

## プロジェクトの概要と全体構造

# 年金管理システム再構築プロジェクトの 開発事例

金融システムビジネスユニット  
年金業務ソリューション第二構築センター  
年金管理システム再構築プロジェクト

## 1. はじめに

住友信託銀行殿（以下STB殿）の年金管理システムの全面再構築において、当社はアプリケーション開発、本番・開発インフラの構築、ユーザーの集合研修、操作マニュアル作成にいたるまで、広範囲かつ大規模な案件を受注した。本稿ではこのプロジェクトの構築事例を紹介する。

## 2. 年金管理システムの概要

年金管理システムは、信託銀行が企業年金の加入者や受給者の記録を管理し、給付業務を支援するものである。退職者の詳細な情報は各人で異なるため給付額の算定や実際の給付処理には非常に複雑な計算が必要だ。そのため、多くの企業年金は信託銀行に年金の管理や事務処理を委託している。

年金管理システムでは、委託している企業年金を「委託者」と呼び、企業年金から年金管理業務を受託している信託銀行のことを「受託者」と呼んでいる。

企業年金から委託されている業務の主な内容は大きく以下の3つである。

### 1) 従業員データの管理業務

企業に所属しその企業の年金に加入している従業員のデータ管理、年金掛金の計算などの業務。またそれらのデータの照会業務。

### 2) 受給者の管理業務

企業を退職し年金を受けている人およびその権利を持っている退職者の情報の管理や給付額の算定などの業務。またそれらのデータの照会業務。

### 3) 支払業務

支払の指図をもとに、年金や一時金の該当者で支払の対象者に対して、支払額および税額・手数料を算出する業務。

上記の業務以外に、厚生労働省に対して基金の業務運営状態を示す各種資料の作成や、事業所別の統計資料の作成などの業務も含まれる。委託された対象業務範囲は委託者と受託者との契約により決定される（図1）。

## 3. 再構築の背景

STB殿では、代行返上や新型企業年金など企業年金を取り巻く変化、インターネット社会への対応、電子政府への対応など様々な変化に対応するため、年金管理システムの全面的な再構築に着手することとなった。再構築に関しては新技術の導入による開発コスト、運用コストの削減も併せて検討することとなった。以下に再構築を行うきっかけともなった年金管理業務を取り巻く変化を列挙する。

### 3.1 総報酬制

保険料負担の公平性の考え方から、賞与額の年収額に占める割合にかかわらず、同じ年収額であれば同じ保険料額を負担することになった。これにより毎月の給与と同じ保険料率を賞与にも用いて保険料を負担することとなる。この方式は平成15年4月からスタートしている。

### 3.2 新型企業年金

確定給付企業年金法により、新型の確定給付企業年金として、規約型企業年金と基金型企業年金が創設される。規約型は、年金規約に基づき事業主が運営主体となり、基金型は、別法人として設立される基金が運営主体となる。

