

短期大規模開発におけるシステム分割構築戦略

System Partitioning Construction Strategy in Short-term Large-scale Development

堀 内 衛

要 約 システム開発は、顧客へ提供する機能要件と信頼性・効率性等の非機能要件を定義し、開発するアプリケーション・プログラムと稼働に必要となるハードウェアとミドルソフトウェア等の見積りによってその規模を確定する。

アプリケーション・プログラム個々の規模を合計したシステム全体が大規模となっても、いくつかの小さなサブシステムに分割し、それぞれを比較的独立したサブプロジェクトとしてチーム編成することで期間短縮を可能にする。関連性を持つ複数のサブシステム間のインタフェースの境界と責任分界点を早期に確定することにより、各サブシステムは、サブプロジェクトとして疎結合な開発を進めることができるようになる。

分割したサブシステム・サブプロジェクトが独立して開発を並走することのリスクは、操作性やアーキテクチャが不統一になることである。システムテストや受入テストでの大きな手戻り発生を予防するためには、基本設計の段階から標準化関連の規約を整備すること、詳細設計レベルで設計構造を強化すること、チェックリストを作成してアーキテクチャの不統一を予防・検出・対策することが有効である。

Abstract In the system development, development scale is determined by defining functional requirements and non-functional requirements such as reliability and efficiency to be provided to the customer, and by estimating application programs to be developed and the hardware and middle software required for running.

Even if a scale of the whole system became oversize, development period can be shortened by dividing the system into several smaller sub-systems and by organizing the team as a relatively independent sub-project. By determining the boundary of interface and burden sharing between sub-systems that are relevant at an early stage, each sub-system will be able to develop as a loosely coupled sub-project.

Risk is that when divided sub-systems or sub-projects develop system in parallel and independently, the operability and architecture become disunity. In order to prevent the system test and acceptance test from a large setback, it is good to establish the rules for standardization from the basic design phase, and to strengthen the design structure at the specific design level, and to create checklists in order to prevent architecture from disunity.

1. はじめに

日本郵便株式会社（以降、日本郵便）の取引管理システムとは、ゆうパックや郵便の引受、切手やはがき等の販売・在庫等を一元的に管理するシステムである。取引管理システムには、郵便局の窓口で使用する端末、内務事務員がオフィス等で使用する端末、および外務員が携帯して使用する端末から情報が入力される。そのため、利用者は全国で約 21,000 局の職員と非常勤職員をあわせた約 38 万人である。郵便事業の根幹を支えるミッションクリティカル・基

幹系システムである。

この取引管理システムを稼働させる「取引管理システム（後続稼働）プロジェクト」（以降、本プロジェクト）の特徴は、10,000人月を超える大規模システム開発を、ピーク時1,000人体制、約2年という短期間で完成させることである。

本稿では、取引管理システムという大規模・短期間での業務アプリケーション開発の実現について、2章で背景とシステムの概要、3章でサブシステム分割と五月雨開発、4章でアーキテクチャ不統一の予防と除去、5章で考察（まとめ）を述べる。

2. 背景とシステムの概要

本章では、日本郵便の次世代郵便情報システムの背景と取引管理システムの位置づけや概要を説明する。さらにシステム開発の規模と工期の関係について述べる。

2.1 次世代郵便情報システムの背景と二段階開発の位置づけ

日本郵便の次世代郵便情報システムは、システムの機能改善と操作性向上を図るとともに、配達予告メールの導入や、お客さまの利便性向上を目指し、次世代郵便情報システム（先行稼働）と次世代郵便情報システム（後続稼働）の二段階で開発を進めた。それぞれの稼働時期と位置づけを表1に示す。

表1 次世代郵便情報システムの二段階開発の位置づけ

| 段階開発 | 稼働時期 | 位置づけ |
|--------|----------|---|
| (先行稼働) | 2013年4月 | 宅配便事業の統合により発生したゆうパック処理の二重管理を解消すること 取扱方法を荷物の追跡情報と運賃を紐付けて管理する方式に統一すること |
| (後続稼働) | 2015年10月 | 損益改善、サービス改善などの施策に柔軟に対応できる次世代システムを構築すること システム基盤整備等の生産性を向上させること |

2.2 取引管理システムの概要

取引管理システムは、日本郵便の次世代郵便情報システムの中で、フロント基幹系システム（窓口端末・携帯端末・追跡・Web ゆうびん・e内容証明等）と、バック基幹系システムとを数多く接続し連携を行う、日本郵便の次世代郵便情報システムの中核システムに位置づけられる。取引管理システムの構成を図1に示す。

取引管理システムは、郵便事業を支える中核として業務領域が広く、他の基幹系システムとの連携数が多いため、他の基幹系システム単位に実施されるシステム運用やメンテナンスに協調できる高い運用性が求められる。さらに社内基幹系システムとしては、利用者数とトランザクション数がともに多く、高いオンライン処理性能とバッチ処理性能が求められる。

