

オープン国内線旅客システム開発の考慮点

Points for Consideration in Domestic Passenger Service System Development on Open Platform

金子 時生

要約 2013年2月に全日本空輸株式会社の基幹システムである国内線旅客システムが稼働した。同社においては約30年ぶりの全面更改となった。

当該システムは、WEB予約または旅行代理店における予約業務、発券/決済業務、及び搭乗業務のサービスを担っており、一企業の基幹システムとしてのみならず、社会基盤としての安定した稼働が要求されるシステムである。

今回の開発では、オープンプラットフォームの環境でメインフレームと同等のサービスレベルを実現すること、所与の時間内で移行・稼働させることがゴールであった。その達成により、生産性の向上、及び迅速な新サービス提供が可能となった。

本稿は、ミッションクリティカルな大規模システムを稼働に導くにあたり、バイブルとしてきた基本方針と開発内容をご紹介します。開発の上で考慮してきたポイントを本特集号のイントロダクションとして取り纏めたものである。

Abstract All Nippon Airways Co. Ltd. has developed their core business system, Domestic Passenger Sales and Service System (PSS System), to go live in Open Platform environment on February, 2013. This is their complete renewal from previous system after more than 30 years.

In the Airline Industries, there are some systems with partial use of Open Platform, but this is the first in the World to have the complete PSS system running on Open Platform.

The Domestic Passenger Sales and Service System handles; reservation through Web; Reservation, Ticketing, Sales through Travel Agency; checking in at Airport; Operational Support to the destination. This core system is in the center of domestic air travel for customers and its having stable operation is essential not only to the business but as an infrastructure.

The goal for this project was set as to establish "same Business Practice on Open Platform", "Same Service Level as previous mainframe" and "100% safe migration". This target was reached and the IT Cost Structure as well as the service delivery time were improved.

In this report, how Mission-Critical and Super Large System implementation project used, as bible, the basic strategies and development is introduced. The important points outlined in this special edition are summarized.

1. はじめに

2006年4月に日本ユニシス株式会社（以下、日本ユニシス）は全日本空輸株式会社（以下、ANA）と同社の中核業務である“予約～発券～搭乗”の国内線旅客サービスを処理する基幹系システム「able-D」をオープンシステムに移行することで合意し、プロジェクト「Project AI^{*1}」を発足、2013年2月に新システム（開発コード名「ANACore」）を無事稼働させるこ

とに成功した。

かねてより日本ユニシスは、長年 ANA の IT パートナーとして、国内線旅客システムの開発・運用に携わっており、以下のシステムの稼働に寄与してきた。

1978年 米国 Unisys 社のエアラインパッケージ「USAS」をベースに「RESANA」稼働

1988年 RESANA を発展させ大幅な機能強化を実現した able-D 稼働

しかしながら、当初の RESANA 稼働から 28 年が経過し、長年に渡り改修開発を続けてきたことによるシステムの柔軟性の低下や、古い世代の言語（Fortran）を使用していることによる技術者の固定化など、ANA の国内線事業戦略の推進に対して、柔軟かつ迅速な対応が難しくなってきた。その解決に向け、技術面、コストなど様々な観点から検討した結果、ANA と日本ユニシスは米国 Unisys 社が保有しているオープンエアラインパッケージ「AirCore」をベースに国内線予約システムを再構築することとした。

2013 年現在、世界中のほとんどの航空会社における予約システムは、メインフレーム上で構築されており、同様の課題を抱えているが、旅客、貨物輸送など社会基盤として業務停止が困難であること、現行システムが大変複雑であること等の理由から、オープンシステムへの移行を検討するも、実現には至っておらず、今回の ANA 国内線旅客システムは、世界初の成功事例となった。

本稿では、新システム ANACore について、その開発の背景と目的およびゴールを 2 章と 3 章で述べ、開発の基本方針と具体的な開発内容を 4 章と 5 章で説明し、6 章で開発の際の考慮点を挙げる。

2. 開発の背景

2006 年当時の現行システム able-D（図 1）には、二つの課題があった。一つは「システム固有」の課題、すなわちレガシー技術（メインフレーム、Fortran）上での保守・開発コストの増大、約 30 年間の事業改革に伴う改修開発の影響によるアプリケーションの老朽化・メンテナンス性の悪化である。もう一つは「IT を取り巻く環境変化」への対応、すなわち Web2.0 などに代表されるコンシューマ IT の拡大（携帯電話の普及とインターネット利用拡大）による顧客の利用チャネルの多様化、オープンプラットフォーム技術の台頭、システム開発技術の多様化への対応である。これらの課題に対峙することは、able-D のオープンシステムへの移行を後押しするきっかけにもなった。

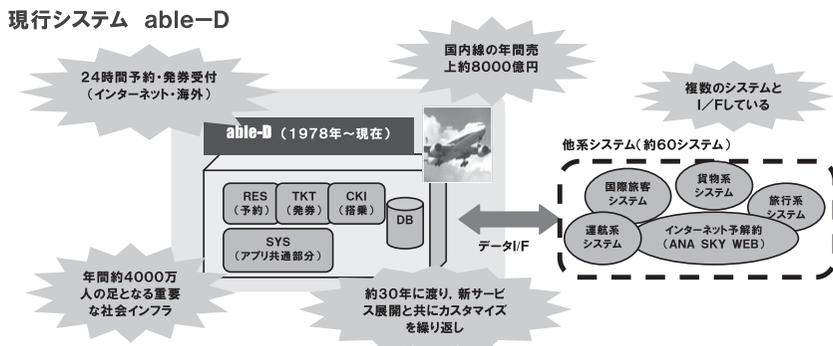


図 1 現行システム able-D の特徴

一方、able-D が持つ強みである、24 時間・365 日稼働、高性能（大量トランザクション・高レスポンス）への対応、きめ細かなユーザ向けサービスは維持することが求められた。

以上のことから、able-D が抱える問題・課題を解決し、また優位な点を踏襲しつつ、今後の環境の変化に耐えうる拡張性、柔軟性を備え持つ新システムを構築すべきという気運が高まり、これらの要件を実現する新システム ANACore の開発/導入に向けて、本プロジェクト ProjectAI が発足した。

3. 開発の目的とゴール

ANACore を開発するにあたり、ProjectAI は前章で述べた背景に根ざした開発の目的と、目的を果たしたことを確認する指標としてのゴールを定めた。本章ではそれらについて説明する。

3.1 目的

ProjectAI では以下の 3 点（詳細は表 1、表 2、表 3 を参照）をシステム化コンセプトとし、これを実現することを ANACore 構築の目的とした。

- IT コスト構造の改革
- システムの拡張性・柔軟性の確保
- 迅速なサービスの提供

表 1 IT コスト構造の改革

No	項目	内容
1	システムの維持・保守費および更新費の削減	H/W・S/Wのオープン製品化により、システムの維持・保守費および更新費の削減を図る。
2	業務AP開発費、保守運用費の削減	業界標準的な開発言語を採用し、ソフトウェア変更の容易性とオフショアを含めた技術者の容易な確保を可能にすることにより、業務AP開発費、保守運用費の削減を図る。
3	本番稼働後の保守生産性	本番稼働後の保守生産性を、able-Dより向上させる。

