失敗しないオブジェクト指向教育とは

～ 効果的な初学者教育へのアプローチ ～

2005/12/16

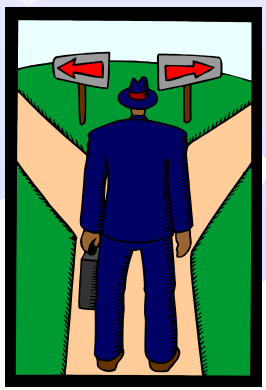
㈱ 豆蔵

http://www.mamezou.com

チーフコンサルタント　藤井俊彰 (fji@mamezou.com)

目次

- オブジェクト指向教育現場の状況
- 成功と失敗の分かれ目とは
- 失敗しないオブジェクト指向教育のために
- 効果的な教育のために



オブジェクト指向教育現場の状況

オブジェクト指向教育現場の状況(目次)

- Case1
  - 早く新技術を習得させたい
- Case2
  - COBOL技術者をJava技術者に転換したい
- Case3
  - 即戦力となる技術者を育成したい

オブジェクト指向教育現場の状況

Case1：早く新技術を習得させたい

- 背景
  - Strutsの案件が増えてきた
  - 「Strutsを使える開発技術者を増やしたい」
    →現場からの要求
  - オブジェクト指向未経験者に対して新技術を習得させて、現場の要求に対応したい
    →人材育成担当者

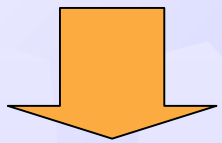
新技術スキルを習得するカリキュラムを中心にトレーニングを計画

Services

Case1：早く新技術を習得させたい

対応方法

- 早急にStrutsでの開発ができる技術者を育成したいので、Strutsの知識習得を中心にカリキュラムを組んだ
  - 新技術の使い方を中心とした内容



結果、Strutsを使える技術者が育成できた・・・

Services

Case1：早く新技術を習得させたい

結果の考察

- システム開発において講座で取り扱わなかった状況が発生すると対応ができない
  - 他の技術との組み合わせ
  - システムトラブルへの対処
- 新技術を学ぶときの学習コストが下がらない
  - 毎回最初から学習しなおし

応用が利かない技術者
成長しない技術者

Services

Case1：早く新技術を習得させたい

成功・失敗の分かれ目

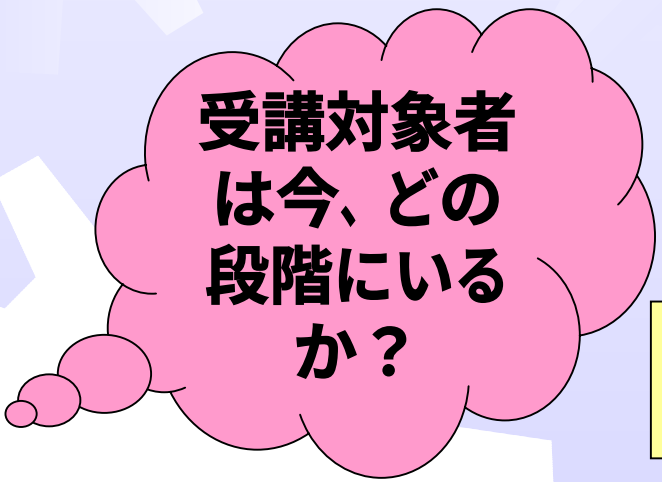
- 「要素技術の使い方中心」のカリキュラムを作成した
- 技術のステップを踏まえていなかった
  - 基礎技術教育を考慮しなかった
  - 基礎技術を理解していないのに、要素技術を上に積み重ねようとした

