

リHOST・リライトによる モダナイズに潜むリスク

保険業界ではリHOST・リライトでの脱HOSTの取り組みが進んでいる。しかし、コスト優先で安易にプロジェクトを進めると思わぬトラブルが起きかねない。実際に難航しているケースも目立つ。それを回避するには現行システムについて周到に分析し、プロジェクトを進めるべきである。

リHOST・リライト手法の採用が増えている

大手ハードベンダーのメインフレームのサポート終了や保守費用の値上げにより、脱HOSTに向けた取り組みが本格化している。保険会社の多くの基幹システムで利用しているメインフレームも例外ではなく、各社で取り組みが加速している。

限られた時間の中で安心・安全にメインフレームからの脱却を図るため、リHOST¹⁾やリライト²⁾によるモダナイズへの取り組みを進めている保険会社が多いが、難航している脱HOSTプロジェクトが増えている。その理由は、後述するアーキテクチャ変更に対する準備不足、データ移行・周辺システム連携の考慮不足、システム全体品質を確保するためのテスト準備不足にある。

リHOST・リライトする際の要注意点

リHOST・リライトは一般的に低コストと認識されているモダナイズ手法であるが、コスト目線で安易に採用してしまうと、プロジェクトが進むにつれて次のような問題が露呈してくる。

<アーキテクチャ変更に対する準備不足>

リレーショナルデータベースに置き換えることによって、プログラム構造を変更しない場合、新旧アーキテクチャの違いによる性能問題が発生する。

また、COBOLをJava等オブジェクト指向型言語に変換した場合、COPY句が冗長なJavaオブジェクト構成になることで移行後環境のメモリ不足が発生し、追加の

ハードウェア導入が必要となることがある。

<データ移行・周辺システム連携の考慮不足>

HOSTの固定長データをそのまま移行した場合、文字コード変換をしないため、一般的なSQLによるデータ操作ができずに運用や他システムとのデータ連携に支障をきたす。最悪の場合、計画していないモダナイズ対象システム以外の改修を余儀なくされる。

<システム全体品質を確保するためのテスト準備不足>

プログラムソースコードの変換品質はベンダーが保証するものの、チューニング・リビルドを含めたシステム全体の品質確保は保証されず委託元である企業マターとなる。このため、特に結合テスト以降のテストが結果的にリビルドと同等のコストを要してしまうことになり、逆に省略し過ぎた場合は品質問題が起こり得る。

これらの問題により、結果としてプロジェクト期間の延長や追加コストが必要となるプロジェクトが増えている。

まず現行システムの周到な分析を

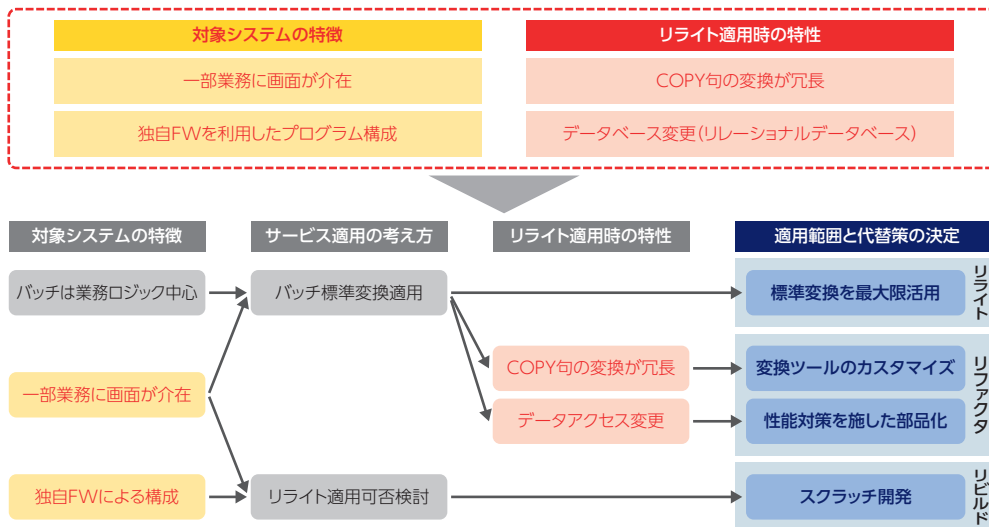
リHOST・リライト採用において重要なことは、①適用範囲を適切に選定しているか、②周辺システムを含めた全体アーキテクチャが整理できているか、③移行後システム全体品質の確保に向けたテストが計画されているか、といったことがポイントである。

これらの施策を確実に実行していくために現行システム分析が重要となるが、リHOST・リライト採用の場合は業務ロジックの調査ではなく、データフローやアプリケーション構造といったアーキテクチャ観点での分析が重要である。得られた分析結果をもとに、以下の点を考慮したプロジェクト計画を立てていく。

NOTE

- 1) OSや基盤を新しいものに置き換えアプリケーションやデータに手を加えずにシステム移行する手法。
- 2) 仕様は変えずに新しいプログラミング言語でアプリケーションを置き換えシステム移行する手法。
- 3) Data Flow Diagram (データ・フロー・ダイアグラム) の略。システムにおけるデータの流れを現した図。データの流れを表現することでシステムの機能を洗い出すことができ、主に上流工程におけるシステム設計ドキュメントの一つとして用いられる。

図表 リライト適用範囲の選定フロー（例）



(出所) 野村総合研究所

をするのか」を、費用対効果を踏まえて明確にしておく。一つの解としては、サブシステムの境界線（内部IF）や外部システムとのやり取り（外部IF）、といったポイントが挙げられる。

これらの対策は、いずれもリホスト・リライト

①適切なリホスト・リライト適用範囲の選定

事前にトライアル実施を通じてコード変換率以外の観点、具体的にはアプリケーション構造がどう変わるかの評価を行い、適用範囲を明確にする。データアクセスは当初からリビルドを前提とした性能対策を行い、COPY句の変換を共通化し変換後リソースの圧縮を図る（図表参照）。

②周辺システムを含めた全体アーキテクチャ整理

モダナイズ対象システム以外の関連システムとの全体像を把握し、データ連携の種類及びプロトコルを把握することで、外部システムとの接続性（外部システムを含めた現行機能保証）を考慮し、実現方針を検討・整理する。

③移行後システム全体品質の確保に向けたテスト計画

DFD³⁾等を用いてデータフローを整理することで、現行システムのデータアーキテクチャを把握し、変換後システムの結合テストの段階で「どことどこの現新比較

サービス提供ベンダー任せではうまくいかない。アーキテクチャ観点での現行システム分析を進め、対象となるレガシーシステムに応じた適切なプロジェクト計画を立てるためには、現行システムの保守担当者の参画を含めた自社内の体制を整えておくことが重要である。



Writer's Profile

檜 歩 Ayumi Naradate

札幌ソリューション開発三部 部長
専門はレガシーモダナイズにおける
アーキテクチャ構想・計画
focus@nri.co.jp



Writer's Profile

鳥居 和博 Kazuhiro Torii

SAシステム事業部 部長
専門は再構築やビジネス施策におけるシステム化構想・
計画支援及びプロジェクトマネジメント
focus@nri.co.jp