
# 人事給与システムの再構築事例に基づく
# メインフレーム・マイグレーションの注意点

JR四国情報システム株式会社
情報システム部
田中 一人




# 目次

1. 事例概要の説明
　1　企業紹介
　　2　既存システム概要
　　3　プロジェクトの背景と目標
　　4　体制と役割分担
2. 資産移行の詳細
　　1　移行前後のシステム構成
　　2　変換要件と仕様
　　3　苦労した点と解決策
3. プロジェクトを振返って
4. 最後に


# 1－1.事例概要：企業紹介

四国旅客鉄道株式会社
Shikoku Railway Company

- 社名　四国旅客鉄道株式会社
- 設立　昭和62年4月1日
- 資本金　３５億円
- 社員数　２,７５６人（平成２１年８月現在）
- 事業内容　旅客鉄道事業、旅行業、その他の関連事業

JRSIS JR四国情報システム株式会社

- 社名　JR四国情報システム株式会社
- 設立　平成16年4月1日
- 資本金　１,０００万円
- 株主　四国旅客鉄道株式会社（１００%出資）
- 事業内容　JR四国基幹システム運用・保守、JR四国グループ会社の
  システム開発、ハード、パッケージ販売、バーチャルモール「夢四国」

  の運営管理、等

# 1－2.事例概要：既存システム概要

★ プラットフォーム　：メインフレーム（ACOS4）

★ システム　：人事給与システム

★ プログラム　：約　800本

★ JCL　：約　500本

★ ファイル　：約　250本【RIQS(RDB)、VSAS(VSAM)、DIR、SEQ】

★ 帳票　：約　500フォーム
電子帳票　ReportViewer

# 1－3. 事例概要：プロジェクトの背景と目標

## ① プロジェクト検討の理由

### 課題

- メインフレーム系システムからオープン系インフラ基盤への刷新
- オープン系のデファクトスタンダードな技術の採用、開発環境の導入
- 既存資産の有効活用と開発期間の短縮

・Open化に移行したい
・独自の人事
　給与体系・・・
・開発期間は・・・

**人事給与システム**

・既存資産の有効活用
・移行リスク＆コストの最小化
・短期間での移行が可能
・Open連携容易

### 解決策

**COBOLマイグレーション　＋　一部リライト**

# 1－3. 事例概要：プロジェクトの背景と目標

## ② 検討した選択肢

### 手法比較

| | システム機能 | コスト | 期間 | 移行リスク | 保守メンテ | 既存資産有効活用 | 拡張性 |
|---|---|---|---|---|---|---|---|
| 再構築 | ◎ | × | × | △ | ◎ | × | ◎ |
| パッケージ | ○ | ○ | △ | ○ | △ | × | △ |
| マイグレーション＋リライト | ○ | ◎ | ◎ | ○ | ○ | ◎ | ○ |
| メインフレーム継続 | ○ | ◎ | ◎ | ◎ | × | ◎ | × |

### 比較のポイント

- 既存資産の有効活用が可能であること
- 開発期間と予算が上限内に収まること
- 開発環境やデバッグ環境が整備されていること
- Web化等オープン連携が容易であること
- システム拡張性に優れていること

# 1－3. 事例概要：プロジェクトの背景と目標

## ③ 移行にあたっての目標

既存システムが持つ高い運用性と信頼性に加え拡張性を実現する。

- **Oracleへの移行要件**
  既存のロジック、レイアウトを変更無しでOracleへ移行すること。
  将来のデータ活用の基盤を構築すること。
- **画面機能要件**
  現行オペレーション及びレスポンスを踏襲すること
- **帳票機能要件**
  現行Report Viewerを利用することで、現行の操作性、運用性を継承する。
- **その他要件**
  ファイル転送、ジョブネットワーク、ジョブ監視等の精度を高め信頼性、運用効率を向上させる。