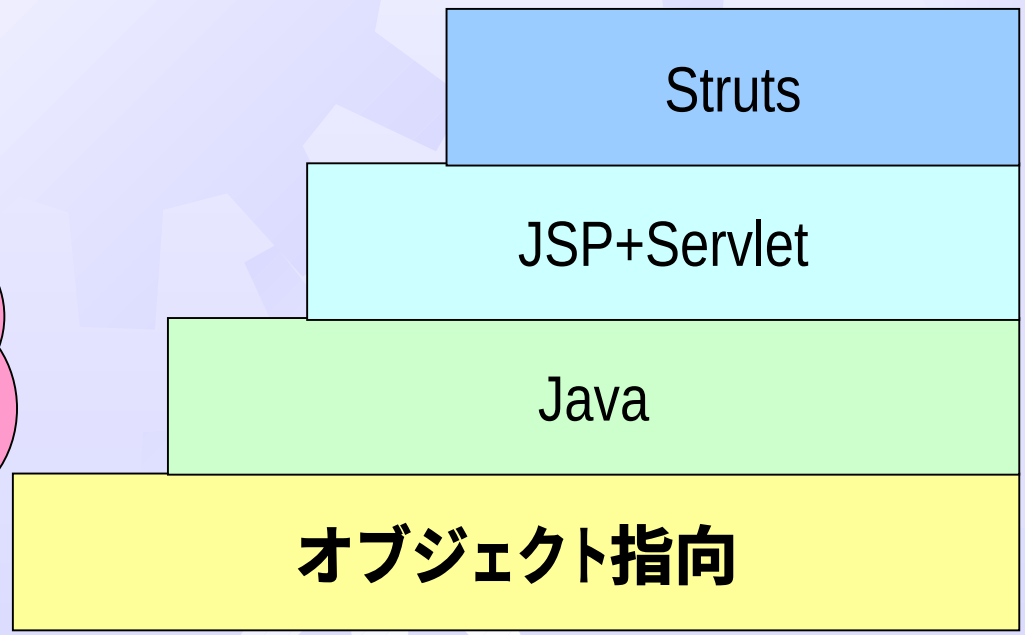
Services

Case1：早く新技術を習得させたい

対処法

- カリキュラム企画時には**技術のステップ**を考慮すること
- 技術の習得は**積み重ね**である





Case2：COBOL技術者をJava技術者に転換したい

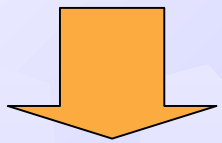
- 背景
  - Javaを使用する開発案件を拡大するため、Javaプログラマが大量に必要になった
  - COBOLによる開発は今後はメンテナンス中心となり、COBOL技術者が余るおそれ大
  - 社内のCOBOL技術者は、オブジェクト指向技術については初心者

COBOL技術者に対して
Java技術者への転換教育を計画

Case2：COBOL技術者をJava技術者に転換したい

対応方法

- COBOLとJavaの機能の対比中心のカリキュラムを作成
  - COBOLプログラミングの知識はあるため、Java言語習得は必要最小限におさえる



結果、ＣＯＢＯＬ技術者はＪａｖａのプログラミングができるようになった・・・

Case2：COBOL技術者をJava技術者に転換したい

結果の考察

- COBOLと直接繋がりのない、オブジェクト指向特有の概念は理解されなかった
  - クラスとは？
  - インスタンスとは？
- Javaなのに、無理やりCOBOLの作法でコーディングする技術者になってしまった
  - クラス名、メソッド名のネーミングルールなど

