

オブジェクト指向技術とその現状

～ 求められる変化とソフトウェア会社の生き残りの条件～

2004年2月20日

ネクストデザイン株式会社

<http://www.nextdesign.co.jp/>

技術セミナー

コンサルティング

システム開発

UMLツールの開発・販売

普及度

- 認知度
- ツール、認定資格の普及
 - JAVA
 - UML
 - C#
- 技術者数の増加
 - 他言語未経験者の場合(無条件)
 - 経験者の場合(パラダイムシフト)
- 事例数の増加
 - 新規開発での使用言語(JAVA)

(補足) 当社設立当時

- 1998年オブジェクト指向の話題
- 福岡
 - 福岡と東京
- 業務アプリケーション(COBOL、VB)
 - 組み込み系ソフトウェア
- 2000年頃からJAVA案件

いつ頃から

- 1950年代
 - COBOL、FORTRAN
- 1960年代
 - 構造化プログラミング、RPG、PL/I
 - Simula67(シミュレーション用言語がはじまり)
- 1970年代
 - ウォーターフォールモデル、構造化分析・構造化設計
 - Smalltalk
- 1980年代
 - C++、OOCOBOL
- 1990年代
 - VisualBasic
 - 方法論 OMT、OOSE、Booch
 - JAVA(1995) UML(1997)統一プロセス(1998)C#(2000)

(補足) これまでの何に置き換わる

- オブジェクト指向は従来の何を置き換える
 - ウォーターフォール(開発方法論)
 - 構造化分析・設計技法
 - プログラミング手法・言語
 - 開発体制、組織

