

大容量データ移行を実現する移行戦略

Migration Strategy Leads to the Large-volume Data Transfer

岸 実

要 約 システム開発工程は一般的に、業務アプリケーションの設計・製造・テストを主体として定義される。移行については、業務の移行仕様が確定しない中で移行開発の工程を進める必要がある。

本プロジェクトでは、業務の移行仕様の変更はリハーサル期間で取り込み、本番移行作業開始に向けては、ツールや手順を変更せずに対応する方針とした。最終的に数回の移行リハーサルを経て、移行作業全体の精度の向上をはかり、本番移行作業を計画どおりに完了することができた。

Abstract Generally, system development process is based on design, implement, and tests of the business application. We need to start migration development process before we decide the migration specification of the business.

In case changes in migration specification of the business are necessary, we defined migration strategy to take them within the migration rehearsal stage. We didn't accept tool and procedure changes towards the cutover. Finally, we improved the accuracy of all the migration tasks by rehearsing many times and that led to a perfectly completed migration.

1. はじめに

システム開発工程は一般的に、業務アプリケーションの設計・製造・テストを主体として定義される。移行については、業務の移行仕様が確定しない中で移行開発の工程を進める必要がある。日本郵便株式会社（以降、日本郵便）の取引管理システム（後続開発）プロジェクト（以降、本プロジェクト）は、2013年4月に本番稼働した取引管理システム（先行開発）が24時間稼働している本番環境への移行が前提であった。移行対象となる業務データは総量約3.7TB、約200テーブル、約20億件であり、本番稼働前日から新システムで稼働する当日までの移行作業（以降、当日移行とする）を、システム停止時間約6時間で確実に完了させることが命題であった。この要件を実現するために、①確実に移行作業を完了するための移行期間（時間）を早期に見極めること、②移行作業の品質を確保すること、の2点を重要成功要因として移行関連作業を進めることとした。

本稿では、2章にて移行関連作業スケジュールと検討プロセス概要、3章にて移行計画の策定、4章にて移行開発における品質確保の考え方、5章にて移行リハーサルにおける品質向上と移行結果について述べる。

2. 移行関連作業スケジュールと検討プロセス概要

本プロジェクト開始時点で策定した本番移行直前までの作業スケジュールを図1に示す。

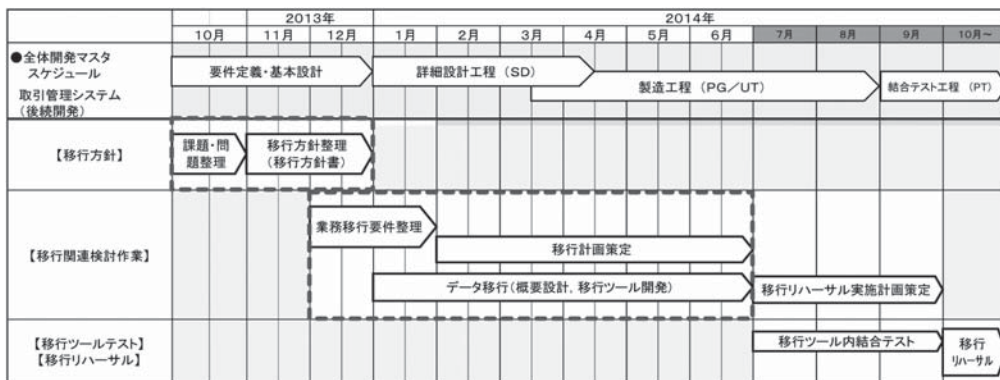


図1 移行関連作業スケジュール

まず、移行関連作業の検討を進めるにあたって、検討対象を『業務移行要件』、『移行計画策定』、『データ移行』の三つに大別した。また、『移行計画策定』と『データ移行』については、図2に示すプロセスを定義して検討を進めた。