表 2 システムの拡張性・柔軟性の確保

No	項目	内容
1	システムインフラの拡張性	システムインフラ（H/W・N/W）を容易に拡張できる。
2	最新IT技術への対応力強化	最新のIT技術への対応力を強化する。（able-Dよりも最新のIT技術を適用しやすくする。）
3	業務内容や規模に応じた最適なシステムを構築	業務内容や規模に応じた最適なシステムを構築できる。
4	ベンダー選択範囲拡大	システムインフラを更新する際のベンダー選択範囲を拡大する。

表 3 迅速なサービスの提供

No	項目	内容
1	改修影響範囲の局所化	改修影響範囲の局所化を実現し、機能改修・追加の早期リリースを目指す。
2	既存業務AP資産の有効活用	既存業務AP資産の有効活用を実現し、機能改修・追加の早期リリースを目指す。
3	旅客系全体で共有可能なモジュール構造化	モジュールの構造化を行うことで、インターネット系、顧客情報系、国際旅客との連携といった他の旅客系システムとのモジュールの共有化を目指し、業務AP資産のスリム化を図る。

3.2 ゴール

ANACoreの開発, 移行, 保守のゴールとして以下の1) から8) をプロジェクト計画書で定め, それらをクリアすることを目指した。

(以下, 「ProjectAI プロジェクト計画書」記述から抜粋)

1) 開発期間

現行国内旅客系システム able-D が稼働するメインフレームのサポート停止 (2013 年) までにオープン系システム基盤への完全移行を実施する。

2) 新運用・保守方式の導入

構築するオープン系システム基盤上で, 新アーキテクチャに基づいた運用・保守方式を確立する。

3) ANACore プラットフォームの構築

将来の環境変化に向けて OS, ミドルウェア (J2EE コンテナ, RDBMS) を複数の候補の中から選定可能なオープン基盤をアーキテクチャとして確立する。

バージョンの選定を含め最適なプロダクトセットを採用し, ANACore のプラットフォームを確立する。

4) ANACore アプリケーション基盤の構築

AirCore の機能を活用し, 非機能要件を含め ANACore アプリケーション基盤を確立する。able-D にはない AirCore 独自機能で業務に有効なものについても, 可能であれば活用する。これにより効率的な ANACore アプリケーション開発を実現する。

5) 現行業務機能の移行

able-D で提供している業務機能と同等の業務機能を ANACore として構築する。また able-D の業務データを ANACore に移行する。

6) 外部システムへの新サービス提供方式の導入

外部システム連携は, Web サービスによる連携方式を可能とする。

7) ANACore の保守・改修体制の確立

移行開発を通じ, ANACore を保守・改修可能な要員を育成する。

オフショア開発を実践し, 開発スキームを確立する。これを活用することで開発・保守コストを削減する。

4. 基本方針

前章で述べたプロジェクトの目的・ゴールを目指すにあたり, 本章の各節で述べる基本方針を定め, 種々の環境変化に左右されない計画を設定した。

4.1 開発全般

開発全般に関わる基本方針として, 以下に挙げる 3 点を定めた。

1) 業務品質とサービスレベルの維持

able-D の機能を ANACore に構築することで業務品質を維持し, 現行メインフレームで実現している基盤・運用管理機能をオープン系システム基盤上に構築することでサービスレベルを維持する。

2) 開発効率の向上

既存パッケージである AirCore のフレームワークや開発ツールを可能な限り利用するとともに、AirCore の SOA アーキテクチャを踏襲して拡張性と柔軟性を確保する。

3) 既存環境との整合性の確保

ゲートウェイ機能により、外部システムや端末といった既存環境の現行仕様を変えずに通信方式の差異を吸収する。

4.2 システム構成

拡張性と可用性を確保することをシステム構成の基本方針とした。すなわち、OS/ミドルウェアを柔軟に選定可能（プラットフォームフリー）とし、AP サーバはスケールアウトにより、DB サーバはスケールアップにより拡張性を確保する。また、可能な限りオンライン無停止でのリリース方式を採ることと、able-D と同等以上の冗長性を持たせることにより、可用性を確保することとした。

4.3 運用方式

以下の3点で示すように、保守運用ツールを最大限活用し、安定的かつ効率的なサービス提供を実現する。

- 開発支援ツール/フレームワークの継続活用による、効率的な保守作業の確保
- ANACore 開発プロセスに則ったドキュメント体系の維持運用による、改修作業の生産性確保
- 稼働状態の継続的なモニタリングによる、プロアクティブな対応の実施（予防保守の充実）

また、各ベンダーのノウハウ・スキルを十分に引き出し、確実に連携可能な協業体制（＝ベンダーコンソーシアム）を確立する。すなわち、課題発生時に原因追求をたらい回ししない体制や、製造メーカの技術力（ノウハウ・スキル）と、十分な検証・保守によりリスクを回避するベンチマーク環境を確保する。

加えて、ANA 標準 ITIL をベースとした運用プロセスを確立し、アプリケーション/インフラ部門の最適な役割分担を設定して ANACore の稼働品質を保証し、継続的に維持管理できる運用保守体制を確立する

4.4 移行方式

便の正常運航に影響を与えず、許容された時間内（最終便から初便までの4.5時間）で切替える。また、切替えにおけるコンティンジェンシープランを用意する。これにより、100%安全確実な移行を実現する。

4.5 開発方式

大規模開発に対応するために、管理可能な規模である1000人月以下に管理スパンを分割する。すなわち、業務系（予約・発券・搭乗・座席管理・共通）、フレームワーク、ゲートウェイ、基盤、移行のサブシステムにてチームを分割し、さらに開発グループによりベース仕様開発グループ1～3と現行改修取り込み1～3の計6グループに期間を分割する。開発のマスターズ