新型企業年金の導入は、厚生年金基金を持つ企業の代行返上による積立て不足拡大リスクの軽減だけでなく、企業

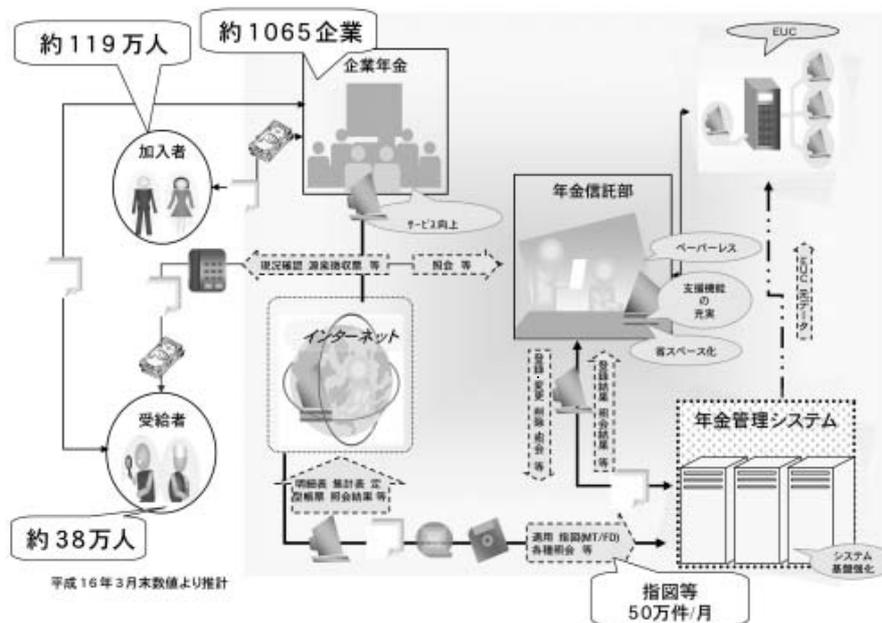


図1 システム概念図

の成熟度や従業員の多様な就労形態、そして事業再編に応じた人事報酬制度の大きな柱として柔軟に機能させていくことができる。そのためにも、既存の企業年金制度からのスムーズな移行措置が重要となっている。

### 3.3 厚生年金基金の代行返上

厚生年金基金による「代行返上」が本格化している。厚生年金基金は大企業などに多い企業年金であるが、国の代わりに公的年金の運用や給付もしていた。その業務を国に返すのが「代行返上」である。厚生年金基金は、主に国内外の株式、債券などに投資して運用されているが、近年高い利率での運用が不可能になり、予定利率を下回る分は企業が負担しなければならない。この負担を回避するのが代行返上の目的であるが、ここ数年で従来の基金から新型企業年金への移行が多発している。

### 3.4 キャッシュバランスプランの導入

キャッシュバランスプランとは、確定拠出型と確定給付型双方の特性を備えた混合型の年金である。確定給付企業年金法において認められた給付設計の一種である。確定給付型年金の1つでありながら、掛金に加えて利息とする期待運用収益を固定化することなく、国債等の客観的指標による利回りに連動させるため、企業にとっては退職給付債務の発生を少なくすることができる。

## 4. 新システムの特徴

今回の再構築プロジェクトでは、3で述べた企業年金を取り巻く変化に対応すると同時に、大量の帳票出力を全

的に電子媒体に移行するペーパーレス化、インターネットを利用した企業年金での入力作業によるデータ登録負荷の軽減化を推進している。さらに、低コストサーバー導入による導入経費とランニングコスト双方の削減を目論んでいる。以下に新システムの特徴を列挙する。

### 4.1 インターネット基盤を利用したのオンライン画面Web化

年金加入者、年金受給者のデータの登録、照会や、Excelを利用したの大量データの投入業務において、ユーザーインタフェースのWeb化を実現した。従来のシステムでは、企業年金と信託銀行に専用端末を設置し専用回線を用意する必要があった。インターネット基盤でのオンライン画面のWeb化によって、これが不要となった。さらに、専用ソフトのインストールが不要なシステムを構築することで、基本的に、PC、インターネットに接続できるすべての場所での利用が可能となった。

### 4.2 ワンタイムパスワードおよびデータ暗号化による万全なセキュリティ対策

オンライン画面のWeb化とインターネット基盤の利用を行うには、堅牢なセキュリティを確保することが絶対条件である。本システムでは、セキュリティ対策の1つとしてワンタイムパスワードを利用したログイン方式を採用した。

企業年金では、信託銀行が配布する専用のキーホルダ型カードによりシステムへのログインを行ってもらうことで本人確認が可能になり、個人データのセキュリティ保護の実現を可能にした。データ暗号化については、STB殿のインターネット基盤ではSSLに対応しており、侵入防止の観

点でもすでにリバースプロキシ技術を導入しているため、本システムでの個別対応は不要であった。セキュリティ強度に関して第三者による脆弱性テストを実施し、特段の重要な指摘事項なしという結果を得ている。