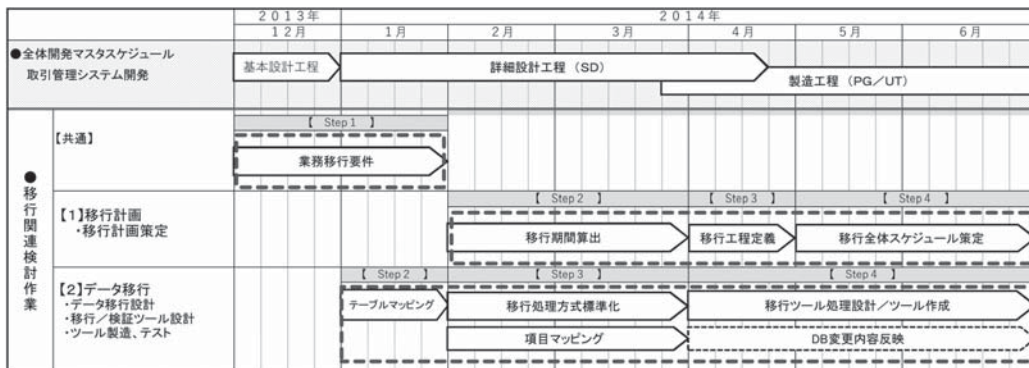


図2 移行検討プロセス

『業務移行要件』では、以降の移行計画策定の検討プロセスにおける移行期間の算出やデータ移行の設計に必要な要件を中心に抽出して整理する。この検討の実施時期は業務アプリケーション開発の設計工程であり、業務の移行仕様は必ずしも確定しないため、移行対象データやその範囲など、大幅な変更を伴わない大枠の要件を一旦整理して進めた。

3. 移行計画の策定

本章では、2章で挙げた移行計画策定の検討プロセスの各ステップの内容を述べる。

3.1 移行期間の算出

業務移行要件の整理で明確化した内容に基づき、本番移行に向けた移行準備作業期間、本番稼働前日までに実施する移行作業の期間、当日移行期間（時間帯）を算出する。尚、データ抽出ミスやデータ移送障害など、何らかの要因で移行作業の再実施が必要となった場合の、戻し作業や作業のやり直しの時間、期間も想定しておく。

3.2 移行工程定義

本番移行の工程定義は、3.1節で算出した期間に基づき、以下の四つの工程で実施する作業を明確化する。

- 1) 先行作業：本番移行作業を開始するにあたっての準備作業
- 2) 事前移行：本番稼働前日までに実施する移行作業
- 3) 当日移行：本番稼働前日から当日までの業務停止時間帯に実施する移行作業
- 4) 事後移行：本番稼働当日以降に実施する移行作業

3.2.1 先行作業

先行作業は、移行作業に必要な各種環境構築作業をさす。移行作業用データベース環境の構築や、移行ツール自体の導入及び実行環境設定、ジョブ登録などを実施する工程である。移行処理にてネットワークを使用してデータを移送する場合は、ネットワークの疎通確認作業も実施する。本番業務が稼働している本番システム環境での作業は、本番業務への影響を考慮して作業実施期間や作業時間帯を制約する。業務の繁忙期やオンライン処理のピーク時間帯、バッチ処理ピーク時間帯などの業務特性を踏まえて作業期間や時間帯を設定する。

3.2.2 事前移行

事前移行は、本番稼働前日までに実施する移行作業をさす。データ移行処理の初回データ移行や差分データ移行、マスタデータ構築作業、業務アプリケーションの事前のリリース、アプリケーション機能の一部を使用したマスタ登録や利用者権限登録などを実施する工程である。事前移行作業は、本番業務への影響を考慮して、本番システム環境上での作業と、別環境（検証環境）での作業を分類し、極力別環境で実施する。

3.2.3 当日移行

当日移行作業は、本番稼働前日の業務停止から当日新システムで業務開始するまでのシステム停止時間帯に実施する作業をさす。本番環境上でのインフラ環境の切替作業や差分データの最終移行処理、データベース切替処理、連携先他システムとの切替や疎通確認、切替作業完了後の業務稼働確認作業などを実施する。

当日移行作業については、作業タイムチャート（時間単位の作業詳細スケジュール）を作成し、進捗チェックポイントの設定や稼働判定（新システムでの業務稼働承認）ポイントなどを設定する。