スケジュールを図2に示す。

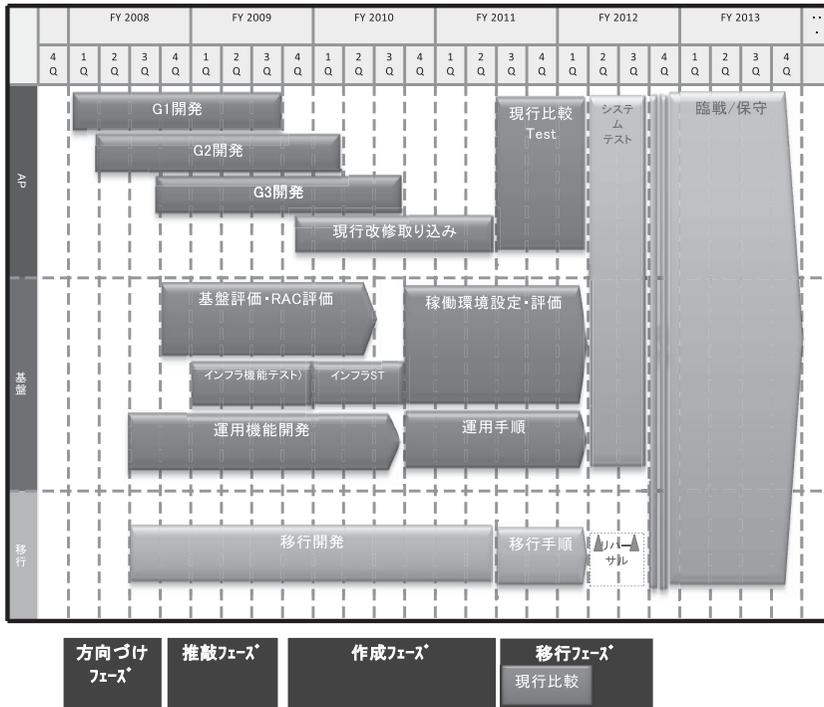


図2 開発計画

長期間の開発に対応するために、開発期間を“方向づけ”、“推敲”、“作成”、“移行”の4フェーズ(図3)に分割し、フェーズ毎に適切なタイミングでチェックポイントと完了条件を設定し計画及び実績を評価して、未完了項目や改善すべき課題に対して必要な対策を実施し、結果を次フェーズ以降の計画へ反映する。

現行改修取り込みは、要員の習熟度が向上した後半の作成フェーズに計画する。

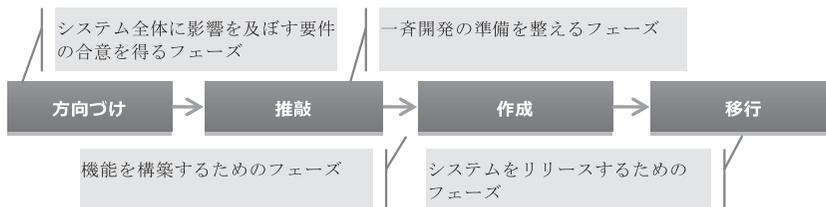


図3 フェーズの位置づけ

また、品質面での要求に対し、次に示す基本方針を定めた。

- 1) 本開発の開始前に、事前に検証・確立した開発手法/アーキテクチャ (OS, ミドルウェア等)/品質管理方式 (品質メトリクス, 評価基準)を採用した本開発を計画する。
- 2) 現行同等の機能性を確保するために、十分な現行比較テスト期間を設定し、本番トラン

- ザクションデータの取り込み機能等、ツールを活用して品質を向上させる。また、able-Dのテストケースやシナリオを利用する等、運用/改修ノウハウを活用した品質確認を行う。
- 3) フェーズ/グループ開発の単位で検証テストを行い、早期に仕様認識や操作感の齟齬を確認し、開発にフィードバックする。
 - 4) 本番移行時と同等の移行リハーサルや、稼働後フォールバックのリハーサル及び訓練を実施し、移行やフォールバックの手順を実証し成熟させる。

4.6 プロジェクト運営

以下の三つの開発の特性を考慮し、プロジェクト運営について1)から5)の基本方針をたてた。

- 大規模開発（大人数・作業量大・多様なステークホルダー）…1), 2)
 - 長期に渡る開発のため、外部・内部環境ともに変化する可能性が大きい…3), 5)
 - 意識モチベーションの低下…4)
- 1) PMOによるプロジェクト管理を強化

プロジェクト全体のPMO機能の設置に加え、サブチーム毎のPMOを設置し、進捗・課題・品質のプロジェクト管理を強化した。
 - 2) アーキテクトによる品質確保

開発工程の標準化を定めるとともに、コード解析ツールによるコード規約への準拠、ふさわしくないコードを検出し管理することにより、開発チームの是正対応の追跡運用を実施し、プログラムコード品質を確保した。

また、システム全体としての性能目標を達成するために、個別機能単位でのレスポンス・処理時間目標を定め、機能テスト段階～システムテスト段階で目標達成状況を監視し、未達成の機能については改善方法の検討、及び改善支援を実施し、性能品質を確保した。
 - 3) 情報の共有化促進

プロジェクト管理における進捗・課題・品質の情報については、プロジェクト全体に開示し、それぞれの視点による状態分析～課題の気づきができるようにした。
 - 4) 一体感の醸成

プロジェクト開始時期から、月一回の全体ミーティングを設け、マスタースケジュール上の進捗状況とステアリングコミッティの意思決定内容を共有するとともに、月間 MVP を選出し、プロジェクト全員で表彰することでモチベーション向上を図るなどの運営を実施した。
 - 5) 外部ノウハウの有効利用

システム構成の中でプロダクトを提供している主な外部ベンダーとベンダーコンソーシアムを構成し、定期的なテクニカルミーティングにて開発状況を共有することで、稼働に向けたリスク項目の早期発見と回避策の検討及び対応を推進した。

5. 開発内容

本章では、ANACore 開発の具体的な内容を、アプリケーション、システム構成、システム運用、移行の四つに大別して説明する。

5.1 アプリケーション

5.1.1 業務アプリケーション

able-D では、メインフレーム上で複雑に関係あった構成で配置されていた機能（図4上）を、ベースとする AirCore のモジュール構成及び SOA の考えを踏襲し、ANACore としてのモジュール構成として構築した（図4下）。

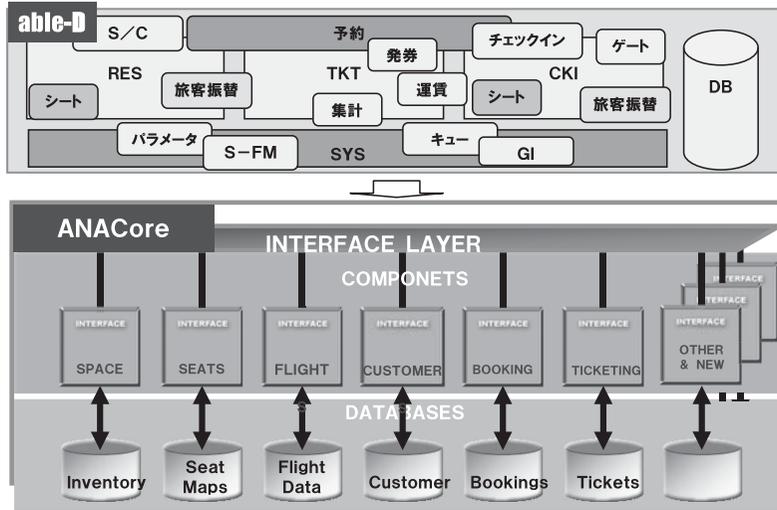


図4 業務アプリケーション構成

5.1.2 インターフェイス・ゲートウェイ

ANACoreの接続端末、及び外部システムについては、接続仕様を現行通りとする基本方針であることから、端末ゲートウェイ及び外部システムゲートウェイのインターフェイスレイヤー機能を構築し、インターフェイスの差異を吸収した（図5）。