#### 4.3 .Netアーキテクチャを用いた高品質かつ低コストでの開発

初期導入コスト、運用コストの削減を狙い、本システムのハード構成を全面IA（インテルアーキテクチャ）とした。またソフトウェアの基本アーキテクチャに関してはMicrosoftが提唱する .NET Frameworkを全面的に導入した。開発言語には .NET Frameworkの1つであるVB.NETを採用し、開発・保守の生産性向上を目指した。

#### 4.4 電子申請や完全帳票電子化を脱んだ帳票PDF化

政府のe-Japan構想のもと、企業年金が政府に提出する報告書類が電子申請できる制度が導入されており、本システムでは対応が可能となっている。従来、紙ベースで出力/保存されていた帳票のすべてをPDF化または電子ファイル化し、PCからのPDF出力または帳票アウトソース会社からの出力のみとして、完全電子帳票化が実現した。長期保存が必要な帳票についても、紙ベースではなく電子媒体での保存が可能となり、データの保管、検索、印字の効率が向上した。

#### 4.5 ユーザー主体の柔軟なジョブ運営

従来、運営部署で行われていたジョブのスケジュール管理事務機能を業務システムの機能として取り込んだ。たとえば、業務ユーザーが掛金計算日などの処理基準日を設定することが可能である。このように、業務ユーザーが運営作業の大部分を担うことによって柔軟なジョブ運営が可能となった。

さらに、統合監視、JP1によるジョブ管理を導入し、運営作業の多くが自動化できている。これにより、大幅なシステム運用コストの削減およびユーザー主導のシステム運用が可能となった。

## 5. 開発プロジェクトの概要と特徴

### 5.1 開発期間

2001年12月～2004年1月（新型企業年金と厚生年金基金のサービス開始、図2）

### 5.2 開発規模とフェーズ

CAC担当分は約3,600人月。これは、インフラ、業務とも、提案活動からサービス開始（新型企業年金、厚生年金基金）までの工数である。

### 5.3 開発体制と会議体

開発メンバーは、ピーク時に国内で120名以上になった。このような大規模な開発を推進するため以下のようなチーム体制と会議体を設置した。

#### 1) チーム体制

業務チームを8～10のサブシステムに分割し、各チームにリーダーを置いた。その他、インフラチーム、ビルドチーム、環境運営チームを設置した。これらのすべてのチームを統括する管理チームが全体を統括した（図3）。

#### 2) 会議体

STB殿とのコミュニケーションは週1回の進捗ミーティング、月1回の月次進捗ミーティングが基本となる。加えてSTB殿とCACの双方の経営層が参画したステアリングコミッティを開催し、制作レベルから経営層までの認識の一致を目指した。

### 5.4 オフショアでの開発

短期間での大規模開発、低コストでの開発、および新技術の導入をスムーズに行うことを目的として、開発の多くの部分のオフショア開発を決断した。

画面の定義、画面クラスの制作担当には新技術である.NETの開発に強い技術者を多数擁するインドのNIIT社を選択した。200～300の業務画面の内部設計から単体テストまでを、NIIT社の本社のあるインドにて開発した。帳票系は、上海を営業拠点とする当社の子会社である

