

# 本社基幹系システムの完全オープン化対応

## Legacy Migration of Mission-critical Systems of Headquarters

市川 公義 ICHIKAWA Kimiyoshi JFE スチール IT 改革推進部 主任部員 (課長格)  
田上 貴啓 TAGAMI Takahiro JFE システムズ 東京事業所 販生流システム開発部 営業グループ長

### 要旨

近年のオープン系システム運用技術の進展は目覚ましく、日本企業においてもシステムオープン化の動向は顕著となりつつある。これに伴うホストコンピュータメーカーの相次ぐ事業撤退やホスト系 IT 技術者供給の先細り傾向から、従来どおりホストコンピュータ上でシステムを維持管理することにリスクが生じてきた。このため、本社基幹系システムすべてをオープン環境にマイグレーションするプロジェクトを実行し、このたび完遂した。これにより DX 推進の基盤も整備された。

### Abstract:

In terms of host computer, it is said that there is a risk that sustainable operation and maintenance will not be possible in the future because of successive withdrawal of host computer makers and decreasing trend of host computer engineers. In addition, host computer has low affinity with advanced ICT technology. JFE Steel decided to migrate all the systems of headquarters from host computer environment to open environment and accomplished the project in 2021. Also, it means that digital transformation infrastructure in JFE Steel has been completed.

## 1. はじめに

経済産業省が警鐘を鳴らす『2025 年の崖』をクリアするためには、まずレガシーシステムのオープン化が大前提となる。本社基幹系システムは、システム利用者が 1 万名を超え、日々システムの改善要望も寄せられている高可用性システムである。従って、オープン化プロジェクトには、システム改造凍結期間を短縮した上で、短工期で完工することが求められる。以下に当プロジェクトにおける工夫点を紹介する。

## 2. 開発の背景

当プロジェクトの狙いは、リスク回避と DX 推進基盤整備による企業価値の向上の 2 つである。

### 2.1 ホストコンピュータ市場の縮小とメーカーの相次ぐ撤退

ホストコンピュータの国内出荷金額は 90 年台半ばには 1 兆円規模であったが、至近では 500 億円を割り込む水準が継続している。このため、大手ホストコンピュータメーカーの日立製作所、富士通が相次いで事業撤退を表明しており、従来どおりホストコンピュータでシステムを稼働させ続けることには大きなリスクを伴う状況となってきた。

また、COBOL や JCL (JOB Control Language) といったホスト系のプログラム言語は、現在では IT 技術教育の現場でも取り上げられず、今後のホスト系 IT 技術者の供給という面でも多大な不安要素を抱えている。

### 2.2 DX 推進の基盤となる IT 構造改革の断行

今後の企業の成長の一翼は DX の推進が担うことになる。しかし、最新の ICT 技術はすべてオープン系技術に準拠しており、ホストコンピュータ上のシステムやデータベースへの直接適用は困難である。DX を効率的に推進するためにはレガシーシステムのオープン化という IT 構造改革が必須となる。

## 3. 本社基幹系システムの概要と特徴

### 3.1 本社基幹系システムの概要

当プロジェクトで移行対象としたシステム資産の規模を表 1 に示す。

本社基幹系システムは、経理・人事・購買等経営管理系システムおよび販売・生産・物流というビジネスに用いる販生流系システムから構成される。

特に販生流系システムはデータの量も多く、高負荷なシステムである。

### 3.2 販生流系システムの特徴

販生流系システムの特徴は以下のとおりである。

2022 年 9 月 1 日受付

表1 移行資産の規模  
Table 1 Migration target assets

Migration target assets		
Java COBOL	Number of steps Number of steps	17 million 24 million
Data Base	Number of databases Data volume	15 824 5 TB
Files	Number of files	57 511
Report	Number of form types	7 642
Interface	Number of interfaces	7 326
CPU	MIPS	15 000

- (1) 鉄鋼業のシステムの特徴として、保有する項目数およびデータ量が極めて多いという点が挙げられる。鉄鋼業ではお客様の要求に応じて製品を作り分けるケースが多いため、一般的な契約に関わる項目のほか、品種ごとに異なる要求仕様項目を設定して製鉄所に製作指図を送付する。また、製品の荷揃以降は個々の現品単位にハンドリングする都合上、現品ごとにデータを保有している。
- (2) 品種ごとのビジネスモデルに即したシステムとなっているため、プログラム資産も巨大であり、かつ高負荷なシステムである。
- (3) データベースを中心とした DOA (Data Oriented Architecture) を採用しているため、各システム間が密結合状態となっている。
- (4) お客様からの改造要望のみならず、法改正対応や通商対応の改造頻度が高いシステムであり、プロジェクト対応中のシステム改造凍結が長期化する場合にはビジネスに多大な悪影響を及ぼす。