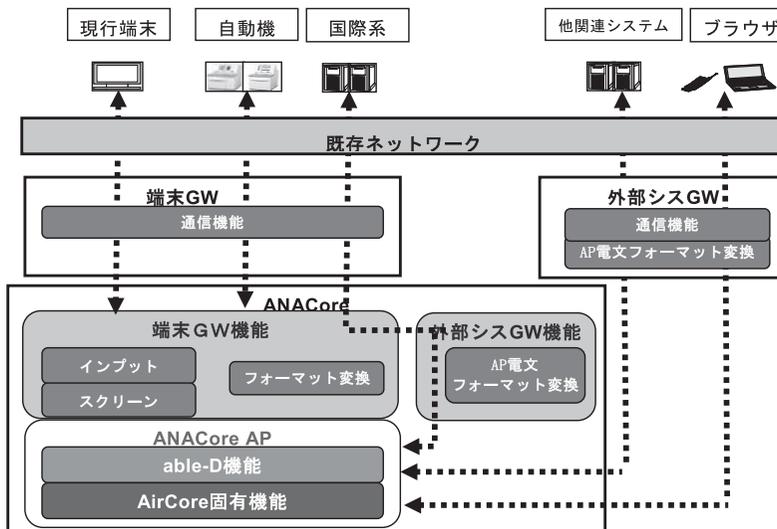


図5 インターフェイス・ゲートウェイ構成

5.1.3 フレームワーク

able-D の稼働環境であるメインフレームと同等のサービスレベルが要求されていることから、米国 Unisys 社製メインフレーム上では HV-TIP, IR, シンビオントコントロールとして実現されていたトランザクション制御, 障害リカバリ制御, プリンター制御を実現するためのフレームワーク機能を構築する必要があった。

開発のアプローチとして、AirCore フレームワークの上に、日本ユニシスのオープン系トランザクションアーキテクチャ標準の AtlasBase に沿った Maia フレームワークを基盤として拡張機能を補完し、再実装した。(機能一覧については表 4 参照)

表 4 フレームワーク機能

Maia フレームワーク機能		ANACore フレームワークとして拡張した主な項目
プレゼンテーション	サブレット	
	アクション	認証アクセス制御
	JSP	
ビジネスロジック	DTO	
	Façade	
データアクセス	Dao	DB 情報キャッシュ管理機能
共通基盤	イニシャライザ	
	メッセージ	サーバー斉通知：対象サーバ指定（停止サーバ認識）及び通知結果
	例外	
	ログ出力	AP 及び各種ログの採取と見易さ向上 入出力ログ編集
	モジュール情報	端末閉塞/機能閉塞機能
	トランザクション制御	ユーザセッション管理 分散処理：複数サーバへの分散実行と状況監視 エラー時考慮 サーバ構成考慮：サーバ負荷を考慮したロードバランス 実行スレッド数・コネクション数の均一化 流量制御：サーバの優先度設定を反映した TRX 制御 同時実行スレッド数制御 AP パーティショニング：処理 DB サーバの振分け、 DB サーバ障害時の自動再配分
エラーダンプ		
バッチ・非同期	適時バッチ	バッチ FW
	非同期	即時/定刻起動の選択、リトライ要否の選択
	—	並行処理：処理の分割・平行実行及び状況監視
帳票	帳票	PUSH 出力：送達保証有/無の選択、0J/DJ 応答
Web	Web サービス	端末 (CUI/GUI) FW
外部システム連携	例外	外部システム呼び出し制御 外部システム FW
ユーティリティ	文字関連	端末 (CUI/GUI) FW
	日付関連	業務日付制御

5.2 システム構成 (実行環境)

ANACore は、able-D で利用中のメインフレームと同様のサービスレベルの実現が要件であることから、メインフレームでの XTPA 機構によるノーダウン（障害発生時の仕掛り中処理を除く）を、AP サーバでは複数ノードによる障害時の縮退運用で、DB サーバでは複数ノードの Oracle 社 RAC 機構を利用した障害時の縮退運用で実現した。システム構成イメージは図6、図の凡例とシステム構成一覧は表5の通りである。

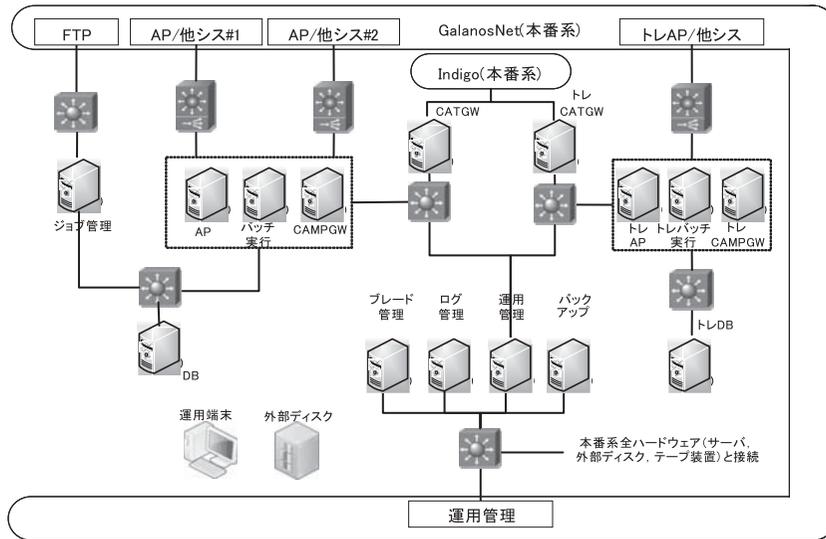
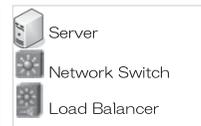


図6 システム構成概要図 (図中、トレはトレーニングの略)

表5 システム構成一覧

Subsystem	冗長化	機種	用途
DB Server	○	HP Superdome	Database Management
AP Server	○	Unisys rE6000	Online Application
Batch Server	○	(Blade Server)	Batch Application
Job Mgmt Server	○		Job Control, FTP Service
Log Mgmt Server	○		Manages log files from all servers
Training AP Srv	○		Online Application for Training
Training Batch Srv	○		Batch Application for Training

Subsystem	冗長化	機種	用途
Blade Mgmt Srv	○	Unisys rE5000	Manages Blade Server
CAMPGW	○	HP rx2660	Gateway to External System
Training CAMPGW	○		Gateway to External System for Training
Backup Server	○	HP rx3600	Backup Tape Management
Operational Mgmt.	○		Monitoring, Failure Detection and Notification
Training DB Srv	○		Database for Training Environment



5.3 システム運用

ANA 全社標準の ITIL をベースとし、アプリケーション及びインフラ分野のそれぞれに、サービスマネジメント統括機能、サービスサポート、サービスデリバリのプロセスを確立した(図7)。以下に説明する。また、必要となる運用機能項目を表6にまとめた。

- サーマネジメント統括機能

ANACore システムの運用保守において、PDCA サイクルを確立し、全プロセスを管

理・統括する。(サービスレベル管理, 財務管理)

● サービスサポート

ANACore システムの日々の運用に関する管理業務の具体的な手順をプロセス毎に確立し, 各プロセスを管理する。

(インシデント管理, 問題管理, 変更管理 (リリース管理含む), 構成管理)

