中／大規模開発の泥沼からの脱出

シナリオで理解する開発プロセスの実際

藤井 拓也
FUJII, Takuya

Technology Tools

- ❑ Visual Basic .NET
- ❑ Visual C# .NET
- ❑ SQL Server 2000
- ❑ Oracle 9i
- ❑ Access 2002
- ❑ ASP.NET
- ❑ Internet Information Services
- ❑ Other:

Level

★ ★ ★ ★ ★

Samples

はじめに

「.NETによる中／大規模開発の実際を……」。最初にこのお題をいただいたときには、正直“う〜ん”と、うなってしまいました。「.NETってなに？」というところからして、風の向くまま気の向くまま、マーケティングが思いつくままのように見えますし。

実装言語も複数あるし、新しいフレームワークやら、CLRやらといった新技術もてんこ盛り。概念を習得するだけでも大変です。普段接する現場のエンジニアのみなさんも現時点では“大規模システム開発の本番”プロジェクトよりも、“大規模システム開発の可能性を検討する評価”的扱いのプロジェクトが多く、立ち上がったばかりといった感じです。

では、これらの“新技術のお勉強”が終われば、システム開発はすぐにでも成功させることができるのでしょうか？

要素技術は確かに重要です。しかしながらシステム開発の規模が大きくなると、個人芸よりも集団芸の世界、単純に中核となる要素技術だけで成否が決まるわけではありません。言語仕様だフレームワークだという仕様を踏まえた上で、プロジェクトの運営やら分析設計手法やらという考慮点が顔を出してきます。

そこで本稿では、「VBは知っているけどVB.NETのプロ以前」かつ「UMLは勉強したけどオブジェクト指向技術者以前」のエンジニアの方を対象に、

VB.NETを使った中／大規模開発を手がける上での泥沼をどう避けるか？

について論じてみたいと思います。ベースとなるプロセスとしては、世に言う“統一プロセス（Unified Process）”を考えたいと思います。

“中／大規模である”ということは何を意味するか？

さて、あっさりと“中／大規模”とは言ってみたものの、開発するシステムの規模が大きくなるということは、

具体的にはどのような課題が顔を出してくるのでしょうか？

モノ作り
～分析やら設計やら

.NETアーキテクチャでは.NET Frameworkの使用が前提であり、開発者は、“継承”という概念の理解を求められます。継承はオブジェクト指向の世界で再利用を実現するための主要な仕組みですが、再利用を実現するためには、それ以外にもさまざまな手段が用意されています（図1）。代表的なところでは次のモノを挙げることができます。

①.NET上のオブジェクトから非.NETのCOMコンポーネントを利用する（既存COM資産の再利用）
②（VB.NET）前述した“実装の継承”
③Webサービスを使って他システムが提供するサービスを再利用する

多くの場合、比較的大規模なシステムは、その投資に見合うだけの期間使い続けられ、継続した機能追加が行なわれると考えても間違いではないでしょう。

こうなると、何を再利用の単位として見なすか、どの手段で実装するかという判断が重要になってきます。基幹系システムの場合は、後ほど継承のところで紹介するように、ひとつの概念があちこちのサブシステムで使われます。うかつな選択をすると、“再利用”や“リファクタリング”ではない、単なる“作り直し”が頻発することとなります。

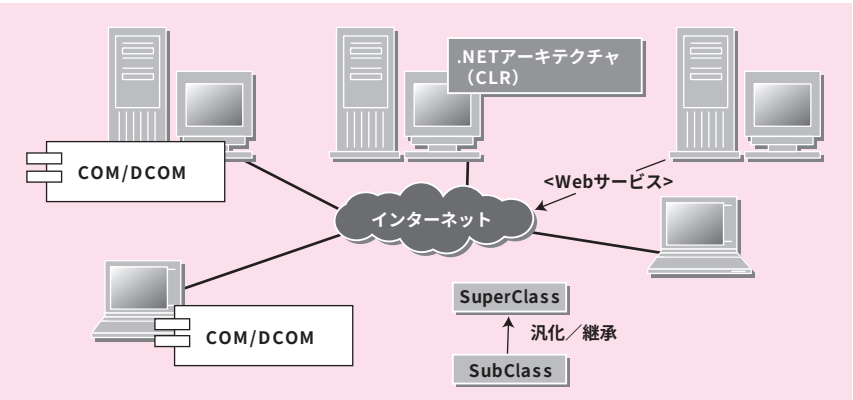
プロジェクトの計画運営

俗に管理といえば“人”“モノ”“金”が対象に挙げられます。開発規模の拡大は、これらすべての不確定要因を増大させます。

管理対象1　人

関係者が増えれば、システムに対する要求も際限なく拡大し、変更依頼もとどまるところを知りません。開発者が増えれば、いろんなスキルレベルの人が集まる中でシステムの品質を確保しなければなりません。

図1：再利用のための各種手段



管理対象2　モノ

開発に必要な資源（場所、機材、関連するソフトウェアツール類）の計画および手配に時間の多くを取られます。

管理対象3　金

……。解決策なんてあるんでしょうか？

これらを制御する（解決ではない）アプローチとして近年注目を集めているのが統一プロセスに代表される反復型の開発プロセスです。どんな手順で計画を立てたり運営してゆくのかは、後述のシミュレーションのところで議論したいと思いますので、今しばらくお待ちを。

人的課題

目に見えて変わるのは、“開発者が増える”という点です。“人がたくさんいて楽しい”というメリットはあるにせよ、注意しないと次のような問題が起こりえます。

・スーパーマンばかりは集まらない
・情報が共有されない、徹底されない
・他の開発者が何をしているかわからない
・責任範囲があいまい

コミュニケーションをしっかり取るということは、開発するシステムの規模／内容にかかわらず非常に重要な「成功のためのキー」です。比較的少人数で運営される“アジャイル”的アプローチではエンドユーザー側の担当者も含めて密なコミュニケーションを取