

Seasar Conference 2007 Spring



半歩先行くSeasar2の実践活用

2007.5.27

株式会社 ティーアンドエフカンパニー

出羽 健一 (dewa@tafc.co.jp)

- 名前

- 出羽 健一

- 所属

- 株式会社 **ティーアンドエフカンパニー**

- (<http://www.tafc.co.jp/>)

- 日本初のコンピュータ専門大学で有名な会津大学の卒業生が中心となって作ったWebアプリ開発会社

- 普段は、CTOとしてWeb+DBアプリの開発に従事

- 金沢工業大学 大学院 客員准教授 として「サーバーサイドJava特論」の授業も担当

- キャンパスは東京の虎ノ門



- 最近の執筆活動・メディア紹介記事
 - WEB+DB PRESS vol.38
 - 特集1:「無駄なコードを**書かない技術**」



– ITpro

- 「HTML画面をそのまま仕様書に」, **5カ月で1000画面**を構築した就職サイトPuffの高速開発手法
(<http://itpro.nikkeibp.co.jp/article/COLUMN/20070214/261859/>)

– コンポーネントスクウェア コラム記事

- Javaコンポーネント技術解説: **Seasar2**
(<http://www.c-sq.com/modules/cbdsqcont/index.php?category=8>)

- Seasarプロジェクトとの関わり
 - Seasar Conference 2006 Autumn で登壇
 - 「突撃！隣のSeasarプロジェクト
～現場で役立つ実践Tipsを教えてください～」
(<http://event.seasar.org/sc2006autumn/Session>)
 - 一応コミッターなんですが、
あまりコミットできていません m(__ __)m
 - 本日は業務でSeasarに触れる機会が多い
1ユーザーとしてお話を聞いて頂ければ幸いです



- **第1部**

Seasarは進化が早い。今のトレンドは？

～ サポート用メーリングリストによる傾向分析 ～

- **第2部**

メーリングリストで活発なプロダクトを
組み合わせたアプリケーション開発のデモ

- **第3部**

フレームワークの有効活用には足回り重要

～ 半歩先行く開発インフラ ～

第1部

Seasarは進化が早い。今のトレンドは？
～ サポート用メーリングリストによる傾向分析 ～



- 「半歩先行く」の意味するもの

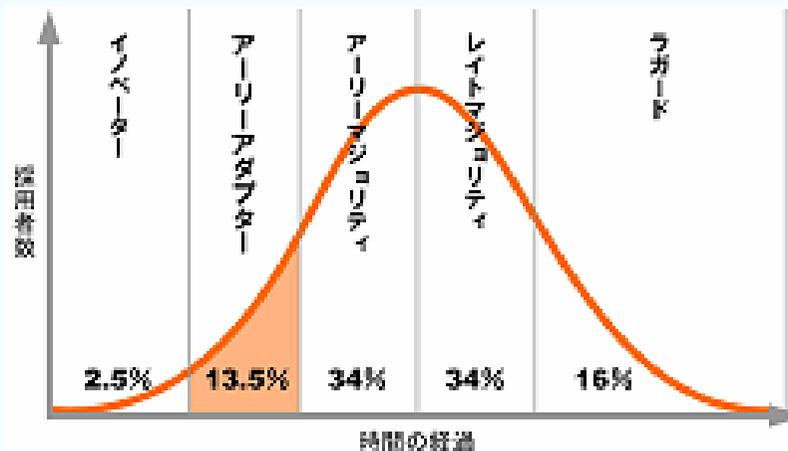
⇒ 技術を採用する際の方針：

「**ちょっと新しい**位が丁度良い」

－なぜなら、

- 新しすぎる技術 ⇒ リスクが高い
- 古すぎる技術 ⇒ ライバルに負ける

- 半歩先行く人は、
「ロジャーズの普及モデル」における
アーリーアダプターに相当



ロジャーズの採用者分布曲線(出所:「イノベーション普及学」)

<http://www.atmarket.co.jp/aig/04biz/earlyadapter.html>

	Seasar採用者	一般的説明
イノベータ (革新的採用者)	コミッター。	冒険的で、最初にイノベーションを採用する。
アーリーアダプター (初期採用者)	MLでの発言者。 S2関連ブログを書く人。	自ら情報を集め、判断を行う。多数採用者から尊敬を受ける。
アーリーマジョリティ (初期多数採用者)	MLの購読者。 S2関連ブログを読む人。	比較的慎重で、初期採用者に相談するなどして追隨的な採用行動を行う。
レイトマジョリティ (後期多数採用者)	デファクトのStruts, Spring等の採用者。	うたぐり深く、世の中の普及状況を見て模倣的に採用する。
ラガード (採用遅延者)	オープンソース否定論者。	最も保守的・伝統的で、最後に採用する。