2.3 システム開発の規模と工期の関係

取引管理システムのような大規模システムの開発であっても、十分な開発期間を確保できる条件（顧客要件と合意を含む）が整う、もしくは大規模（10,000人月）を分割したサブシステムの要件（新規機能・機能数・操作性・外部インターフェース・性能等）の大きさを均等にできる条件が揃う場合、その開発の難易度とリスクは低減できる。



図1 取引管理システムの構成図

日本情報システム・ユーザー協会（JUAS）では、毎年ユーザ企業のプロジェクトを調査した「ソフトウェアメトリックス調査」^[1]を発表している。その中で、プロジェクトの規模と工期の関係を分析し標準工期を求め、適正工期の判定方法を導いている。2013年の調査結果から導き出された標準工期は、「投入人月の立方根の2.58倍」であった。また、「ソフトウェアメトリックス要点ハンドブック」^[2]では、この標準工期との乖離が0～25%短い場合は、「優秀なプロマネやSEを集める、ユーザー（顧客）の協力を仰ぐ等の対策が必要」、30%以上短い場合は、「開発規模を見直す、工期を再設定し長くする等の対策が必要」としている。JUASの2013年調査結果から、標準工期とそれを30%短くした限界の工期（以降、短縮工期と呼ぶ）をシステム規模別に算出したものを表2に挙げる。

表2 標準工期の比較

| システム規模 | 投入人月 | 標準工期（計算式） | 短縮工期（限界30%短縮） |
|--------|----------|---|---------------|
| 小 | 100人月 | 12.0ヶ月（ $= 2.58 \times \sqrt[3]{100}$ ） | 8.4ヶ月 |
| 中 | 200人月 | 15.1ヶ月（ $= 2.58 \times \sqrt[3]{200}$ ） | 10.6ヶ月 |
| | 500人月 | 20.5ヶ月（ $= 2.58 \times \sqrt[3]{500}$ ） | 14.3ヶ月 |
| 大 | 1,000人月 | 25.8ヶ月（ $= 2.58 \times \sqrt[3]{1000}$ ） | 18.1ヶ月 |
| | 2,000人月 | 32.5ヶ月（ $= 2.58 \times \sqrt[3]{2000}$ ） | 22.8ヶ月 |
| | 2,300人月 | 34.1ヶ月（ $= 2.58 \times \sqrt[3]{2300}$ ） | 23.8ヶ月 |
| | 3,000人月 | 37.2ヶ月（ $= 2.58 \times \sqrt[3]{3000}$ ） | 26.0ヶ月 |
| | 10,000人月 | 55.6ヶ月（ $= 2.58 \times \sqrt[3]{10000}$ ） | 38.9ヶ月 |

（期間については小数点以下第二位を四捨五入して算出）

表2によると、本プロジェクト（10,000人月）の標準工期は55.6ヶ月（4年超）と導出される。本プロジェクトに与えられた24ヶ月は、JUASが提言する短縮工期（限界30%短縮）38.9ヶ月よりさらに短い。しかしながら、開発期間2年間は顧客からの必要要件である。この

2年間（24ヶ月）で「標準的に完遂できる」投入人月は1,000人月弱であり、「完遂できる可能性がある」短縮工期からみた投入人月は約2,300人月である。本プロジェクトを成功させるためには、投入人月を削減する対策が必要であった。

3. サブシステム分割と五月雨開発

本章では、本プロジェクトのサブシステム分割とそれぞれのサブシステムの特徴、そして五月雨開発について紹介する。

3.1 なぜサブシステム分割をするのか

前章で述べたとおり、投入人月に対して開発期間が短すぎることを理由に、開発対象とする機能を1,000～2,300人月まで削減・廃止することは、顧客業務が成立せず、受け入れられることはない。また、次世代郵便情報システム（後続稼働）の各システムが足並みをそろえて稼働する必要があるため、開発期間を延ばすということも選択できない。そのため、大きなシステムの一部を構成する、より小さな単位のサブシステムに分割することを選択した。

小規模もしくは中規模に分割したサブシステムを担当する体制を作り、それらがサブプロジェクトとして疎結合かつ独立した開発を進める。サブシステム分割のタイミングは、システム開発対象の機能一覧（画面一覧、帳票一覧、バッチ一覧等）を確定する要件定義後半（もしくは基本設計前半）が一般的である。サブシステム分割のアプローチは、顧客利用部門による業務の単位とシステム機能の特性（類似性と関連性）から、サブシステム定義を行うのが通常である。業務の単位とサブシステムの単位を同じにする事由は、顧客（業務部門と情報システム部門）と開発ベンダともに、仕様管理が合理的であることによる。

3.2 取引管理システムのサブシステムの特徴

本プロジェクトも、業務単位にサブシステムを分割している。各サブシステムの特徴を考察し、それぞれの規模と工期の妥当性を評価した。各サブシステムの投入人月、機能数、工期の妥当性と特徴（1日1000万件超の画面入力・外部IF入力トランザクション性能要求）を表3に示す。なお、表中の投入人月は厳密な見積りや実績によるものでなく、規模感を示している。

