

# Ruby On Rails

## ～ラクに開発できるのか？～

平成20年度 OISA 「技術研究会」  
Ruby On Rails部会 報告書

2009年 2月

## 部会員およびアドバイザー・技術委員

## 【部会委員】

(順不同)

川上 隆幸	大分シーイーシー(株)
佐藤 洋一	大分シーイーシー(株)
兒玉 清幸	大分大学
小野 可奈子	(株)富士通大分ソフトウェアラボラトリ
田中 康夫	九州東芝エンジニアリング(株)

## 【アドバイザー・技術委員】

(順不同)

築城 久敏	(株)システムトレンド
清水 太	(株)ワイズ・システムズ
三嶋 信太郎	大分ケーブルテレコム(株)
佐藤 清孝	(株)ウィルウェイ
安達 裕之	(株)システムトレンド

# 目次

## 1. はじめに

- 1. 1 はじめに
- 1. 2 活動目的

## 2. Ruby と Rails

- 2. 1 Ruby
- 2. 2 Rails
- 2. 3 国内／海外での動向

## 3. 検証・評価

- 3. 1 サンプルシステム作成
- 3. 2 Railsの利点／欠点

## 4. おわりに

- 4. 1 まとめ
- 4. 2 おわりに

## 参考資料

# 1. はじめに

# 1. はじめに

- 最近、**Ruby**や **Ruby On Rails** というキーワードを見たり、聴いたりしませんか？

Railsって  
スゴいらしいよ。

Javaの10倍の開発  
生産性がある！

本当に**ラク**に開発できるの？

## 1. 2 研究目的

### 【活動目的】

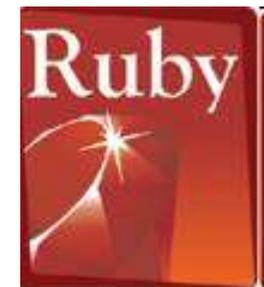
- Ruby On Railsの持つ開発生産性を調査
  - なぜ、支持を集めているのか？
  - 国内・海外での動向
  - ラクに開発できるのか？（開発生産性は高まる？）

本当にラクに開発できるの？

## 2. Ruby と Rails

## 2. 1 Rubyとは？

- インタプリタ型オブジェクト指向プログラミング言語
- まつもとゆきひろ氏によって開発された、国産言語
- オブジェクト指向、正規表現、ガーベージコレクタなど主要なプログラミング言語が持つ機能を実装
- 最新バージョンは、1.9.1



# 文法構造～Javaとの比較～

## ◆クラスの定義

Ruby

```
class クラス名 [<スーパークラス]
  文
end
```

Java

```
[修飾子]class クラス名
    [extends スーパークラス]{
  文
}
```

## ◆メソッドの定義

Ruby

```
def メソッド名
  文
end
```

Java

```
[修飾子] 戻り値の型 メソッド名{
  文
}
```

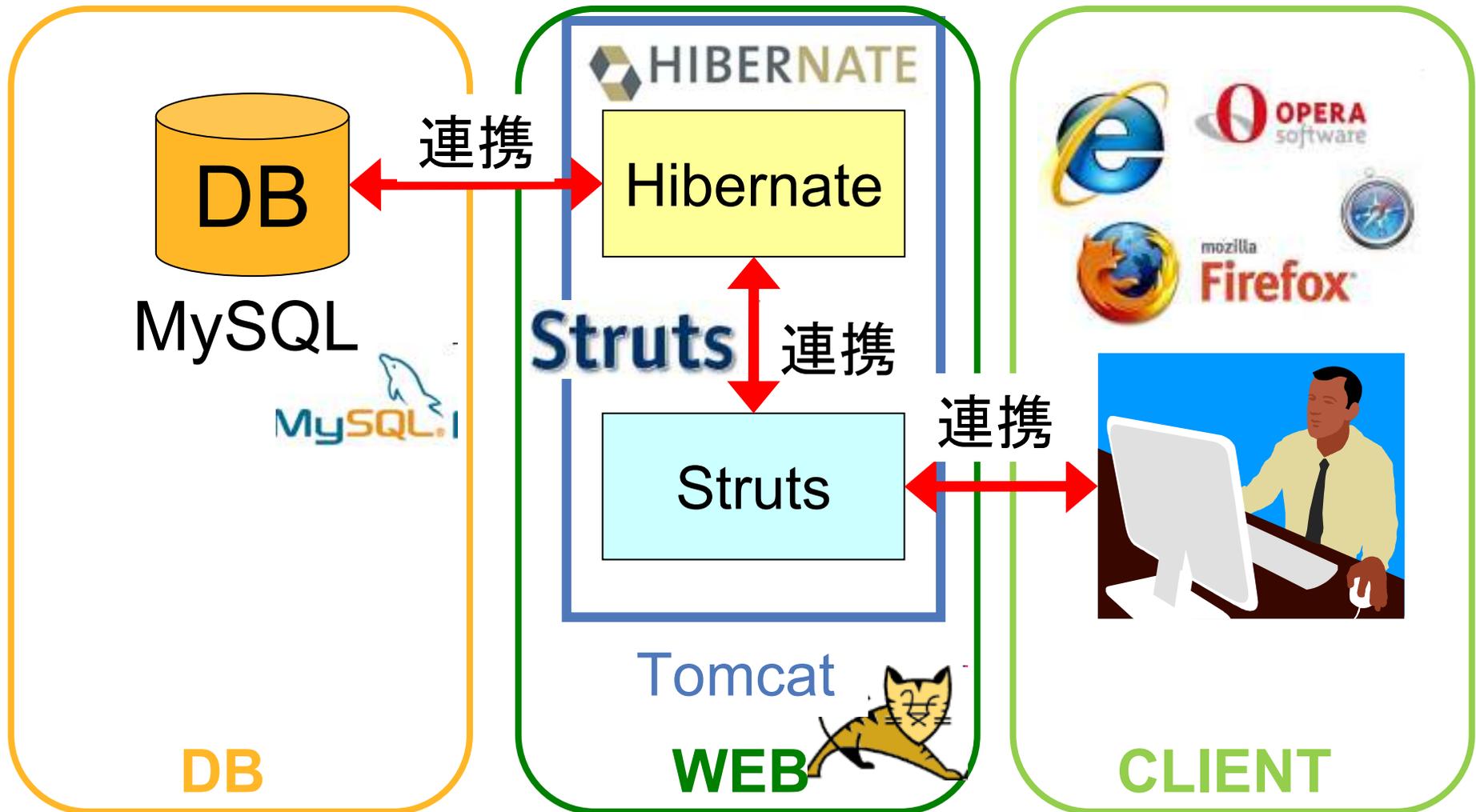
# Rails (Ruby On Rails) とは？

- Rubyを用いたWebアプリケーションフレームワーク
- オープンソースで開発（現在の最新versionは2.2）
- 高い開発効率が注目を集めている！
  - Railsなら、ブログが10分で作れる！
  - Railsなら、Javaより10倍早く開発できる！