# 1－4. 事例概要：体制と役割分担

## ① プロジェクト体制図　及び　各担当の役割・連携

JR四国

JR四国情報システム（JRSIS）

全体統括、総合試験実施

NEC

NECシステムテクノロジー

システム総合サポート

日本コンピューター・システム

調査分析～結合試験実施及び
総合試験サポート

東京システムハウス

変換サービス・サポート

# 1－4. 事例概要：体制と役割分担

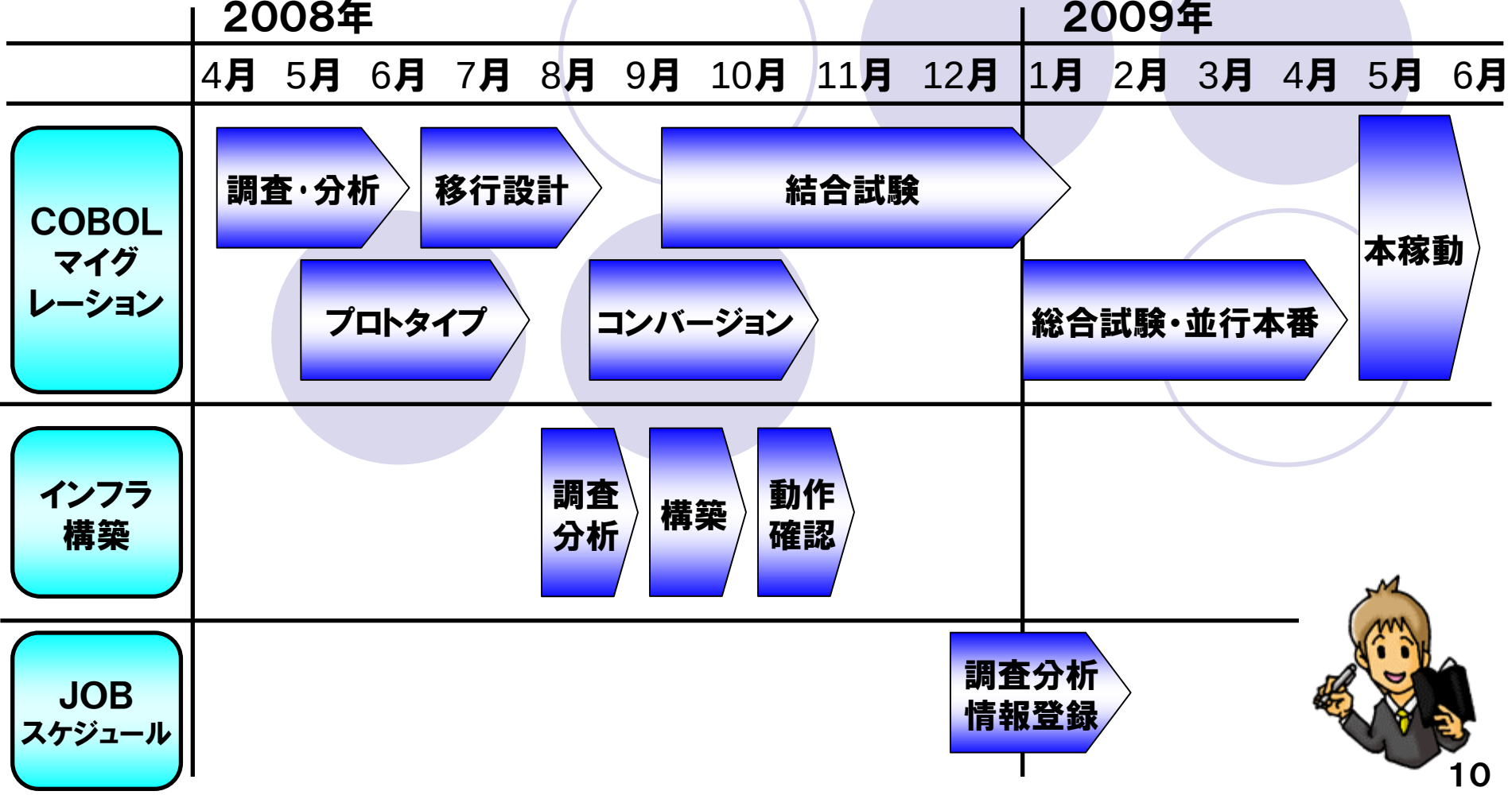
## ② 体制作りでの注意点

**POINT ： プロジェクト内の情報共有**

- 打合せ前の事前テーマ通知
- 議事録の配布
- 定期的な進捗報告と進捗会議、各フェーズの終了をプロジェクトで確認
- 情報交換票による連絡（書面による連絡）
- メーリングリストの活用（関係者に漏れなく連絡）

# 1－4. 事例概要：体制と役割分担

## ③ スケジュール



# 2－1. 移行資産の詳細　移行前後のシステム構成

## ① インフラ比較

### 移行前

- ACOS4
- CGMT装置
- ディスク装置
- センターコンソール
- プリントサーバー
- ファイル転送サーバー

### 移行後

- 人給本番サーバー（AP・Webサーバ）
- DBサーバー
- ストレージディスク
- 運用管理サーバー
- 人給開発サーバー（バックアップサーバ）
- プリントサーバー
- ファイル転送サーバー




# 2－1. 移行資産の詳細　移行前後のシステム構成

## ② ミドルウエア・アプリケーション比較

### 移行前

**OS ： ACOS4**

| | |
|---|---|
| 運用管理 | FIPS |
| ファイルシステム | RIQS(RDB)、VSAS(VSAM)、DIR、SEQ |
| 言語 | COBOL ・ COBOL／S |
| 帳票 | FORM－EX ・ ReportViewer |
| ファイル転送 | FTP |

### 移行後

**OS ： Windows2003**

| | |
|---|---|
| 運用管理 | Senju |
| ファイルシステム | Oracle ・ SEQ |
| 言語 | MFCOBOL |
| 帳票 | ReportViewer |
| ファイル転送 | HULFT |

# 2－2. 移行資産の詳細　変換要件と仕様

① COBOL資産移行に関する要件
以下に示す機能がメインフレームと同等であること

- ■ COBOL標準部品やユーザマクロ
- ■ 世代管理ファイル
- ■ VSAS(VSAM)ファイル
- ■ CGMTでのバックアップ運用
- ■ FORM使用帳票
- ■ JCLでのジョブネット
- ■ 画面運用

# 2－2. 移行資産の詳細　変換要件と仕様

## ②　要件を吸収して移行を行うための仕様設計

| | |
|---|---|
| ■ COBOL標準部品やユーザマクロ | ・ TSH社の蓄積資産で既存のものは活用、不足分は新規開発<br>・ ファイルリネーム方式の既存代替処理 |
| ■ 世代管理ファイル | |
| ■ VSAS（VSAM）ファイル | ・ Oracleへ移行 |
| ■ CGMTでのバックアップ運用 | |
| ■ FORM使用帳票 | ・ ReportViewerに全て移行 |
| ■ JCLでのジョブネット | ・ AJTOOL Batch Framework・Senjuにて定義 |
| ■ 画面運用 | ・ Weblogic・Tuxedoにて実現 |