3.2.4 事後移行

事後移行作業は、新システム稼働後に実施する移行作業をさす。本プロジェクトにおいては本番稼働の前月分業務締め処理までを現行システムで実施する前提であったため、本番稼働後の現行システムとの平行稼働期間においても、現行システムで更新されたデータの移行処理（差分）や他システムとの連携、切替作業などを実施した。

尚、新システムで稼働後に業務システムにて障害が発生した際の解析に用いる、移行済みデータの保管期間や、移行作業で使用した移行ツールや一時的に使用したデータベース・テーブルなどの削除作業についても、移行後の作業として検討しておく。

3.3 本番移行全体スケジュール

3.1節及び3.2節に基づき本番移行全体のスケジュールを作成する。作成にあたっては、以下の点も考慮に入れた。

1) 本番移行作業を開始するタイミング

業務アプリケーションのテスト工程における不具合修正等に伴い、データ移行の仕様変更（データ項目編集仕様や切替処理など）が発生する可能性が高いため、業務アプリケーション開発のシステムテスト工程完了以降とする。

2) 再実施可能な予備期間の設定

移行の先行作業工程の本番システム環境上で実施する作業や、事前移行工程のデータ移行作業（移行元システム側でのデータ抽出からデータの移送、移行データ作成まで）は、やり直しが発生すると後続作業への影響が大きいため、予備期間を設ける。

本プロジェクトにて策定したスケジュールを図3に示す。事前移行が必要な移行対象データの容量、テーブル数、毎月のデータ更新の静止点や月次締め処理のタイミングなどを基に、業務アプリケーションの統合テスト期間となる本番稼働当日の3.5ヶ月前から移行の先行作業を開始し、事前移行期間を3ヶ月間として当日移行を迎えるスケジュールとした。

		2015年								
		6月	7月	8月	9月	10月				
イベント		夏期繁忙期 (6/20~8/20)								
システム開発 全体工程		統合テスト		運用テスト				本番稼働		
移行全体 スケジュール	本番移行 (4ヶ月)									
	先行 作業	予備期間	事前移行 (3ヶ月)					当日 移行	事後移行	
			データ移行 (初回)	予備期間 (データ移行初回)	データ移行 (2回目)	予備期間 (2回目)	データ移行 (3回目以降)			先出期
						過渡期	切替			

図3 本番移行全体スケジュール

4. 移行開発における品質確保について

本章では本プロジェクトにおける重要成功要因の一つとした、移行作業の品質について、移行開発工程での品質確保の考え方と、移行処理標準化の内容を基に述べる。

4.1 移行開発工程の品質確保の考え方

移行開発では、移行ツール、移行データ、移行手順の品質確保の考え方を整理した。表1に内容を示す。移行ツールについては、設計工程での設計書レビュー、リハーサルでの本番データを使用した検証により品質を確保する。

データ移行処理にて作成した移行データについては、データの検証範囲と検証内容を設計工程で明確化したのち、移行データ検証ツールを開発し、移行結合テスト工程からのデータ正当性を検証する。移行データの検証範囲や内容は、例えばデータ移行処理で編集仕様に基づき設定した項目の親子関係や、条件に基づく値の計算結果を設定した項目などの業務観点でのビジネスルールに従った検証などである。データ移行処理で作成した移行データは、最終的には業務アプリケーションのテストで使用することにより品質を確保する。

移行手順については、設計工程で移行作業のフローのレビュー、テスト工程以降はテストやリハーサルにて移行手順を検証し品質を確保する。

表1 移行開発工程における品質確保の考え方

	設計工程	製造工程	結合テスト工程	移行リハーサル
ツール	移行設計書の顧客レビュー	手積みデータで検証	手積みデータで検証	本番データで検証
移行データ	検証範囲/内容の顧客及び業務有識者レビュー	— (検証用ツール製造)	・検証ツールでの検証 ・業務 AP 結合テストで検証	・検証ツールでの検証 ・業務 AP システムテストで検証
移行手順	移行作業フローの顧客レビュー	— (手順書作成)	結合テストで検証	リハーサルにて検証