Seasarにおけるロジャーズの普及モデル③

- ロジャーズの普及モデル
 - 最も重要だとされるのがアーリーアダプター
 - SeasarにおけるML発言者とブログを書く人

 - イノベーターとアーリーアダプターの総数が16%を超えると急激に普及・拡大する



サポートメーリングリストによる傾向分析①

- Seasarプロダクト群を知らない人から見た印象
 - プロダクト数が多い(約70種類)
 - どのプロダクトが良いのか? 使われているか? が分かりにくい
 - トレンドの変化が早い



- イノベータとアーリーアダプターの情報交換が頻繁に行われているSeasarのサポート用メーリングリストをプロダクト別に集計し、その分析を通じて**プロダクトのトレンドを探る**



サポートメーリングリストによる傾向分析②

- 分析対象のメーリングリスト

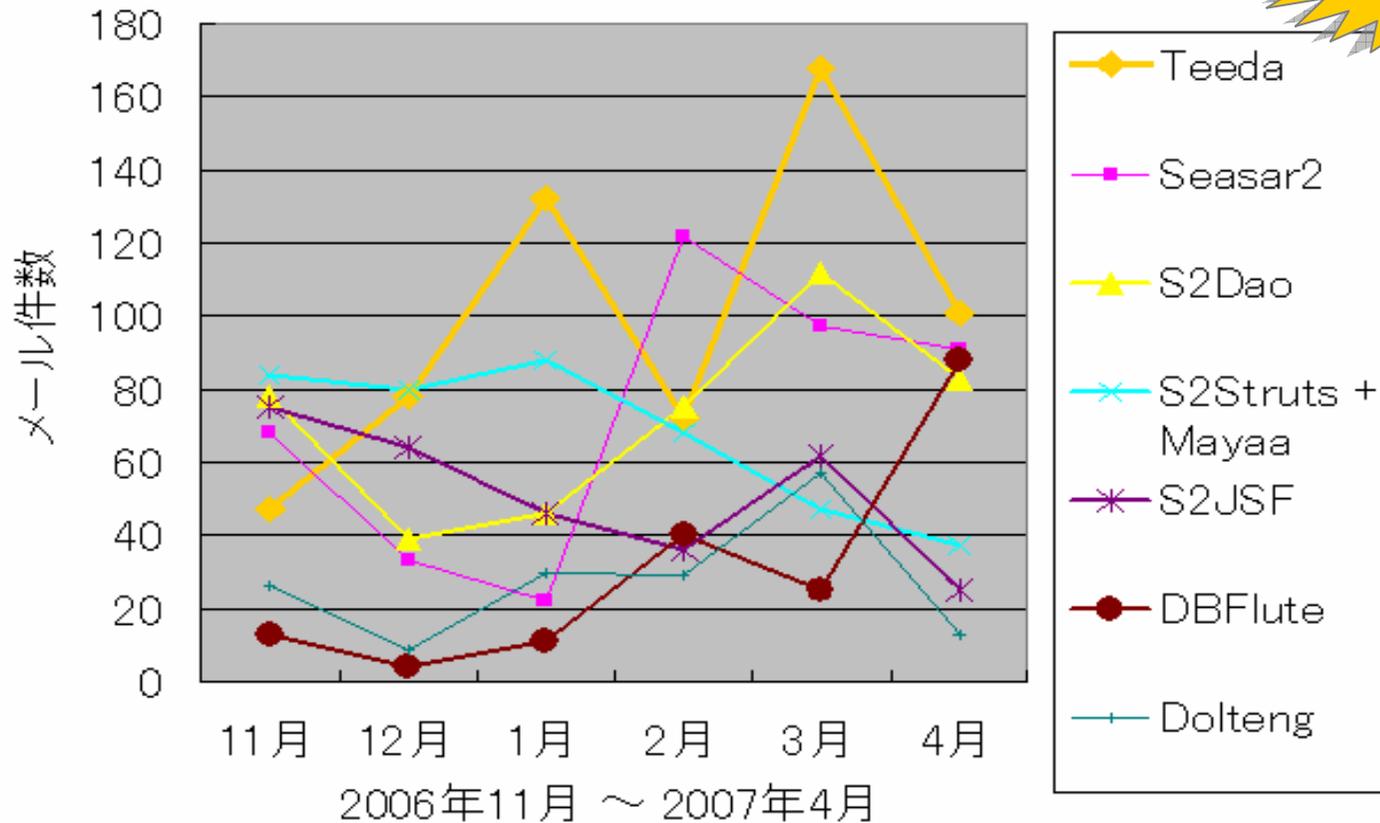
名称	Seasar-user
メールアドレス	seasar-user@ml.seasar.org
目的・用途	Seasarを使っていて困ったこと・ 質問・要望など
案内ページ	http://ml.seasar.org/mailman/listinfo/seasar-user/