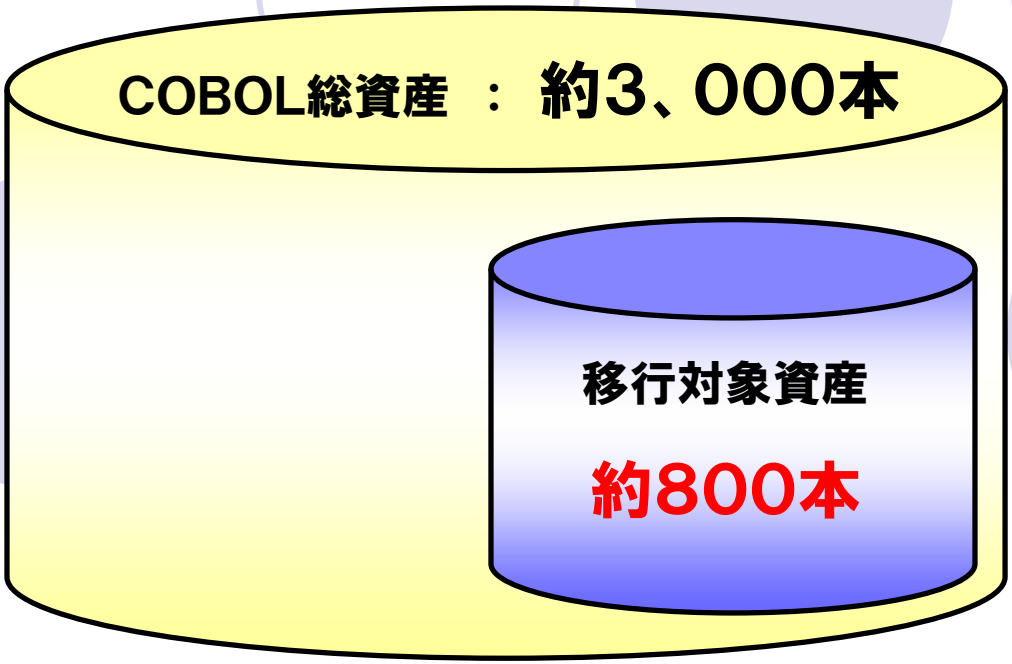
# 2－3. 移行資産の詳細　苦労した点と解決策

## ①　現行資産調査

### ①－1　移行資産の絞込み

実行時ログ調査


ユーザヒアリング



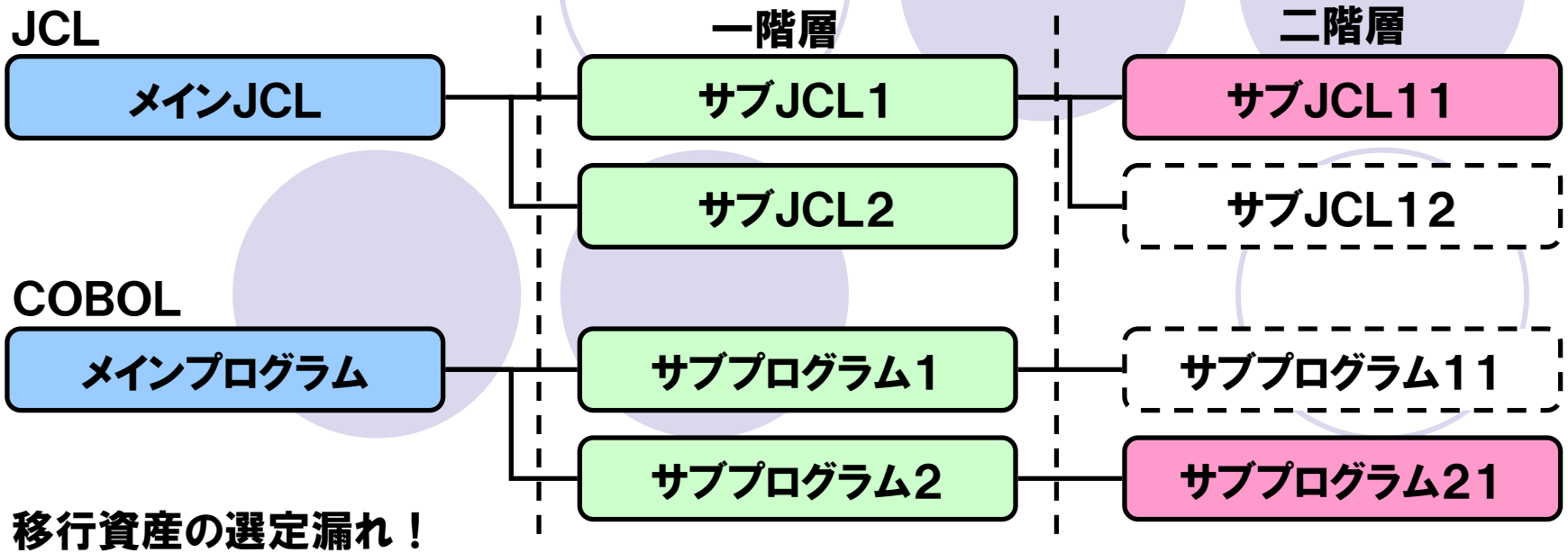

無駄な資産は移行対象外！

- 移行資産の洗い出し　：　過去実行ログ調査・ユーザヒアリング
  無駄な資産の移行は工数・期間の無駄使い

# 2－3. 移行資産の詳細　苦労した点と解決策

① 現行資産調査

①－2 稼動資産調査（サブプログラム・階層JCLの洗い出し）



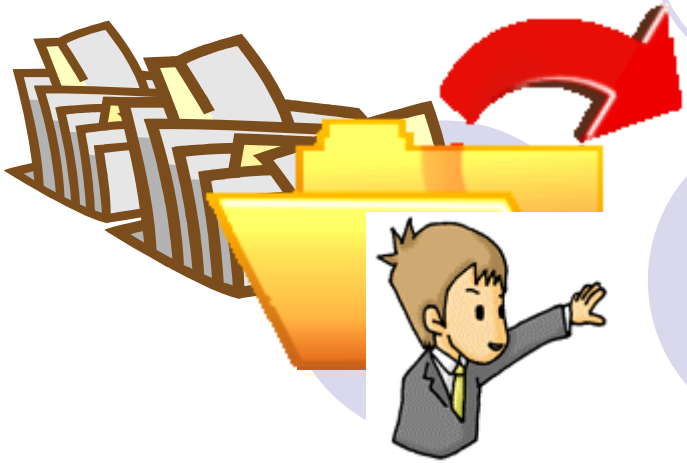

移行資産の選定漏れ！
調査分析段階での洗い出しが必須！
後工程で発見されると、思わぬ工数と時間が発生する。

- 全稼動資産の洗い出し ： ツール（エディタ、ExcelVBA）を利用した調査を実施し、洗い出し漏れを無くす

# 2－3. 移行資産の詳細　苦労した点と解決策

① 現行資産調査

①－3 実装されているメインフレーム ＋ 当社独自機能の確認のため
プロトタイプ選定



Batch系

Online系

プロトタイプ実行

プロトタイピング選定に際しては、システム上重要な箇所・技術的なポイントを意識し選定。

Batch系　：　正確性調査のため計算ロジックを含むものを選定
Online系　：　処理速度調査のため照会系を選定

- プロトタイプ選定　：　今後の移行の指針となるため選定は慎重に行う

# 2－3. 移行資産の詳細　苦労した点と解決策

## ①　現行資産調査

### ①－4　実装されているメインフレーム　＋　当社独自機能の確認

メインフレームのリソースには、本来開発者が意識すべきことをOSが代替してくれるようなものがある。
長い開発期間を経るうちにそれらが標準となってしまったシステムもある。