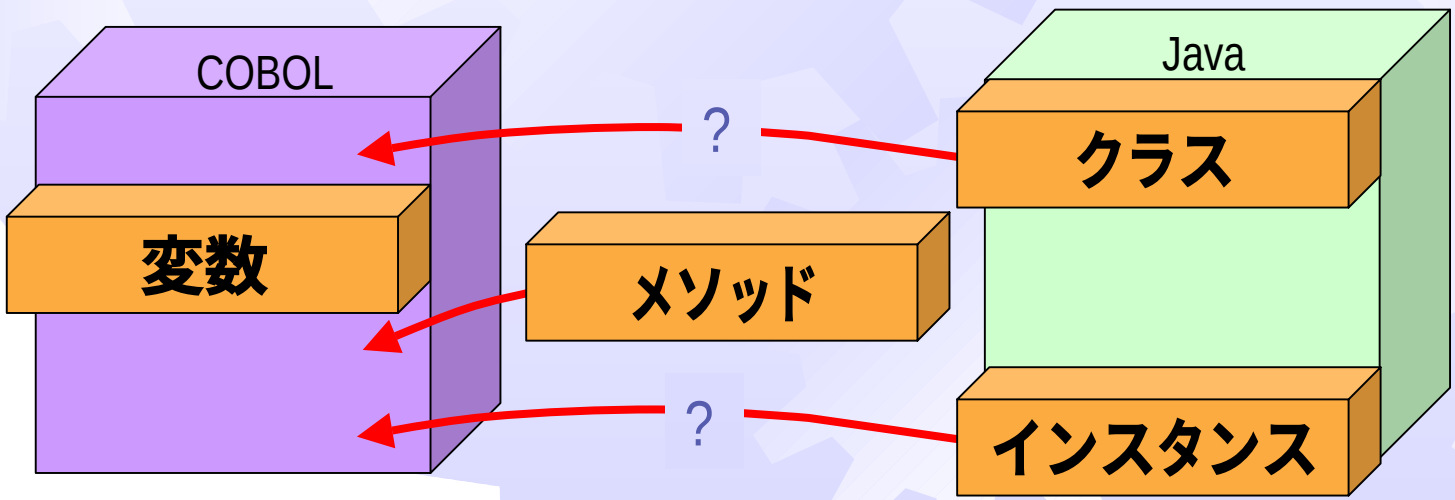
オブジェクト指向の
メリットが活かせない技術者

Services

Case2：COBOL技術者をJava技術者に転換したい

成功･失敗の分かれ目

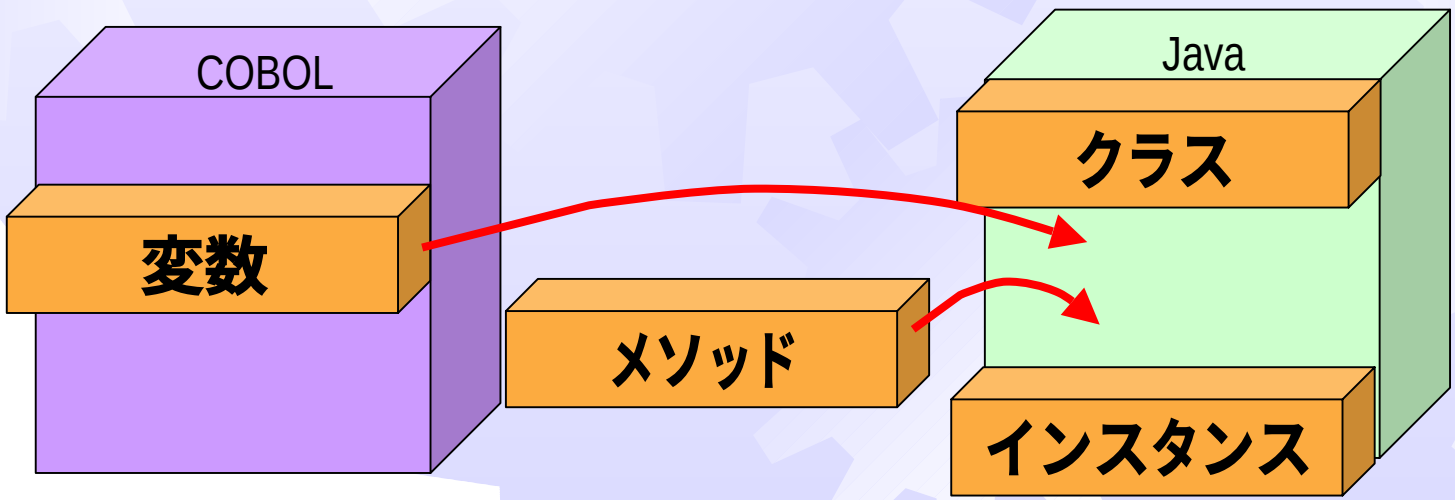
- COBOL(手続き型)からJava(オブジェクト指向)への**考え方の切り替え**に失敗した
  - COBOLの土台の上に無理やりJava言語の知識を当てはめようとした



Case2：COBOL技術者をJava技術者に転換したい

対処法

- まず**考え方の転換**を行う
  - オブジェクト指向は「**考え方**」。受講者に考え方の違いを気づかせること
  - 手続き型の考え方は、オブジェクト指向にも応用できる



オブジェクト指向教育現場の状況

Case３：即戦力となる技術者を育成したい

- 背景
  - 「研修後の技術者は即戦力として役に立たない」
    →現場の声
  - 今年度の受講者は、JavaとCOBOLを組み合わせた教育カリキュラムを実施した
  - 来年度の受講者に対しては、教育効果が現場でも認められるようなカリキュラムを実施したい
    →人材育成担当者

