

基幹システムのリフォーメーション

— 最適策を導くための羅針盤 —

アブストラクト

1. 研究背景

なぜ基幹システムのリフォーメーションが脚光を浴びているのか。その背景には、企業の再編成や事業変化に伴い、「基幹システムに対する要件が多様化してきたこと」が挙げられる。また、長年維持してきたシステムが「複雑化」「運用負荷の増加」といった問題を抱えていることも要因と言える。

リフォーメーションの必要性や有意性は認識していても、最適策が見つからずに現行システムを継続利用している企業も多い。当分科会では、最適策を導くための「羅針盤」が必要と考え、研究を行った。

2. 研究アプローチ

最適策を導くには何をすべきか。まずリフォーメーションの進め方を整理した(図1)。

事例研究や当分科会での議論の結果、通常のシステム開発と同様に要件定義を確実に行うことが重要であると結論に至った。

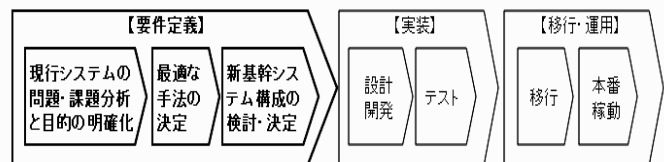


図1 リフォーメーションの進め方

当分科会では要件定義工程の以下3点を柱とし具体的に研究を進めていくことにした。

- (1) 現行システムの問題・課題を分析し「リフォーメーション目的の明確化」を行うこと
- (2) リフォーメーション目的に沿った「最適な手法の選択」を行うこと
- (3) ITインフラの最適化を目指した「基幹システム構成案の検討」を行うこと

3. 研究成果

3.1 目的の明確化を行う「現行システム課題・要件チェックシート」

様々な課題を抱えている企業にとって、目的を決めることは容易ではない。そこで現行システムの課題・企業要件を客観的に分析し、真の目的を導き出す「現行システム課題・要件チェックシート」(以下「チェックシート」)を作成した。現行システムの課題を「運用負荷・新技術・要件変化・情報活用・人・費用・複雑化」の7つに分類したチェックシート(表1)と、企業要件を「企業戦略・費用・運用サービス」の3つに分類したチェックシートに回答すると、問題・課題分析チャート(図2)に集計される。この結果を分析することで、重要な課題・要件すなわち「リフォーメーション目的」が導き出せる。

【現行システム課題チェック表】	
1. 運用負荷	
課題例: 多岐に渡るプラットフォーム、アプリケーション、先進技術などで構成されたシステムの運用負荷が増大している	
現行システムの問題点	YES=1 チェック
WEB-EDIが普及してきており、対応をせまられている	1
システムの構築が各部署単位で行われており運用が困難となっている	
業務サーバが分散配置されており、相互の補完体制が欠如している	1
権限システムの運用拠点が複数あり、相互のシステム運用、バックアップ体制が必要である	
独自開発した運用管理ツールの保守・運用負荷が増加している	
基幹システムの一部がオープン化している事で運用負荷が増加している	1
マルチベンダー環境の為、インタフェースのトラブルや異常処理など問題の特定が困難となっている	
事業所毎にマスター管理している為、同期ズレや二重管理が発生している	

表1 現行システム課題チェック表 (抜粋)

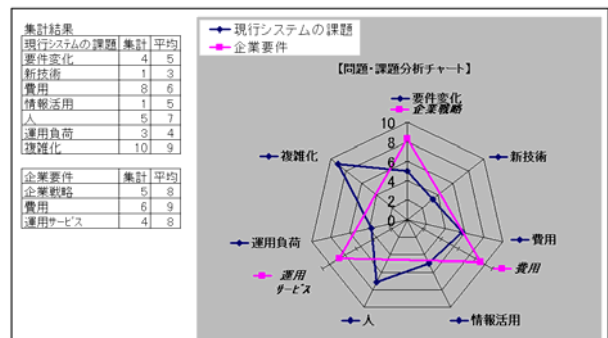


図2 問題・課題分析チャート

チェックシートは全部で10シート用意している。詳しくは活用ツールを参照されたい。

3.2 目的に沿った最適な手法の選択を補助する「手法選択チャート」

リフォーメーション目的を明確にした後、手法(リビルド、リライト、ラッピング、リホスト)の選択を行う。4つの手法には、それぞれ一長一短があり一概にどれが良いとは言いきれず、手法選択を難しくしている。そこでリフォーメーション目的に沿った最適な手法選択を補助する「手法選択チャート」を3種類作成した(図3)。

