

開発技術と開発プロセス

実装フェーズでのプロジェクト内サポート

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

藤島 一月 男

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

岡村 智 幸

1. はじめに

年金管理業務は、年金制度の複雑性を始めとして、システム構築を行う際に深い業務知識が要求される領域である。その複雑な業務要件を理解し、適切なシステム構築を行うために今回のプロジェクトでは、外部設計フェーズでは業務系技術者を集めることが重要成功要因であった。このため、今回開発に従事したのは、多くがメインフレームとCOBOLを中心に経験してきた技術者であった。プログラミングを行うメンバーにはオブジェクト指向言語やVisual Basicの経験者も参加したものの、プロジェクト編成に当たり実装技術よりも業務経験を優先させたことは否めない。そのため当初から、開発時にいかにスムーズに品質の良いプログラムを開発するかがハードルとなることが予想されたが、プロジェクト内サポート体制を充実させることでこの問題を乗り切った。今後も、特に金融系システムでは、業務系技術者が重要となるプロジェクトにおいてはメインフレーム経験者が大半を占めるケースも想定される。今回.NETを利用した大規模開発の実装フェーズで採用した体制やサポートプロセスは今後の参考となると考え、本レポートを作成した。

なお、本誌別稿でも詳細に触れているが、内部設計段階ではオブジェクト指向設計を実施し、Visual Basic.NET(以

下VB.NET)のクラスを本格的に利用したシステム実装を行っている。プロジェクトメンバーは、オブジェクト指向設計に関するMicrosoftのカスタムトレーニングコース、VB.NETのカスタムトレーニングコースを受講済みであり基本的な知識は修得していた。

2. 想定された問題点と実施した施策

基本訓練は終えていたとはいえ、豊富な実装経験があるわけではないため、プログラミングテクニックに関するサポートが必要になると最初から考えられていた。加えて、業務が多岐にわたり複雑なため、実装フェーズに至るまでサブプロジェクトを業務単位で構成したことから、どのサブプロジェクトでもオンラインあり、バッチあり、DBアクセスありと同種のパターンの実装が各チームに分散されるという実装戦略上の不利があった。

実装の段階では、例えばDBアクセスのプログラムを繰り返し開発するような実装パターン別のチーム割になっていれば、品質も一様になりやすく生産性も徐々に向上することが期待できる。反面、その実装に必要な業務知識が複雑な場合は、アプリケーションとしての機能バグの入り込む危険は拡大する。今回は、業務面での機能品質向上を最大テーマと考えたため、機能割のチーム編成を選択した(図1)。

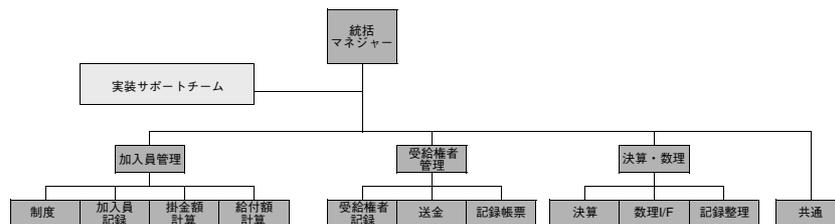


図1 年金管理システムの開発体制図

実装フェーズ開始時に想定したリスクは以下の3点である。

- ・プログラミング生産性が想定値を満たせないリスク
- ・技術的な問題により、容易に解決できないエラーが発生するリスク
- ・想定アーキテクチャの能力を引き出せない設計・実装を行ってしまうリスク

このリスクの発現を回避するための施策として以下の3点を計画・実施した。

- ・標準化推進タスクの実施
- ・情報共有のためのポータルサイトの導入
- ・実装サポートチームの編成

これらの施策を行っても上述の3つのリスクは一部発現し、プロジェクトを苦しめた部分もあったが、少なくとも被害の軽減を行うことができプロジェクトの所期の目的の達成に貢献できたものと評価している。