● サービスデリバリ

ANACore システムの品質向上のために, 中長期的な視点において運用管理業務をプロセス化し, 管理する。(継続性管理, 可用性管理, キャパシティ管理)

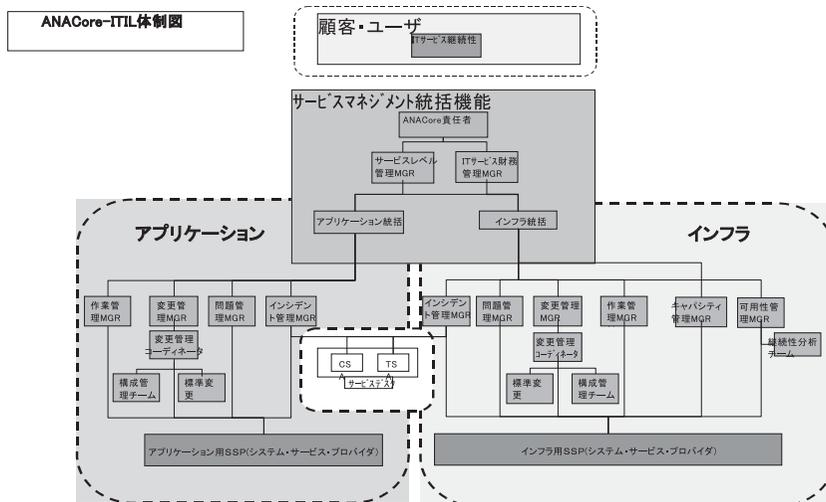


図7 ANACore システム運用体制

表6 運用機能項目一覧

運用項目	概要
監視運用	インシデントの発生を管理するための監視運用を行う。
障害時運用	障害発生時のリカバリを迅速にする運用を行う
オンライン入力制御運用	オンライン業務への可用性を高める運用を行う。
リリース運用	OS やプロダクト S/W, AP のリリースに関する運用を行う。
キャパシティ運用	性能面での監視, システム分析, 性能データの管理と運用を行う。
オペレーション運用	サービスデスクの運用を行う。
ログ運用	発生した障害を速やかに解析する運用を行う。
バックアップ運用	システムの継続性計画に基づいた運用を行う。

5.4 移行

5.4.1 移行要件

移行の要件は「所与の時間内での 100% 確実な移行」であり, 具体的には以下の各項目を実現する。

- システム切替えによって, 便の正常運航には影響を与えない
- システム切替えが漏れなく実施できる
- 許容されたシステム停止時間内 (4.5 時間) でシステム切替えを完了できる
- 切替えたシステムで able-D と同等のレベルでの業務を継続できる

- ノーリターンポイントを経過する前であれば、able-D への戻しを実施しても便の正常運航には影響を与えない
- 万一システム切替えに失敗した場合のコンティンジェンシープランが用意されている
- 切替え後の繁忙期も問題なく業務を実施できる

今回の移行の特徴は、able-D の前身である RESANA 以来、約 30 年ぶりの初のデータ構造の全面変更を伴う移行となること、事業における中核システムであり、周辺との接続システムが多いため、システム一斉移行が必要なことである。

5.4.2 切替え対象

able-D から ANACore に移行する際、切替えの対象となる項目を表 7 に挙げた。

表 7 切替え対象項目

対象項目	説明
ハードウェア	OS、ミドルウェアを含む、国内旅客系ハードウェアおよび OS、ミドルウェアのプロダクトセット群 先行設置～セットアップを行う。
アプリケーションソフトウェア	ANACore アプリケーション 先行リリースを行う。
運用・保守ソフトウェア	運用および保守に使用されるソフトウェア 先行リリースを行う。
データベース	本番業務用データベースおよび運用保守用データベース 新規開発する移行プログラムを使って、先行セットアップ・事前セットアップ・当日セットアップを行う。
他関連システム (オンライン)	able-D にオンライン・アクセスするホスト/サーバ群 (自社/他社) 先行で ANACore 接続ネットワークとも接続できる構成を設定し、当日は切替えのみの作業とする。
他関連システム (バッチ)	able-D からバッチ形式でデータ提供を行うホスト/サーバ群 (自社/他社) ANACore への経路変更を行う。
端末	able-D にアクセスする端末群 先行で ANACore 接続ネットワークとも接続できる構成を設定し、当日は端末からのログイン先指定で新システムに切替える。

5.4.3 移行イメージ

移行当日のシステム停止時間を最終便から当日初便までの 4.5 時間とするために、以下に示す able-D からの段階的な移行 (図 8. 移行データの説明は表 8) を組み立てる必要があった。

- 1ヶ月前から1週間前までの先行移行：本番ネットワーク環境に HW 環境のセットアップ及び AP プログラムのリリース展開を実施し、当日はネットワーク切替え作業のみで ANACore に振り向けるようにした。また、ANACore として初期設定が必要なデータ (able-D では持っていないデータ) は先行してセットアップした。
- 稼働までの1週間の事前移行：able-D から ANACore へのデータ移行について、初回に全体移行を行った後、2回目以降は差分移行を実施する形とした。また、当日の切替え時間の中では十分な稼働確認が困難であることから、この事前移行期間の中で、主要空港での稼働確認を計画した。
- 当日移行：当日分の移行データ反映と、ネットワーク切替え (外部システム接続切替え・端末接続切替え) により 4.5 時間での切替えを実施した。

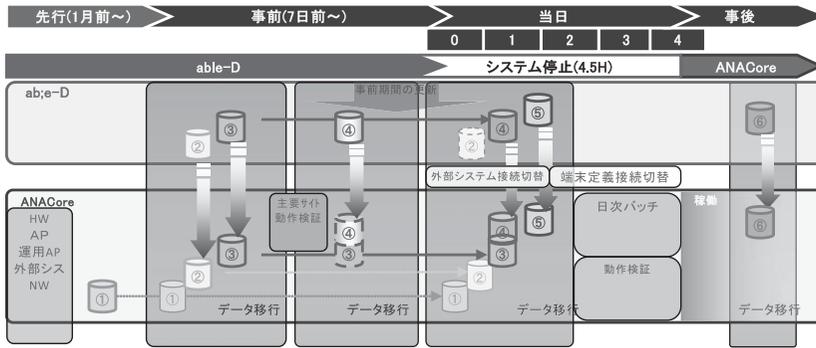


図8 移行イメージ

表8 移行データ分類

#	① ⑥のデータの分類	主な対象
①	「事前」前から変更が発生しないデータ (旧システムに存在しないデータ)	静的データ (マスターデータ)
②	「事前」に移行すれば、以降は変更が発生しないデータ ※本番業務運用を制限し、データ変更を抑制	
③	「事前」から「当日」の間に変更が発生するデータのうち、 「事前」時に移行する分のデータ	動的データ (トランザクションデータ)
④	「事前」から「当日」の間に変更が発生するデータのうち、 「事前」後の差分データ	
⑤	「当日」のみ移行可能なデータ	
⑥	稼働後に移行するデータ (新システムでバッチ処理を行う前までに移行すればよいデータ)	保存データ (履歴データ)