規則を無視した場合でもOSがカバー

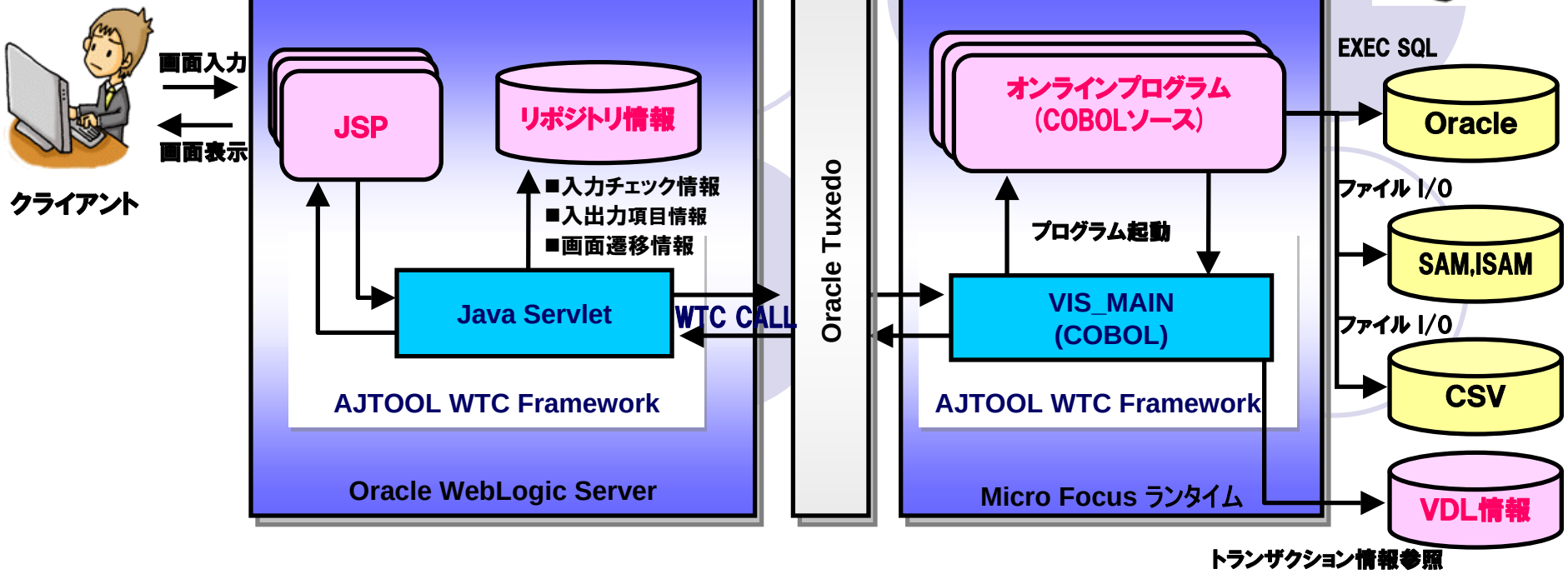
「領域初期化」、「特殊SORT」、etc・・・

規則遵守の形へ変換

- メインフレーム　独特の機能の洗い出し　：　調査分析・プロトタイプで確認し、変換ツールで対応するか、手修正で対応するか判断する

# 2－3. 移行資産の詳細　苦労した点と解決策

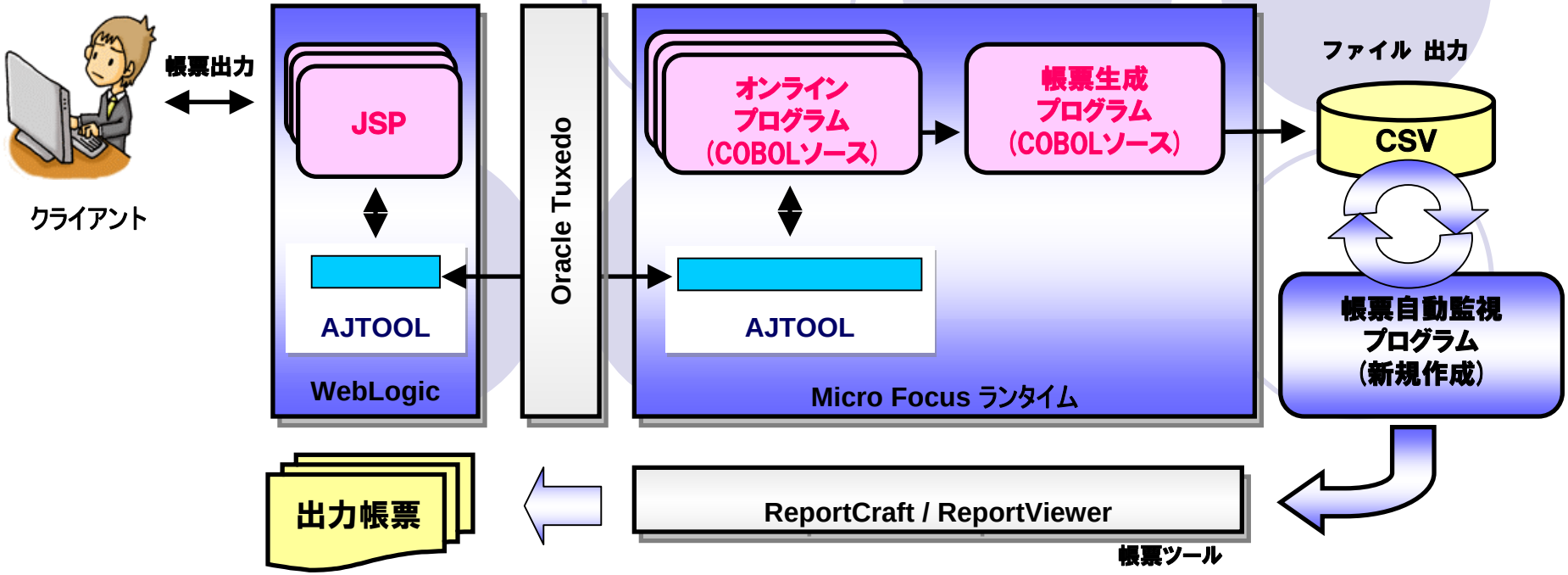
## ② ONLINEプログラム（画面入力）
### 移行後イメージ




- 違和感の無い画面　：　カラー選定、ファンクションキーの選定、最大最小化等