4.2 移行処理標準化

移行処理標準化は、以下のステップにて検討を進める。

- 1) 移行処理環境の決定
- 2) 一括移行方式か差分移行方式かの決定
- 3) 処理方式のタイプ別の分類
- 4) 本番環境へのデータ投入方法の決定
- 5) 処理のグルーピングと処理順序の決定

上記2)の一括移行方式とは、データ更新の静止点が明確で全量を一回で移行可能なデータが対象となり、マスタデータなどが該当する。また、差分移行方式の対象は、定常的にデータが更新されるトランザクションデータなどが対象となり、ある時点での全量データを初回に移行し、その後更新分のデータ（差分データ）を移行する。

また、上記5)の処理のグルーピングと処理順序とは、例えばあるマスタデータの値を参照してトランザクションデータを編集する場合などに、マスタデータを最初に移行しその後トランザクションデータを移行する手順を一つの処理単位として定義することである。

本節では顧客の業務特性やシステム特性によって決定手順が異なる2)、4)、5)は割愛し、1)移行処理環境の決定と、3)処理方式タイプ別分類について記載する。

4.2.1 移行処理環境の決定

移行対象データが大容量で複数回にわたって移行処理を実施する場合（初回移行+差分移行など）や、複雑な移行編集・加工処理を必要とする場合は、業務システムが稼働している本番環境で処理を実行すると、システムリソースを圧迫し本番業務に影響する懸念がある。このため、本番環境での実行は必要最小限にとどめ、別の環境にて移行データを事前構築し、本番環境へは構築した移行データを投入するのみとすることが望ましい。

本プロジェクトでは、3章でも述べたとおり、本番稼働前日までに事前に実施可能な移行作業は別環境にて実施する方針とし、事前移行対象としたデータの移行処理は図4の①に示す本番環境とは別の環境（検証環境）を経由して移行データを構築する方式を標準方式とした。また、何らかの原因で移行処理の再実行が必要になった場合でも、作業調整や再実行日時の制約などが少ないことも、別環境を使用するメリットである。

検証環境で構築した移行データの本番環境への投入・反映処理は、本番業務の業務処理量が

少ない曜日や時間帯（夜間帯など）に実施する計画とした。

本番稼働直前（前月静止点以降）の日次の差分データ反映処理や当日移行の最終の差分データの反映処理などは、差分データのデータ容量が少ないため、検証環境を経由するよりもデータ移行作業が短時間になることから、本番環境にて直接データベースへ反映する方式とした（図4の②）。