5.4.4 稼働後のコンティンジェンシープラン

本システムは国内運輸事業の中核となる社会基盤であることから、切替え失敗により運航に影響を発生させることは絶対に避けなければならなかった。そのため、不測の事態の発生によるANACoreシステムの停止に備え、コンティンジェンシープランを備えておく必要があった。プランとは具体的には、ANACoreに切替えた後1週間の期間はable-Dに切戻すことができる仕組みのことである。

考え方としては、切替え時点以降、able-Dにも差分更新を反映していき、コンティンジェンシー発動時には、ネットワークを切替えて(逆移行のイメージ)、切戻す方式を採った。

6. 開発の考慮点

本章では、ProjectAIでANACoreを開発するにあたり、特に考慮が必要だった点について解説する。

6.1 プロジェクト管理

300人〜ピーク時800人のメンバーによる開発のプロジェクト運営として、プロジェクト管理項目を管理作業に具体化した上で、PMOの組織化と管理作業(表9)の割当てが必要であった。プロジェクト全体のPMO及び、サブチーム毎のPMOのメンバー構成として、30人〜50人によるプロジェクト管理体制で運営した。

表9 PMO 作業項目

エリア	全体	スコープ	タイム	コスト	コミュニケーション
管理対象	プロジェクト	要件管理	計前・進捗管理	要員・財務管理	プロジェクト運営
管理計画	プロジェクト計画	要件・案件・管理計画	進捗管理計画	要員・財務管理計画	プロジェクト運営計画
作業内容	フェーズ計画(作成/変更) フェーズ終了報告(作成) フェーズ計画変更承認 ステミ材料作成	要件管理運用(作成) 要件管理運用 要件追加/変更見積り	進捗管理運用(作成/変更) マスタースケジュール管理運営(作成/変更) フェーズスケジュール管理運営(作成/変更) 進捗進捗管理運営(更新/集計) 日次進捗査核・確認運営 【各種計画/実績の管理対象】 ・WBS ・設計ドキュメント ・稼進 ・プログラム/スクリプト/SQL/JOB/ ・テストケース設計・作成 ・テスト実施 ・テスト障害(個体の状態管理)発生・対応 ・テスト障害対応生産性 ・TQ 社内PMR事務局	プロジェクト予算/見通/実績 フェーズ予算/見通/実績 月次予算/見通/実績 要員計画/実績(くりり表) 発注(SOW)/稼収業務 作業実績報告書受領(協企-DNP) 社内財務報告	【金銭・連絡会・レビュー運営】 ・フェーズ会議(プロジェクト) ・成果物承認会議 ・各種計画レビュー ・月次会議(プロジェクト) ・進捗会議(チーム/プロジェクト) ・マネジメント定例会議(プロジェクト) ・課題推進会議 ・内部連絡会 ・パートナー連絡会 ・日次障害連絡会(障害個体管理) 各種会議議事作成

エリア	品質	構成	課題/リスク	人的資源	ファシリティ
管理対象	品質管理	構成管理	課題/リスク管理	要員管理	ファシリティ管理
管理計画	品質管理計画	構成管理計画	リスク管理計画		
作業内容	品質管理運営要領作成 進捗品質データモニタリング 進捗個別機能定モニタリング 進捗品質状況報告 進捗品質アクション管理 フェーズ品質評価 コード内部品質モニタリング 単体テスト受入れ 障害対応フロー整備 機能テスト障害対応プロセスレビュー システムテストリソース判定レビュー 個別性能レビュー 個別性能モニタリング リグレーション実施計画/実績管理 品質確認テスト実施計画	構成管理運営要領作成 開発サーバ/リソース管理 構成管理サーバストリーム管理 テスト環境構築推進 ・初期値/データセットアップ ・テスト機器調達 ・テスト用サブライ調達 テスト環境割当管理 各種管理ツール整備 ライブラリ管理 リリース管理	課題管理運営要領作成 各週会議議オプザブ 課題ヒアリング 課題対策ファンリティー 課題対策スタッフ推進管理	人退管理・手続き 要員台帳(作成/更新) スキルマップ整理 PCアカウント管理 倉庫管理 体制図(作成/更新)	スペース調整 LAN環境運営 開発PC管理 プリンター環境運営 コピー機管理 各種媒体管理 用紙管理 事務・生活用品管理 社内事務運営 ビル管理窓口 健康管理窓口

表9のプロジェクト管理項目として、プロジェクト計画（人材計画，コスト計画も含む）及びスコープを定義してプロジェクト立上時のベースライン計画とし，その通りに開発を推進することでゴール（稼働）を迎えられると想定した。コミュニケーション，構成管理，ファシリティは，管理計画及び運営要領を定め，通常運営化することで統制を計った（図9）。プロジェクト推進時の問題点は，最初は進捗・品質上の課題として表面化してくることから，これらを週次での主な監視対象とした。状況悪化の傾向が出てきた際には，上記ベースライン計画を立てた際の前提条件との差分要素が生じていると仮定し，それらを課題とした。その課題の原因がベースライン計画のずれにあるのか，人材の質・要員数にあるのか，または，プロジェクト運営のコミュニケーション，ファシリティにあるのかを分析することで，有効な対策の手立てを講じる方針とした。詳細は本特集号の他の論文に譲るが，本稿ではこの進捗・品質・課題の管理で考慮したポイントを説明する。

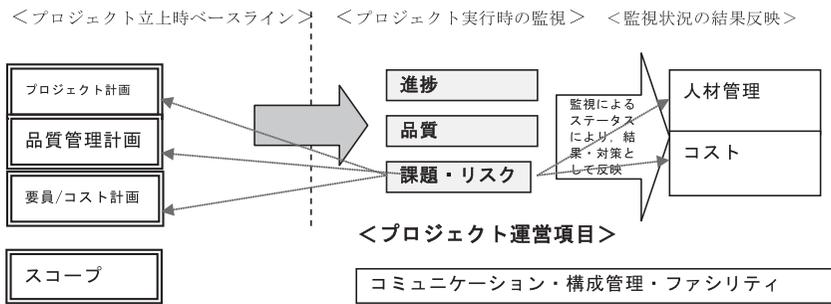


図9 プロジェクト管理項目

6.1.1 進捗管理

進捗については，WBS 及び成果物の台帳で管理した。工程の開始ごとの計画作成時には通

常、計画生産性（時間/人または、成果物件数/人、等）を設定する。成果物または作業の完了数の計画/実績を監視対象とするのはもちろん、計画時に前提とした計画生産性を日次/週次のKPIとして監視し、グラフ化等の可視化をしてプロジェクト内で共有し、傾向分析をすることで、工程の計画に影響が出るような大幅な進捗遅延を可能な限り防ぐ手立てとした。