ぜひ、活用しよう！！

サポートメーリングリストによる傾向分析③

seasar-userのメーリングリストを製品別に集計

Top 7



※ 1ヶ月の平均が20件を超えていない製品は省略

※ S2StrutsとMayaaについては、組み合わせて使うことが多いので、便宜的に1つの製品として集計

※ Mayaaはmayaa-userのメーリングリスト(<https://ml.seasar.org/mailman/listinfo/mayaa-user>)の件数を集計

- グラフから読み取れる内容(その1)
 - プレゼンテーション層のプロジェクト
 - 最もホットなプロジェクトは「Teeda」
 - データアクセス層のプロジェクト
 - 依然として「S2Dao」が強い
 - S2Dao以外のドメイン層のプロジェクト(KuinaやS2Hibernate など)はあまり話題に上っていない
 - Sandboxの「DBFlute」がTop 7入り&上昇傾向

※ JPAメーリングリスト

<https://ml.seasar.org/mailman/listinfo/jpa>

※ DBFluteはS2Daoを支援する目的のツール

<http://dbflute.sandbox.seasar.org/>



サポートメーリングリストによる傾向分析⑤

- グラフから読み取れる内容(その2)
 - 全体の傾向
 - 去年(2006年)の11月に正式リリースされたばかりの Chura(Super Agile)系のプロダクトが上位7件中4件を占めている

※ Chura(Super Agile)系のプロダクト : Seasar 2.4 + Teeda + S2Dao + Dolteng

第2部

メーリングリストで活発なプロダクトを
組み合わせたアプリケーション開発のデモ



- メーリングリストで盛り上がっている
プロダクトを組み合わせたデモ
- 使用するプロダクト

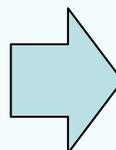
ベースフレームワーク	Seasar 2.4
プレゼンテーション層	Teeda
データアクセス層	S2Dao + DBFlute
Eclipseプラグイン	Dolteng

- 画面仕様

売上検索

売上日: ~

商品名: (前方一致)