上記の特徴を持つ販流系システムの移行を、工期の短縮に加えて本番移行作業時間も短縮することで移行作業に伴うシステム停止時間を抑制し、ビジネスに影響を及ぼさずに実現するということが、当プロジェクトに課された最大の難題であった。

## 4. レガシーマイグレーションの手法

レガシーマイグレーションでは、ホスト環境からオープン環境へ移行した場合に発生する非互換への対応が必須となる。

### 4.1 マイグレーション手法の選択

#### 4.1.1 Java 等オープン系プログラム言語

Java などオープン系のプログラム言語で作成されている資産はそのままオープン系基盤に移植しても非互換発生は微小である。従って、オープン系資産については原則として単純移行とした。

#### 4.1.2 COBOL や JCL などホスト系プログラム言語

ホスト系プログラムのマイグレーション手法としては下記の2つが存在する。

##### (1) リホスト

オープンサーバ上に互換ソフトウェアを稼働させたいという利点はあるものの、ホスト技術者の知見が将来的にも必要となり、ホスト系 IT 技術者の枯渇という問題の解決策にはならないため、当プロジェクトでは採用しなかった。

##### (2) リライト

ホスト系プログラム言語をオープン系プログラム言語に変換する手法である。非互換の発生率も高く難度は高くなるものの、レガシーマイグレーションを行う所期の目的を達成できる唯一の手法であり、本手法を採用した。

#### 4.1.3 データの文字コード

ホストコンピュータで幅広く使用されている文字コードである EBCDIC (Extended Binary Coded Decimal Interchange Code) をオープン系文字コードの S-JIS (Shift JIS) にツールにより変換した。

表2にEBCDICからS-JISへの文字コード変換による影響を示す。主な影響として下記の点が挙げられる。

##### (1) 文字コードの値自体の変化

正しく変換されない場合、画面や帳票の文字化けに繋がる。

##### (2) 並び順の変化

EBCDIC と S-JIS とでは文字の並び順が異なる。これに

表2 文字コードの変換による影響

Table 2 Influence of character code type conversion

Character code type	EBCDIC	Shift-JIS	Possible failure
Change of code value (ex. "A")	0XC1	0X41	Garbled characters
Change of sort order	Lowercase letter ↓	Lowercase letter ↑	Output error of following JOB
	Uppercase letter ↓	Uppercase letter ↑	
	Numeric character	Numeric character	
Double-byte character control code	Exist	Not exist	Print misalignment

より文字の持つ値の大小関係も異なることになる。前のJOBの出力結果を入力データとしているJOBの場合には出力結果が現行通りにならないという不具合の原因となりうる。