  Windowsボタンの抑制、オペレータへ移行後の画面確認
- 応答レスポンス　：　現行のレスポンスをキープ

# 2－3. 移行資産の詳細　苦労した点と解決策

## ③ ONLINEプログラム（帳票）
即時発行を要求されるONLINE帳票

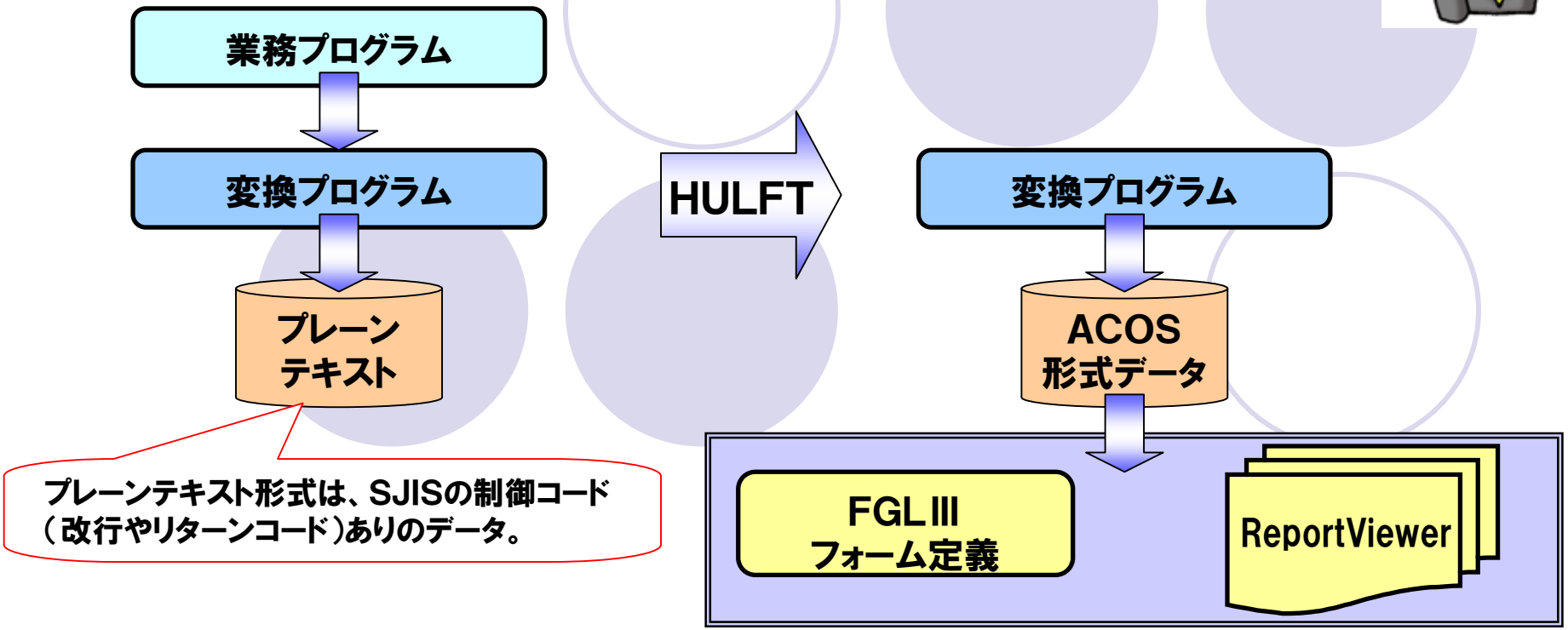



- ONLINE帳票（即時性を要求）　：　帳票データをCSV出力に変更
  CSV格納フォルダを自動監視するシステムを新規開発、
  即時に帳票ツールへデータの受渡しを実現