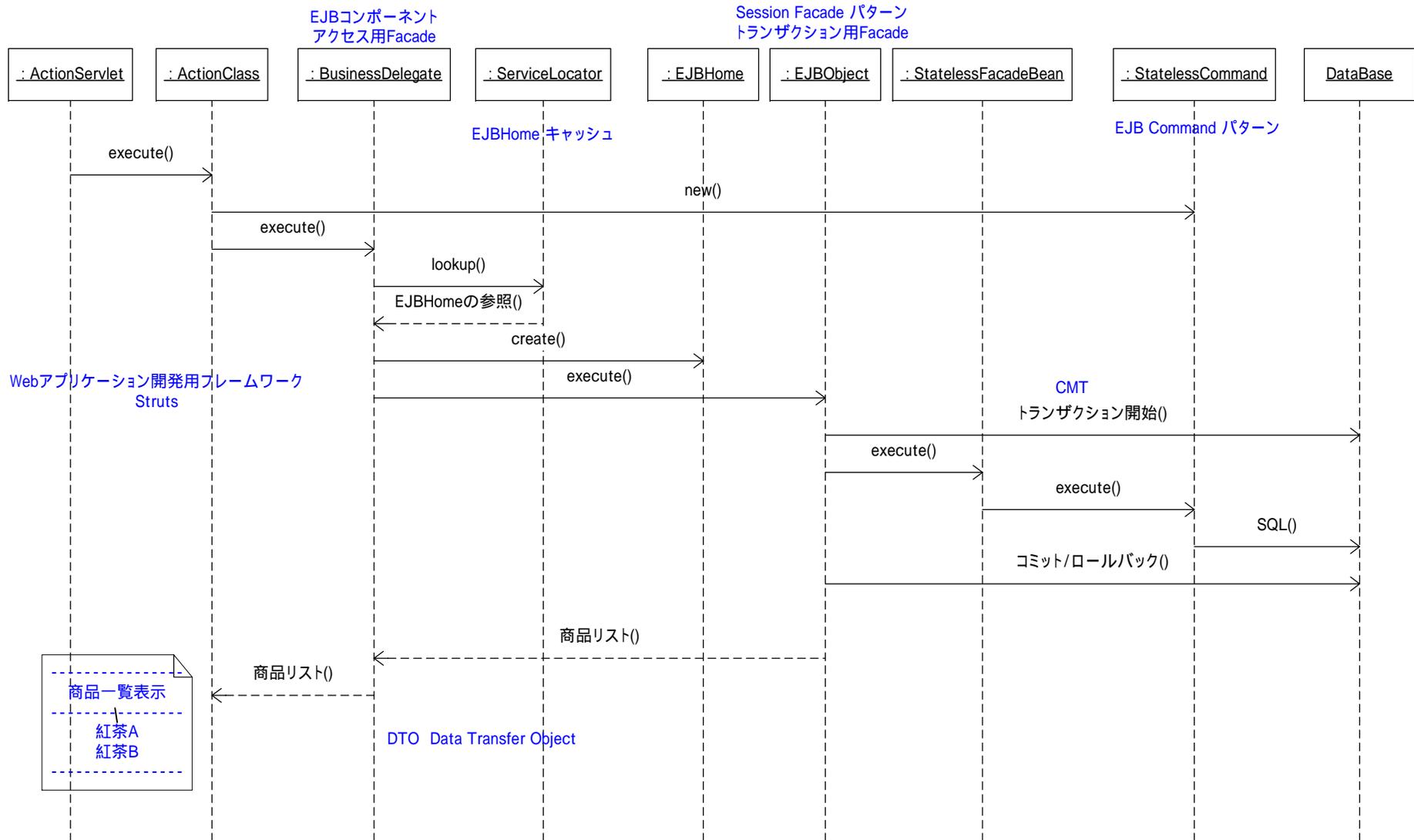
導入の動機と効果

- 導入の動機(普及の要因)
 - 生産性・品質の向上
 - 再利用の期待
 - ユーザ(発注元)の要求
 - トレンド
 - 複雑な要件への対応
- 効果
 - 成功事例と失敗事例の比率

導入例1 (初期の問題)

- 簡単なシステム
- 豊富なサンプル(雑誌、インターネットなど)
- 導入効果を実感できるか
 - **未経験者**
 - 特に疑問はない
 - **経験者**
 - デメリットを感じる人が多い
 - パラダイムシフトを敬遠する
 - 学習コストは認められない
 - 導入目的を確信できない

(補足) サンプル



導入例2 (開発時の問題)



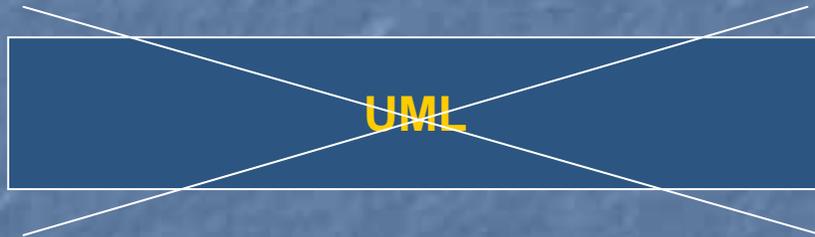
COBOL
C

VisualBasic

~~JAVA~~

導入例3 (開発時の問題)

要件定義	基本設計	詳細設計	プログラミング	結合テスト	システムテスト
------	------	------	---------	-------	---------



開発プロセスと取引構造



ウォーターフォールの問題点

- 問題が少ないケース
 - 目新しい要件がない、経験済みのシステム
 - 同じメンバー、同じツール
- 変化に対応できない
 - ユーザ要求(問題)、とりまく市場環境の変化
 - 開発ツール、プラットフォームの変化
- 文書の作成と凍結が**過度**に重視される
 - 包括・網羅する(書き尽くす)ことは非常に困難
 - 実行できないので正確な検証はできない
 - コストがかかる

オブジェクト指向開発プロセス

- 統一プロセス
 - 反復型
 - アーキテクチャ中心
 - ユースケース駆動

反復型の開発



アーキテクチャルベースライン

従来よりも早い時期にソースコードを書く。ただし「書いて直せ」ではない。

変えなければならないもの

- UMLを使えばオブジェクト指向か
- JAVAを使えばオブジェクト指向か
- ウォーターフォールを使えるか
 - 同じ要件、同じ環境、同じメンバー
- 現場の技術者だけに変化が求められるのか
 - 新しい環境について行くのが精一杯
 - 学習コストは認められない
 - サンプル使いにならざるを得ない
 - 従来の設計者とプログラマの関係
 - 開発体制、取引構造