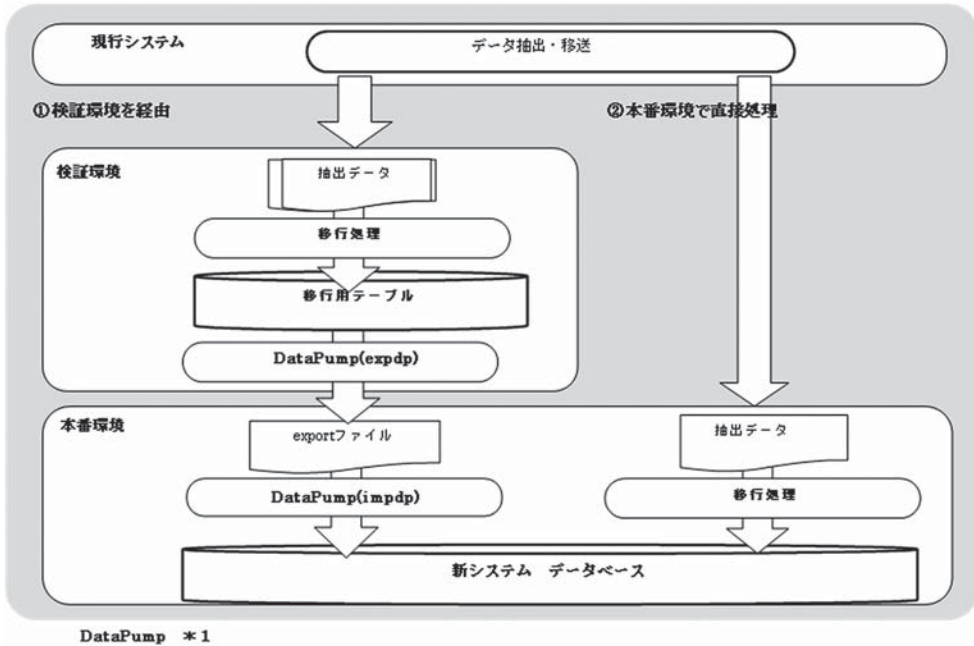


図4 移行処理環境

4.2.2 移行処理方式タイプ別分類

移行対象データの移行仕様に応じて、処理方式を以下に示すA～Dの4タイプに分類する。基本的に移行ツールの開発においては、独自開発する移行プログラムの範囲を最小限にし、データベース製品で提供されているユーティリティを極力使用してツール化（自動化）する前提とした。

- ・タイプA：移行元データを直接移行先のテーブルへ格納
- ・タイプB：移行元データを加工・編集して移行先テーブルへ格納
- ・タイプC：移行元データを人手で加工・編集して移行先テーブルへ格納
- ・タイプD：業務システムの利用者がアプリケーション機能にて移行先テーブルへ設定

また、処理方式の分類にあたっては、分類指針を定めて実施した。四つのタイプの処理概要を図5に、分類指針を表2に示す。タイプBに分類される複雑な編集仕様を伴う対象データのみ、独自の移行プログラムを開発することとした。また、移行プログラムの設計・製造においても設計規約やコーディング規約を作成し標準化を図った。