売上検索結果一覧

売上日: 2007/05/01 ~ 2007/05/27

商品名: 酒

売上日	顧客名	商品名	単価	販売数	金額

54件中21件~40件を表示中

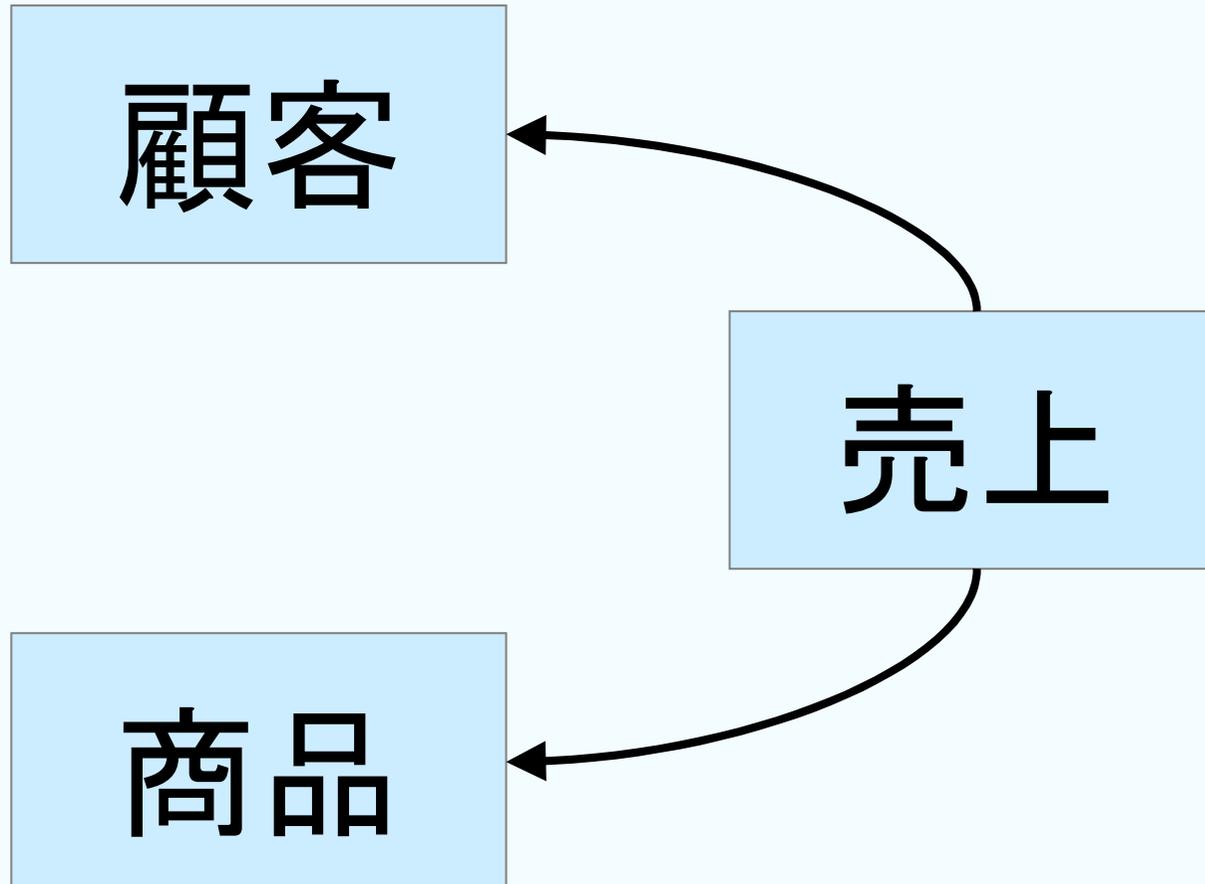
search.html

(⇒ SearchPage.java)

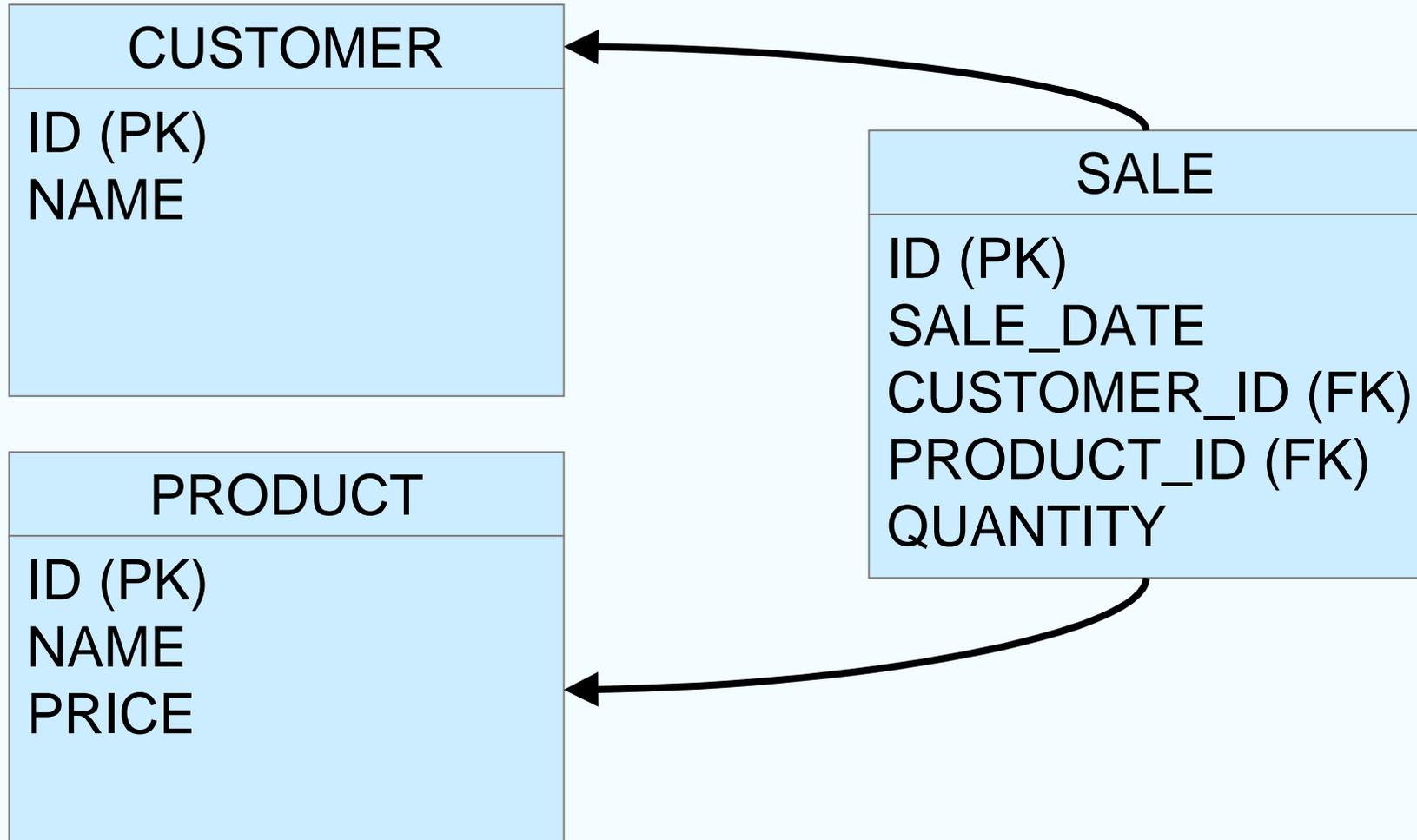
resultList.html

(⇒ ResultListPage.java)