サブシステム分割後の投入人月から導出する標準工期が24ヶ月を超過するのは、引受と実績と在庫と業務共通の4サブシステムである。しかしながら「完遂できる可能性がある」短縮工期は、引受26.0ヶ月・実績17.4ヶ月・在庫23.1ヶ月・業務共通17.4ヶ月となり、顧客要件24ヶ月を超過するサブシステムは引受（2ヶ月超過）のみとなる。このサブシステム分割により、独立したサブプロジェクトとして投入人月と「完遂できる可能性がある」短縮工期を目標に開発を進めることが可能になった。

各サブシステム内の設計開発とテスト終了後、他サブシステムとの連携テストと検証を行い、取引管理システム全体としての品質を確保する。その後の統合テスト・運用テスト（6ヶ月）にて、他システムとの連携テストと検証を行い、日本郵便の次世代郵便情報システムとしての品質を確保する。その期間を確保するために、本プロジェクト（システムテストを含む）の工期は実質18ヶ月となる。次節以降では、「完遂できる可能性がある」としたものの、工期18ヶ月では実現性に乏しい引受と在庫サブシステムにおける対策および他サブシステムとの境界・責任分界点について説明する。

表3 サブシステムの妥当性

| サブシステム | 投入人月 (規模感) | 機能数 (画面/帳票/ バッチ・IF/API 他) →平均投入人月/機能 | 上段：標準工期 (判定)/短縮工期 (判定) (○ 18ヶ月以下, △ 18～24ヶ月, × 24ヶ月超) 下段：特徴 |
|--------|---------------|--|---|
| 引受 | 3,000 人月 | 203 (35/57/56/55) → 14.8 人月/機能 | 37.2 ヶ月 (×)/26.0 ヶ月 (×) 性能要求 |
| 実績 | 900 人月 | 168 (35/44/79/10) → 5.4 人月/機能 | 24.9 ヶ月 (×)/17.4 ヶ月 (○) 性能要求 |
| 計器 | 400 人月 | 36 (11/10/10/5) → 11.1 人月/機能 | 17.3 ヶ月 (○)/12.1 ヶ月 (○) |
| 販売 | 900 人月 | 90 (20/12/26/32) → 10.0 人月/機能 | 24.9 ヶ月 (△)/17.4 ヶ月 (○) 性能要求 |
| 在庫 | 2,100 人月 | 226 (62/34/44/86) → 9.3 人月/機能 | 33.0 ヶ月 (×)/23.1 ヶ月 (△) 性能要求 |
| 選挙 | 300 人月 | 37 (17/11/4/5) → 8.1 人月/機能 | 17.3 ヶ月 (○)/12.1 ヶ月 (○) |
| 精算 | 900 人月 | 91 (14/13/25/39) → 9.9 人月/機能 | 24.9 ヶ月 (△)/17.4 ヶ月 (○) 性能要求 |
| 手数料 | 300 人月 | 62 (5/8/43/6) → 4.8 人月/機能 | 17.3 ヶ月 (○)/12.1 ヶ月 (○) |
| 現金 | 300 人月 | 41 (5/4/16/16) → 7.3 人月/機能 | 17.3 ヶ月 (○)/12.1 ヶ月 (○) |
| 業務共通 | 900 人月 | 127 (47/10/26/44) → 7.1 人月/機能 | 24.9 ヶ月 (×)/17.4 ヶ月 (○) |
| 全体 | 10,000 人月 | 1,081 (251/203/329/298) → 9.3 人月/機能 | 55.6 ヶ月 (×)/38.9 ヶ月 (×) |

3.3 引受サブシステムの特徴と他サブシステムとの境界・責任分界点

引受サブシステムは、郵便局職員の操作性向上と誤入力防止のために、クライアントサーバシステム*1のような高い操作性を提供している。jQuery/Ajax 関数*2を採用すれば、単純な Web ページよりも操作性と自由度を高めることが可能になる。利用者視点でメリットがあるものの、システム開発視点では仕様定義の量は拡大する。「引受を行う画面」1機能の基本設計書ページ数は、中小規模の計器・選挙・手数料・現金の1サブシステム全量と同等となり、顧客ヒアリングとレビューに多くの時間を必要とする。基本設計書ページ数の増加は、詳細設計書ページ数とプログラムステップ数やテスト量に比例する。標準画面と比較すると30倍を超える量になる。引受サブシステムには「引受を行う画面」以外にも、標準画面と比較して20倍を超える「配達を行う画面」も提供している。さらに、引受サブシステムは、画面入力情報だけでなく、窓口端末と携帯端末から連携される引受電文も一元的に管理している。