チェックシートと手法選択チャートの有効性を確認するために当分科会メンバ企業3社にて検証を行った。結果自社の課題を整理するには有効なツールである、と回答を得た。

また、チェックシートの表記に無い課題も連鎖的に書き出すことができたなどの意見も得られ、十分有効なツールという検証ができた。

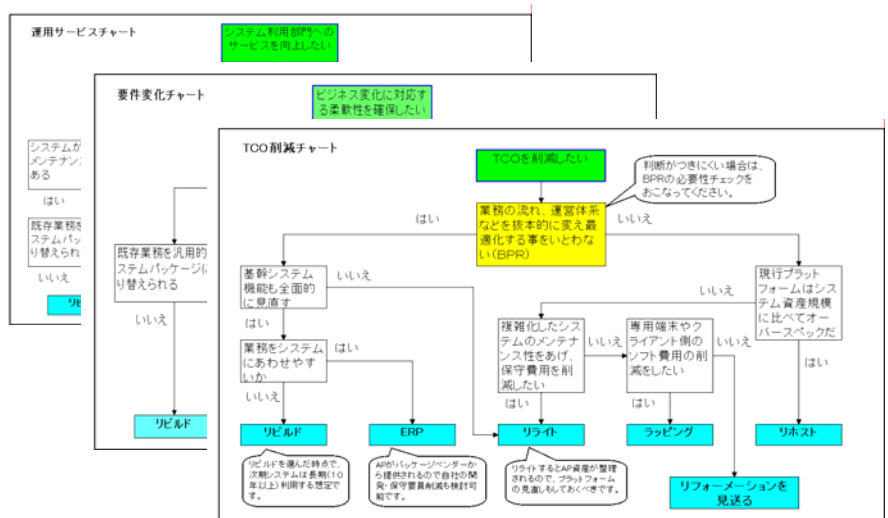


図3 運用サービスチャート・要件変化チャート・TCO削減チャート

3.3 ITインフラの最適化を目指した「新基幹システム構成案」

現行システムは少なからず部分最適の観点で構築されている。そのためインフラ面において、サービスレベルや運用ルールが統一されていないなどの問題が顕在化してきている。これらの問題を解決するためにはITインフラの最適化が必要である。最適化のポイントは、複雑さを排除してシステム統合を推進すること、業界標準の技術や製品を使用すること、標準的な開発・運用のプロセスを構築することである。

当分科会では新基幹システムの構成案(図4)を3パターン選出し、11項目で評価を行った(表2)。また、評価結果の劣る案については対応策の検討を行った(図5)。評価結果だけを見ると現時点で各要件を全て高次元で満足するシステム構成はないことが明らかになった。しかし当分科会で検討した対策を実施するか、弱点を補完しあうシステム構成案を組み合わせることで、十分満足するシステム構成にすることは可能である。

表2 評価結果

	案1	案2	案3	
サーバ	メインフレーム	メインフレーム	オープンサーバ	
DB	メインフレーム	オープンサーバ	オープンサーバ	
構成図				
評価項目	内容	案1	案2	案3
信頼性	サービス時間の保証	◎	○	△
保守性	アプリケーション保守の容易性	△	○	◎
安全性	バックアップの容易性	◎	◎	◎
機密性	セキュリティの確保	◎	○	△
親和性	企業間での情報共有の容易性	△	△	◎
継承性	アプリケーション資産の長期的な継承	◎	△	△
拡張性	柔軟なシステム拡張・縮小の容易性	◎	◎	◎
情報活用	迅速や高精度な情報提供の容易性	○	○	◎
運用コスト	システム固定費用の削減	△	△	◎
操作性	操作性の高いUIの提供	△	◎	◎
要員の確保	将来性のある技術の採用	○	△	◎

図4 システム構成案

(1) 継承性

OSやミドルウェアのバージョンアップに伴うバージョンアップ時のアプリケーション側の非互換性

① 評価

OSサポート期間・リリース間隔から見るとメインフレームにある。他のミドルウェアについてもOSと同様の傾向にある。一般的にハードウェアはソフトウェアと比較してライフサイクルが短い傾向にある(図3-12参照)。ハードウェア交換や更新等の理由でハードウェアが変更されることで、既存のソフトウェア資産に一切手を加えることなく移行できるケースは稀であり、通常はソフトウェアの改造が必要となるケースが多い。当然、動作確認のためのテストも必要となり移行コストが増加する。

② 対策

ソフトウェア資産の継承性向上策として、「アプリケーションソフトの導入」があげられる。

a. アプリケーションインターフェースの隠蔽
通常、バージョンアップ時の非互換は、OSやミド

図5 評価と対策案(抜粋)

継承性	OSサポート期間	リリース間隔
案1	10年超 +上位互換	長
案2	5年~10年 -上位非互換	2~5年
案3	5年~10年 -上位非互換	2~5年

4. まとめ

リフォーメーションを成功に導くためには、目先の問題だけに着目してはならない。全体最適の観点で真の問題を追求し、システム構成案を検討することが重要である。当分科会で作成した「チェックシート」「手法選択チャート」「構成案評価と対策案」を活用し、基幹システムのリフォーメーションへの第一歩を踏み出して頂きたい。