3. プロジェクト内技術支援の構造

本プロジェクトの実装フェーズにおける支援構造を説明する。

プロジェクト内の技術スキームを設計し構築する技術者からなるチームを本稿ではアーキテクトと呼ぶ（プロジェクト実施中はインフラのリーダーが兼務していたためアーキテクトという呼称は用いていなかった）。当プロジェクトのアーキテクトは2名で、適宜Microsoftのコンサルティングを受けながらプロジェクト全体のアーキテクチャに関して決定を下していった。このアーキテクトが決定した環境に基づき共通チームがコアコンポーネントを作成した。

標準化作成チームはアーキテクトと協力して標準化を決定し、展開する。DA/DBA、Microsoft Consulting Service（以下MCS）およびMicrosoft Premier Support Service（以下PSS）*1はアーキテクトの指揮下に編成した。DBのスキーマの定義、変更はDBAからアナウンスされた。

開発チームはサブシステム単位のチームであったため、いわば横串を刺す形に実装サポートチームと呼ばれるチームを編成し、また情報共有を推進するためのシステムツールとしてポータルサイトを立ち上げて一貫性を確保し生産性、品質を向上させる役割を担わせた。

実装サポートチームは、若干メンバーの入れ替えはあったものの内部設計フェーズから存在しており、そのフェーズでは内部設計レビューと呼ばれていた。内部設計レビューは、オブジェクト指向の設計経験が十分でない設計者を支援するために設計支援の目的で結成されたチームであ

る。実装サポートはこのチームの実装フェーズ版であった。ただし実装フェーズでは残った問題をすべて解決しなくてはならないため、内部設計レビューに比べて格段に広い範囲をサポートする必要があった。実装サポートチームはMCSおよびPSSをバックオフィスとして利用することでサポート効率を向上させることができ、これらの配置は重要な成功要因となった（図2）。

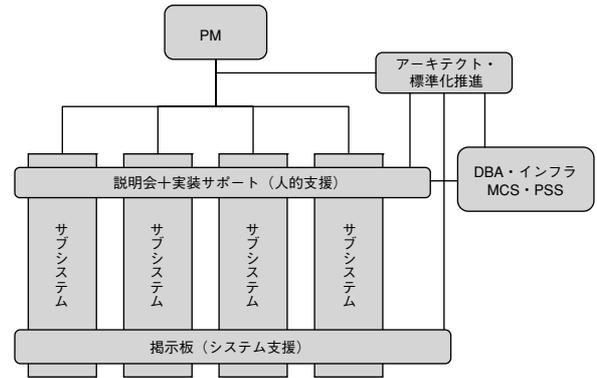


図2 プロジェクト内技術支援の構造

4. 具体的な施策

4.1 標準化推進タスクの実施

標準化推進タスクは、標準化担当およびアーキテクトが基準を設定し、プロジェクト内周知徹底にはポータルサイトを利用、さらに実装サポートチームがプロジェクトメンバーへの解説や指導を行った。具体的に実施した内容は以下の通りである。

- ・プログラミングの標準化

プログラミング標準ドキュメントの作成をメインに行い、プログラミングが進むに従ってドキュメントの内容に不足部分が出てくれば、メンバーから依頼を受け、標準を決定しドキュメントをアップデートしていく。ただし、ドキュメントは共有されたワープロファイルであり、通常、メンバーは印刷したものを参照していることが多いため、Web揭示板のQ&Aを通じてメンバーへの更新のアナウンスも行った。

4.2 情報共有のためのポータルサイトの導入

ポータルサイトの導入により、標準化情報の周知徹底が促進されたが、それだけではなく実装サポートやアーキテクトとのQ&Aの記録、データベーススキーマや共通コン

*1) Microsoftで提供しているサービス。専任担当者による「障害の迅速かつ的確な解決策の提供」「一般未公開技術情報や最新技術情報の提供」など運用段階のみならずシステム構築段階から支援を行い、システムの確実性向上、運用コスト削減をバックアップする。

ポーネントの変更情報のアナウンスでも効果を発揮した。加えて、アンケート実施のプラットフォームとしても機能させることができ、現場からのフィードバック機能としても有効に機能した。以下に実施内容の評価を含めて紹介する。