システム要件と規模が大きい引受サブシステムの開発を遅延させないこと、後方サブシステム(実績・手数料・精算)へ遅延が連鎖するリスクを低減する対策が必要であった。

3.3.1 引受サブシステムの細分化

引受サブシステムの開発を遅延させない対策として、五つの機能群に細分化した。各機能群の特性を考察し、それぞれの規模と工期の妥当性を評価した。表4に示す。

表4 引受サブシステムの機能群の妥当性

| 機能群 | 投入人月 (規模感) | 機能数 (画面/帳票/ バッチ・IF/API 他) →平均投入人月/機能 | 上段：標準工期 (判定)/短縮工期 (判定) (○ 18ヶ月以下, △ 18～24ヶ月, × 24ヶ月超) 下段：特徴 |
|----------------|---------------|--|---|
| 引受を行う 画面 | 500人月 | 8 (7/0/0/1) → 62.5人月/機能 | 20.5ヶ月 (△)/14.3ヶ月 (○) 性能要求 |
| 引受 (その他) | 800人月 | 69 (4/29/26/10) → 11.6人月/機能 | 24.0ヶ月 (△)/16.8ヶ月 (○) 性能要求 |
| 配達を行う 画面 | 300人月 | 14 (13/0/0/1) → 21.4人月/機能 | 17.3ヶ月 (○)/12.1ヶ月 (○) 性能要求 |
| 配達 (その他) | 700人月 | 65 (5/28/26/4) → 10.8人月/機能 | 22.9ヶ月 (△)/16.0ヶ月 (○) 性能要求 |
| 引受共通 | 700人月 | 49 (6/0/4/39) → 14.2人月/機能 | 22.9ヶ月 (△)/16.0ヶ月 (○) 共通 API, 子画面 |
| 引受サブ システム全体 | 3,000人月 | 203 (35/57/56/55) → 14.8人月/機能 | |

この細分化により、すべての機能群は、工期18ヶ月で妥当性ありとなり、それぞれを担当するチームを編成し、設計開発を進めることが可能になった。

3.3.2 後方サブシステムの遅延リスク対策

後方サブシステムの実績・手数料・精算へ遅延が連鎖するリスク対策として、境界・責任分界点となる「引受」「配達」テーブルについて、基本設計後半から詳細設計前半にて、全商品毎のデータベース更新の詳細仕様を確定し、後方サブシステムに公開した(図2)。これにより、後方の実績・手数料・精算サブシステムは、引受サブシステム全体(および各機能群)の詳細設計完了を待つことなく、独立して詳細設計と製造と単体テストを進めることが可能になった。

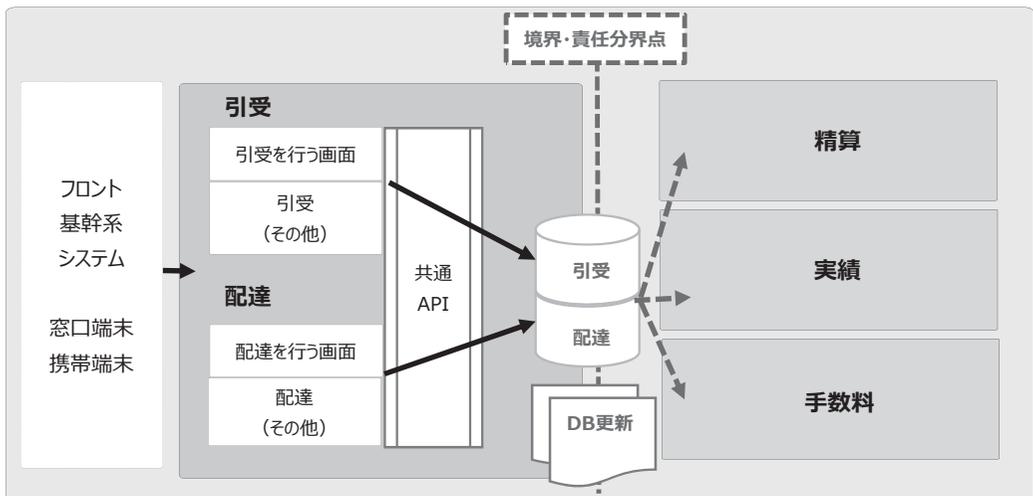


図2 引受と後方サブシステムの責任分界点

3.4 在庫サブシステムの特徴と他サブシステムの境界・責任分界点

在庫サブシステムは、引受サブシステムと同様に、大規模であり、窓口端末と携帯端末から連携される在庫移動電文を一元的に管理している。規模が大きい在庫サブシステムの開発を遅延させないこと、関連するサブシステム（販売・精算）へ遅延が連鎖するリスクを低減する対策が必要であった。

3.4.1 在庫サブシステムの細分化

在庫サブシステムの開発を遅延させない対策として、六つの機能群に細分化した。各機能群の特性を考察し、それぞれの規模と工期の妥当性を評価した。表5に示す。

表5 在庫サブシステムの機能群の妥当性

| 機能群 | 投入人月 (規模感) | 機能数 (画面/帳票/ バッチ・IF/API 他) →平均投入人月/機能 | 上段：標準工期 (判定)/短縮工期 (判定) (○ 18ヶ月以下, △ 18～24ヶ月, × 24ヶ月超) 下段：特徴 |
|----------------|---------------|--|---|
| 補充請求 | 500人月 | 30 (15/5/10/0) → 16.7人月/機能 | 20.5ヶ月 (△)/14.3ヶ月 (○) |
| 受払 | 300人月 | 30 (8/8/14/0) → 10.0人月/機能 | 17.3ヶ月 (○)/12.1ヶ月 (○) 性能要求 |
| 切手箱 | 500人月 | 29 (9/2/10/8) → 17.2人月/機能 | 20.5ヶ月 (△)/14.3ヶ月 (○) 性能要求 |
| 仮出 | 200人月 | 19 (10/5/4/0) → 10.5人月/機能 | 15.1ヶ月 (○)/10.6ヶ月 (○) 性能要求 |
| 点検等 | 200人月 | 23 (15/6/1/1) → 8.7人月/機能 | 15.1ヶ月 (○)/10.6ヶ月 (○) |
| 在庫共通 | 400人月 | 92 (3/6/9/74) → 4.3人月/機能 | 19.0ヶ月 (○)/13.3ヶ月 (○) 公開 API, 共通 API, 子画面 |
| 在庫サブ システム全体 | 2,100人月 | 223 (60/32/48/83) → 9.0人月 | |

