



全日空システム企画

航空旅客システム事業部 CRMシステム部  
第一チームリーダー 阿部恭一郎

アジャイル開発手法  
「Scrum」を  
RUP に組み合わせる

全日空システム企画では、全日空の大規模基幹業務システムの構築プロジェクトに RUP を適用した。さらに、状況変化への対応に主眼を置いたマネジメント基準である「Scrum」を行動規定に取り入れ、リスク軽減を図りながらシステム構築を成功させた。

「RUP はあくまで指針。どう活かすかが重要」と RUP 経験者は口を揃えるが、当社の事例でも、推敲フェーズで進捗が滞るといふつづきを経験した。これを打破したのは、マネジメント方式の見直しだった。

開発プロセスにオブジェクト指向を導入し、効果を十分に引き出すには、マネジメント方式を工夫することが肝心だと考えている。

行動規定の見直しと  
迅速な変化対応で問題を解決

基幹システム構築プロジェクトの概要は、以下のようなものだった。

開発期間は 2001 年春からの約 10 カ月で、500 人月超の工数が見込まれた。最大の課題はリスクヘッジだった。既存システムを停止させずに、新システムへの大量データの移行が可能なのか、全社規模の運用に耐えるレスポンスは確保できるのか、といった点が危惧されていた。

開発方式に RUP を採用した理由は、「計画を基本に置いた検証」、「反復型開発による工数増加の抑制」や「リスクを前倒しして吸収できる点」を

評価したからだ。ウォーターフォールやスパイラル開発の適用も検討したが、前者はリスクヘッジの問題、後者はメソッドや詳細手順が確立していない点がネックとなった。

しかし、膨大な RUP ドキュメントのうちどれが最適かといったノウハウはなかった。本プロジェクトでは 8 つの開発チームを同時並行で設置したが、複数チーム間のマネジメントも手探り状態だった。

まず、反復の回数を計画した。推敲フェーズは 2 カ月で 1 回、作成フェーズは 3 カ月で 1 回、移行フェーズは 3 カ月で 5 回反復することにした。

次に、必要最小限のドキュメントを図 1 のように選定した。「システム要件一覧」「ユースケースモデル」「分析設計モデル」「インフラ基本設計書」の 4 つと、マネジメント用の「リスクリスト」および「課題管理表」を重視することにした。

マトリクスは大きく 3 つに分けた。ユースケース単位の「進捗マトリクス」、エラー密度を管理単位としたレビューの「品質マトリクス」、従来型と同様にバグ摘出率などを単位とした「テスト品質マトリクス」だ。

開発段階では推敲フェーズに力を入れた。反復計画書では 1 回だったが、実際には複数回の評価を重ねた。また、アーキテクチャ検証を前倒しして、動作保証を図った。

作成フェーズでは、ウォーターフォールをベースとした上で、期間短縮のためのアレンジを加えた。作成フェーズの反復は 1 回だが、複数チームからのフィードバックによるリグレッションテストの品質保証、工数削減のためには検証の自動化が必須となり、テストツ

ールを導入した。移行フェーズでは、5 回の反復を実施した。

精神論を排し  
従来型の管理手法から脱却

方向付けフェーズを終え、プロジェクトはそのまま推敲フェーズへと移行した。だが、そこで問題が起こった。2 週間を過ぎたころから、計画どおりに進捗しなくなってしまったのだ。

原因は、計画と評価が不十分だった点にあった。数値化された管理内容と実際の作業が乖離し、週に 1 度の報告では、状況変化へ対応するための管理作業が膨らんでしまった。

また、RUP のドキュメント規定も負担になっていた。マネジメントが従来型から完全には脱却できていなかったのだ。当初は、ドキュメント規定とプロセス規定に重点を置く一方、ドキュメントの利用法やワーキングプロセス管理などを規定していなかった。