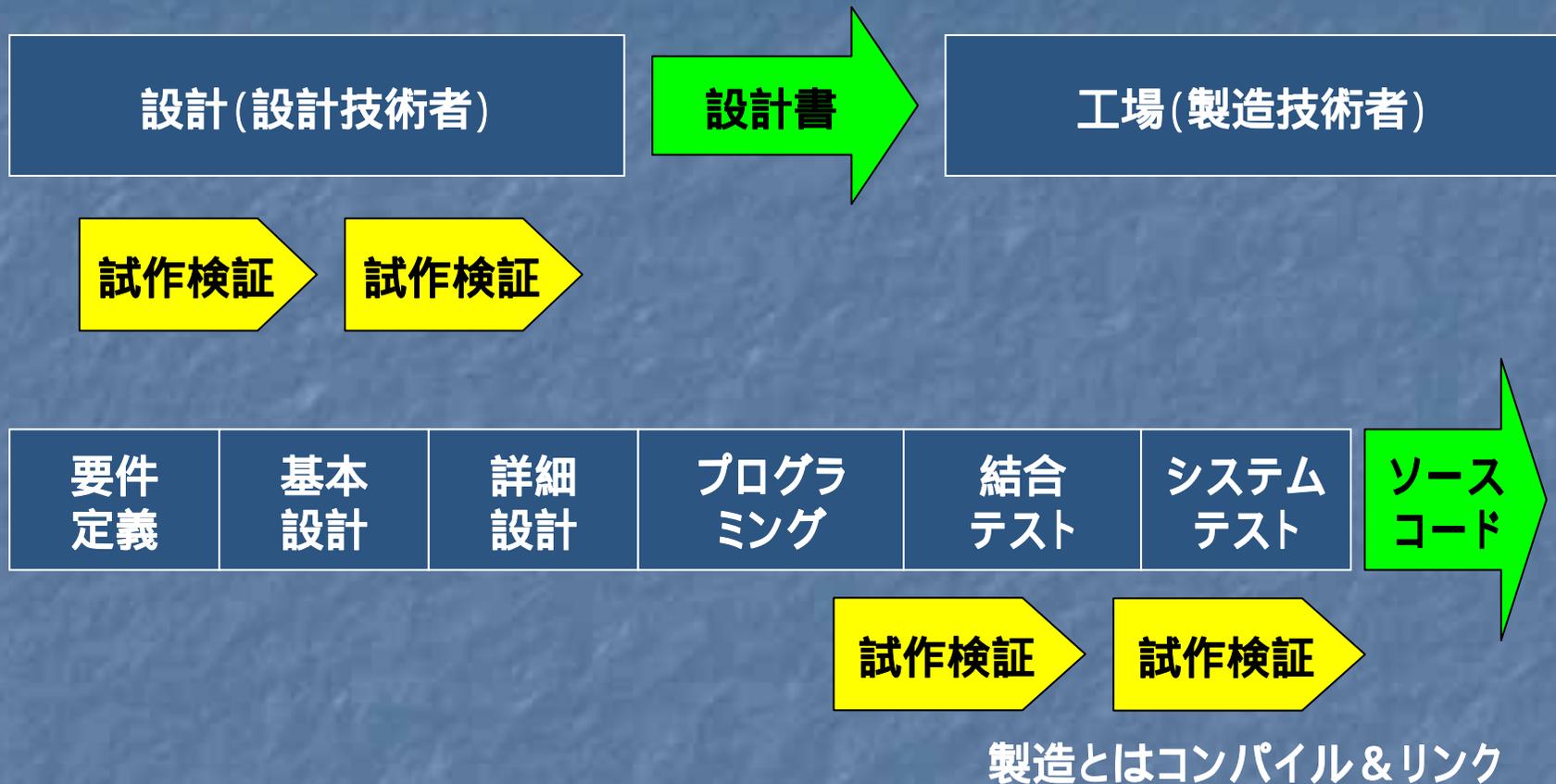
得られるもの

- 複雑なシステム要件の解決手段
 - 企業
 - 技術者
- 再利用効果
 - パターン
 - フレームワーク
 - コンポーネント
- オープンな環境
 - ツール、プラットフォーム
 - 情報
- より工学的な手段
- **技術者、会社としての競争力**

これから

- アジャイル・プロセス
 - 提唱者
 - 過度な文書化は推奨しない
- MDA (OMGモデル駆動型アーキテクチャ)
 - 実行できないモデルはスケッチ
 - 実行可能なモデル
- UML 2.0
- アスペクト指向

(補足) 設計とは何か



1992 C++ Journal Jack W.Reeves

(補足)設計とは何か2

- 検証可能で動く設計モデル
 - MDA
 - モデル 3GL 2GL 1GL(抽象化)
- ソースコード中心
 - アジャイル
 - UML(スケッチ) + ソースコード

(補足) 書籍などの情報の変化

- オブジェクト指向プログラミング
 - 方法論
 - JAVA、C#
 - UML
 - デザインパターン
 - J2EE、.NET
 - オープンソフト How-To系
- 順を追って習得する必要がある**

(補足) アーキテクチャとは

- 論理ビュー
 - 対エンドユーザ(機能、Web、SWTなど)
- 実装ビュー
 - ソフトウェアリソース管理など(CVSなど)
- プロセスビュー
 - 並列性、スタートアップ、シャットダウン
 - フォールトトレランス、分散・デッドロック
- 配置ビュー
 - インストール、パフォーマンス
- ユースケースビュー