引受サブシステムと同様に、この細分化により、すべての機能群は、工期18ヶ月で妥当性ありとなり、それぞれを担当するチームを編成し、設計開発を進めることが可能になった。

3.4.2 関連サブシステムの遅延リスク対策

密連携するサブシステムの販売・精算へ遅延が連鎖するリスク対策として、サブシステム間の境界・責任分界点となる販売・精算サブシステム向けに「公開 API」を提供することとした。この API の仕様を、基本設計後半から詳細設計前半にて確定し公開した (図3)。これにより、販売と精算サブシステムは、在庫サブシステム全体 (および各機能群) の詳細設計完了 (物理 DB 確定) を待つことなく、独立して詳細設計と製造と単体テストを進めることが可能になった。

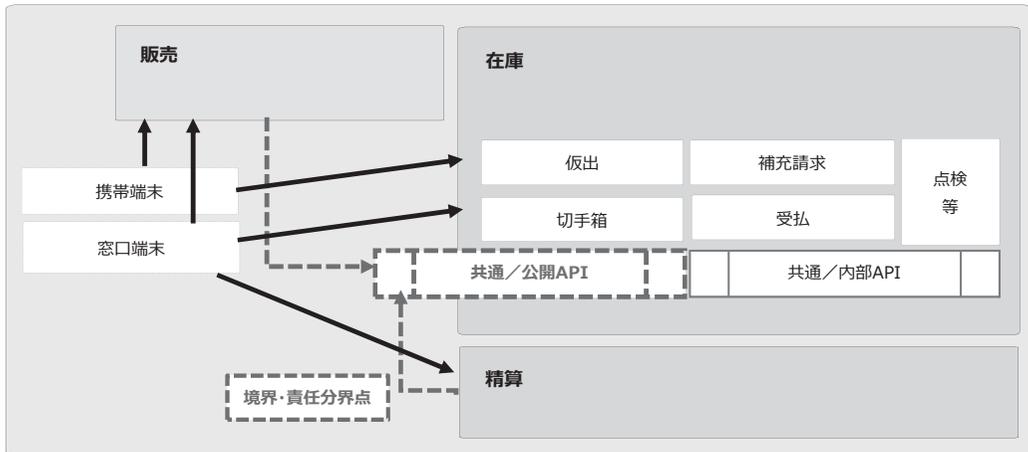


図3 在庫と関連サブシステム責任分界点

3.5 五月雨開発

取引管理システムは、郵便事業の根幹を担う基幹系システムであることから、全ての基本設計成果物について、情報システム部門に仕様確認とレビューを依頼した。これは、情報システム部門が、旧システムを含めた取引管理システムの一番の理解者（システム仕様の指示・承認者）であり、日本ユニシスの設計開発要員が情報システム部門からシステム仕様をヒアリングしレビュー指摘とその対応により本システムを理解すること、理解した仕様に基づいて設計情報を成果物へ落とし込むことが最も重要であることによる。また、後々の詳細設計・製造・単体テスト・結合テスト・システムテストにおける品質確保のため必須と判断したためである。

前節までのサブシステム分割と機能群への細分化により、工期18ヶ月で妥当性ありという条件は整った。しかしながら、開発するアプリケーションそれぞれは不均等である。全ての成果物作成を一斉に開始し、一斉に顧客レビューまで終了することは不可能である。とくに特性（操作性・新規性・性能・他システム連携等）が複雑かつ規模の大きい機能は課題が多く発生する。さらに関係する他システム確認と調整に時間をかける必要がある。

そのためシステム特性が複雑で規模の大きい引受と在庫サブシステムの基本設計完了時期を最遅（2014年3月）とするマイルストーンを設定した。基本設計から結合テストまでの五月雨^{*3}開発スケジュール（2014年1月～12月）を表6に示す。

五月雨開発の目的は、常に最遅となる大規模サブシステム（引受・在庫）の遅延拡大リスクを低減・予防するために、小規模サブシステム（計器・選挙）が先行して次工程を進めることである。次工程成果物の作成とレビュープロセスを回して、作業の品質や開発プロセスの問題を抽出し対策立案する。大規模サブシステムが次工程に着手する前までに、その対策をガイド・作業手順・チェックリスト等に反映することにより、引受と在庫サブシステムの遅延拡大を低減させることを可能にした。