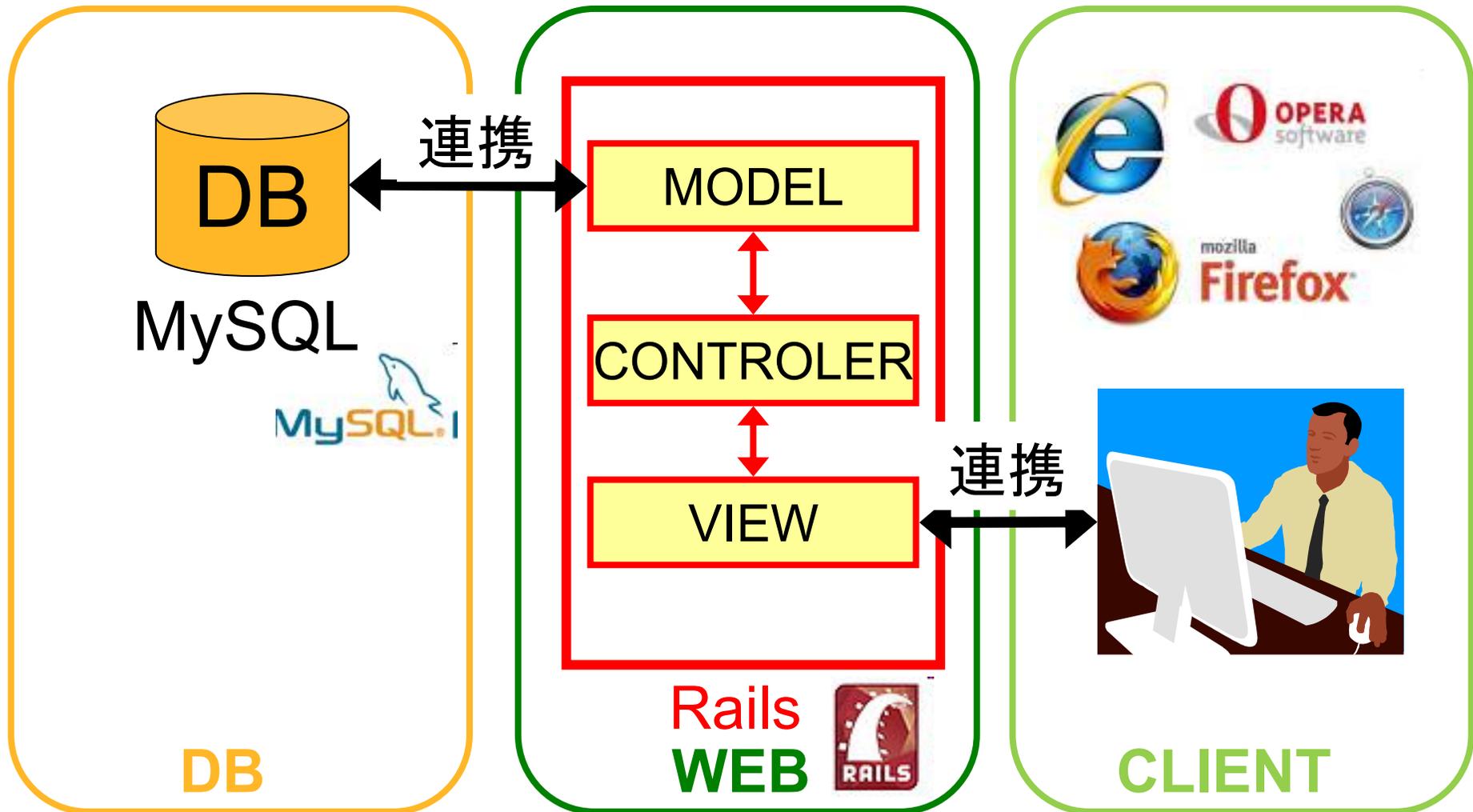
## 【疑問】: Railsの開発効率はなぜ高い？

1. 環境設定がラク
2. コード量が少なくてラク
3. 実行がしやすくてラク

# 環境設定【Javaの場合】



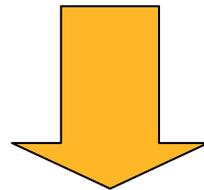
# 環境設定【Railsの場合】



# 環境設定がラク！

## 1. 環境設定がラク

- 後付でフレームワークを導入する必要なし！
- 多様なフレームワークを扱う必要なし！
- Railsのみに注力することができる！



**開発効率が上がる！！**  
**(ラクに開発できる！！)**

## コード量比較【Ruby vs. Java】

- ユーザに表示する部分（HTML）の記述は、RubyとJavaで大きな差はない。

```
<html>
....
<body>
....
<%= "Hello,World" %>
....
```