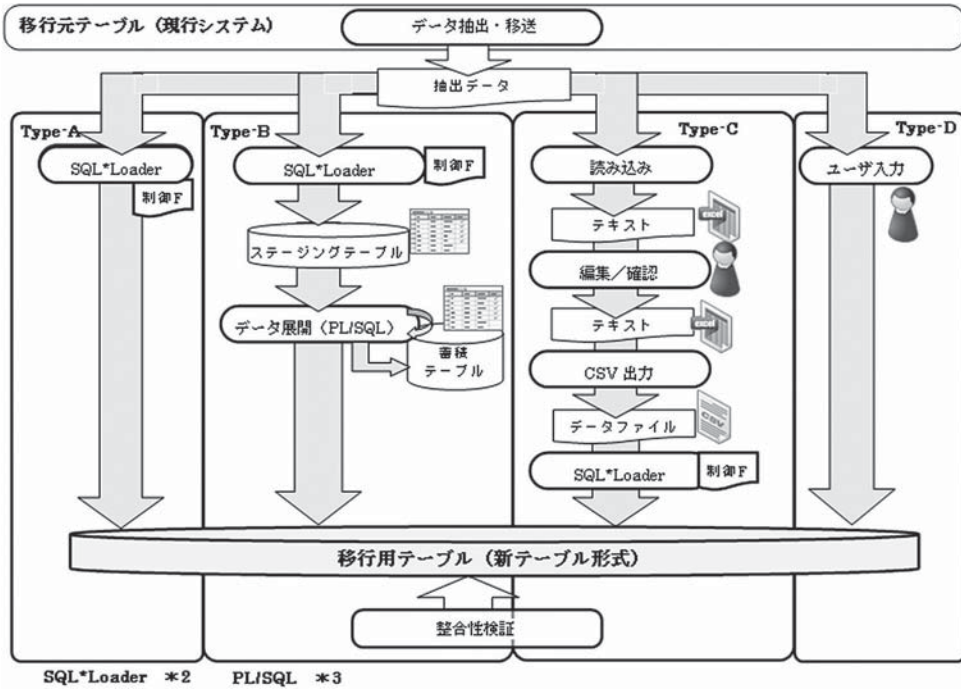


図5 移行処理方式のタイプ別分類

表2 処理方式タイプの分類指針

移行 Type	処理方式	分類指針
Type-A	製品ツール (SQL*Loader)	下記の3条件を全て満たす場合 ①抽出データの1レコードが移行先テーブルと1対1 ②抽出データの編集が不要、または簡単な編集 ③差分移行が不要、または差分はレコード追加のみ
Type-B	・製品ツール (SQL*Loader) ・移行プログラム (PL/SQL)	①抽出データの1レコードが移行先の親子テーブルに分割 ②複数の異なるフォーマットの抽出元から1テーブルに統合 ③差分移行対象で更新/削除等の判定が必要
Type-C	製品ツール (SQL*Loader)	①初期値設定 ②抽出データに初期値設定する
Type-D	ユーザ入力	業務アプリケーションを利用してユーザがデータ登録する

5. 移行リハーサルでの品質向上と移行結果

本章では、移行リハーサルにおける移行作業の品質向上と移行結果について述べる。

5.1 移行リハーサルでの品質向上の考え方

移行リハーサルの計画段階では、リハーサルの目的・目標、リハーサル回数及び検出すべき問題を設定する。移行リハーサルでは、開発環境での各種テストでは検証が難しい環境的な問題や時限性（性能）、想定外データ・異常データの検出、移行手順、人の動きなど、本番移行作業を想定して実施し検証する。

本プロジェクトには、日本郵便の次世代郵便情報システム（後続稼働）プロジェクトの全体計画として、表3に示す三つの移行リハーサルが設定されていた。

一つめの自システム内の移行リハーサルは、環境不備や移行ツールの不具合や仕様変更による修正が発生する可能性が高いため、リハーサル期間内で2回実施する計画とした。

二つめの他システムとの連携を含めた全体リハーサル①においても同様に、他システムとの仕様齟齬や連携処理における問題の対応後の再検証が必要と考え、リハーサル期間内で2回実施する計画とした。

最終の全体移行リハーサル②については、本番移行作業と同等の作業タイムスケジュールに準じて全ての移行作業を実日数、実時間で実施する計画とした（一部、データ内容の違いのみで繰り返しの作業を除く）。

表3 移行リハーサル計画

リハーサル	目的・目標	回数/期間	検出すべき問題
自システム内	移行ツールと移行作業タイムスケジュールの確認及び手順の確立	2回/2ヶ月 (各1ヶ月)	環境不備、ツール・抽出データ不正、手順不正・考慮不足、処理時間想定誤り、想定外データ・異常データ検出
全体リハーサル①	移行元システムや連携先システムを含めたデータ移行、システム切替作業のタイムスケジュールの確認及び移行手順の検証、作業の習熟	2回/3ヶ月 (各1.5ヶ月)	移行元システム・連携先システムとの手順不正、調整不足項目、連携処理時間誤り、移行作業体制・役割不備
全体リハーサル②	本番移行のタイムスケジュールに沿った移行作業全体の手順最終確認及び作業の習熟	1回/2ヶ月 (本番移行作業の実日数、実時間)	チェックポイント過不足と内容不備、連絡ルート誤り、各種本番移行運営不備

また、表4に示すとおり、各リハーサルは、移行タイムスケジュールの妥当性・整合性、移行手順の正当性、移行作業時間の妥当性（人間系の確認作業や証跡採取などの時間も含む）、移行作業に関連する人の動きを評価の観点として実施した。

5.2 移行リハーサルで検知した問題への対応

移行リハーサルでも、想定していない様々な問題が発生しうる。データ移行の仕様誤りや想定外のデータによる編集処理の障害等が発生した場合の対応として重要なのは、これまでテストやリハーサルを重ねて品質を向上してきた移行ツールや移行手順を極力変更（改修）しないことである。これまで確立した移行処理の前後へのデータバッチ処理の追加や、必要となった処理のみのツールを追加作成して、問題のあったデータのみに個別対応することにより、既存の移行ツールや移行手順の変更による、移行作業全体の品質低下を防止する。