- ERD(概要)



- ERD(詳細)



- この仕様をどの位の時間で実装できますか？
- 前提：
 - 画面のHTMLあり
 - DBあり
 - 好きなフレームワークの使用OK
 - プロジェクトは生成済み
 - 個別部分の実装のみ

- 選択肢

—A : 半日以内

—B : 1日以内

—C : 2日以内

—D : 3日以上

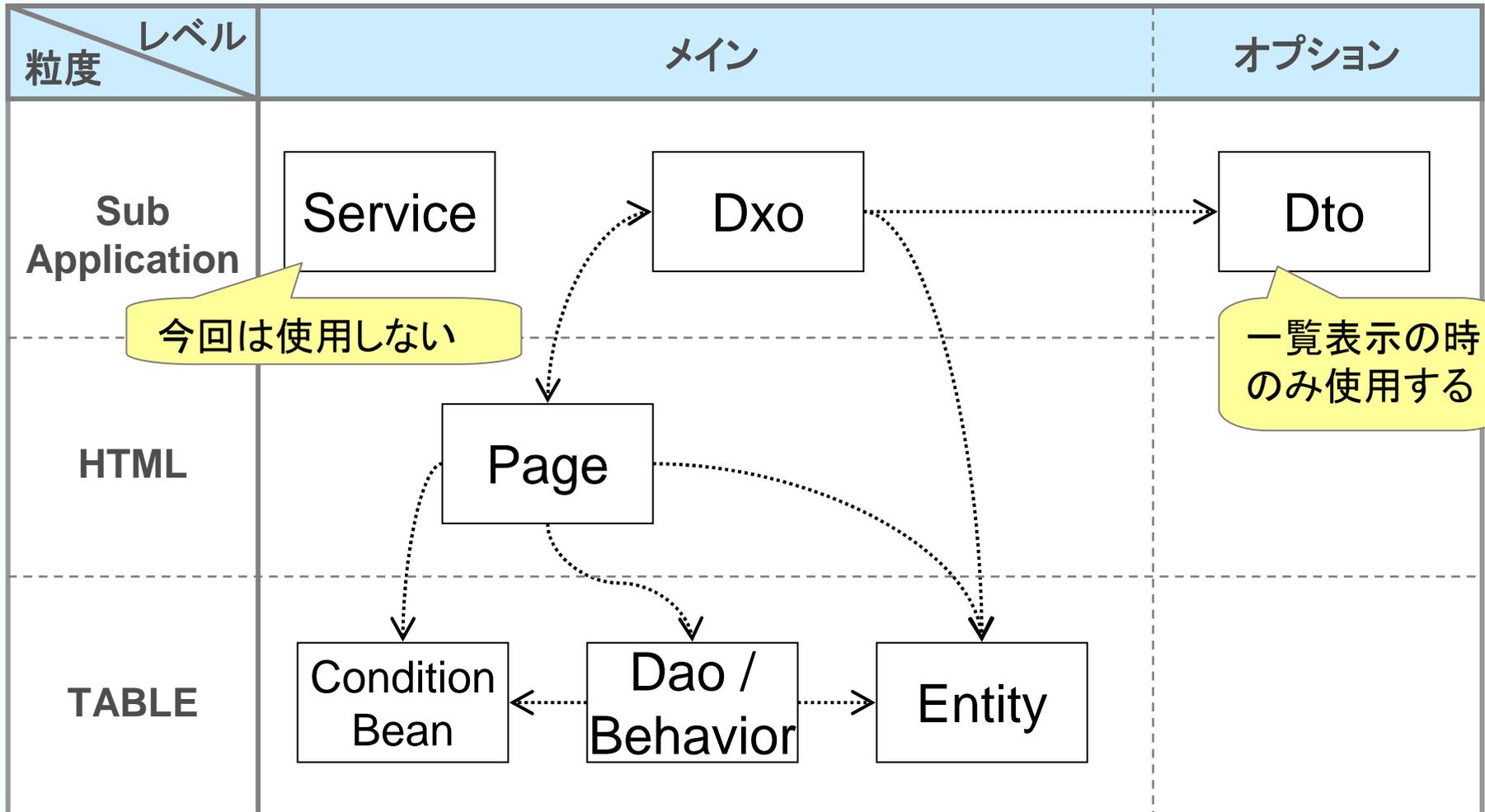
- 答え:

- 私の答えは、

- これから行うデモに掛かった時間
です。

⇒ 当日は約20分で
コーディング完了！！

• クラス図 (概念レベル)