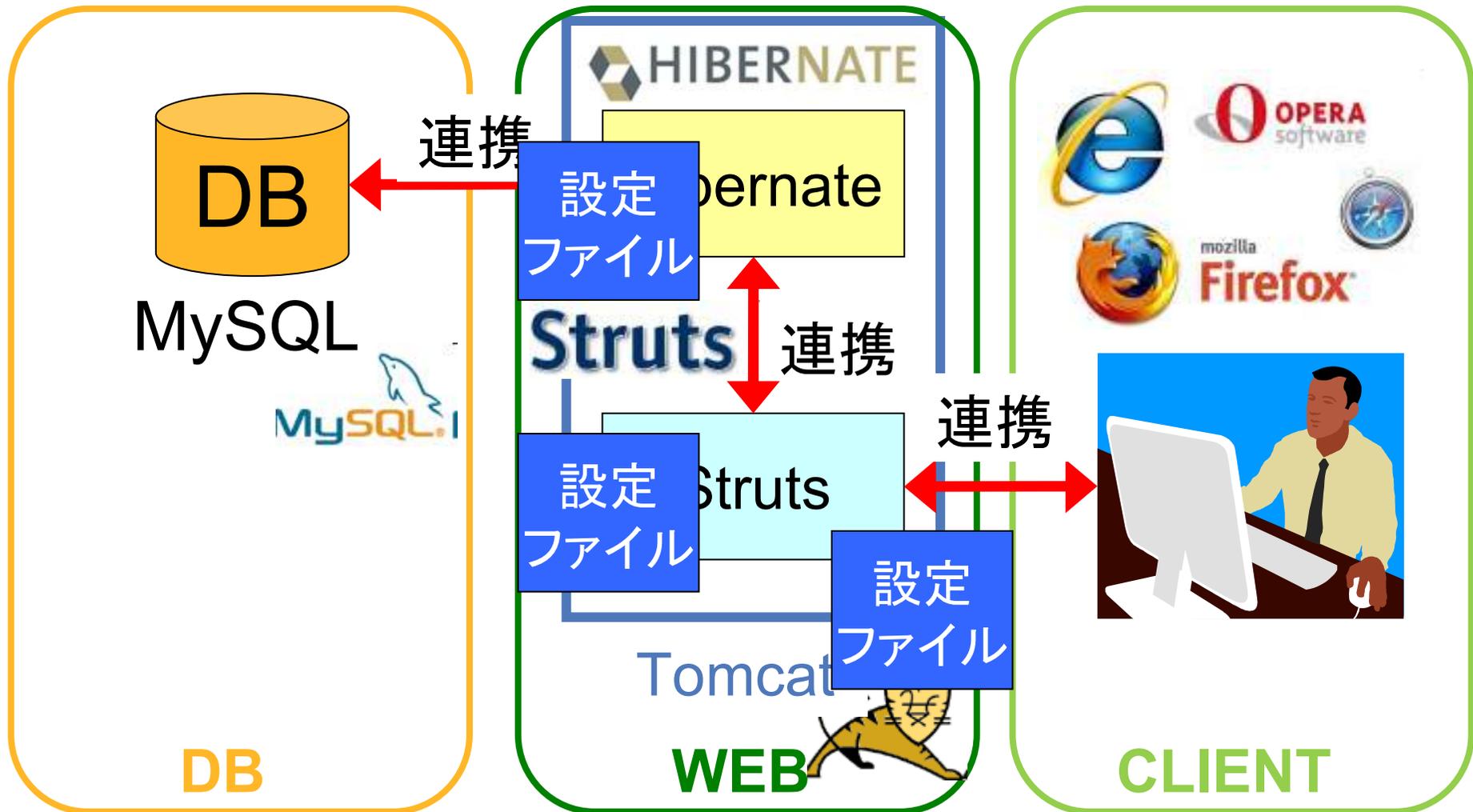
RHTMLファイル

```
<html>
....
<body>
....
<%out.println("Hello,World") %>
....
```

