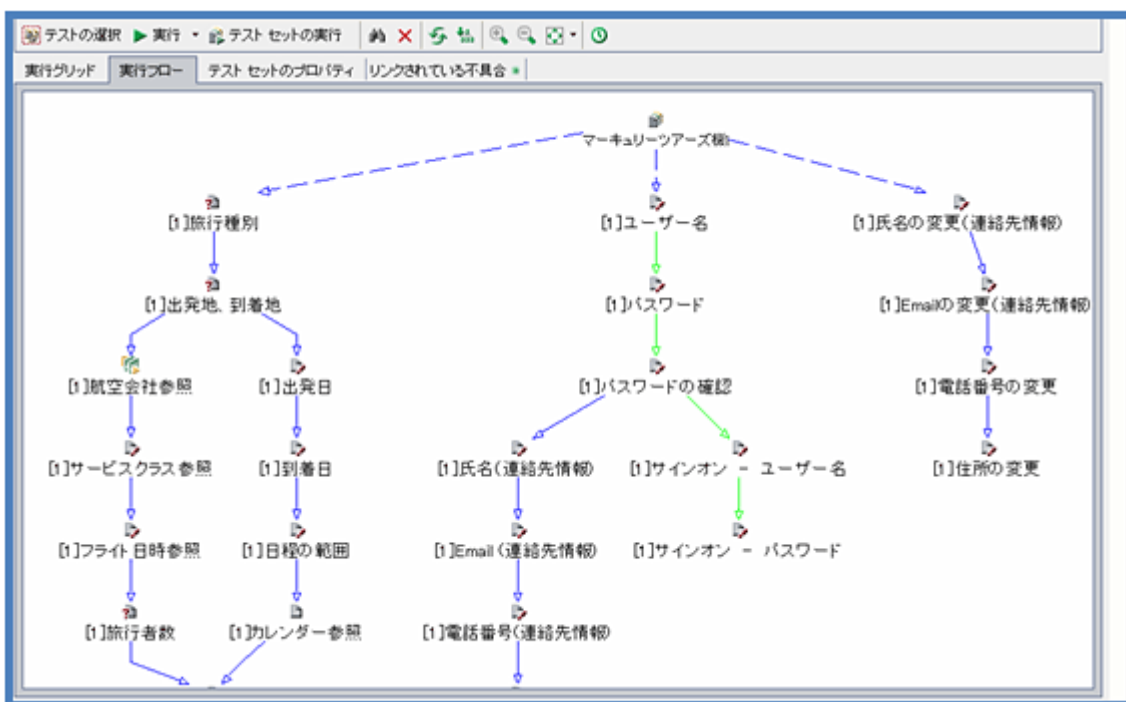


図 5

図 5 の説明の中で自動実行と書きましたが、HP ではキャプチャ・リプレイツールとして HP QuickTest Professional (HP QTP) があります。アプリケーションの機能テストを自動化することで繰り返し(回帰)テストにかかる工数負荷を効率化することができます。

HP QTP と HP QC を連携した場合、HP QTP で作成したキャプチャ・リプレイ用スクリプトを HP QC で一元管理し、紹介した実行順序(フロー)、実行スケジュールで自動実行(リモートマシン上の HP QTP を利用して実行することが可能)しかつ、結果をリモートマシンから HP QC へ自動でアップロードし管理する事が可能になります(図 6)。

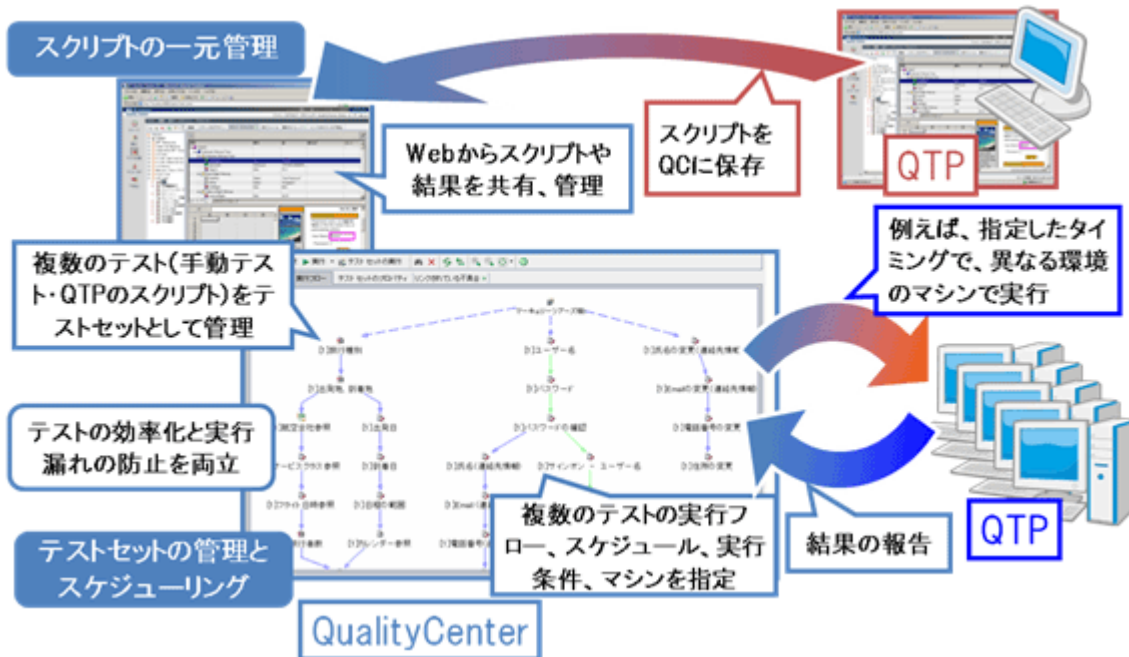


図 6

他にもご紹介できなかった機能が多数実装されていますが、それは別の機会にしたいと思います。

まとめ

前回第二回で、テストマネジメントツールの大きな特徴2つを説明させていただき、そのうちの1つ「過剰作業の抑止による作業量の最適化」について HPQM(HP Quality Model)の観点も交えながらお話をさせていただきました。

そして今回はもう一つの特徴である「情報連携・自動実行による生産性向上」について実際のテストプロセスのフローから情報の一元化と情報連携による生産性の向上、そして、自動実行による生産性向上について HP 社のテストマネジメントツールである HPQC の機能をベースにどのような事が実際にできるのかをお話をさせていただきました。

次回は、今回ご紹介した HPQTP や Manual Runner 以外のツール連携することによる効果について説明したいと思います。