

# Seasar Conference 2006 Autumn



## Seasar2.4

Seasarプロジェクト/チーフコミッタ  
電通国際情報サービス/統括マネージャ  
ひがやすを



## 最近のWeb開発の二大潮流

- Agile
  - 代表はRuby on Rails
  - スピード感を重視
  - 独自仕様で徹底的に生産性を追及
- Enterprise
  - 代表はEJB3、JPA、JSF
  - 大手ベンダー間で仕様作成に時間をかける
  - できた仕様はベンダーの思惑の入った玉虫色
  - でも標準という安心感



## 最近のWeb開発の二大潮流 - 2

---

- 開発者の立場として
  - 独自仕様でイノベーションを徹底的に追求したい
- ユーザの立場として
  - 標準という安心感を求める気持ちもわかる



- Super Agile
  - Agileを徹底的に追求
- Easy Enterprise
  - 標準仕様で使いにくいところを機能拡張して使いやすくする
- 軸足はSuper Agile
  - Super Agileで確立した成果をEnterpriseのほうにフィードバックする



- Super AgileとEasy Enterpriseの  
共通の方向性
  - All in One
  - 規約重視
  - さくさく感のある開発



- All in One以前

- 複数のフレームワークを自前で組み合わせる
- 独立したフレームワーク同士を組み合わせるのはそれなりのノウハウが必要
- フレームワーク間で思想が統一されていない



- All in One後
  - フレームワークと開発哲学を1つにパッケージングして提供
  - Railsで効果が認められポピュラーに
  - 統一された哲学、手順をもとに開発
    - 迷うことが少なく開発が効率的
  - 開発、メンテナンスが1つのチームで行われているので安心感がある



- Chura(ちゅら)
  - Seasar2.4を中心にしたAll in Oneパッケージ
  - Dolteng(どうるてん)を通じて配布
    - DoltengはEclipseのPlugin
  - Super AgileとEasy Enterpriseをサポート





- Super Agileのプロジェクト構成
  - Seasar2.4
  - Teeda(ていーだ)
  - Uji(うーじ)
  - S2Dxo(えすつーだっくすおー)
  - Dolteng(どうるてん)
  - Diigu(でいーぐ)



- Easy Enterpriseのプロジェクト構成
  - Seasar2.4
  - Teeda(ていーだ)
  - KuinaDao(くいなだお)
    - S2Hibernate-JPA
    - Hibernate-JPA
  - S2Dxo(えすつーだっくすおー)
  - Dolteng(どうるてん)
  - Diigu(でいーぐ)



- 設定ファイル地獄
  - これまでのフレームワークは柔軟性を確保するために設定ファイルを使うことが多かった
  - 開発の規模が大きくなると設定ファイルを書くことが負担
- 規約重視で設定ファイルを減らす
  - 規約に従っていれば設定ファイルがなくてもフレームワーク側が自動的に処理
  - Railsの「Convention over Configuration」が有名



## さくさく感のある開発

- スクリプト言語の良さ
  - ファイルをサーバに置けばすぐに認識される
  - トライ&エラーを繰り返しながら「さくさくっ」と開発できる



- Javaの問題点
  - 面倒な手順をふまないと修正を認識できない
    - ソースをコンパイル
    - アプリケーションサーバにデプロイ
    - アプリケーションサーバを再起動
  - トライ&エラーを繰り返すと非効率
    - 最初から完成度を上げる必要がある
      - 心理的なプレッシャー
      - 開発が楽しくない
    - デプロイや再起動時の無駄な待ち時間の発生
    - 無駄な待ち時間で思考が中断されリズムが悪くなる



- まとめ

- Javaはさくさく感のある開発ができないので
  - 最初から完成度を上げるプレッシャー
  - デプロイや再起動時の無駄な待ち時間の発生
  - 無駄な待ち時間で思考が中断されリズムが悪化
- スクリプト言語はさくさくっと開発できるので
  - 開発効率が良いと感じる



- 硬い言語(Java)にも良いところはある
  - コンパイル時の最適化でパフォーマンスが向上
  - コンパイル時の型チェックで単純なコーディングミス  
を自動検出
  - IDEの強力な支援
    - コードの自動補完
    - リファクタリング
    - 保存時のインテリジェントなコンパイル



- Javaで
  - 「さくさく感のある開発」ができないことによる生産性の低下は高機能なIDEによる生産性の向上でカバー
  - 「さくさく感のある開発」が可能になったなら、スクリプト言語のメリットと硬い言語のメリットをどちらも得ることができ、かなり生産性が向上する





## さくさく感のある開発 - 6

- HOT deploy
  - Javaでさくさく感のある開発を可能にする
  - アプリの稼動中にクラスの変更を自動認識
  - スクリプト言語のメリットと硬い言語のメリットをどちらも得ることができ、かなり生産性が向上する



- ページ駆動開発
  - UI(ページ)を起点に開発する手法
  - GoyaにおけるUCD(User Centered Design)の具体的開発手法
  - 詳しくはTeedaのセッションで
- テーブル駆動開発
  - テーブルのメタデータを元にアプリケーションを自動生成
  - 業務的な画面から補助的に使うデータのメンテナンス画面



- 3分でできるWebアプリケーション
  - プロジェクトの作成
  - scaffoldの作成
  - HOT deploy
- Churaの詳細は下記のセッションで
  - Teeda
  - KuinaDao
  - Super Agile Web Development with Seasar2



Thank you!

ありがとうございました