### (3) 2バイト文字を制御するコードの有無

EBCDIC では漢字など2バイト文字の前後に、2バイト文字の始点と終点を示す制御コードを有するが、S-JISには制御コードは存在しない。

従って、EBCDICをS-JISに変換する際、2バイト文字の前後の制御コード分がなくなることで、帳票の印字位置ずれなどを惹起する場合がある。

これを防止するには、レコード長の現行保証が必要となるので、当プロジェクトでは半角スペース2バイト分を帳票データの後ろに付加する共通部品を開発して対応した。

## 4.2 リライトによるマイグレーションの作業工程

図1にリライトによるマイグレーションの作業工程を示す。各工程の内容は、以下のとおりである。

### ① 資源精査

システム資産の棚卸を行い、移行対象を確定させるとともに資産別の移行難易度を分析する。

### ② 変換設計

移行資産、技術要素ごとに変換方式・変換仕様を設計する。

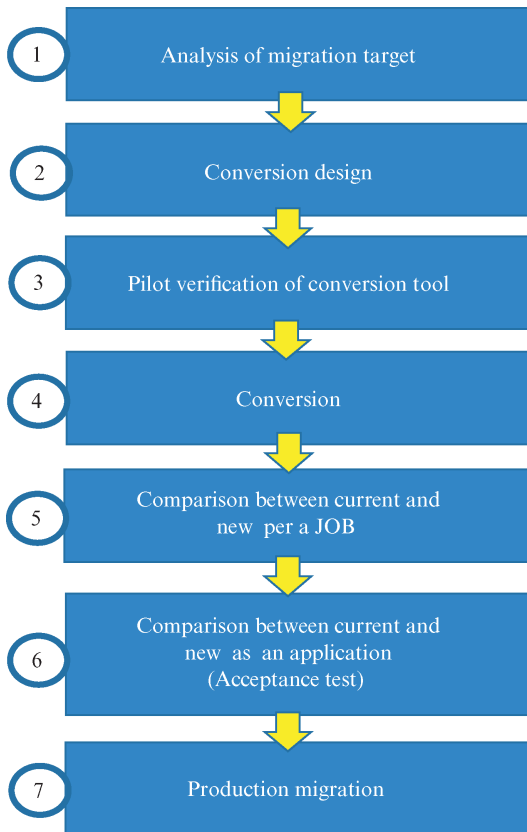


図1 レガシーマイグレーションの作業工程

Fig. 1 Process of legacy migration

### ③ パイロット検証

上記の変換設計を踏まえた自動変換ツールを作成し、意図したとおりの変換がなされていることを検証する。

### ④ 変換

自動変換ツールを活用して、大量の移行対象資産の自動プログラム言語を変換する。

### ⑤ 現新比較テスト

JOB1本ごとに変換前、変換後の出力結果が一致することを確認する。

### ⑥ 受入テスト

JOBを連結し、システム全体として稼働させた場合に変換前、変換後の出力結果が一致することを確認する。レスポンスなど非機能検証も実施する。

### ⑦ 本番移行

本番データを文字コード変換して移行し、変換後のプログラムをオープン環境に配置して稼働させる。

上記工程の中で最も工数を投入する必要があるのは、⑤現新比較テストと⑥受入テストである。これはプログラム言語の変換と文字コードの変換による非互換の発生が多く、テスト→不具合修正→テストというプロセスを繰り返すためである。

本社基幹系システムについては、先述したとおり長期のシステム開発の凍結はビジネスに多大な影響を及ぼすため、いかに短工期で完工するかという点がポイントとなる。

## 5. 短工期実現の施策

短工期での完工に向けて種々の施策を考案し、実施した。

その中から、⑤現新比較テストと⑥受入テストの工期圧縮に多大な効果のあった3つの施策と、本番移行作業に伴うシステム停止時間の抑制に寄与した膨大なデータベース群の移行に関する工夫点を紹介する。

### 5.1 テスト自動化ツールの構築による工期短縮

通常のシステムテストでは、テスト担当SEがテストデータを準備し、それを入力データとして出力結果が正しいことを確認していくまでのすべての工程がマニュアル作業である。

通常どおりのテスト方法では、短工期での完工は困難であるため、テストを自動で行うツールを構築した。

図2にテスト自動化ツールの概念図を示す。

テスト自動化ツールは、テスト管理データベースを起点としたツールで、テスト対象のJOB名、テスト実行順、JOBの入力データ名、出力データ名、現新比較に必要な情報(コンペア情報)等、自動実行に必要な情報をテスト管理データベースで管理している。