作成済み成果物と手戻り率は増加する。また、小規模開発であれば、手戻り工数も小さくリカバリは難しくないが、システム規模が大きくなるほど、類似機能や共通処理へと連鎖し、手戻り工数は指数的に拡大する。その例を表7に示す。

表7 規模と手戻り工数の例

| システム規模 | 工数 | 手戻り「小」 (下流工程で不統一を組み込み) | | 手戻り「大」 (上流工程で不統一を組み込み) | |
|--------|----------|---------------------------|-----------|---------------------------|-------------|
| | | 手戻り率(例) | 手戻り工数 | 手戻り率(例) | 手戻り工数 |
| 小規模 | 100人月 | 1% | 1人月 | 5% | 5人月 |
| 中規模 | 500人月 | 1～2% | 5～10人月 | 5～10% | 25～50人月 |
| 大規模 | 1,000人月 | 2～4% | 20～40人月 | 10～15% | 100～150人月 |
| | 10,000人月 | 4～8% | 400～800人月 | 15～30% | 1500～3000人月 |

4.2 アーキテクチャ不統一の予防と除去

アーキテクチャ不統一リスクの予防には、新しい技術を取り入れる必要はなく、標準化の規約を網羅的に定義・強化すればよい。本プロジェクトでは、要件定義と基本設計前半の段階で、画面と帳票の規約や、画面仕様書と帳票仕様書のガイドとサンプル設計書を整備した。

また、新規着任者は、規約やサンプル設計書を確認し、それらに準拠した基本設計書を作成したつもりでも、必ず漏れや誤りは発生すると考えるべきである。その漏れや誤りの混入を設計担当者とレビュー自身が点検し、除去するためのチェックリストを整備した。それでも、短期開発では、すり抜け（漏れや誤りを除去する活動でも漏れる）が必ず発生するため、専門部隊（標準化・性能）による説明会と技術検討質問会、第三者レビューを繰り返し実施した。

特に、取引管理システム（先行稼働）で発生したシステム障害の再発防止のために、基本設計の段階からシステムの設計構造を考慮すること、特に性能懸念のある重点機能を特定し上流から下流工程を通して十分な検証と対策を行う個体管理をすること、設計ガイドとチェックリストに検討すべき事項と留意点の盛り込みを行った。本プロジェクトの標準化関連規約の中から画面規約の目次を表8、画面共通仕様チェックリストの一部を表9に示す。

5. 考察（まとめ）

大規模でも小規模でも、要員が所属するチーム（10人～20人単位のサブプロジェクト）が短い期間で設計開発できる規模は限られている。そのため、工期（顧客要件）に見合う担当規模まで分割・細分化を行う必要がある。各サブプロジェクトが疎結合・独立的に開発を進めるために、境界・責任分界点をできるだけ早期に確定させること、小規模サブプロジェクトが、大規模サブプロジェクトの遅延予防のために、次工程を先導する五月雨開発が有効である。

また、アーキテクチャの不統一は必ず埋め込まれるという前提のもと、早期に標準化関連規約を整備すること（予防）、チェックリストと第三者検証（除去）を繰り返すことが、特に短期かつ新規要員が多い開発プロジェクトには必須である。

短期開発であっても、仕様を定義する基本設計の顧客レビューを省略することは、後々のシステムテストや受入テストへと問題を先送りするリスクとなる。本プロジェクトのような、大規模かつ短期開発では、基本設計を全件顧客レビュー・承認を得ること（ベースライン＝正解を固めること）が最も重要と考えている。