- ・ディスカッション掲示板によるQ&Aの利用

Q&AをWeb掲示板で行うことにより1つの問題を全員で共有でき、同じ質問に対応する回数を減らすとともに過去に起きたトラブルに関して検索が容易に行えた。

しかし、掲載される順序などはあくまでも質問する順序となるため、後になって過去の情報から自分が抱えている問題が既にあるかどうかを確認しようとすると手間がかかることがある。後からQ&AをWeb掲示板の発言から整理しようと試みたが、未整理の状態からは手がつけられず、残念ながらいまだに整理できていない。またWeb掲示板では発言がスレッドとして連なっていくが、最後、提案なのか結論なのかはっきりしない形で収束したスレッドも多々あったように思われる。Web掲示板を使用する際は注意しなければいけない点である。

- ・共通クラス/DB変更の変更情報アナウンス

不特定多数に影響が出ると思われるクラスやDBレイアウトの変更を周知徹底することを目的としてWeb掲示板に掲載した。

残念ながらいつ更新されるかがわからない掲示板に対して常に注意を払う人は少なく、さらにプロジェクト全員あてメールでも同様の内容を配信した。

ただ、メールだけでは自分が必要となったときに過去のメールを探し出すのが困難な場合があるので、Web掲示板で確認するという使い方がメインとなった。

メールと掲示板がお互いの機能を補完する使用方法として定着し、混乱を未然に防ぎスムーズな運営につながったといえる。

4.3 実装サポートチームの編成

トレーニングコースを受講したとはいえ、大人数で新しい言語を用いた開発を行えばコーディング時にサポートが必要になるのは当然と思われていたが、実際にはそれ以外の問題もサポートを要することとなった。

慣れていないオブジェクト指向設計開発手法や、PCインフラ周りで発生する問題についてもメンバーだけでは対応が難しく、これらについても実装サポートチームが対応し効率化した。

実装サポートの実際の作業は設計の支援、制作支援からインフラ周りの問題解決と対応範囲は広い。このため要員の強化が行われることになり、最終的には7名~8名の体制となったが、それでも全体をカバーするにはまだまだ不足している状況であった。

実装サポート要員の不足に対応するため、業務プログラ

ムをカテゴリに分割し、サポートメンバーはDBアクセス関連（業務セット）、バッチ関連（バッチスーパークラス）、帳票関連（帳票スーパークラス）、画面関連（画面スーパークラス）などに特化して作業分担し、それぞれが各機能割のチームすべてをサポートすることにした。各担当はまずサンプルとなる1本を作成することから始め、必要に応じてコアコンポーネントへの変更要求を出すケースもあった。

当初想定していたコーディング時の技術問題解決として特に影響の大きかったと評価しているものは以下の3点である。

- ・同時実行違反

更新用途でSelectしたデータを修正しupdateする時にselect時と値が変更されている場合に発生する（通常Webアプリケーションではロックをしない）。

今回はDBのトリガー設定や.NETのDBアクセス方法の組み合わせにより発生した。

- ・ネットワークの一般エラー

DBアクセス時に不定期に発生。特定のコーディングをした時の.NET Frameworkのバグであった。

- ・クエリタイムアウト

SQLServerの応答が一時的に遅くなることもある。それらを見越したタイムアウト時間の設定が必要であった。

このような問題を解決するには、実装サポートチームだけでは難しいケースもあり、しばしばMCSやPSSの助けを借りて問題を乗り切った。

その他、生産性を向上させるために実装チームでは以下のような作業も担当した。

- ・テスト環境標準化作成

- ・テストツール作成

- ・コーディングQ&Aサポート

- ・デバックQ&Aサポート

- ・個人環境（マシン）Q&Aサポート

- ・サーバー環境Q&Aサポート

- ・サーバー環境トラブルサポート&運営

- ・代表パターンのコードレビュー

- ・Microsoft支援依頼の窓口

- ・スケジュール遅延プロジェクトのコーディング

- ・共通クラス使用方法Q&Aサポート

- ・共通クラストラブル受付窓口

5. MCSの利用形態

Microsoftとは週1回のミーティングを開催し、週単位で問題点の整理を行う形式でスタートした。しかし、1週間はサイクルとして長すぎたため、実際にはメールでリアルタイムのやり取りとなった。今回、この形式で受けるサ