テスト自動化ツールの機能は以下のとおりである。

(1) テスト対象のJOB名、テスト実行順、JOBの入力デー

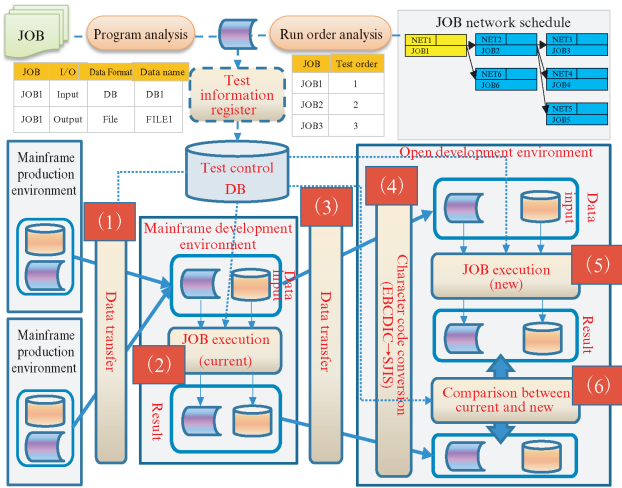


図2 テスト自動化ツール概念図

Fig. 2 Conceptual diagram of test automation tool

- タ名、出力データ名、現新比較に必要な情報（コンペア情報）等、自動実行に必要な情報をテスト管理データベースで管理している。テスト管理データベースのJOBの入力データ名に従ってテストに必要なデータを本番環境より抽出し、ホスト開発環境に移送する（テストデータの自動取得）。
- (2) テスト管理データベースのテスト実行順に従い、入力データのセットアップ→JOBの実行→実行結果データの保管を繰り返し行う。
  - (3) (2) で使用したテストデータおよび実行結果データをオープン開発環境に移送する。
  - (4) オープン開発環境にデータを取り込む際にホスト系文字コード（EBCDIC）からオープン系文字コード（SJIS）に変換する。
  - (5) (2) と同じ実行順でJOBを実行し、実行結果データを保管する。
  - (6) テスト管理データベースのコンペア情報より、(2) の実行結果データと (5) の実行結果データを12パターンの観点で比較し、比較結果、差分データを出力する（表3参照）。

このテスト自動化ツールの適用を、JOB1本ごとの現新比較テストのみならず、システム全体としての出力結果の正しさを確認する受入テストにも拡大した。これは複数のJOBを繋げたJOBネットワークをテスト管理データベースに登録し、システム全体として同様のテストを実施することで実現した。

この取組みによりテスト工数を大幅に圧縮することに成功した。

## 5.2 効率的なテスト方案の案出と実行

短工期での完工という制約のもとで、効率的に一定のシ

表3 12パターンでの現新比較テスト内容

Table 3 Current-new comparison tests for 12 patterns

		Handling of half-width space		
		Do nothing	Trim half-width space	Trim both half-width and double-byte space
Pretreatment of current-new comparison tests	As it is	Pattern1	Pattern5	Pattern9
	Mask some items	Pattern2	Pattern6	Pattern10
	Sort in order of S-JIS	Pattern3	Pattern7	Pattern11
	Sort in order of S-JIS and mask some items	Pattern4	Pattern8	Pattern12

ステム品質を確保するためのテスト方案を案出し、実行した。テスト方案策定の考え方は以下のとおりである。

- (1) 全てのシステム機能をテストすることは長工期化を招き、本社基幹系システムの特性と照らし合わせると実質的に不可能である。
- (2) テストの重複を極力排除し、業務影響の観点から重要機能を選別し、当該機能のテストは手厚く行う。
- (3) 重要機能以外の潜在的な不具合については業務影響上軽微であるため、本番稼働後、発覚した時点で速やかに対処すると割り切る。

図3にテスト方案の概念を示す。

- (1) JOB1本ごとの現新比較テストは全数実施する。この際、文字コード変換により出力結果のソート順が異なる等の不一致の現象が見られたJOBについては「条件付き一致」としJOBネットワーク一貫のテストを行い、その結果で合否を判定する。このテストは受入テストを

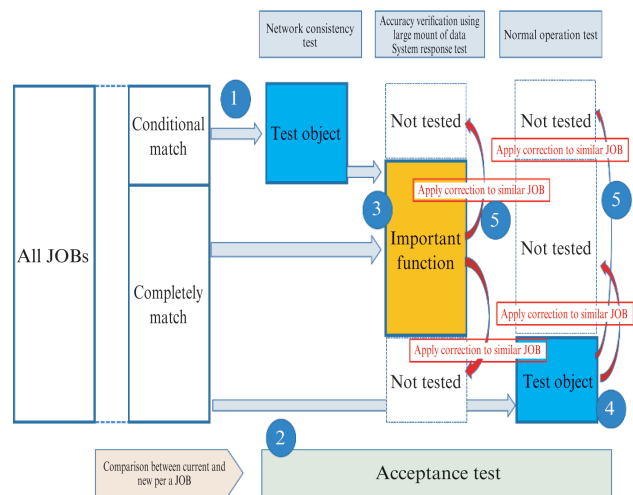


図3 効率的なテスト方案の概念図

Fig. 3 Conceptual diagram of efficient testing

兼ねて行う。

- (2) JOB1本ごとの現新比較テストがすべて合格した後、システム全体として稼働させた場合の実行非互換の有無をテストする。
- (3) 重要機能については、大量データを投入して出力結果を確認することでシステム品質を高める。
- (4) 重要機能以外の機能については、正常稼働することの確認のみ実施する。
- (5) 各テストで発覚した不具合については、当該テストの対象外としている機能の類似プログラムに対して同様の修正を適用する横展開により、テスト対象外の機能の品質も向上させる。