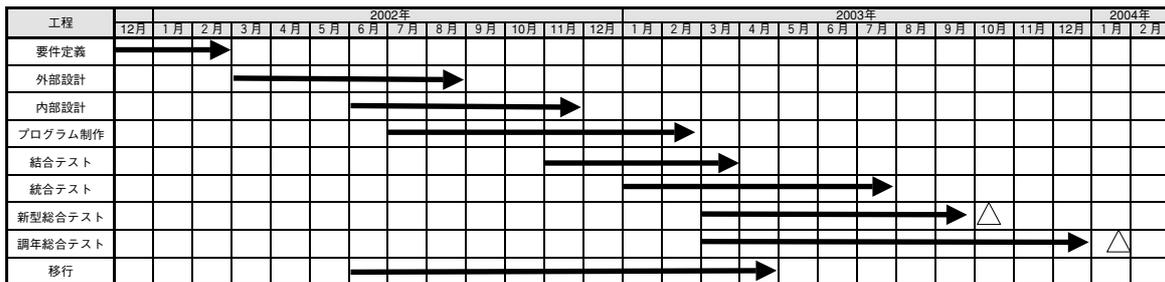


図2 マスタースケジュール

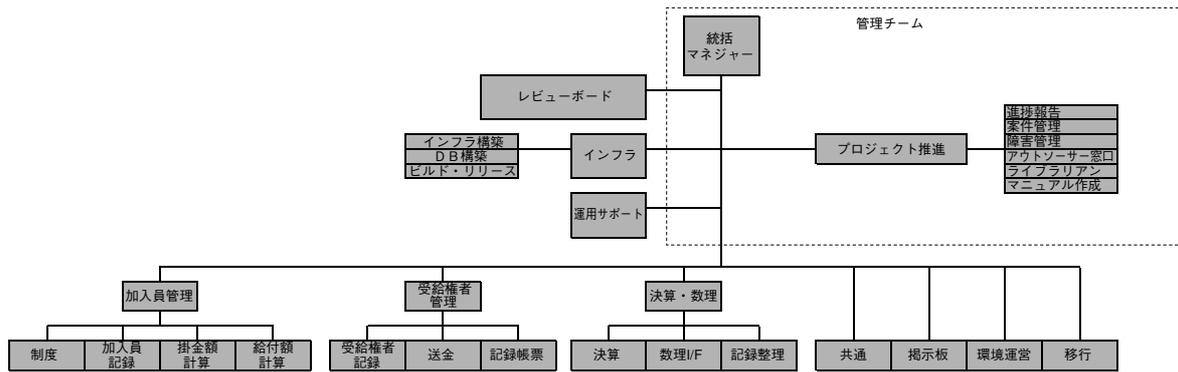


図3 年金管理プロジェクト社内体制図

CAC上海に制作を委託した。400~500種類もの帳票について、帳票の定義、帳票クラスの制作のほぼすべてを中国・上海で行った。帳票系については、開発期間中、日本人技術者を現地に常駐させて日本との連絡を密にとり品質の管理に努めた。インドでのオフショア開発については、月に1回~2回の割合で主要なメンバーが来日して日本の開発部隊と連絡をとりあい開発を進めた。

## 6. 開発における教訓

### 6.1 Microsoftとの連携

新しい技術の導入リスクを最小限に留めるための案として、Microsoftのコンサルティングサービスとプレミアサポートサービスを求めることとした。

当初、.NET Frameworkに精通しているメンバーがほとんどおらず、ピーク時に国内外あわせて170名を超えた開発メンバーへの教育・設計支援・開発支援・問題解決については、Microsoftのコンサルティングによる支援が欠かせないと考えた。要件定義フェーズから.NETの概要を説明するワークショップを全メンバーに対して実施。設計フェーズでは設計者およびプログラマ全員を対象としたVB.NET研修を開催した。同時に、週1回、コンサルティング担当者と当社アーキテクトとの定期的な会議を開催し、技術的な問題、設計手法についての標準化案を策定した。

制作・テストフェーズでは深刻な問題が何度か発生し、その度に問題解決のための検討会議を共同で開催した。Microsoft製品の動作に起因する障害の局面では、コンサルティング以外にプレミアサポートサービスのサービスリクエストを利用して、早期の問題解決を図った。

今回の開発を通して、Windows環境の大規模なシステムを構築するプロジェクトではコンサルティングサービスのサポートは不可欠であると感じた。しかし、効果的なサポートを得るためには、ただ単に発生した問題を投げるのではなく、解決の近道を探るためのヒントを貰えるように