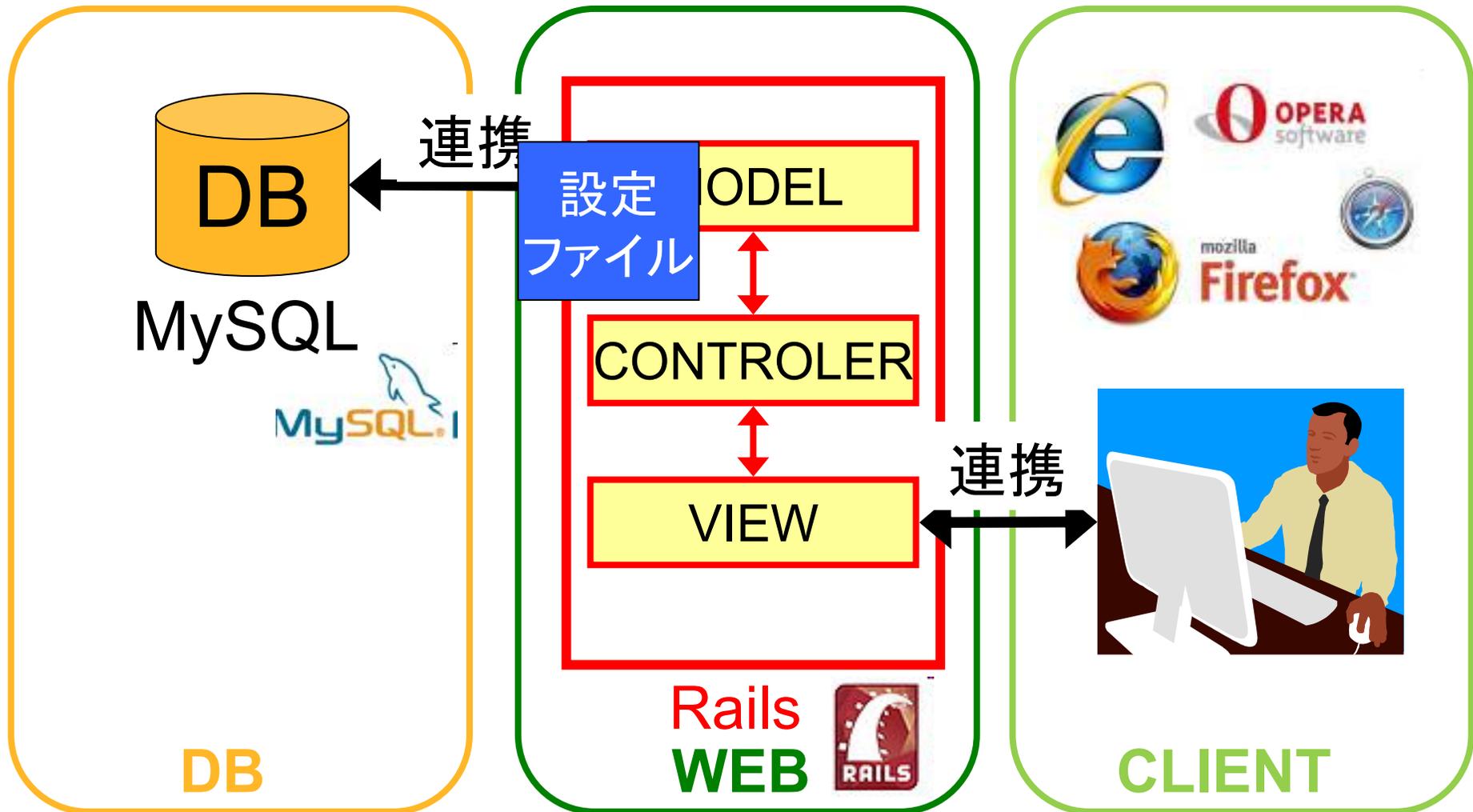
JSPファイル

- 設定ファイルの記述量に大きな差ができる。

# コード量の比較【Javaの場合】



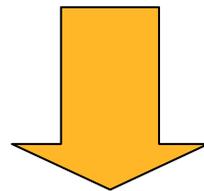
# コード量の比較【Railsの場合】



コード量が少なくてラク！

## 2. コード量が少なくてラク

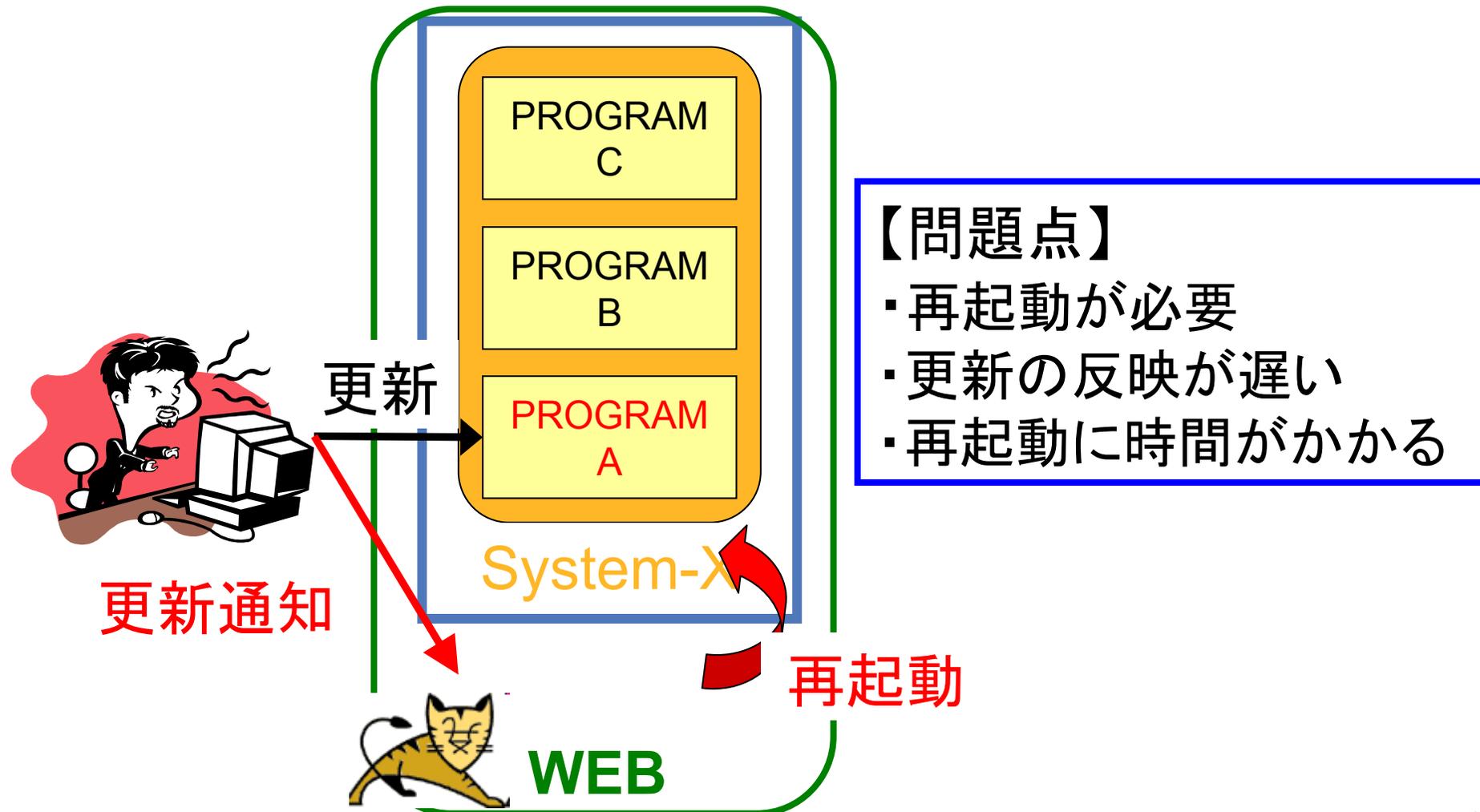
- 複数の設定ファイルを作成する必要なし！
- 設定ファイルの記述もとても簡単！
- 余計な設定を行う必要なし！



開発効率が上がる！！  
(ラクに開発できる！！)

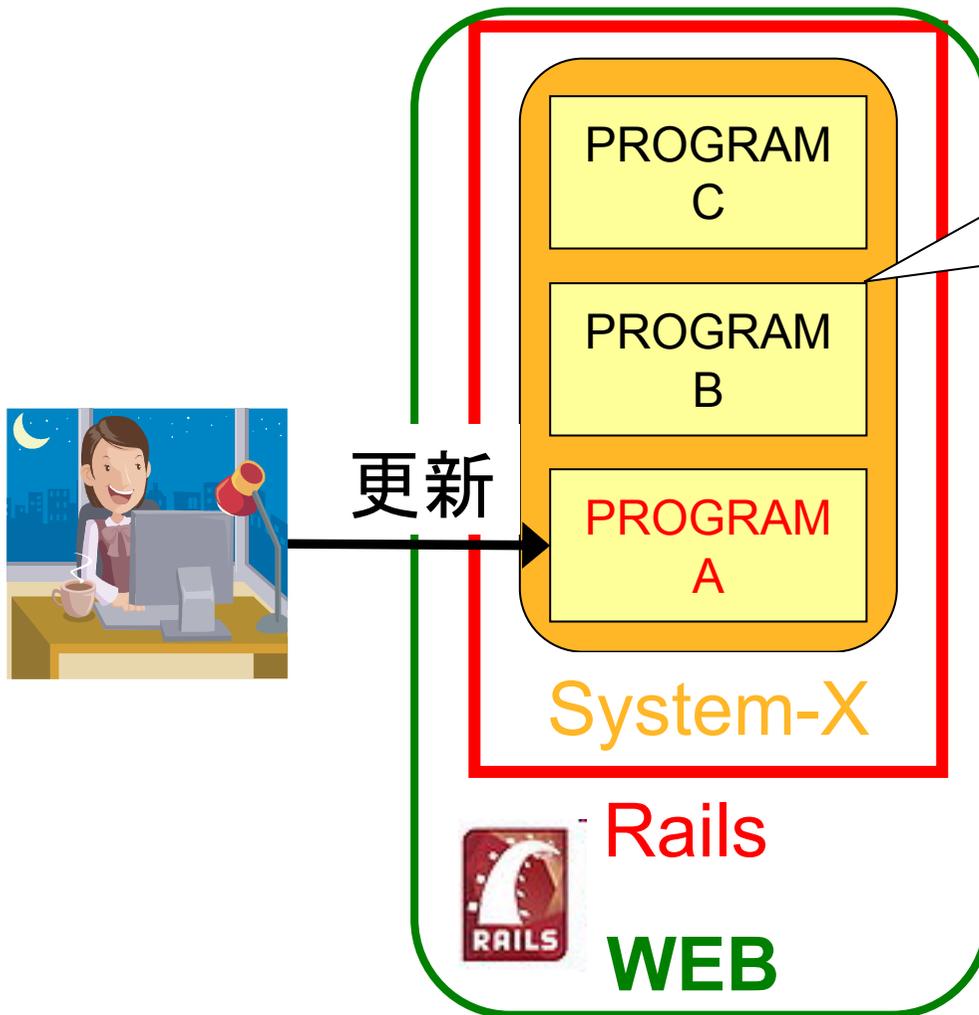
## 開発のやり方【Javaの場合】

WEBアプリケーションの作り方:



# 開発のやり方【Railsの場合】

WEBアプリケーションの作り方:



実行時に評価  
最新状態に更新

PROGRAM  
A

## 【利点】

- ・再起動は必要ない
- ・更新の反映が早い

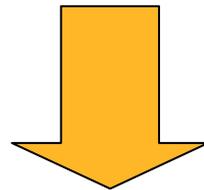
## 【問題点】

- ・実行時のオーバヘッド

実行がしやすくてラク！

### 3. 実行がしやすくてラク

- システムを再起動させる必要なし！
- 更新がすぐに反映される！
- 実行⇒デバッグのサイクルが早くなる！



開発効率が上がる！！  
(ラクに開発できる！！)

【疑問】: Railsは使われているのか？

ビジネスでは…

活用事例が増えてきた！

# 活用事例紹介

## ①: 国内企業編

- ニフティ株式会社の事例
- 株式会社リコーの事例

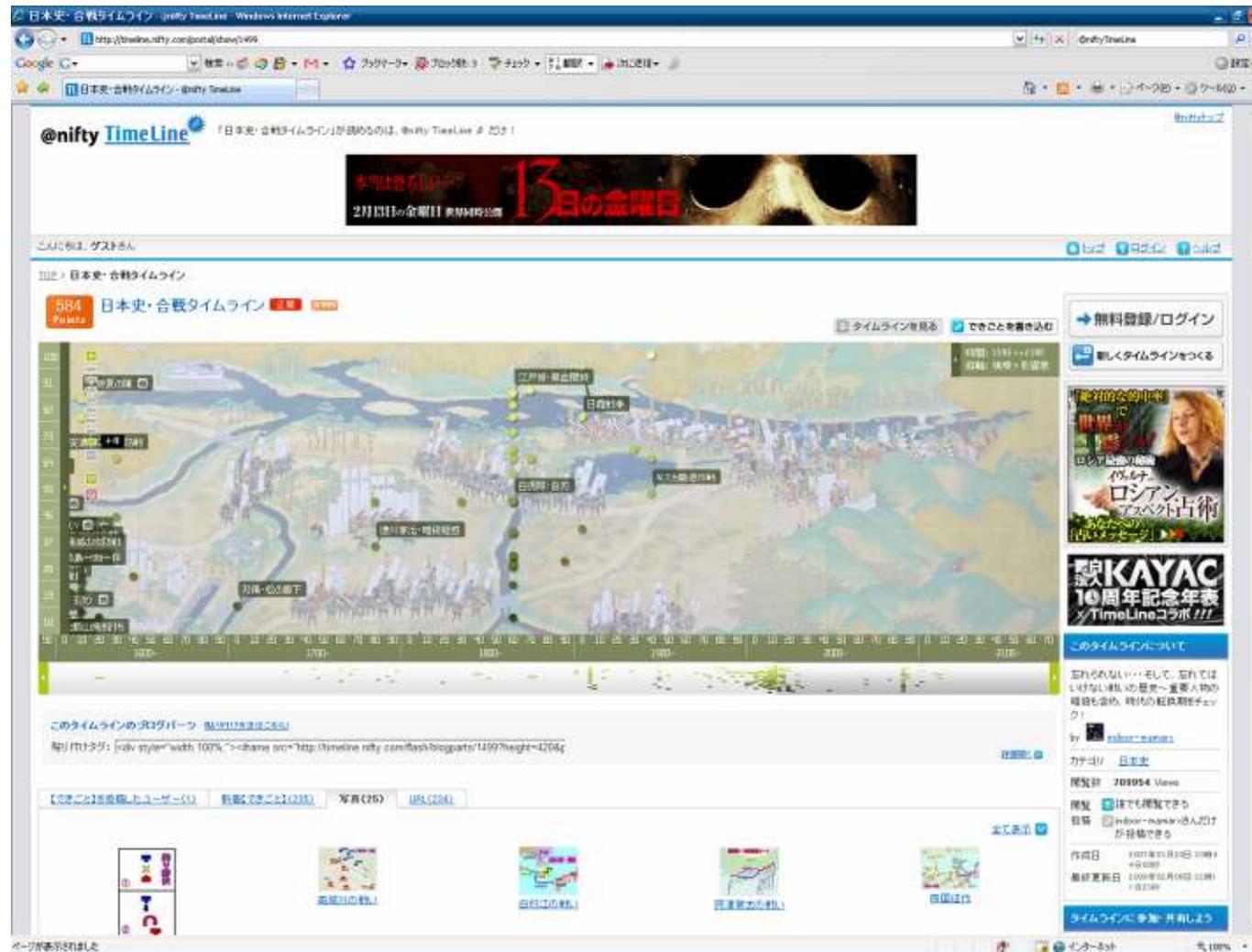
## ②: 海外企業編

- Twitter社の事例

## 国内企業編：ニフティ株式会社

- WebアプリケーションをRailsで作成・公開
  - @nifty TimeLine  $\beta$  (誰でも年表が作れるソフト)
  - Web開発プラットフォームの選定のために開発
  - 4~5人月くらいで作成
  - @nifty開発後も積極的Railsを利用している。

# @niftyTimeLine



## 国内企業編：株式会社リコー

- WebアプリケーションをRailsで作成・公開
  - quanp (ファイル共有サービス)
  - 有料サービス
  - 現在も、Ruby/Railsを使用して開発を続けている。

quanp



## 国内企業とRails

- まだまだ、導入実績は少ない。
- 着実に増加している。有料サービスでの導入実績も見られるようになった。
- 現在は、啓蒙活動の段階ともいえる。  
⇒ Ruby Business Commons

# 米Twitter社

- WebアプリケーションにRailsを利用
  - Twitterは、海外で人気を誇るコミュニケーションサービス
  - 2006年7月にサービス開始、2008年4月より日本語版も登場している。
  - Railsベースのキラーアプリケーション

## Twitter

Twitter: What are you doing? - Windows Internet Explorer

http://twitter.com/

Google 検索

Twitter: What are you doing?