6.1.2 品質管理

品質については、各工程毎の定量的な品質目標値を設定した定量分析を評価軸の一つにしている。その目標値は、プロジェクトの特性である言語表現性、機能毎の複雑度を反映して設定した。また、不具合分析では、従来の原因分類に加えて前工程でのすり抜けを原因とする部分にも注目し、定量化した目標値を設定した。分析として前工程でのすり抜け要因が多い機能については、前工程の見直しまで戻る対応も考慮してきた。

この定量分析に加え、定性分析としては、徹底した現物・現場主義を方針として、“成果物がどうなっているのか？”“現場ではどのように感じているのか？”を目視確認・ヒアリングし、週次で監視する運営とした。

品質管理という観点とは異なるが、製造工程での品質を担保するために、製造を担当するパートナーでの単体テスト結果について、テストデータの生成からテスト実行までをスクリプト化し納品してもらうことで、オフショアにより再実行した結果を再検証する受入れ手続きが可能となった。

6.1.3 課題管理

進捗・品質ともに計画通りの実績となるのが理想ではあるものの、実際には、大小さまざまな要因で、進捗・品質状況に色々な課題が現れてくるものである。そのため、課題管理に一番焦点を当てたPMO 要員アサインを行ってきた。

課題管理要員はおよそ全ての進捗・品質の打ち合わせに出席し、状況認識および課題の洗出しを行う。課題対策については、仮説による原因分析及び仮説検証後、対策をステップ化していく活動をファシリテートする運営とした。

課題対策を打つにあたり工夫した点は以下の二つである。

- 課題対策については解決作業のステップ化（誰が、何を、何時までに）を設定し、重要課題についてはプロジェクト内で対応の進捗を共有できるようにした。
- 原因の所在については、開発側にある場合、顧客側にある場合、外部要因にある場合とステークホルダーを跨る可能性があるため、課題管理要員には、第三者の立場を取れるパートナーを割り当てた。

6.2 アーキテクチャ

ANACore は国内交通手段の基盤であり、サービスを停止することが許されないシステムであるため、開発前のアーキテクチャ検証及び開発開始後のインフラ・プロダクト検証を徹底的に実施したうえで、アプリケーションを構築した。

6.2.1 事前アーキテクチャ検証

オープンプラットフォームでのインフラ・プロダクトセットを構成するため、サーバ処理性

能（スループット性能）/レスポンス/バッチ処理性能/簡易ロングラン/代表障害運用に関してベンチマーク検証及び評価を実施し、稼働時のHW・SW構成を決定した。

6.2.2 事前開発プロセス検証

開発標準化とガイドラインを整備し、主要機能のベンチマーク開発を実施して、開発プロセスを検証した。

6.2.3 インフラ・プロダクト検証

事前アーキテクチャ検証及びインフラ構成設計で決定したHW・SWプロダクトでの対障害性動作については、インフラ環境構築前に本番稼働相当の環境を用意し、想定できる詳細ケースについて以下1)と2)の徹底検証を実施し、プロダクトリスクを排除した。

1) 可用性検証（RAC・DBサーバ）

HP-UX上でのOracle RACがAirCoreミッションクリティカル環境に適合した可用性を保持していることを検証

2) サーバ運用性検証（APサーバ・DBサーバ・DISK・バックアップサーバ・運管サーバ・ログ管理サーバ）

日本ユニシスは、過去30年以上に及ぶable-Dのサポート経験、その他数多くのミッションクリティカルシステムのサポートに際して発生した各種障害をもとに、メインフレーム出荷前に、対象プロダクトスタックに対してのQA（品質保証）を実施するメトリクスとプロセスを保有していた。

今回のプロダクト検証では、当該メトリクスを基本に、商用Unix、RDBMS、Webサーバが内包する脆弱性の仮説をたて、過去実績に基づく観点から生成したプロダクト単体、あるいはセットとしての潜在不具合をあぶりだすための検証活動を約1年にわたって実施した結果、各プロダクトが抱えていた30件以上の潜在不具合を発見し、各ベンダーとともに対応を進めることにより、プロジェクトでの製品利用開始以前に安定的なプロダクトセットを構築することに成功した。これにより、システムテストや稼働前後でインフラにまつわる致命的な不具合が発生することを未然に防止することができた。

6.3 テスト戦略

4.5節の開発方式の基本方針でも示した通り、ANACoreのAP開発では、機能特性により3回（グループ1～3）に分割したベース機能開発と、ベース機能開発後に、その期間中のable-Dで開発された仕様の反映として3回（現行取込1～3）に分割した現行改修の取込開発を実施する計画とし、設計～テストの開発のプロセスを合計6回の開発グループに分けて繰り返し実施した。

また、接続する外部システムも約60システムと多く、開発グループに合わせて接続性と機能性を作り込んでおく必要があった。

一方、インフラ開発においても、業界初となる全てオープン技術による国内線旅客システム構築であるため、求められる堅牢性、安定性を実現するべく、プロダクトの検証～運用の機能性の作りこみを実施する計画であった。

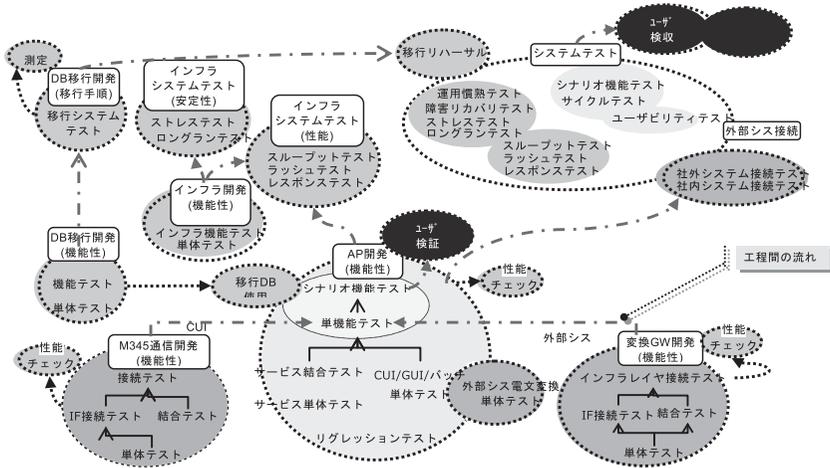


図10 テスト関連図

図10に示すように、6回に分割したAP開発グループにわたる機能テスト、各開発グループ毎にレベルアップしていく品質や、多くの接続先との接続テストの組み方など、2000機能を超える機能単位でのテストをどのように組み立てていくか、また稼働前の総合テスト（業務面・運用面）として、システムテストやユーザ検収をどのように実施するのかについて、テスト全体のシナリオを定義し、個々のテスト計画に反映させていく必要がある。そのため、テスト全体像の設計（全体テストの基本方針）をテスト戦略書（図11）として作成するに至った。