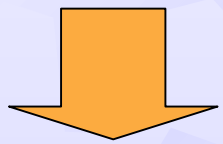
研修期間や内容を増やす方向で
トレーニングを計画

Services

Case3：即戦力となる技術者を育成したい

対応方法

- 前年度のカリキュラムよりも研修期間を長くする
  - 研修期間中に、オブジェクト指向のより深い内容を学ばせる
  - プログラミング言語をJavaのみに絞る



結果、研修期間を増やしたため
受講者からの評価は高くなった・・・

結果の考察

- 現場の声
  - Javaのプログラムは書けるようになった
  - しかし、即戦力としてそのまま現場に投入できるレベルではない

即戦力としての技術力までは
身につかなかった

Services

Case3：即戦力となる技術者を育成したい

成功・失敗の分かれ目

- **講座のトピックだけ**に注目していた
  - 実践的で使えるスキル習得を意識していない
- **講座だけ**で何とかなると考えている
  - **時間だけを増やして**も効果が出ない
  - 講義後も学習を**継続する習慣**がない

Services

Case3：即戦力となる技術者を育成したい

対処法

- 実践的なスキルを考慮する
  - 構成管理、テスト、演習題材、コミュニケーションスキルなど
- 継続して学習する習慣やしくみを作る
  - 受講者の**継続学習を支援**するため、経験のある技術者をメンターにする
  - 講義後に学習を**継続する習慣**を身に付ける

Services

成功と失敗の分かれ目とは

豆蔵
mamezou

成功と失敗の分かれ目とは（目次）

- カリキュラム作成時
- カリキュラム実施後

成功と失敗の分かれ目とは

- カリキュラム作成時
  - **段階を踏んだ教育**をすること
  - 現在の**スキルを把握**すること
  - オブジェクト指向の**考え方を重視**すること
  - **講座だけで即戦力**の育成を考えないこと
- カリキュラム実施後
  - 現場での受講者**受け入れ体制**をつくること

まとめ

知識は外から与えられるもの
技術は自分自身で育て上げるもの

技術を伝える教育・・・ではなく
技術者を育てる教育が重要

Services

失敗しない
オブジェクト指向教育のために

失敗しないオブジェクト指向教育のために(目次)

- Case1
  - 早く新技術を習得させるには？
- Case2
  - COBOL技術者をJava技術者に転換するには？
- Case3
  - 即戦力となる技術者を育成するには？

Services

失敗しないオブジェクト指向教育のために

早く新技術を習得させるには？

「与える教育」ではなく、「根付かせる教育」で実践する

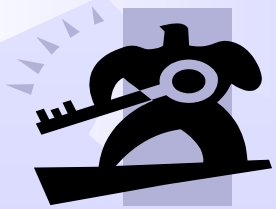
- 「与える教育」とは？
  - 受講者の状況に関わらず、提供側の都合で**知識**を伝えようとする**提供者中心の教育**
- 「根付かせる教育」とは？
  - スキル習得状況を常に確認しながら、受講者に**技術**を根付かせる**受講者中心の教育**
  - アセスメントや確認テストを通して、受講者の現状把握（受講前→受講中→受講後）が重要

Services

早く新技術を習得させるには？

「根付かせる教育」のために

- 講師スキルが成功の鍵
  - 技術は受講者が**受講後の効果**を意識しなければ身につかない
    - 受講者にその効果を意識させることができる
  - 受講者の**状況によって対応を変える**ことができること
    - 受講者によって理解できるツボが違うため、1つのトピックについて2つ以上の例え話ができる
    - 咀嚼して伝えることができる
- 「根付かせる教育」は簡単にショートカットできない！



失敗しないオブジェクト指向教育のために

COBOL技術者をJava技術者に転換するには？

技術を根付かせる土台（下地）が作りが重要

- 技術を学ぶときの受講者の誤解
  - 受講者が現在持っている技術体系に新技術を当てはめると近道（特に中堅技術者に多い）
  - 結果、当てはまらないオブジェクト指向技術のトピックについては理解に拒否反応を示す傾向あり