表 8 画面規約の目次

| | | | | | |
|--------|------------------|--------|----------------------|---------|-----------------------|
| 1 | 前提 | 5 | 例外画面パターン | 11 | HTMLコーディング基準 |
| 1.1. | 採用するHTMLのバージョン | 5.1. | 例外画面 | 11.1. | 準拠HTML |
| 1.2. | 対象ブラウザ | 5.1.1. | 入力チェック例外ダイアログ | 11.2. | コピーライト |
| 1.3. | OSのバージョン | 5.1.2. | 業務例外ダイアログ | 11.3. | 言語コード |
| 1.4. | ブラウザのセキュリティ設定 | 5.1.3. | 例外ダイアログの再表示 | 11.4. | 文字コード |
| 1.5. | 本ガイドラインの影響範囲 | 5.2. | 共通例外画面 | 11.5. | キャッシュの抑止 |
| 2 | 画面構成 | 5.2.1. | 業務共通例外画面 | 11.6. | HTMLフォーマット |
| 2.1. | 標準スタイルシート | 5.2.2. | システム共通例外画面 | 11.7. | 特殊文字の記述 |
| 2.1.1. | フォント | 5.2.3. | 認証エラー画面 | 11.8. | フォーム |
| 2.1.2. | 背景色 | 6 | 画面設計基準 | 11.9. | イメージ |
| 2.1.3. | オートコンプリート | 6.1. | ディスプレイの設定 | 12 | 共通フレーム |
| 2.2. | 画面レイアウト | 6.2. | ブラウザ(ウィンドウ)の設定 | 12.1. | 共通フレームの実装 |
| 2.2.1. | 全体イメージ | 6.3. | ブラウザのスタイル | 12.2. | 業務画面の共通フレーム |
| 2.2.2. | 画面構成エリア | 6.4. | 基本画面 | 12.3. | 子画面の共通フレーム |
| 2.3. | ボタンレイアウト | 6.5. | 顧客名・ロゴ | 13 | 項目タイプ別使用指針 |
| 2.3.1. | ボタンの名称 | 6.6. | ウィンドウ | 13.1. | 文字列関連 |
| 2.3.2. | ボタンの色 | 6.7. | 文字 | 13.1.1. | 電話番号・郵便番号 |
| 2.3.3. | ボタンの並び順 | 6.7.1. | コンテナ | 13.1.2. | コードと名称 |
| 2.3.4. | ボタンの配置 | 6.7.2. | アイデンティティエリア | 13.2. | 数値関連 |
| 2.4. | 画面装飾 | 6.7.3. | パンくずエリア | 13.2.1. | 金額 |
| 2.4.1. | サブタイトル,表・見出しセルの色 | 6.7.4. | コンテンツタイトルエリア | 13.2.2. | その他数値関連 |
| 2.4.2. | 明細行の色 | 6.7.5. | ナビゲーションエリア | 13.3. | 日時関連 |
| 2.4.3. | 必須入力項目 | 6.7.6. | メッセージエリア | 13.3.1. | 年月日 |
| 2.4.4. | 注釈 | 6.7.7. | コンテンツエリア | 13.3.2. | 時刻 |
| 3 | 親画面と子画面 | 6.8. | 表 | 13.4. | その他 |
| 3.1. | 親画面と子画面の関係 | 6.9. | ボタン | 13.4.1. | パスワード |
| 3.2. | 親画面 | 6.10. | アイコン | 13.4.2. | ファイルアップロード |
| 3.2.1. | 基本構成 | 6.11. | オブジェクト種類 | 14 | 画面部品 |
| 3.2.2. | 対象となる画面パターン | 6.12. | 表示規則 | 14.1. | 共通部品の配置場所 |
| 3.3. | 子画面 | 6.13. | 各項目入力方法 | 14.2. | 画面作成時の注意点 |
| 3.3.1. | 基本構成 | 6.14. | テキストボックス入力時のIME状態指定 | 14.2.1. | 部品の配置方法 |
| 3.3.2. | 子画面を使用するケース | 6.15. | 単位表示 | 14.2.2. | コンテナ部品の使用方法 |
| 3.3.3. | 使用に関する注意事項 | 6.16. | 一覧形式 | 14.3. | 画面サイズ |
| 3.3.4. | 一覧形式での明細選択 | 6.17. | メッセージ | 14.4. | 業務画面(spファイル)と画面設計書の管理 |
| 4 | 業務画面パターン | 6.18. | ヘルプ機能 | 14.5. | 提供API |
| 4.1. | メニュー画面 | 6.19. | 機能のグループ化 | 14.5.1. | 共通JavaScript・jQuery |
| 4.1.1. | メニュー画面レイアウト | 6.20. | フォーカス制御 | 14.5.2. | 共通タグファイル |
| 4.2. | サブメニュー画面 | 6.21. | 特記事項 | 15 | 入出力部品 |
| 4.2.1. | サブメニュー画面レイアウト | 7 | 画面遷移 | 15.1. | テキストフィールド |
| 4.3. | 検索画面 | 7.1. | 正常時の画面遷移 | 15.2. | チェックボックス |
| 4.3.1. | 検索画面レイアウト | 7.2. | エラー時の画面遷移 | 15.3. | ラジオボタン |
| 4.3.2. | 入力補助機能 | 7.3. | 子画面から親画面への画面遷移 | 15.4. | ドロップダウンリストボックス |
| 4.3.3. | 入力チェック | 7.4. | 子画面から子画面の呼び出し | 15.5. | テキストエリア |
| 4.3.4. | 検索仕様 | 7.5. | メニュー直下の「戻る」ボタン | 15.6. | テーブル |
| 4.3.5. | 該当データが存在しない場合の対処 | 7.6. | アラート一覧からの遷移 | 15.7. | ボタン |
| 4.4. | 検索結果一覧画面 | 8 | 入出力部品使用基準 | 15.8. | ハイパーリンク |
| 4.4.1. | 検索結果一覧画面レイアウト | 8.1. | テキストフィールド・テキストエリア | 15.9. | ラベル(見出し) |
| 4.4.2. | 検索結果一覧の仕様 | 8.2. | ラジオボタン | 15.10. | hiddenフィールド |
| 4.4.3. | 検索結果一覧画面パターン | 8.3. | チェックボックス | 15.11. | 表示項目 |
| 4.4.4. | 検索結果一覧画面パターン別の特徴 | 8.4. | ドロップダウンリストボックス | 16 | 画面項目の取得方法 |
| 4.5. | 詳細画面 | 8.5. | 日付 | 16.1. | 画面タイトル |
| 4.5.1. | 詳細画面レイアウト | 9 | ボタン | 16.2. | 項目名 |
| 4.6. | 入力(登録・修正)画面 | 9.1. | 標準ボタン | 17 | テストモードについて |
| 4.6.1. | 入力(登録・修正)画面レイアウト | 9.2. | 「前へ」「次へ」ボタン | 17.1. | テストモードの場合の表示 |
| 4.6.2. | 一覧形式の入力 | 9.2.1. | レイアウト | | |
| 4.7. | 確認画面 | 9.3. | 実行ボタン | | |
| 4.7.1. | 確認画面レイアウト | 9.4. | 「トップメニューへ戻る」ボタン | | |
| 4.7.2. | 一覧形式の確認画面 | 9.5. | 「サブメニューへ戻る」ボタン | | |
| | | 9.6. | 参照ボタン | | |
| | | 9.7. | ボタン押下時の注意事項 | | |
| | | 10 | 段組レイアウトの注意点 | | |
| | | 10.1. | 段組適用の考え方 | | |
| | | 10.2. | 段組レイアウトを使用しても良いケース | | |
| | | 10.3. | 段組レイアウトを使用してはいけないケース | | |