# 2－3. 移行資産の詳細　苦労した点と解決策

## ④ 電子帳票連携

### 移行後イメージ



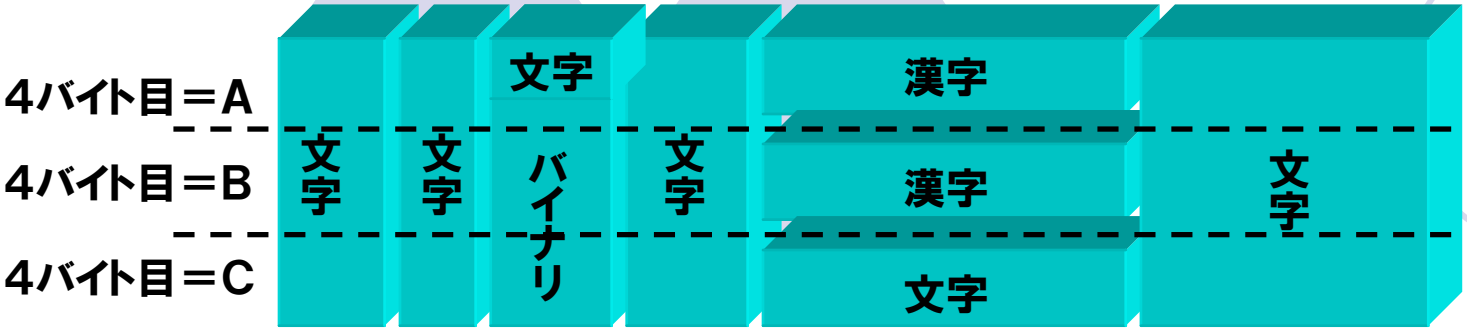

- 既存のフォーム定義を利用　：　変換プログラム
- フォントタイプの違いによる帳票項目の位置決め　：　手作業によるレイアウト照合

# 2－3. 移行資産の詳細　苦労した点と解決策

## ⑤ コード変換

### マルチレイアウト時のコード変換

| | | | | | |
|---|---|---|---|---|---|
| 999 | A | XXXXXXXXX | | NNNNNNNNN | XXXXXXXXX |
| 999 | B | S999 | XXXX | NNNNNNNNN | XXXXXXXXX |
| 999 | C | S999 | XXXXXXXXX | XXXXXXXXXXXXXXXXX | |