Select Language ...

## Twitterとは?

なに?    なぜ?    どうやって?

Twitterは「いまなにしてる?」という質問に短い文章で答えることによって、友だちや家族、職場の同僚とつながり合うサービスです。

[Twitterに登録する](#)

[ビデオを観る!](#)

[ここからログイン](#)

ユーザー名かメールアドレス

パスワード

次回から入力を省略する

[ログイン >>](#)

[パスワードを忘れた? ここをクリック](#)

「ツイッターの魅力はそのつながりのゆるさにある」

「フグの雑炊食べたいな」『うるる』

「ミニブログと呼ばれるサービスが、日本でも流行の兆し」本目社

インターネット 100%

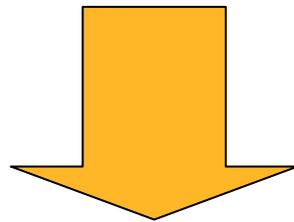
## 海外企業とRails

- ベンチャー企業でよく利用される。
- 日本よりも人気度・注目度は高い
- Twitterのようなキラークラウドアプリケーションも登場している

### 3. 検証・評価

【疑問】ラクに開発できるのか？

■ **ラク**に開発できるのか？

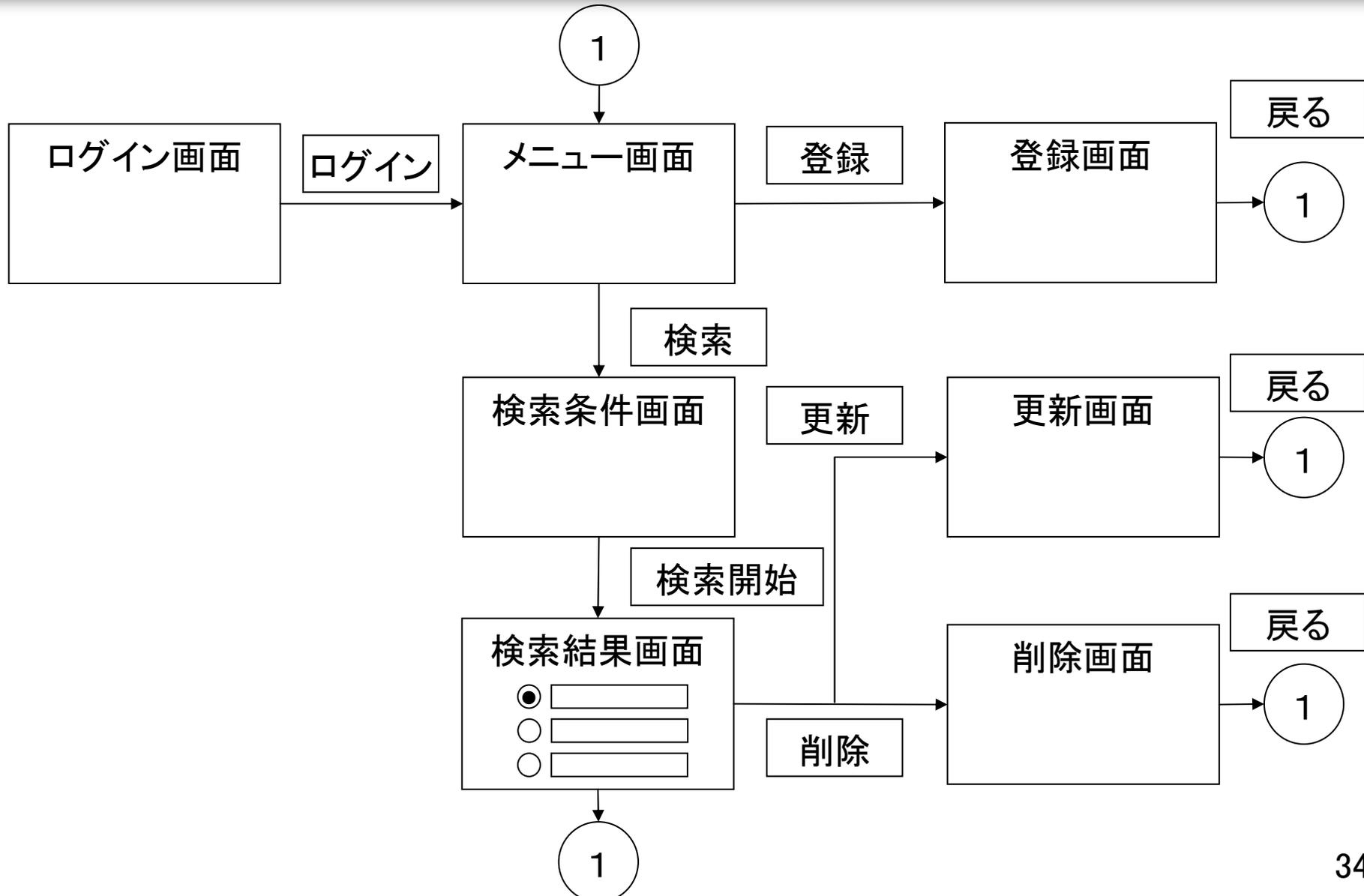


自ら試して評価する！

## 評価方法

- 評価方法：
  - 2つのフレームワークで同じアプリケーションを作る。
  - 比較するフレームワークは、Javaを採用。
  - 二つのアプリケーションの開発で得られた知見から評価する。
  