5.3 移行結果

本プロジェクトが大規模短期開発であったことから、初回のシステム内の移行リハーサルでは多くの不具合や仕様不備が発生した。しかし、期間的に厳しいながらもシステム内及び全体①の各リハーサルを複数回に分割して計画し、回数を重ねることで、移行ツールの品質や移行手順、問題発生時の対応を含めた人の動きなど、移行作業全体の精度の向上が図れ、本番移行作業を計画どおりに完了することができた。

表 4 移行リハーサルにおける品質確保に向けた評価の観点

工 程	評価項目	評価の観点 (内容)
移行リハーサル (自システム内) ※本番データ使用	移行手順	移行手順, 切替手順, 障害発生時の手順に問題がない
	ツール	・データ移行ツール, データ検証ツールが仕様どおり動作する ・処理グループの考え方, 処理順序に問題がない
	処理時間/作業時間	各ツールの処理時間と作業時間が想定した見積り時間内である
	移行データ	想定外データ, 異常データの検出
	業務アプリケーション	ビジネスルールに従ったデータである (システムテストで移行データを使用し検証)
全体移行リハーサル① ※本番データ使用 ※他システムとの連携	移行手順	・移行手順, 切替手順, 障害発生時の手順が問題ない ・移行作業者の移行手順の習熟度 (作業手順ミスがない)
	ツール	・データ移行ツール, データ検証ツールが仕様どおり動作する ・処理グループの考え方, 処理順序に問題がない
	処理時間/作業時間	ツールの処理時間と作業時間が移行作業タイムスケジュールに設定した時間内である
	移行データ	・想定外データ, 異常データが検出されていない ・検出した不正データの処理が明確になっている
	業務アプリケーション	ビジネスルールに従ったデータである (統合テストで移行データを使用し検証)
	他システム	・移行元の現行システムとのデータ連携方式・タイミング・処理時間が問題ない ・他システムとの連携タイミング・連携データ・処理時間が問題ない
	移行処理フロー	他システムとの連携含むデータ移行, システム切替の移行処理フロー全体に問題がない
	移行体制/役割	移行作業体制や役割に不備・不足がない
全体移行リハーサル② ※本番データ使用 ※他システムとの連携 ※移行運営の最終確認	移行手順	次世代プロジェクト全体の移行手順, 切替手順, 障害時手順等各種手順の最終確認
	ツール	これまで発生したツール障害が全て対応されていること最終確認
	移行作業時間	移行作業タイムスケジュールに則った移行全体の作業時間の最終確認
	移行データ	・想定外データ, 異常データが新たに検出されていない ・検出した不正データを正しく処理していること最終確認
	移行処理フロー	データ移行, システム切替の移行作業全体の移行処理フローの最終確認
	移行体制/役割	移行作業体制や役割に不備・不足がないこと最終確認
	本番移行運営	本番移行時を想定した関連各所との連絡含む本番移行運営が問題ないこと

6. お わ り に

プロジェクト開始から移行リハーサル工程の開始まで12ヶ月の短期開発であり, 移行対象のデータは大容量のトランザクションデータという, 難易度の高いプロジェクトであった. 本稿にて紹介した移行方針や品質確保の各種施策が, 今後のプロジェクトにおいて参考になれば幸いである.

プロジェクト遂行にあたり, 現行システムベンダや連携先他システム, その他関連各所との調整に多大なるご支援をいただいた日本郵便株式会社のシステム主管及び関係各位に本稿にて感謝申し上げます.

- * 1 DataPump : Oracle 社が提供する Oracle データベース・ユーティリティ
- * 2 SQL*Loader : Oracle 社が提供する Oracle データベース・ユーティリティ
- * 3 PL/SQL : データベース言語 SQL を手続き型にしたプログラミング言語

執筆者紹介 岸 実 (Minoru Kishi)

1990年日本ユニシス(株)入社。電力会社の汎用機・オープン系のシステムサービス、業務システム開発、公共システム部門のエネルギー関連サービスの企画・開発・保守を担当。2013年より日本郵便株式会社の取引管理後続開発プロジェクトに参画。