※ ConditionBeanとBehaviorはDBFlute専用のクラス

• クラスの説明

Page	画面項目のプロパティと画面遷移、ロジックを扱う。	Doltengで雛形を自動生成
Dto	Teedaでは主に、一覧表示で一行分のデータを格納するために使う。	
Dxo	画面周りのデータオブジェクトとDB周りのデータオブジェクトの変換。	-
Entity	DBテーブルのデータを格納。	DBFluteを使ってDBスキーマからコードを自動生成。 Generaton Gapパターン。
Dao	SQL文を実行する。	
Behavior	EntityとDaoの定番処理をドメインの振る舞いとして定義。Daoの委譲クラス。	
Condition Bean	SQL文の組み立てに必要な情報を扱う。	

- HTML(売上検索)

売上検索

売上日: ~

商品名:

日付入力コンポーネント
と認識される

売上日:
`<input id="fromSaleDate" type="text" class="T_date" /> ~`
`<input id="toSaleDate" type="text" class="T_date" />`

商品名:
`<input id="conditionProductName" type="text" />`
`<input type="button" id="doSearchSale" value="検索" />`

- Page(売上検索)

売上検索

売上日: ~

商品名:

```
public Class doSearchSale() {  
    return ResultListPage.class;  
}
```

「検索」ボタンを押すと、HTML上のボタンのid名と同名のメソッドが呼ばれる

画面遷移の方法:
遷移先ページのクラスを戻り値に指定する

- HTML (売上検索結果一覧)

売上検索結果一覧

売上日: 2007/05/01 ~ 2007/05/27

商品名: 酒

売上日	顧客名	商品名	販売数	単価	金額

54件中21件~40件を

前のページ
次

売上日:
``
 ~
``

商品名:
``

売上検索ページと同名のプロパティ名にしておけば、値を自動で引き継ぐ

• HTML(売上検索結果一覧)

売上検索結果一覧

売上日: 2007/05/27
商品名: 酒

売上日	顧客名	商品名	単価	数量	金額

54件中 1件 ~

前のページ

```

<table border="1" >
<thead>
<tr>
<th>売上日</th><th>顧客名</th><th>商品名</th><th>単価</th><th>数量</th><th>金額</th>
</tr>
</thead>
<tbody id="saleItems">
<tr>
<td><span id="saleDate"> 2007/05/27 </span></td>
<td><span id="customerName"> ほげ商店 </span></td>
<td><span id="productName"> 酒 </span></td>
<td><span id="price"> 350 </span>円</td>
<td><span id="quantity"> 2 </span>個</td>
<td><span id="amount"> 700 </span>円</td>
</tr>
</tbody>
</table>

```

繰り返し:
繰り返し対象を囲むタグの
idが"Items"で終わる必要
がある。

繰り返し単位でDtoを用意

- Page (売上検索結果一覧)

- prerenderメソッド内の処理 (その1)

```
/* 検索条件オブジェクトの作成 */
```

```
SaleCB saleCb = new SaleCB();
```

```
saleCb.setupSelect_Customer();
```

```
saleCb.setupSelect_Product();
```

```
/* 顧客テーブルと結合 */
```

```
/* 商品テーブルと結合 */
```

実は、かなり便利

```
/* 条件: 売上日の範囲指定 */
```

```
saleCb.query().setSaleDate_FromTo(fromSaleDate,  
toSaleDate, new FromToOption().compareAsDate());
```

```
/* 条件: 商品名の前方一致 */
```

```
saleCb.query().queryProduct().setName_LikeSearch(  
conditionProductName, new LikeSearchOption().likePrefix());
```

- Page (売上検索結果一覧)
 - prerenderメソッド内の処理 (その2)

```
/* 並び順 */
```

```
saleCB.query().addOrderBy_SaleDate_Desc();
```

```
/* 検索の実行 */
```

```
List<Sale> sales = saleBhv.selectList(saleCB);
```

```
/* リストの表示 */
```

```
this.saleItems = salesDxo.convert(sales);
```

List<SaleDto>

- SalesDto

```
public class SalesDto {  
  
    private Date saleDate;  
    private String customerName;  
    private String productName;  
    private Integer price;  
    private Integer quantity;  
    private Integer amount;  
  
    << Setter / Getterは省略 >>  
  
}
```

- Dxo

```
public interface SalesDxo {  
  
    @ConversionRule("customerName : customer.name, " +  
                   "productName : product.name")  
    List<SalesDto> convert(List<Sale> sales);  
  
}
```

プロパティ名が異なる場合は、
ConversionRuleアノテーションで一致させる

『変換先のプロパティ : 変換元のプロパティ』



- ポイント

1. 初期プロジェクトを生成

- 拡張されたオリジナルプロジェクト (Dolteng)
 - <http://d.hatena.ne.jp/taichitaichi/20070330/1175273524>
- 洗練された標準パッケージ構成 (Seasar 2.4)