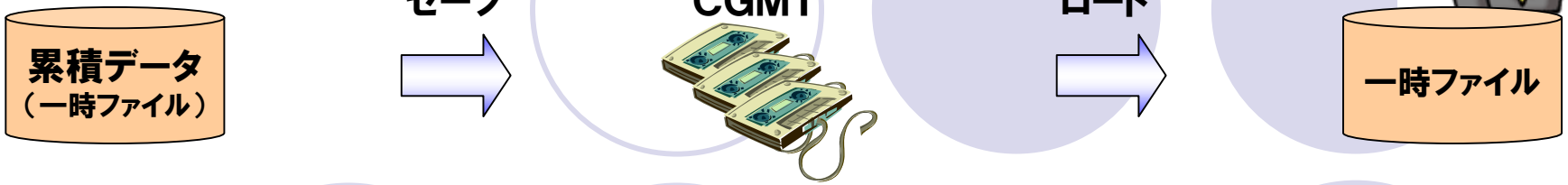

コード変換ソフトを駆使し、レコードの特徴に合わせてコードレイアウトを定義

- マルチレイアウトフォーマット　：　データ構造の理解とコード変換ツールの活用
- テスト時プログラム異常終了　：　情報共有により異常終了時のパターンを把握

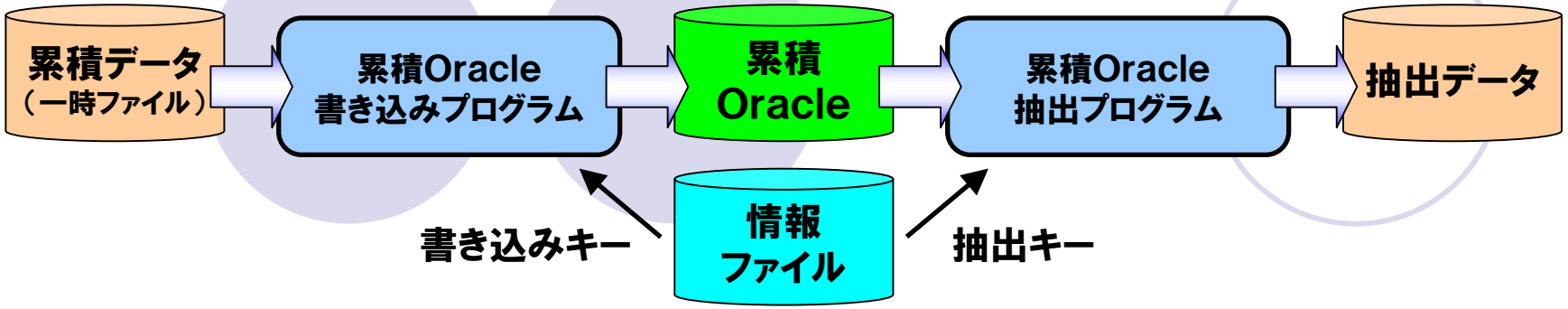
# 2－3. 移行資産の詳細　苦労した点と解決策

## ⑥ CGMTの扱い

メインフレーム




移行後



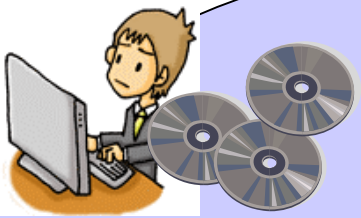

従来、年度別に保管していたCGMT内のファイルをOracleに変更し、日付キーを与えて抽出できるように移行。

- 累積データ管理のためのOracle化　：　管理レイアウトの追加とツール作成
- CSV形式のデータ変換に変更　：　ファイル仕様の作成とDBツールの活用

# 2－3. 移行資産の詳細　苦労した点と解決策

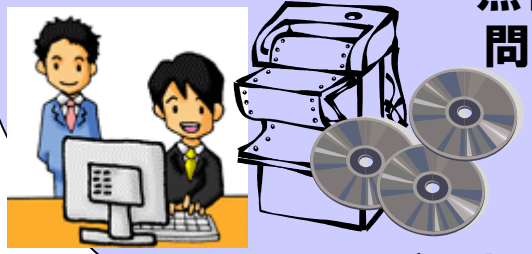
## ⑦ 結合試験の実施

**高松**



- 試験データ作成

- 照合結果＆問題点確認

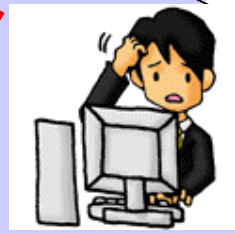


- ReportViewer印刷、PDFデータの送付

**メール・定例会**
**情報の共有**

**大阪**

- 試験実施



- 問題点票（質問票）起票

- 照合作業

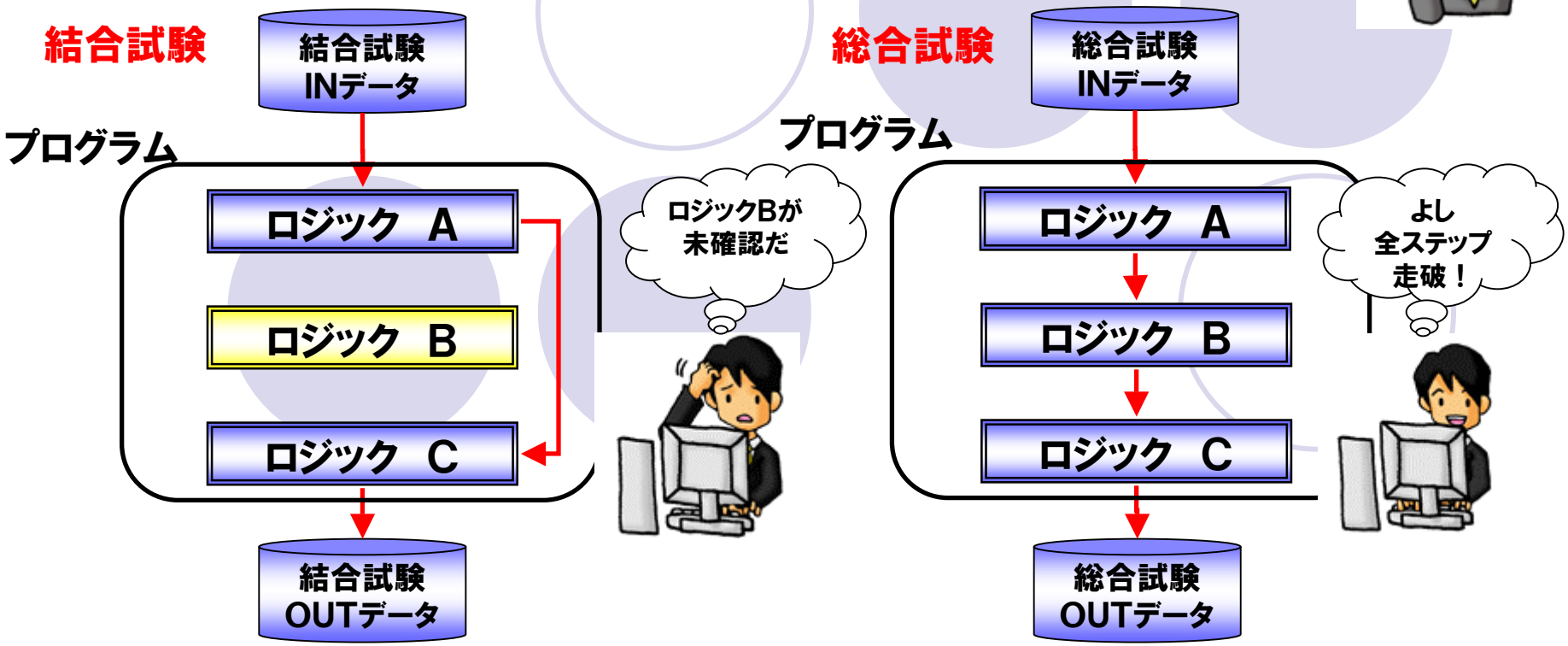
- 帳票出力依頼

| | | |
|---|---|---|
| • 異常終了原因の所在の切り分け | ： | 対応要員の配置、作業者間の情報交換 |
| • 帳票データの照合作業 | ： | 照合作業専門要員の配置 |
| • 連日発生する問題点 | ： | 定期的な内部進捗会議・情報交換票の活用 |