COBOL技術者をJava技術者に転換するには？

新しい技術を根付かせるために

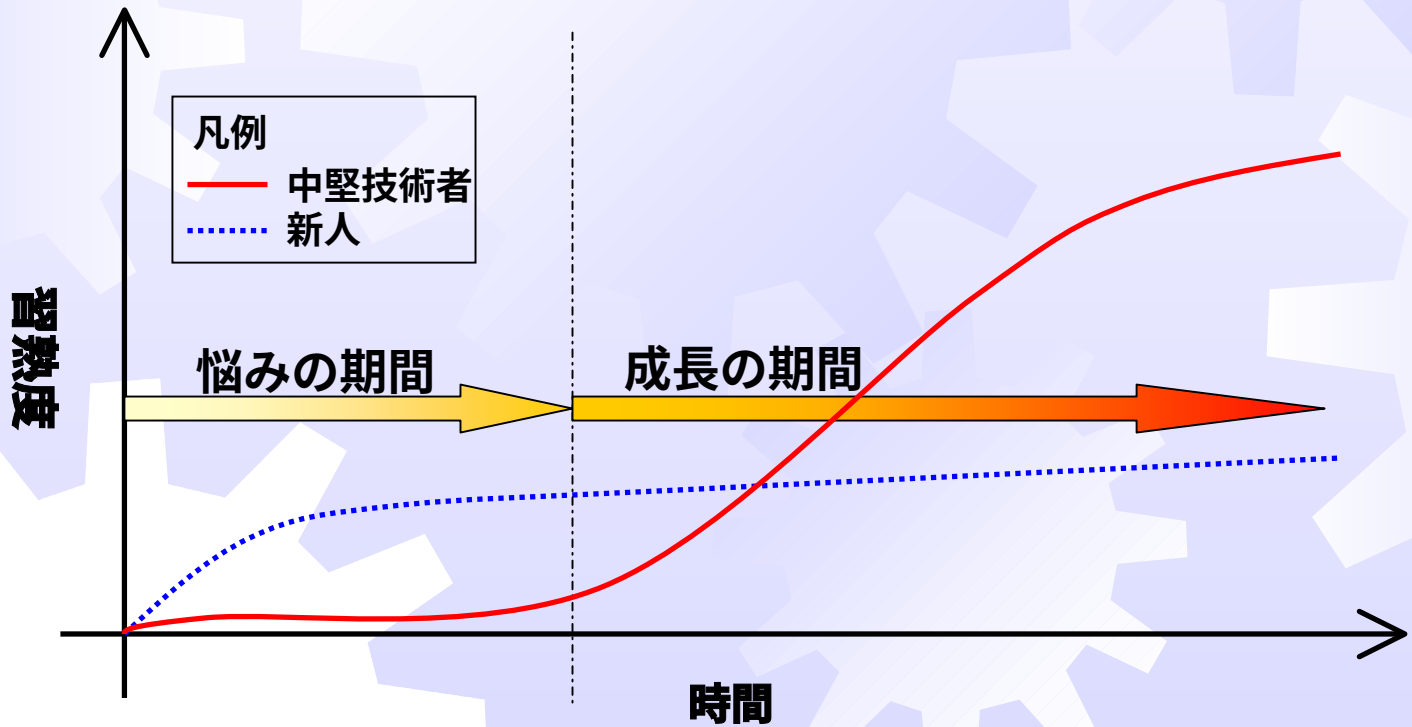
- 新しい技術を根付かせるための新たな土台を準備する
  - 新しい技術の土台ができれば、そこにこれまで身に付けていた技術と共通する部分についてはそのまま流用できることに気づき、以降はスキルの伸びが顕著になる（次のスライド参照）

Services

COBOL技術者をJava技術者に転換するには？

新人と中堅技術者との違い

- オブジェクト指向が理解できるまでの時間と習熟度の関係



失敗しないオブジェクト指向教育のために

即戦力となる技術者を育成するには？

「メンタリング」を使用して教育効果を高める

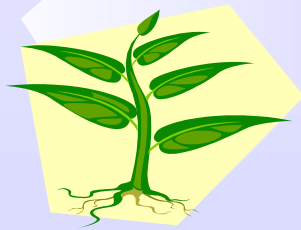
- 「メンタリング」とは？
  - 知識や経験が豊かな人（メンター）が、現時点において未熟な人に対してさまざまな側面から一定期間継続して支援を行うこと。

Services

即戦力となる技術者を育成するには？

講義でできることは・・・

- オブジェクト指向という新しい大地(土台)に種を蒔き**新しい芽**を発芽させるお手伝いをするところまで
  - 新技術の土台作り
  - 考え方の習得など
- 講義で学ぶことが技術者として必要なスキルの全てではない