2. コードを変更してもTomcatの再起動は不要

- ホットデプロイ (Seasar 2.4)



- ポイント

- 3. 画面遷移

- Pageクラスのメソッドの戻り値で指定 (Teeda)
 - **同名プロパティ値の自動引継ぎ** (Teeda)
 - 現在ページとURL表示がズレない (Teeda)

- 4. コード自動生成

- HTMLからPageクラスを生成(Dolteng)
 - DBスキーマからDB関連クラスを生成(DBFlute)



- ポイント
- 5. タイプセーフなDBプログラミング
 - インテリセンスでメソッドを選択するスタイル(DBFlute)
 - SQL文は1行も書いていない(DBFlute)
- 6. ページング処理
 - とてもシンプルに実現(Teeda + DBFlute)

第3部

フレームワークの有効活用には足回り重要
～ 半歩先行く開発インフラ ～



- フレームワークを最大限に活用するには足回り(開発インフラ)の整備が重要
- 開発インフラが脆弱だと、「ルーチンワーク」と「トラブルシューティング」で創造的な活動をする時間が奪われてしまう
- そこで、開発インフラには次の要件が重要
 - 周辺タスクの自動化
 - トラブル発生率を低くする



フレームワークだけの知識では不十分②

- 主な開発インフラの要素
 - 統合開発環境 (IDE)
 - Eclipse (現時点でのオススメはWTP 1.5)
 - ソースコード管理システム
 - Subversionなど
 - ビルドツール
 - Maven2 (ソースコード関連の周辺タスクを自動化)
 - 継続ビルドサーバ
 - Continuum など

• 開発インフラ要素の役割・ポイント

IDE (Eclipse)	<ul style="list-style-type: none">•プロジェクト専用Eclipseの準備•ローカルPC上のディレクトリ構成もメンバー間で統一•プラグインの選定
ソースコード管理 (Subversion等)	<ul style="list-style-type: none">•トランク・ブランチ方式の採用•マージ操作スキル重要
ビルドツール (Maven2)	<ul style="list-style-type: none">•Eclipse, Tomcat等との共存環境を整備•ビルド・テスト・デプロイ・レポート生成タスクの自動化•ライブラリの管理
ビルドサーバ (Continuum等)	<ul style="list-style-type: none">•自動化タスクの定期的実施•タスク実施結果の通知

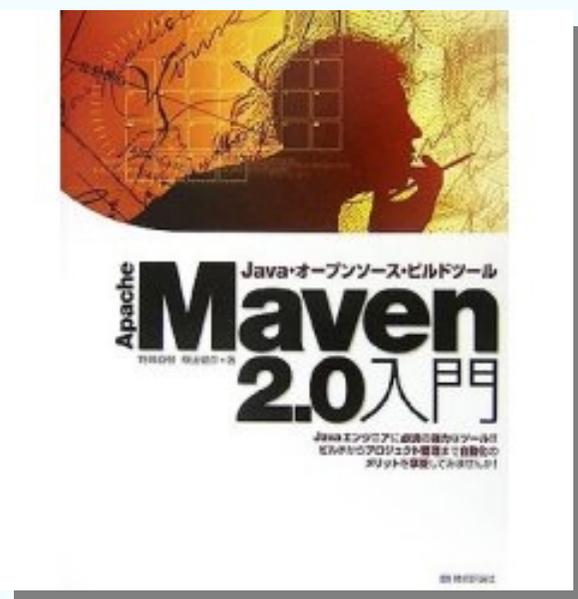
- Maven2とは？

- Apache Foundation Software が開発しているソフトウェアビルドツール
- コンパイル、テスト、jarやwar作成、デプロイ、各種レポート生成、サイト生成などのタスクを自動化
- Java系オープンソースのプロダクトではデファクトとして受け入れられている
- 効果は高いが、導入障壁も高い

- どのような導入障壁があるの？

- 高度な抽象化がなされているが、内部的な仕組みを知らないと使いこなせない
- XML形式の設定ファイルを直接編集できるレベルのスキルが要求されてしまう
- 各プロダクトの相性を考慮しなければならない
- 多くのバッドノウハウを必要とする
- 情報源が少ない

- この本は自信を持ってオススメ！
 - Apache Maven 2.0入門
Java・オープンソース・ビルドツール
 - [野瀬 直樹](#) (著), [横田 健彦](#) (著)



- 万人に受け入れられるツールではなさそう
 - 一部のパワーユーザーには手放せないほど、受け入れられている
 - 覚悟が足りない人は手を出さない方が無難
- 徐々に安定した開発環境が整いつつあるので、一度はチャレンジすべき！！

- Maven2を使ってEclipseで使うJarファイルのバージョンアップ

1. pom.xmlを編集

```
<dependency>  
  <groupId>org.seasar.teeda</groupId>  
  <artifactId>teeda-core</artifactId>  
  <version>1.0.4</version>  
</dependency>
```

