Visual Studioで構築するエンタープライズシステム

Application Architecture for .NETの利用例

第7回 フレームワーク化を始める

株式会社CSKシステムズ
IT生産技術部
中垣 健志
NAKAGAKI, Kenji

はじめに

大きな台風が過ぎ去り、空は青く澄み渡っています。今日、N君の支援していたプロジェクトが無事本番稼動を迎えました。支援中は技術的な課題が次々と発生しましたが、チーム内のメンバーと協力してすべて解決できました。N君の頭の中には多くの.NETに関するノウハウがつまっていまにもこぼれ落ちそうです。そんなN君に、同僚のKさんが声をかけてきました。

「とりあえずおめでとう。ところでふと考えたんだけど、支援の中でさまざまなノウハウが身についたわけだよね。できれば、それを次のプロジェクトに活かせるようにまとめておいたほうがいいと思うんだけど、どうかな？」

至極もっともな意見だとN君も思いました。そこで、早速今回のプロジェクトで学んだアーキテクチャのポイントをまとめることにしました。

Level

1 2 3 4 5

Technology Tools

- ☑ Visual Basic
- ☑ Visual C#
- ☐ Visual C++
- ☑ SQL Server
- ☐ Oracle
- ☐ Access
- ☑ ASP.NET
- ☐ Other:

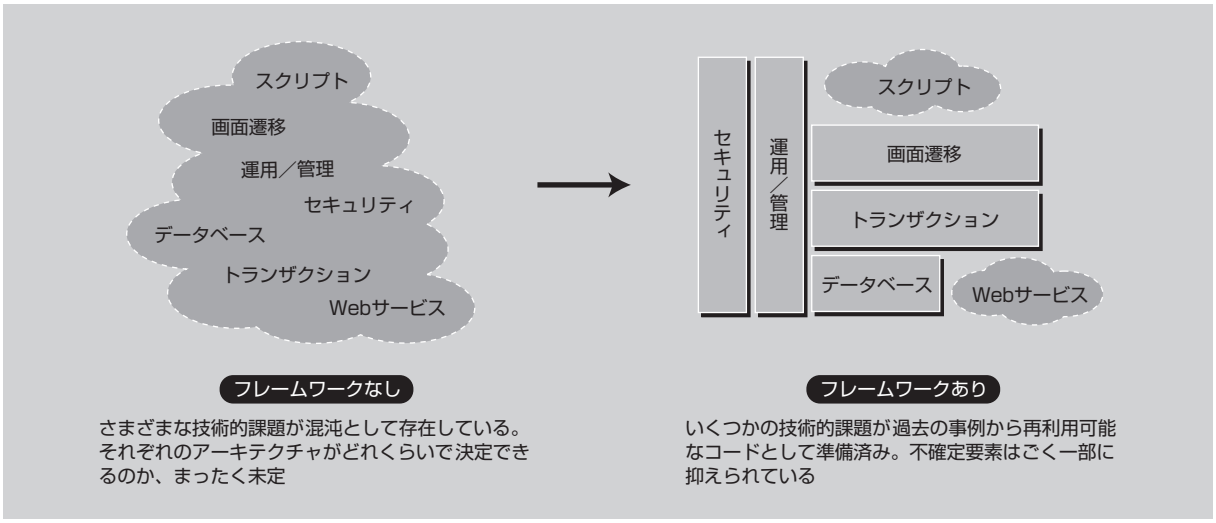
なぜフレームワークが必要なのか？

ここ数年の間に、「フレームワーク」という言葉はエンタープライズシステム構築においてはずせないキーワードとなってきています。そこで、「なぜフレームワークは必要とされるのか」という理由から考えていこうと思います。

エンタープライズシステム構築そのものは「手段」であり、「目的」ではありません。エンタープライズシステム構築の目的は、企業に利益を生み出すことです。健全な企業経営を行なうためには、利益は計画的に生み出されなければなりません。つまり、エンタープライズシステム構築は企業活動の計画の一部として、予定された品質、コスト、納期に従って進められなければならないのです。

一方で、エンタープライズシステム構築は複雑で知的な作業です。ベルトコンベアーで流れてくる部品にねじを止めるだけ、といった単純作業ではありません。エンタープライズシステムを構築するためには、最新のさまざまな技術を評価し、そして組み合わせる作業が必要です。では、求める機能を実現するために、どのような技術をどのように組み合わせればよいのでしょうか？　この課題の解決に要する時間はまちまちです。1時間で解決できるかもしれないです

図1：フレームワークとアーキテクチャの設計



し、3か月かかっても解決できないかもしれません。そもそも、解決できない課題かもしれないのです。つまり、エンタープライズシステム構築の計画を立てるのは、非常に困難なことなのです。

この相反する2つの命題をSIベンダーは常に抱えています。フレームワークの採用は、この課題に技術的な側面から解決を図るための有効な手段なのです[注1]。そこで、続いてフレームワークが果たす大きな役割を示していきます。

◆アーキテクチャ設計難度の低減

エンタープライズシステム構築においてアーキテクチャの設計はもっとも重要、かつ高度な技術を必要とする作業です。アーキテクチャとはシステム開発の技術的基盤となる基本設計です。プロジェクトで作成されるすべてのコードは、定められたアーキテクチャに従わなければなりません。したがって、アーキテクチャが貧弱だとプロジェクト全体の品質や生産性に悪影響を与えてしまいます。

ITの進歩は近年とても早くなっています。これらの最新技術を把握した上で、適切なアーキテクチャを設計しなければなりません。しかし、アーキテクチャの設計を行なうことができるスキルを持った技術者はほんのわずかです。したがって、スキルを持った技術者を集められなかったプロジェクトでは、アーキテクチャの設計の品質や期間を計画するのは非常に困難なこととなります。

この問題を解決するための手段としては「アーキテクチャの再利用」です。アーキテクチャの設計を行なうにあたって決定しなければならないことのうち、いくつかはほぼすべてのエンタープライズシステムで共通の課題となります。先月号までに取り上げてきたテーマ（DBアクセス方法、トランザクション管理手法、画面管理など）はその典型例です。これらのテーマについて、推奨される設計済みのアーキテクチャをフレームワークとして用意します。各プロジェクトでこのフレームワークをアーキテクチャとして採用することで、技術的な不確定要素を少なくすることができます（図1）。

◆プロジェクト間で成果物を再利用

今回作成しなければならない機能は、すでに別のプロジェクトで作成されているかもしれません。優れたアーキテクトであれば、自作ではなく既存の仕組みを探すところから作業を始めます。しかし別のプロジェクトで作成された部品は、大体の場合そのままでは使えないことが多いものです。これは.NET Frameworkが決してエンタープライズシステム専用というものでなく、ビジネス

注1）管理的な側面から「プロジェクトマネージメント」という考え方も注目されていますが、ここではとりあげません。