表9 画面共通仕様チェックリストの一部

| 大分類 | 中分類 | 項目 | 確認内容 |
|------|----------------------|---|--|
| 画面遷移 | 帳票出力 | 条件入力画面（印刷）→条件入力画面 | ・ 検索条件の入力内容が <u>変わらない</u> こと。 |
| | チェックボックス情報の引継ぎ | 検索・一覧（頁次へ）→検索・一覧（頁前へ）→検索・一覧 | ・ チェックボックスの選択状態が維持されること。 |
| | | 検索・一覧→検索・一覧（検索）→検索・一覧 | ・ チェックボックスの選択状態はクリアされること。 |
| | 戻る遷移 | 検索条件がある画面への戻る遷移 | 検索条件入力画面のみの場合 ・ 前回入力時の検索条件が復元されること。 （disable/readonlyの制御が入力内容に合わせて戻ること。） 検索条件・検索結果一体型の場合 ・ 前回入力時の検索条件と、 前回実行時の検索結果が復元されること。 （disable/readonlyの制御が入力内容に合わせて戻ること。） |
| | | 一覧画面→詳細画面（戻る）→一覧画面 | ・ 再検索を行わず、前回の検索結果が表示されること。 ・ 前回検索時のページが表示されること。 （ページング機能がある場合） ・ ラジオボタンの選択行も復元すること。 ・ 前回検索時のソート順が維持されていること。 |
| | 関連チェックエラー | 更新画面（確認）→更新画面 | ・ 入力内容が変更されないこと。 （disable/readonlyの制御が入力内容に従って設定されていること。） |
| | 単項目チェックエラー | 更新画面（確認）→更新画面 | ・ 入力内容が変更されないこと。 （disable/readonlyの制御が入力内容に従って設定されていること。） |
| 検索 | 検索画面→一覧画面（スキップ）→詳細画面 | 検索結果が2件以上の場合 ・ 一覧画面を表示すること。 ・ 詳細画面にて「戻る」ボタン押下時は、一覧画面に遷移すること。 検索結果が1件のみの場合 ・ 一覧画面での選択をスキップして詳細画面を表示すること。 ・ 戻るボタン押下時は、検索画面に遷移すること。 | |

6. おわりに

大規模かつ短期開発となる取引管理システム（後続稼働）は、多くの困難を乗り越え、2016年2月に本番稼働を開始し、本稿執筆時点（2016年9月）で安定稼働中である。また、2016年10月稼働を目指した次フェーズ開発もシステムテスト・統合テスト・移行リハーサルを計画通り完了した状況である。

今後は、本プロジェクトにて蓄積したプロジェクト管理の経験と知識を活かして、今後計画されるであろう第3次郵便情報システムにも挑戦し、日本郵便のパートナーとして、システム開発と顧客サービス向上に貢献していきたい。

- * 1 クライアント側にクライアントシステムを導入し、クライアントシステムがサーバシステムと通信を行いながら処理していくシステム。クライアントにシステムを導入することによりクライアント側で処理が行え、高い表現力や操作性などを実現できるシステムアーキテクチャ。
- * 2 ページ遷移をさせずにコンテンツを書き換えたり、他の処理を行いながらサーバとデータのやりとりを行う技術。
- * 3 案件や要件が一度で完結せず、複数回に分けて連絡、分割納品すること。

- 参考文献** [1] 「ユーザー企業 ソフトウェアメトリックス調査 【調査報告書】2013年版」, 一般社団法人 日本情報システム・ユーザー協会, 2013年6月
<https://www.juas.or.jp/servey/library/pdf/13swm.pdf>
- [2] 「ユーザー企業 ソフトウェアメトリックス = 要点ハンドブック = 第2版」, 一般社団法人 日本情報システム・ユーザー協会, 2014年3月
http://www.juas.or.jp/servey/library/pdf/swm_hb445042_Ver3.2.2.pdf

執筆者紹介 堀 内 衛 (Mamoru Horiuchi)

1990年日本ユニシス(株)入社。社会公共部門にて新聞製作、通信事業の開発、官公庁・自治体の再構築等に従事。2013年より本稿で紹介した日本郵便取引管理システム(後続稼働)プロジェクトに参画。