### 5.3 移行資産のパターン化によるテスト数削減

上述のテスト方案に加えて、レガシーマイグレーションの特徴を踏まえたテスト数の削減対策も実施した。考え方は以下のとおりである。

- (1) 通常開発と異なり、レガシーマイグレーションではビジネスロジックは修正していない。
- (2) 事前の変換設計に基づいて、ベンダーの自動変換ツールによるプログラム言語と文字コードの変換を実施した。したがって非互換発生はパターンごとに同様となる。
- (3) すべてを対象としたテストは実施せず、あらかじめ移行資産をパターン化しておき、パターンのうち代表となる少数のJOBのテストのみ実施した。
- (4) 代表少数のテストが合格の場合、当該パターンはすべて合格とした。代表で発生した非互換は当該パターンに属する他の資産においても発生するものとみなし、ただちに同様のシステム改修を実施する。

表4に通常開発における帳票とインターフェースのテスト件数と当プロジェクトで実施したテスト数を示す。

このような工夫によりテスト実施数を通常開発の半分以下とすることで短工期化を図った。

### 5.4 データベース群の移行における工夫

本番移行における最大の課題は15,000個を超えるデータベースをデータベース間の整合性を確保しつつ、いかに短時間で本番移行作業を完了するかという点である。

表4 パターン化によるテスト件数削減結果

Table 4 Results of reducing number of tests by patterning

	Number of test run	
	Normal system development	Concerned project
Report	7 642	2 540
Interface	7 326	4 204

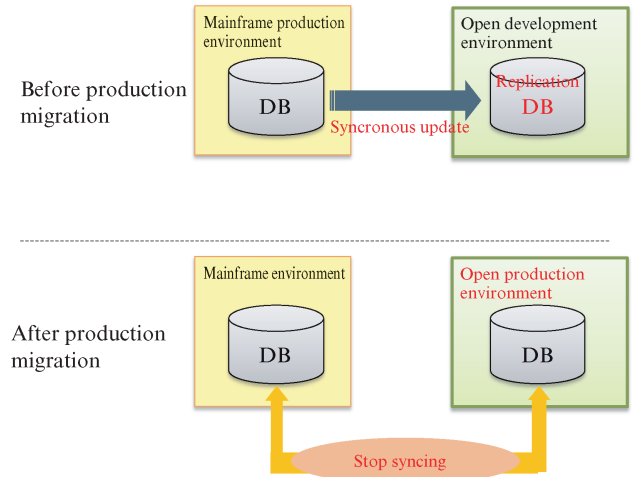


図4 データベースの移行方法

Fig. 4 Production migration of data bases

通常の本番移行の方法に従えば、データベース群の移行作業に1週間近くを必要とし、その間、移行対象システムを全停止する必要がある。1週間の全システム停止はたとえ年末年始休暇期間といえども、本社基幹系システムの上、不可である。

上述の課題解決の手法として下記の移行方法を採用した(図4参照)。

- (1) 本番移行作業当日に移行するデータベースは、更新頻度が低く、移行作業負荷の小さいものに限定する。
- (2) 更新頻度が高く、データベース移行作業の負荷が大きいデータベースについては、本番移行作業日の2週間前からホスト上の本番データベースのレプリカデータベースを移行先のオープン環境に作成する。
- (3) 本番移行作業日までの間、ホスト本番環境のデータベースが更新された場合、オープン環境上のレプリカデータベースも同期をとって更新されるようにする。
- (4) 以上の手順により、本番移行作業開始に伴うシステム停止の直前の時点において、両データベースの内容は完全に一致する。
- (5) 本番移行作業後はレプリカデータベースを正とし、ホスト上のデータベースとの同期を切る。

上記のような工夫により、本番移行作業とそれに伴うシステム停止を約52時間にとどめることに成功した。

上述の短工期化のための各種施策が奏功し、大規模システムのレガシーマイグレーションを約3年間で完工できた。

## 6. おわりに

以上の取組みにより、鉄鋼業界では先頭を切って本社基幹システムの完全オープン化を達成し、システム面での事業継続性のリスクを排除した。

本プロジェクトのもう一つの狙いである DX 推進基盤整備

についても当プロジェクトで実現したが、本当の意味で効果を上げられるかは、ひとえに今後の DX 案件の推進にかかっているのが実情である。

業務部門との更なる連携を深め、知恵を絞り、積極的な DX 案件の推進に努め、企業価値の更なる向上に努めたい。