Archway
Consulting Service

MP3　2005/12/16

シンプルに作る
エンタープライズアプリケーション

株式会社アークウェイ
代表取締役
森屋英治
http://www.archway.co.jp

アジェンダ

- 一般性の追求
- 守破離
- アーキテクチャ
- 適材適所
- .NETらしい技術
- エンタープライズアプリケーション
- まずは、シンプルにつくる

一般性の追求

- フレームワーク、ミドルウェアに再利用性を求める
- 過度のフレームワーク
  - 大きいことはいい事か？
  - 万能フレームワーク
  - 使われない→現場で、またフレームワークを開発
  - 複雑化→単純の流れ
  - ビジネス的にパッケージを適用するケースでも使用する機能を限定
- .NETフレームワークの意味
  - クラスライブラリー
  - フレームワーク（最近は、Foundation）
  - Javaは若干プリミティブ
- 専門性の変化と対応
- 安定している要求と可変性
- パターンの意味（P&P）

守破離

- 守破離
- 自分たちの作りたいアプリケーションにもっともあったアーキテクチャの選択
- 新しい技術、エッセンシャルな技術に対する深い知識
- 正解はないと思い、検証を繰り返す
  - その時点での正解に近いものを模索する
- 固定概念を捨て練り直す
- ゴールは、
  - 実装可能なアーキテクチャ
  - 実行可能なアーキテクチャ
  - 運用可能なアーキテクチャ

アーキテクチャ

- 一枚のアーキテクチャ
  - 一枚のアーキテクチャですべてを表現できるわけがない
  - 箱と線の罪
  - 複数のアーキテクチャを検討
  - チームスキルを考慮
  - 技術的欲求を抑える
- デザインのパターン
  - SOA中心
  - OO中心
  - フレームワーク中心
    - UIP
    - XML
    - ORM
    - TM
    - メッセージバス

アーキテクチャ(つづき)

- 技術の波
  - Visual Studio 2005
    - フレームワークのベースアップ
    - 統合開発環境へ
  - WinFx
    - WPF(Windows Presentation Foundation)
    - Atras
    - WCF(Windows Communication Foundation)
    - WWF(Windows Workflow Foundation)
    - LINQ
  - アーキテクチャはしばらく変わりつづける

アーキテクチャ

- 安定した基盤とは
  - SOA
  - Data
  - フレームワーク
- 分散システムの第一原則
  - 分散させない
- Javaの流れ
  - ミドルウェア→DI
  - MSは逆行　シンプルからコンプレックスへ
- 非機能要件（トレードオフの連続）
  - FURPS+
- アーキテクチャプロトタイプのススメ

適材適所とバランス

- 偏ったデザインは必ずしわ寄せがくる
- アプリケーションタイプを分類し、アプリケーション特性にあった技術を採用する
- 部分適用(限定適用)の技術
  - 分散トランザクション
  - ORM
  - UIP
  - WebService
- マルチパラダイム
  - SOA
  - OOD
  - DOA

.NETらしい技術

- UI
  - D&D症候群
  - イベント
  - コードとデザインの分離
  - ASP.NET(Formスタイルの開発)
- データバインディング
  - データセットをバインドしたい
- データアクセス
  - インメモリーデータベース
  - データアダプタ
- WebService
  - 革命的なASMXのコードスタイル

エンタープライズ・アプリケーション

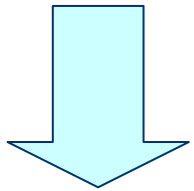
- DBがある
  - OOだけではつめ切れない
  - CRUD中心 + Find XX
  - Transaction（ローカルトランザクション）
  - データバインドしたい
  - モデルをDBにもつケースが多い
  - データ分析、正規化すべき
- コンプレックスな処理
  - プロセス
  - ルール
  - トランザクションスコープ
  - WWFに期待

エンタープライズアプリケーション（つづき）

- フレームワーク開発
  - OO中心
  - パターンの適用
  - 可変性の考慮
- 一番多いコード
  - UIのバリデーションコード
  - データバインド
  - データアクセスコード
    - パラメータ渡し
    - 型変換

エンタープライズアプリケーション

- CRUDベースの開発
- OO経験者の不在
- XSLTが苦手
- SOAのデザインができない
- ローカルアクセスなのにWebService



必要なところに限定！

- OOなるべく使わない
- XSLTなるべく使わない
- 分散トランザクションなるべく使わない
- WebServiceなるべく使わない

Archway Consulting Service

シンプルにつくる意味

- すべてに一枚のアーキテクチャを採用した場合
  - 全体の開発後に最適化(Transaction,WebService)
  - 全員に同一スキル、トレーニング
  - 楽な方に崩れやすい、強制に対する抵抗
  - 開発コストがフラット
  - 自動化、制約を実施できる意味で価値がある。
  - テストできないような複雑なデザインは、検討が必要
  - 意図がなく、どこかの絵を見たようなアーキテクチャは注意
  - これですべて大丈夫も危険な合図

シンプルにつくる意味

- まずシンプルに作ってから必要な場所に構造を持つ
  - UIプロトタイプ（最初からVSで作成）
    - データバインド、データセット
    - テスト用データベース
  - 全体適用の構造（非機能要件、配置）
  - 機能単位で必要な構造を検討
    - ユーザインターフェイスプロセス
    - ビジネスロジック
      - ビジネスプロセス
      - トランザクションコントロール
      - ビジネスルール
    - 外部システム連携意（モックの検討）
      - サービス、API

シンプルにつくる意味（つづき）

- フレームワーク
  - エンタープライズライブラリー
  - DIコンテナ
  - Windows Workflow Foundation
- 技術的な引き出しが多く、臭いところを試す癖
  - アーキテクトと呼びたい
- パフォーマンスチューニング
  - 利用頻度の高いものをチューニング
  - キャッシュ（SODA)
  - トランザクション（ダーティーリードの検討）
  - インデックス
  - ストプロ

まとめ

- 実装、実行、運用可能なアーキテクチャの検討
- 技術は適材適所
- 難しいものが正解ではない、常にシンプルなものを模索しよう
- “作りやすさ“は、.NETらしい技術にヒントがある
- “やわらかいアーキテクト”のススメ。

http://www.archway.co.jp