2. コマンドを実行

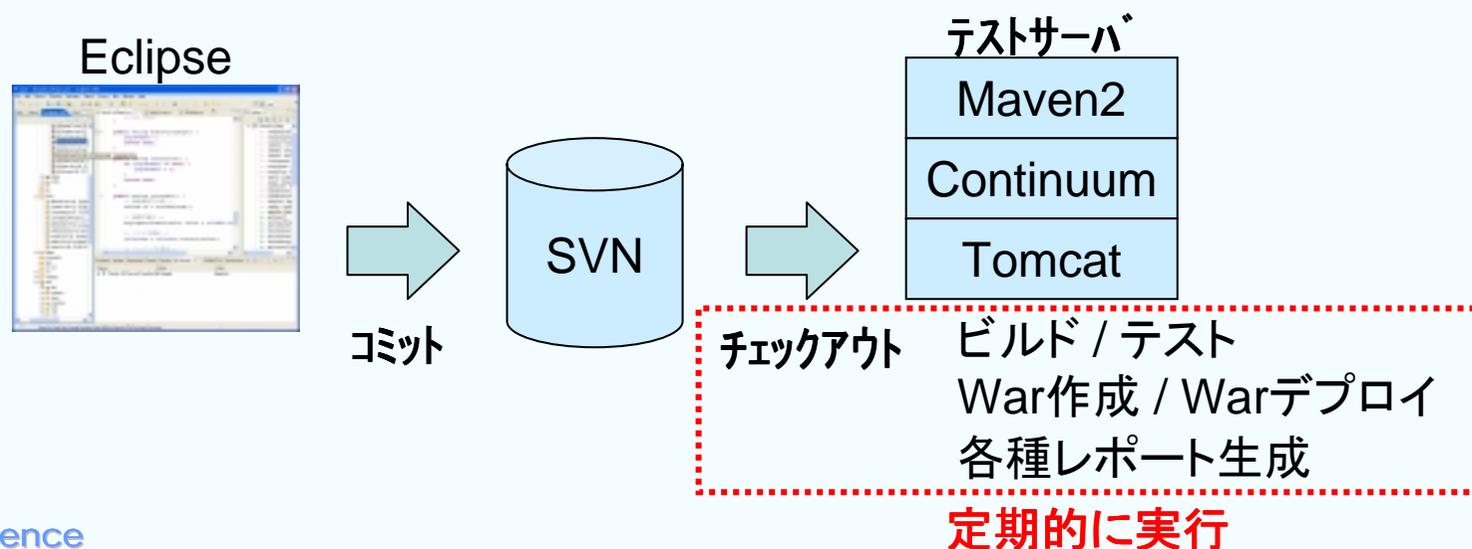
```
> mvn eclipse:eclipse
```

例えば1.0.4 から
1.0.6 に書き換える

3. (F5などで)Eclipseの状態を最新にする

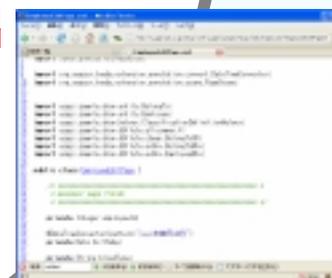
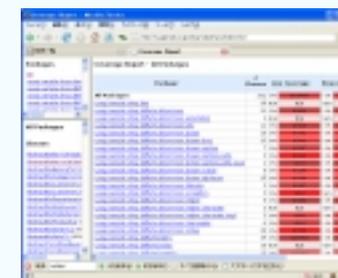
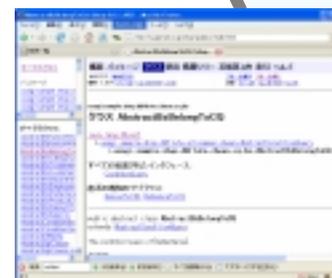
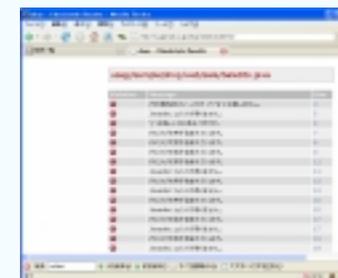
- 自動でインターネット経由でjarをダウンロードして、Eclipseの設定ファイルまで更新！

- Maven2による周辺タスクの自動化
- Subversionにコミットするだけで、定期的にビルド・テスト・war生成・デプロイまでを自動的に行ってくれる様子を見てみよう！



- Maven2が生成する各種レポート

プロジェクト情報
Javadoc
コードブラウザ
コーディング規約チェック
単体テスト結果
カバレッジ
重複コード発見
静的コード解析によるバグ検出
ToDoリスト
その他





ご清聴
ありがとうございます
ございました