ポートはコンサルティングの一部なのでトラブルなどの対応は別途SR（サービスリクエスト）という別の形での依頼となる。

SRはMicrosoft起因の問題であれば無料となるが、ユーザー側の問題で発生したトラブルの場合は料金を請求されることとなる。

開発の中でサポートを依頼する場合は、当然、費用と効果を考えなければならない。新技術を導入するプロジェクトでのMCS作業予算化の目安は5%程度が適切という話を聞く。一概に言える数値ではないかもしれないが、予算計画上十分な確保を行いMCSにプロジェクト状況をきちんと理解しておいてもらわないと、問題発生時の迅速な対応が期待できず重大な事態を招きかねない。今回も、Webにまつわるトラブルや大規模データの扱いに伴うトラブルが発生したが、MCSおよびPSSによるサポートの導入効果は非常に大きなものであったと考えている。おそらく、導入しなければスケジュール通りのリリースは難しかったであろう。

6. 施策の評価と得られた知見

今回の体制構築および作業内容については、概ね正しく効果を発揮できるものであったと評価している。担当技術者のインフラやアーキテクチャへの習熟度により体制の大きさは変動すると思われるが、アーキテクトと実装サポートチーム（メンタリングチームと呼ぶ方が妥当かもしれない）の編成は常に行うべきであると考えている。直接、プログラム作成責任を持たないサポート担当者を配置することにはコスト面からの抵抗感があるだろうが、ある程度の規模以上のプロジェクトであればこのチームの編成なくして効率的なプロジェクト運営は困難と考える。

一方、今回のプロジェクトの経験を通じ、より改善すべきであったと感じられたのは以下の3点である。

・実装サポートの投入時期

実装サポートが結成され、作業を開始した時点で問題となった、前工程が不十分であったためのロスを考えると、内部設計フェーズでより大きなサポート体制を敷いておいたほ

うが得策であったと考えている。まず実装サポートが設計思想に基づいたサンプルコードを作成し、その中で具体的なコーディング上の決定事項を決めておき、実際にプログラミング工程に入った時点で展開できれば効果的である。

・実装サポート権限

実際にアプリケーションの開発でスケジュールがかなり厳しい状況では、サポートチームに強力な権限が与えられないとスムーズな運用は難しい。

体制的には各業務チームと同等の権限しか与えられていなかったためせっかく情報の一元化を行い他のチームにサポートの横展開を図っても、チームに強制力が足りないため指示通りに修正を行わせることに苦労することもあった。

・情報ポータルツールの選定時期

プロジェクトの情報共有ツールは、プロジェクトの計画段階でどのような用途が考えられるかを考えて選定することが大事である。プロジェクトの性質により情報共有手段・コンテンツはさまざまである。また、世の中にはあらゆる情報共有ツールが存在しておりその選定のタスクはその後のプロジェクトの生産性をあげる観点から重要である。当プロジェクトのケースでは外部設計フェーズより導入し、コンテンツもその場その場で作成した。本来、ツールの選定・コンテンツ検討はプロジェクトの予備検討段階から実施すべきだった。

7. おわりに

今回のプロジェクトでは、計画時にもリスク評価を行って、サポート要員の投入、支援ツールの導入を行ったが、想定したよりもさらに多くの工数がサポートに必要なになった。この点から考えても当初から思い切った支援計画を立案する必要があることを実感させられた。幸い、無事にカットオーバーを迎えられたが、同様の条件のプロジェクトを行うケースがあれば、サポート体制とサポートツールの検討には念を入れるべきであるとコメントしたい。本稿が今後のプロジェクトの成功のために参考になれば幸いである。