即戦力となる技術者を育成するには？

教育効果を高めるために

- 講義後のフォローが重要
  - 特に初学者には講義終了後も継続して「メンタリング」によるサポートが必要
    - オブジェクト指向技術を身に付けた技術者を育成するには、新しい芽が大地に根を張り太い幹になるまで大切に木を育てていくのと同じぐらいの配慮が必要ということ

Services

効果的な教育のために

豆蔵
mamezou

効果的な教育のために (目次)

- チェックポイント1
  - 講師スキル
- チェックポイント2
  - コンテンツ
- チェックポイント3
  - アセスメント

効果的な教育のために

チェックポイント1

- 講師スキル
  - エンジニアとしてのスキルと教える人としてのスキルが両立しているか？
    - 特に初学者には受講者が何が分からないかを把握する能力、教えすぎない能力が重要

効果的な教育のために

チェックポイント2

- コンテンツ
  - 考え方を重視しているか？
    - JavaならWeb開発の前にJavaの特性を理解するカリキュラムが盛り込まれているか？
    - 特定プラットフォーム向けに、そのプラットフォームを生かした設計・実装を行うカリキュラムが盛り込まれているか？
  - 技術者の役割にあわせたオブジェクト指向学習カリキュラムが提供されているか？
    - プロジェクト管理者、設計者、実装者など

効果的な教育のために

チェックポイント3

- アセスメント
  - 受講の**前提条件**と**事後条件**が満たされているか？
    - 事前条件
      - プログラミング能力･モデリング能力を客観的にチェックし、適切な講座を受講する
    - 事後条件
      - 受講後、講座受講の効果を測定する