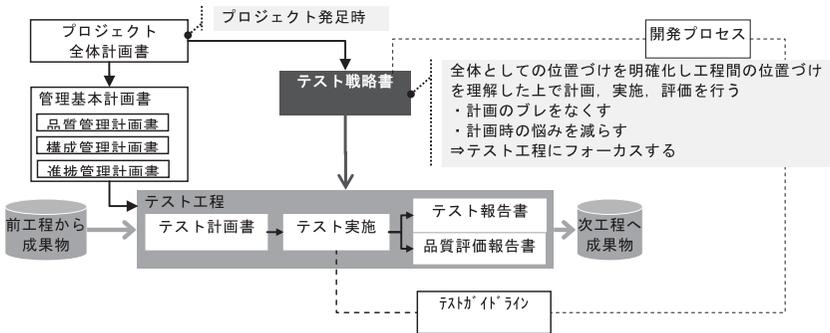


図11 テスト戦略書の位置づけ

6.4 “現行通り”への対応

3.1節で述べた通り、ProjectAIの目的はable-Dの「ITコスト構造の改革」「システムの拡張性・柔軟性の確保」「迅速なサービスの提供」であり、アプリケーション機能のゴールは「able-Dアプリケーション機能と同等」であった。

しかしながら、約30年前の初期稼働から改修を重ねてきているable-Dには整備された機能仕様書がなかったため、現行業務同等の開発ゴールを達成するためには、開発計画策定段階で従来の開発技法（ウォーターフォール型）をベースとした工夫を組み込んでいくことが必要だった。

6.4.1 機能要件の定義とリスク

現行業務を成り立たせている機能要件については、プログラムコードからリバースしてそのベースを作成した。人によるプログラムコードの読み込みからのリバースであるため、担当したメンバーの解釈によっては、異なったアプリケーションルールまたは漏れてしまうルールがでてしまうリスクがあることを想定し、後段の工程でのリスク対策を組み込む必要があった。

6.4.2 早期の品質検証

機能要件のリスクを早期に検知することを目的に、開発機能をグループ分けした中で、最初のグループとしてまず主要機能から開発と検証を行ったところ、定義した機能要件の範囲で、機能仕様は満たしているという機能品質は確認できたものの、機能と機能を組み合わせた業務仕様としては誤り・抜け漏れが散見される結果となった。

6.4.3 現行比較フェーズの設定

ベース仕様開発及び現行改修取り込みでの仕様開発を終えたところで、現行と同等の仕様レベルとなることから、その後のフェーズとして、現行比較フェーズを設定し、業務シナリオに沿った現行業務との比較を徹底的に実施し、仕様差異を埋めていく計画とした。

業務機能の比較については、機能テスト工程の最終ステップとして、業務シナリオに沿った機能単位での確認を実施した。オンライン機能に関しては、入力媒体を現行と共有しているため、用意した業務シナリオに沿った確認ケースの入力を新旧端末で実施し、結果を比較して確認した。またバッチ機能についても新旧システムでバッチ走行した結果を比較して確認した。

現行比較フェーズとして、全機能を対象とした業務シナリオの実行に9ヶ月を想定し、不具合発生率は従来の機能テスト工程同等の率を想定した。

6.5 移行～稼働後臨戦

移行から新システム稼働直後の臨戦（障害対応に備えた要員待機）については、システム都合で運航業務に影響を及ぼさないこと、万が一影響があったとしても最小限にとどめることを目的とした準備を整えておく必要があった。

移行に関しては、計画していた作業を決められた時間内に確実に実施する作業であるため、手順に沿ったりハーサルを複数回繰り返し実施し、想定外の事象に対する追加手順化を含め、作業の正確性を高める計画とした。

稼働後臨戦は、計画段階で、実際に発生する障害の量、質ともに推定することは困難であり、いかなる状況にも対応可能な体制として24時間・延べ500名の体制及びワークフローを準備しておく必要があった。また、本番稼働直後に想定される緊迫した状況の中で、早く、確実な障害対応を実施していくために、各メンバーが役割に応じたタスクを一定品質で実行し、他チームメンバーともスムーズに連携できるワークフローを定着させる必要がある。そのために、全メンバー参加の訓練という形で稼働後臨戦時のシミュレートを複数回繰り返し、安定した障害対応の運用を確立した。

障害対応の基本方針は以下のとおりである。また、稼働後臨戦の体制を図12に示す。

- 通常運用でのITIL規約に即した役割と運用とする。

- 障害対応方法としては、運用制限、DB 修正、プログラム修正、環境修正、対応不要のいずれかの範囲で対応する。
- 障害による業務影響度を五つのレベルに区分し、対応優先度を設定する。
- 障害連絡受付⇒解析⇒対応方針決定⇒対応⇒リリース判定⇒リリース手続き⇒サービスのワークフローに沿った対応とし、一連のステータスについてはプロジェクト全体で共有できる仕組みを準備する。

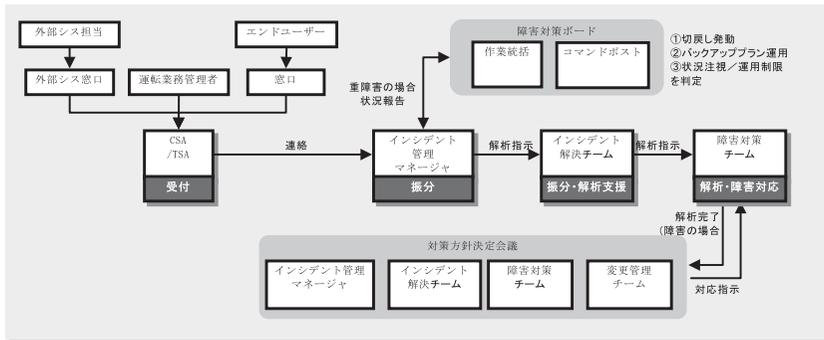


図 12 稼働後臨戦体制

7. おわりに

様々な事業改革に伴う追加改修を入れつつ、約 30 年に渡り社会基盤としての堅牢かつ安定した運用を提供してきた旧システム able-D を、仕様をそのままに全面的に新アーキテクチャの上で構築するにあたり、アプリケーション・インフラともに様々な難問に直面してきた。

その都度、プロジェクトでは会社の垣根を外した協力体制の下で、当初計画で定めたゴールと基本方針をぶれさせることなく課題対策検討を繰り返して難問を乗り越え、本番稼働を迎えることができた。

プロジェクトとともに運営してきた全日本空輸株式会社様、ANA システムズ株式会社様に多大なるご支援を頂いたことは言うまでもないが、この大プロジェクトに最後まで協力頂いた数々のパートナー会社の方々、およびベンダーコンソーシアムで技術支援を頂いた日本ヒューレット・パカード株式会社様、日本オラクル株式会社様、株式会社日立製作所様のご協力あつての本番稼働であった。本稿を借りて深く感謝を申し上げたい。

* 1 ProjectAI : ANACore 開発のプロジェクト名 (Ana passenger system Innovation)。

執筆者紹介 金子 時 生 (Tokio Kaneko)

1986年4月 日本ユニバック (当時) 入社. システムエンジニアとして運輸・ロジスティクス関連を担当. 金融部門の勘定系ならびに証券系の開発を担当・大規模勘定系オープンシステム構築のPMを経て, 2008年1月より国内旅客システムマイグレーションを担当. 現在に至る.