そこで、Scrum と XP (eXtreme Programming) に準拠した行動規定を策定した。これにより、状況変化に応じて役割を変更できるようになった。結果、推敲フェーズの管理方式は、以下ようになった。

**専任のマネジメントグループを独立**  
7 人のマネジメントメンバーを選抜し、あえて管理のための定型的な役割を作らず、自律的で柔軟に対応できるチームとした。RUP による品質最優先のプロジェクトでは、管理工数の多少の増大はやむをえないと割り切った。

**毎日の状況監査**  
15 分間のブリーフィングで、状況変化と対策、コスト等を報告。ここではユースケースごとの全体管理表と、課

表 2 マネジメントドキュメント類

ドキュメント名称	インプット情報	目的
ソフトウェア開発計画書		開発に関する全体の計画方針を記述する。推敲フェーズにて作成する。
開発個別定義書		各フェーズにおけるワークフローと作成ドキュメント類を定義する。
各種ガイドライン		モデリング、言語利用におけるガイドラインを示す。
マトリクス定義		進捗、品質マトリクスを定義する。
反復計画書		方向付け、推敲、作成、移行それぞれのフェーズにおいて最低 1 つの計画書を作成する。
リスク管理計画書	反復計画書	反復計画におけるリスクについての基本的な方針を記述する。
リスクリスト	反復計画書	反復計画におけるリスクを具体的に記述する。
課題管理表	リスクリスト	リスクリストの問題を整理した後、課題としてリストアップおよびスケジューリングする。
進捗報告書	反復計画書	RUP 名: ステータス評価書、週報、スケジュール表など。
問い合わせ管理簿		質問、確認などの文書類。
仕様変更管理簿		仕様変更管理
反復評価書	マトリクス定義、反復計画書	方向付け、推敲、作成、移行それぞれのフェーズにおいて対応した評価書を作成する。収束、持ち越しにおいては次期計画書に反映されていること。
工程判断報告書		反復計画中のマイルストーンの評価報告、次フェーズへの判断材料。
試験状況報告書		各試験の状況と品質状態を記述する。
製品検収計画書		

題管理表のみを使用した。従来は、ガントチャートに振り回される傾向があったが、課題管理表に基づいて対応することにした。ドキュメントを軽くし、変化を迅速に捉えるためだ。

**障害原因を徹底的に追及**  
推敲フェーズ開始時の失敗を教訓とし、障害原因を厳しく究明した。評価・対策には精神論を排除し、数値ベースで実現可能な方法のみを採用することとした。

**週報に課題管理表を最大限活用**  
「ドキュメントのためのドキュメント」を極力軽減するため、リスクリストを課題管理表へ成長させ、作業のスケジューリングによって全体管理表へさらに成長させるというプロセスを徹底させた。

これらの効果はハッキリと現れた。プロジェクトの遅れは、推敲フェーズの中間地点で取り戻すことができ、新しい管理方式が定着していった。「RUP の採用は正しかった」という認識が共有され、プロジェクトの推進方式が固まった。以降のフェーズでは、同様の問題は起こらなくなった。

ユースケースとアーキテクチャを  
中心とするアプローチが重要

本システムは、性能・品質ともに満足のいく結果となった。必要なドキュメントの整理・選択、Scrum を取り入れたマネジメント方式の確立により、今後、同様のシステム構築の管理工数を削減できると考えている。

オブジェクト指向開発本来のコストパフォーマンスを引き出すには、以下の点が重要と認識した。

まず、タスクの管理はユースケース単位とすること。決して機能視点に陥ることなく、ユースケース視点を貫く「ユースケースドリブン」を徹底する。さらに、最重要なユースケースを最初にモデリングして検証することで、リスクを回避する。その後、他タスクを横展開していく。

また、ネットワーク構成とソフトウェア構成、データモデルを早期に構築し、推敲フェーズの半ばでアーキテクチャを検証、チューニング計画を実施する。こうしたアーキテクチャ中心のアプローチが重要と考えている。