アセスメント

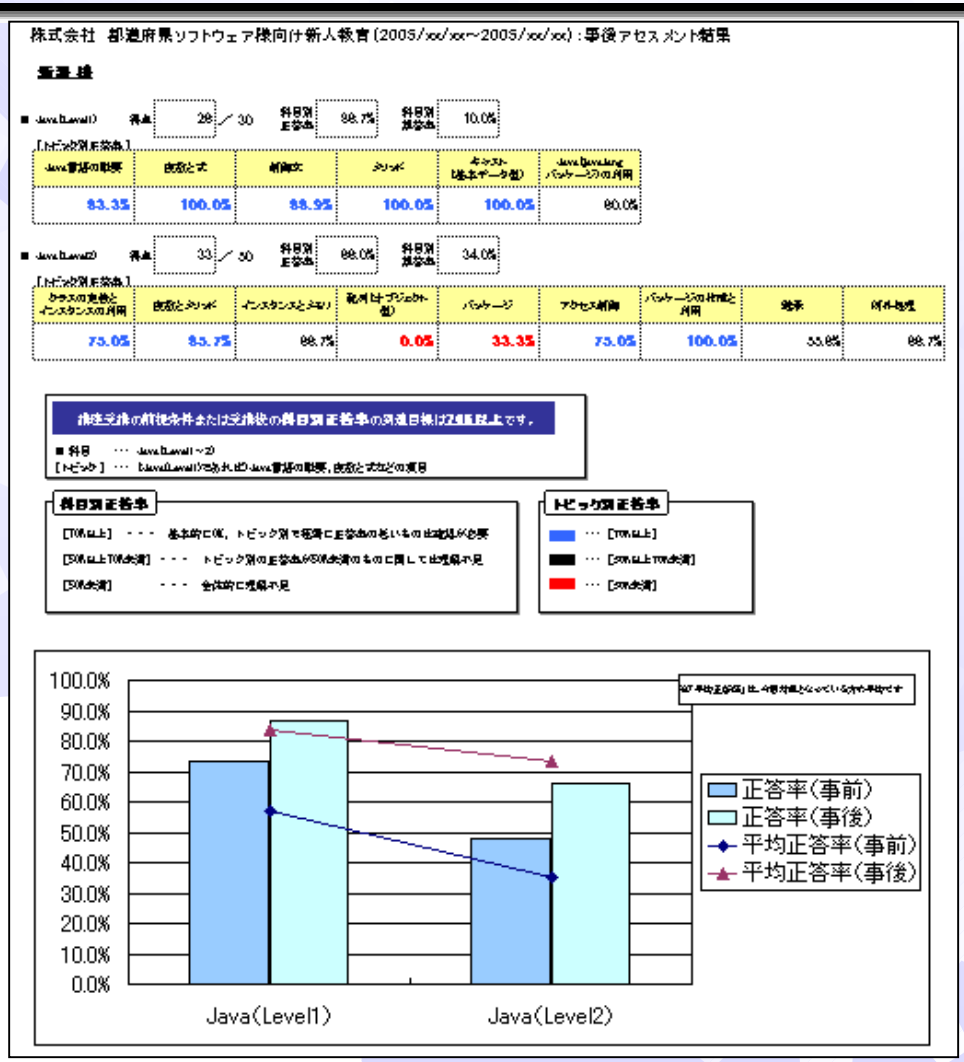
実施結果

- プログラミングアセスメント (Java)
- モデリングアセスメント (UML)

Services

アセスメント

プログラミングアセスメント (Java)



Services

アセスメント

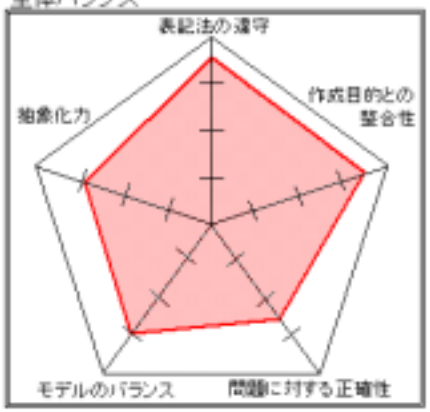
モデリングアセスメント結果 (UML)

アセスメント対象者
A 様

アセスメント日
2005年8月1日

回答モデル
販売情報管理システム

全体バランス



総合評価
潜在的なモデリングスキルをお持ちですが、まだ本質を掴むことに不慣れなようです。抽象化力はモデリングにおける最も重要な能力ですので、良いモデルを参考に、モデリングを繰り返し、重要な概念群とその関係を掴む訓練を継続してください。

表記法の遵守
UMLの仕様に則って描かれており、問題ありません。正しく表記法(Notaion)と意味(Semantics)を理解されています。

作成目的との整合性
関連に誘導可能性が付記されていることから、若干設計よりの視点になっているように見受けられます。概念モデルは重要な概念間の静的構造を表すものですので、この段階ではメッセージの流れる方向まで吟味する必要はありません。

問題に対する正確性
メーカと商品の関係が商品を全体側とした集約となっていますが、全体と部分というよりは対等な関係とみなした方が自然です。また、販売に日付を表す属性が不足しているため、日次の売上集計ができないモデルになっています。日締売上を保存する場合、同様の理由で集計結果の識別ができません。販売には属性「商品別○○」がありますが、属性に多重度が記述されていない場合、属性名に関わらず0.1と見なされるため、商品の購入数を記録できません。

モデルのバランス
適度に責務が分散されていますが、販売は複数の責務を抱えているように見受けられます(明細の概念が混在)。また、日締売上から販売台帳へのロール名が売上明細となっていますが、このモデルでは販売台帳はシステムに1つしか存在しないため、関連自体もしくはロール名が不適切であるといえます。これらの点でバランスを崩した結果、やや分かりづらいモデルになってしまっています。

抽象化力
ほぼ必要なクラスは抽出できています。「モデルのバランス」でも言及したように、販売が複数の責務を負っています。明細を表す概念は別クラスとすることで、商品別の販売数や小計を保持することができます。
また、クラス名やロール名の命名にも気を配ると良いでしょう。

Services

まとめ

- 適切なタイミングで適切な教育が重要

低 ―― 技術者としてのキャリア ――→ 高

ポイント	動機付け	実践的スキル習得	コンサルティングサービス
	基礎スキル習得	アーキテクチャ構築技術習得	技術実践/適用
キー技術	・分析設計手法 ・イディオム ・UML基本 ・プロセス ・プロジェクト管理	・アーキテクチャ構築 ・反復開発実践 ・デザインパターン ・フレームワーク ・ツール	・開発・管理プロセス定義支援 ・アーキテクチャ構築支援 ・フレームワーク構築支援 ・再利用プロセス定義支援 ・組織作り支援
教育サービス内容	・基礎技術要素教育 ・アセスメント	・仮想プロジェクト ・メンタリング ・モデルレビュー	

Services

ご清聴ありがとうございました