# 2－3. 移行資産の詳細　苦労した点と解決策

## ⑧ 総合試験 ・ 並行本番の実施

### ⑧－1 試験データによるロジックの網羅性確認

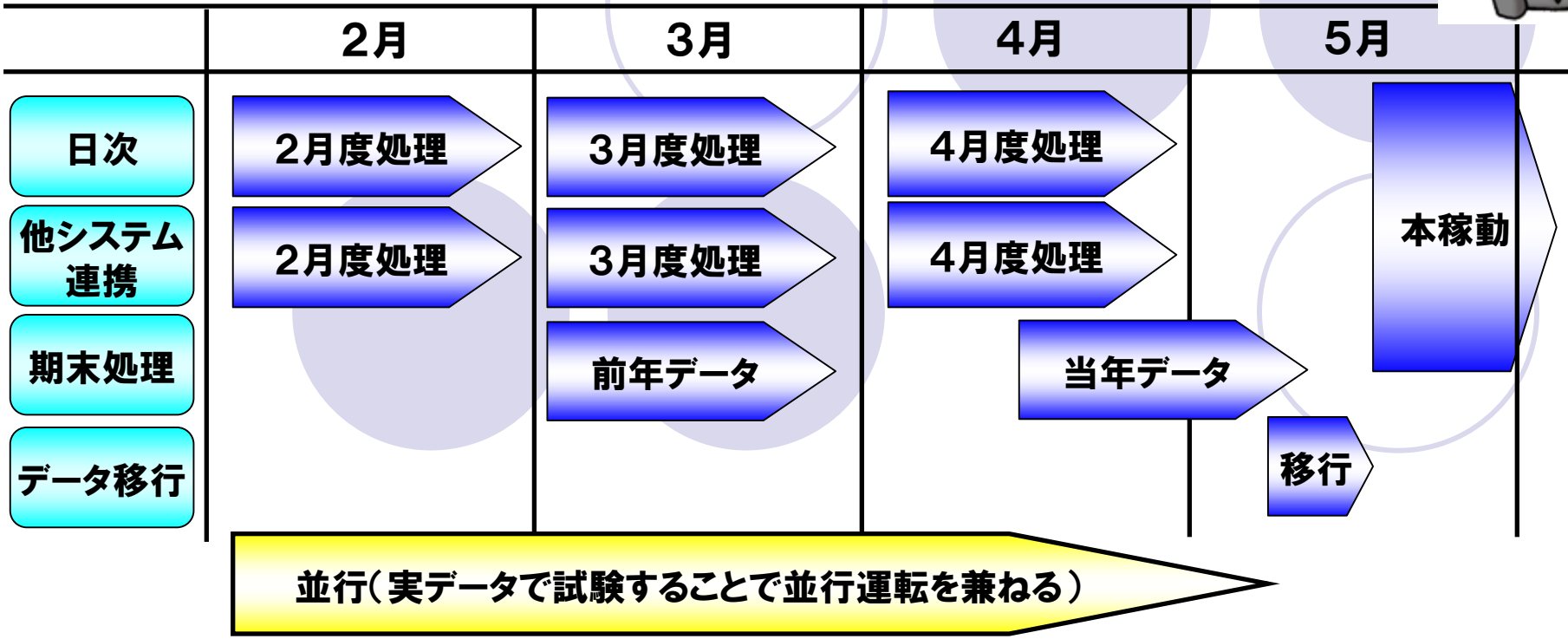



- データ網羅性　：　結合試験では、特定パターンのデータとなる「総合試験」で3ヶ月の実データによる試験を実施することでデータの網羅性を図る

# 2－3. 移行資産の詳細　苦労した点と解決策

## ⑧ 総合試験 ・ 並行本番の実施

### ⑧－2 総合試験計画




- 結合試験では予測できない障害（データ網羅性による障害）　：　障害原因の横展開、関連項目の調査・分析を即時で実施。
- 短期間でのデータ移行　：　照合試験で作成した移行ツールの活用。




# 3. プロジェクトを振り返って

## COBOL及びマイグレーション資産の評価

### 移行後の特徴

- 移行後の言語に違和感がない
- データを全てDISKで管理
- 実行ログ管理の一元化
- デファクトスタンダードな技術の採用と開発環境
- 既存の電子帳票システム利用

### 評価点

- 教育はエディタの使用法のみ
- オペレーション工数の削減
- システム監視の信頼性向上
- 新機能の追加が容易
- 品質の維持、ユーザ負担なし

# 3. プロジェクトを振り返って

## 総合評価

- 予定通りの納期、コストで順調に本番稼動（本番稼動後のトラブル0件）
- 保守・開発・運用効率の向上　（処理時間の短縮）
- ランニングコスト低減　（メインフレーム撤去）
- インフラ（ハード・OS）変更への柔軟性確保
- データベースのオープン化により、統計データ作成の迅速化との利便性の向上
- 次期システムステップアップへの基盤を確立

# 3. プロジェクトを振り返って

## 今後マイグレーションを行われる方へのアドバイス

- 資源の棚卸、改修計画、改修反映

必要最低限を洗い出せ！資産は常に変化している！

- 精度の高いテストデータ準備

正確なINとOUTがマイグレーションの鍵となる！

- 連絡事項・確認事項は文書で残し、プロジェクト全体で情報共有する

目的に向かいプロジェクト一丸となる！

- プロジェクト全体のコミュニケーションが重要

担当者間で意思疎通をはかる！




# 4. 最後に

マイグレーションに際しては、以下のような不安要素もあった。

- 現行のメインフレームと同等以上のオープン系プラットフォームを構築することが出来るのか？
- メインフレームと同等のレスポンスは期待できるのか？

これらの不安要素もマイグレーションサービス（MMS）を選定したことにより杞憂に終わった。

オープン系プラットフォームの場合、複数のハードウェアやソフトウェアの組み合わせとなるため、その組み合わせ次第で機能が不足したり不安定な動作となることが予想される。
しかし、これらもMMSによる基盤設計で安定したものとなっている。
レスポンスについても汎用機の処理時間を大きく下回る数値を示している。
また、試験を通じ各システムを見直すことも出来た。

**システム全体を柔軟性・拡張性のあるアーキテクチャに近代化させることが出来た。**




# 4. 最後に

ご協力いただきました各社様、ありがとうございました。

ご清聴ありがとうございました。

― リスクを負わないのがリスクである ―
ビル・ゲイツ

挿絵 ： ソザイヂテン VOL.1 ビジネス・オフィスシーンより