当社側でも道筋を作っておく、というような工夫も大切だと感じた。

### 6.2 開発メンバー教育とサポート

外部設計終了後、内部設計から総合テストまでのほぼすべてのフェーズにわたって、各開発担当者だけでは解決できない様々な問題が発生した。

まず、Microsoftから回答を得られそうな問題は、6.1で記述したコンサルティングに回すことができた。しかし、それ以外にも、開発環境の理解不足からくる問題や、開発手法の選択ミスによる不具合など、開発を進める上での大小さまざまな問題がすべての開発チームで発生した。その際、問題の切り出しと解消の支援を行い他のメンバーと情報を共有するために、内部設計を支援する内部設計レビューア、および制作を支援する実装サポートという専門の支援チームを立ち上げ、問題解決にあたった。

このチームの活動により、各開発担当者が抱えている問題を解決できたと同時に、同様の問題が再発生するなどの時間のロスによる開発の遅延を防いだ。ただし、このチームの投入決定は、計画段階ではなく、多くの問題が発生してからであったため、その分、専門要員の調達と教育に時間がかかってしまった。この点は、当初から十分に準備と計画ができていればより効率が上がったと思われる。

### 6.3 情報の共有について

前述したように、本プロジェクトは、ピーク時には国内外あわせて170名を越す開発メンバーが参加した。さらに、プロジェクトの途中から参加するメンバーも多く、プロジェクトの概要、現状、既知の問題とその解決状況、フレームワークの更新情報、DBスキーマの変更情報、障害の発生状況、などの情報共有は、プロジェクト遂行に欠かせない。これらの情報は電子メールを使った連絡方法に加えてイントラネットの掲示板に公開し、全メンバーとの情報共有を図った。

また、一方的な情報の発信だけでなく、開発のノウハ

ウ、メンバーからの要請、連絡などにも掲示板を用いて、メンバーの横の連携にも利用した。それでもピーク時には連絡の漏れや周知の遅れなどが一部で発生してしまった。教訓として、一方的な情報発信だけではなく、それが周知されているかをチェックする必要があったと感じている。

#### 6.4 ソフトウェアリソースのすべてを日本語に

開発効率や保守性の向上を目的として全面的な日本語化を導入した。

日本語化の対象範囲は、データベース定義のテーブル名、項目名、ソースライブラリのソースファイル名、クラス名、メソッド名である。これらについて100%日本語化を行い、特段大きな問題なく動作できた。これにより開発効率が向上し、ソースの可読性も高くなったと思われる。

日本語化における問題点として挙げられることが2点あった。1つ目は、本システムではデータの定義をXMLのスキーマ定義方式であるXSDを用いて行っており、一部の文字が定義不能ことがわかった点。これに関しての手戻りが何箇所か発生したが、幸い大きな問題とはならなかった。

2つ目はソースファイル名の問題である。本システムで使用しているソース管理ツール（VSS）が、ファイル名が34バイト以上の場合に一部機能の動作不良を起こした。日

本語が直接の原因ではないが、ファイル名に関しては比較的短くてすむ英語名が無難であると感じた。

オフショア開発の計画段階において、完全日本語化は生産性の低下が懸念されたが、オフショア先の協力も得ながら発注の段階で辞書の制作を行い、オフショア開発にかかわる日本語化についても特に問題なく進めることができた。この点に関しては、他のプロジェクトにおいても積極的に導入すべき項目として自信を持って推奨できる。

## 7. 終わりに

本プロジェクトは、.NETという新技術の導入と開発規模の大きさを見る限りでも、計画段階からあらゆる困難が予想される開発であった。当社は初期段階から多くのリスクを想定して適切な手段を講じ、困難を乗り越える策を数多く打ってきたが、それでもなお厳しい局面が多々あった。最終的にはメンバー各人の努力と、カットオーバーを必ず実現するという信念を貫くことで完了できたと思う。この開発では多くの失敗も味わったが、それ以上に多くを学ぶことができたと思う。ご協力いただいたSTB殿に、この場を借りて心からお礼申し上げますとともに、開発メンバーおよび関係者の労をねぎらいたい。