- 作成するアプリケーション：
  - ログイン管理システム

# ログイン管理システム：画面遷移図



# 使用フレームワーク(1): Rails

## 利用フレームワーク一覧:

- Ruby version 1.8.6
- Ruby On Rails version 1.2.6

## 利用するDB:

- My SQL version 5.0.6.7

## 使用フレームワーク(2): Java

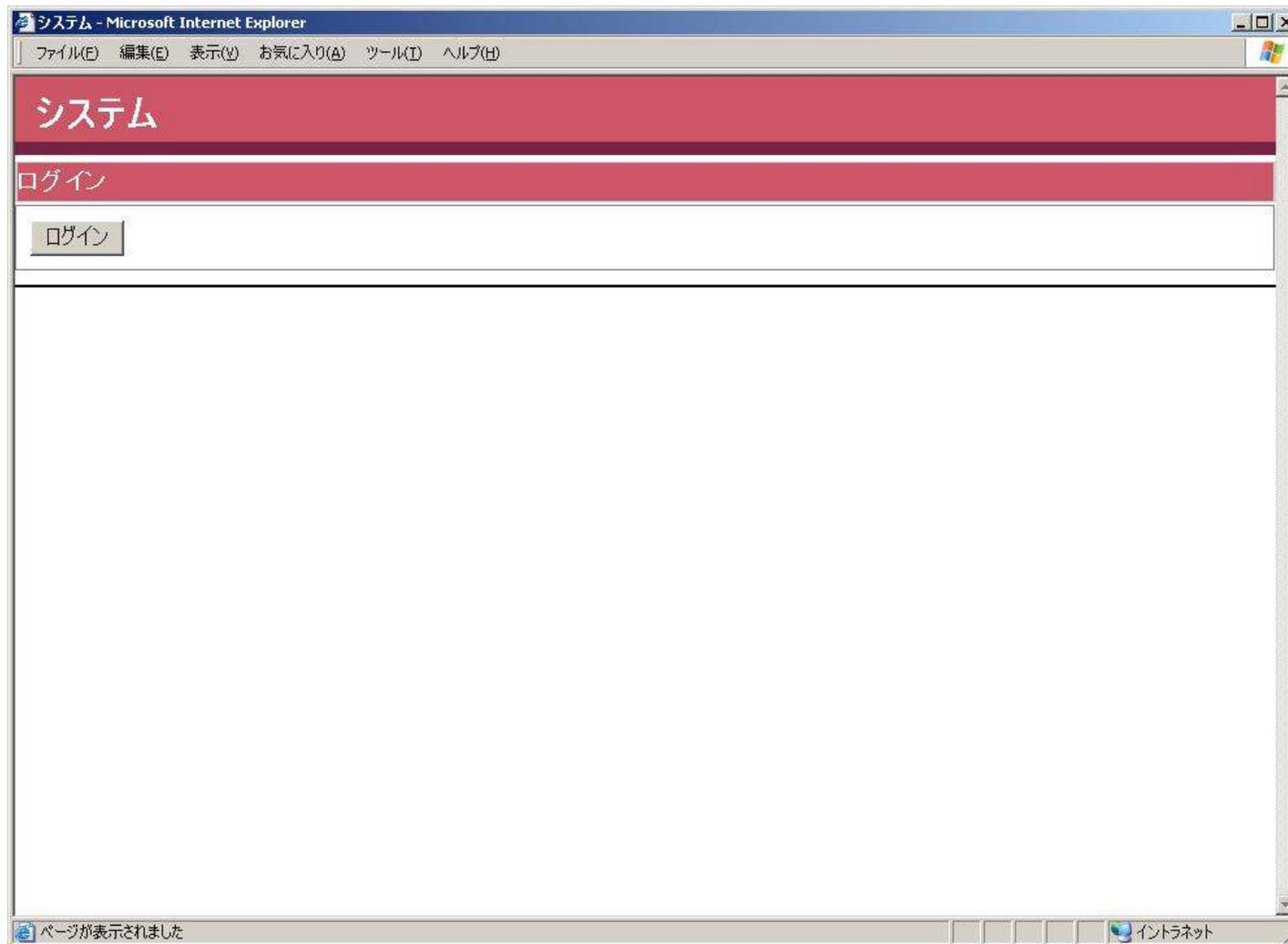
### 利用フレームワーク一覧:

- Java version 1.4.2
- Tomcat version 5.5
- Hibernate version 3.2
- Struts version 1.3.8

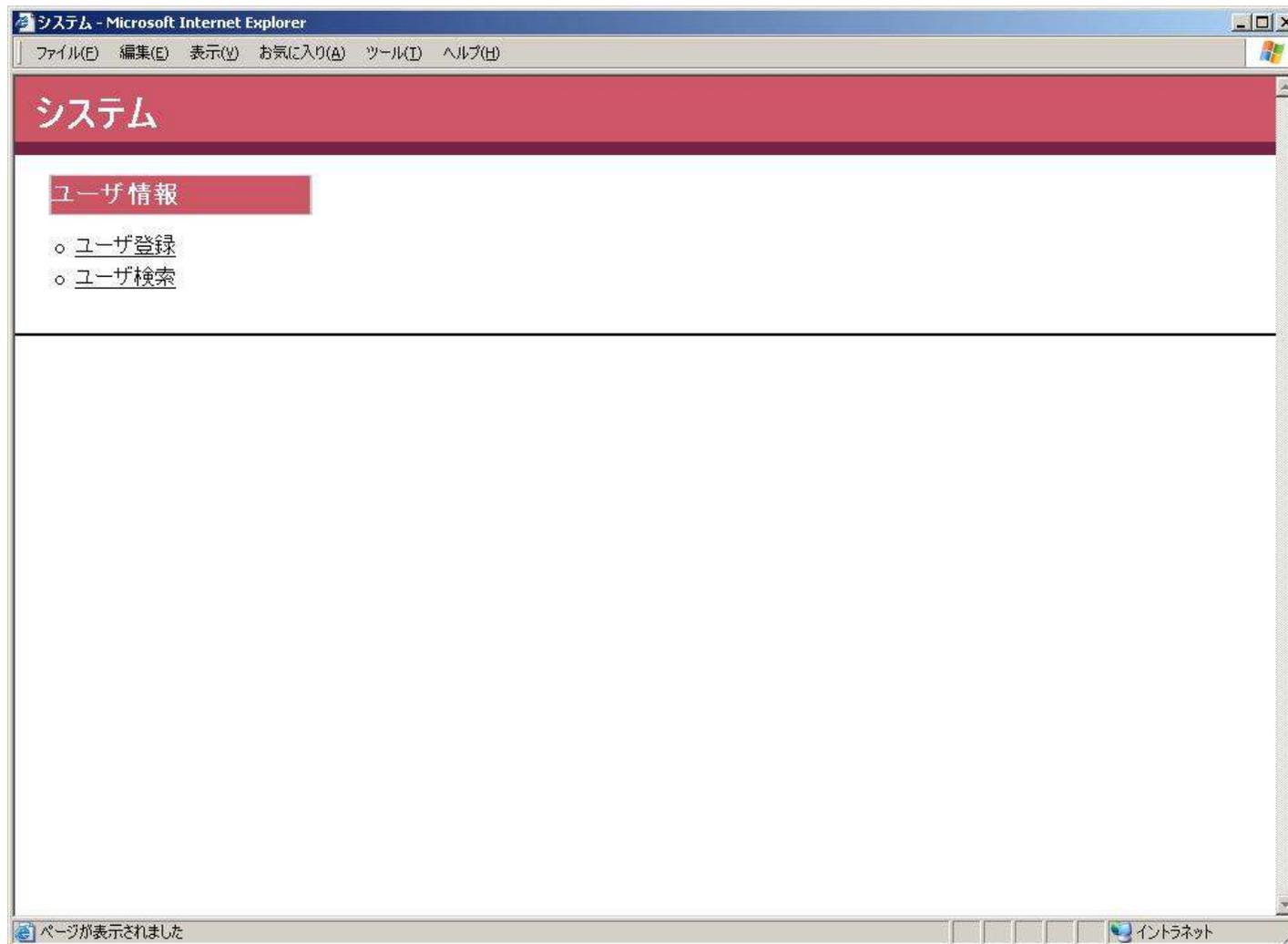
### 利用するDB:

- My SQL version 5.0.6.7

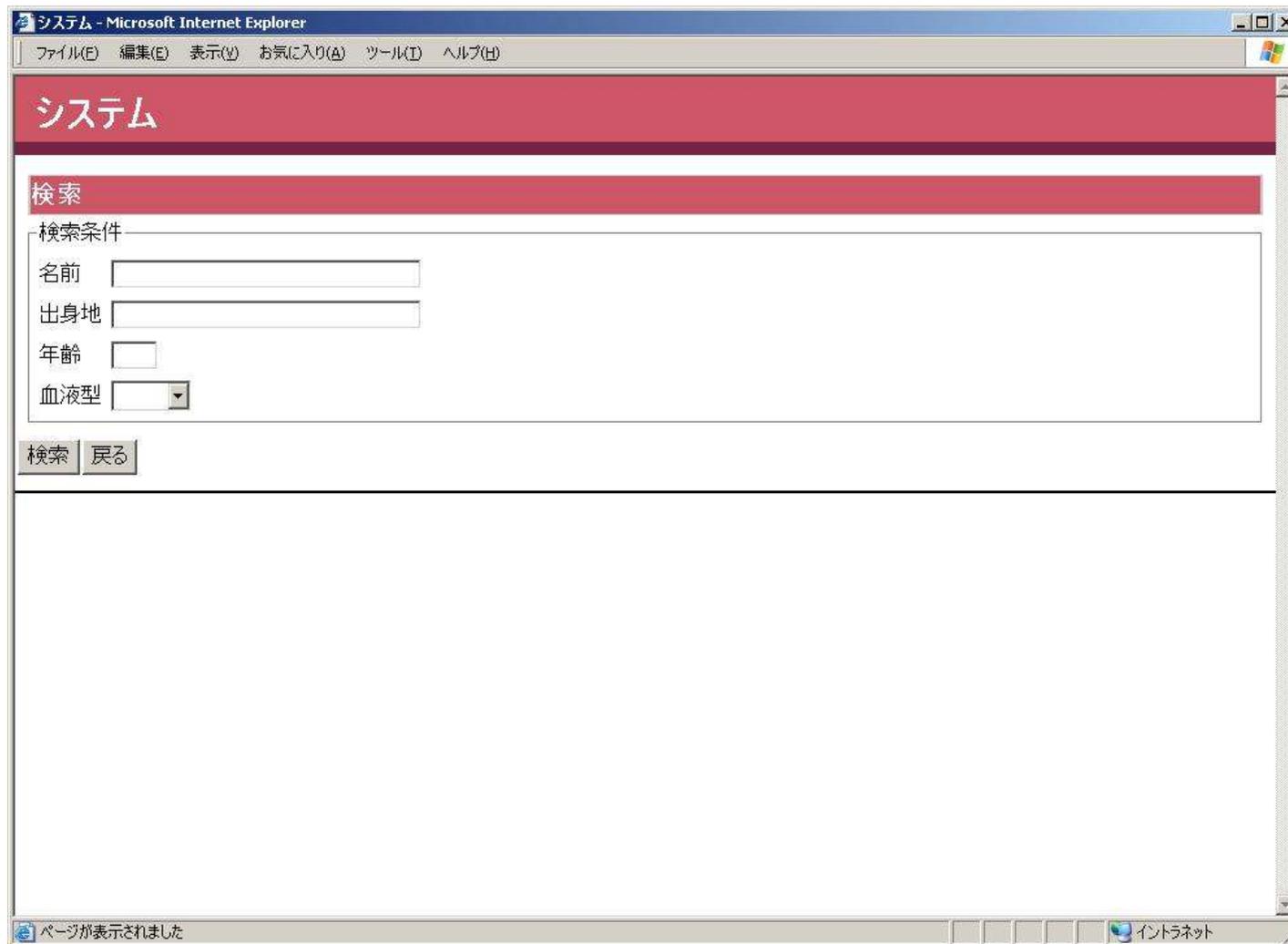
# 作成したシステム: ログイン画面



# 作成したシステム:メニュー画面



# 作成したシステム: ユーザ検索画面



## 作成したシステム: 検索結果

システム

結果一覧

No	選択	ID	名前	出身地	年齢	血液型
1	<input checked="" type="radio"/>	21	大分一郎	大分県大分市	20	A
2	<input type="radio"/>	23	別府次郎	大分県別府市	21	B
3	<input type="radio"/>	24	由布三郎	大分県由布市	22	O
4	<input type="radio"/>	25	中津四郎	大分県中津市	23	AB

戻る 削除 更新

ページが表示されました

イントラネット

## 評価: Railsを使ってみて良かった点

- 1. 準備(環境設定)がラク
- 2. 設定ファイルがラク
- 3. 実行しやすくてラク

## 利点1. 環境設定がラク 【Java編】

- Javaでのインストールから開発までのフロー
    - Javaをインストール
    - Tomcatをインストール & 設定
    - MySQLをインストール
    - Strutsをインストール & 設定
    - Hibernateをインストール & 設定
    - **開発スタート！**
- 

開発までの、環境設定が煩雑！

## 利点1. 環境設定がラク 【Rails編】

- Railsでのインストールから開発までのフロー
    - Rubyをインストール
    - Railsをインストール
    - MySQLをインストール
    - **開発スタート！**
- 

開発までの、環境設定が**ラク！**

## 利点2. 設定ファイルがラク

- Railsでの設定ファイルは記述が少ない！

### 【Java版システム：設定ファイル】

- hibernate.cfg.xml , USERS.hbm.xml
- struts-config.xml
- server.xml , web.xml, etc....

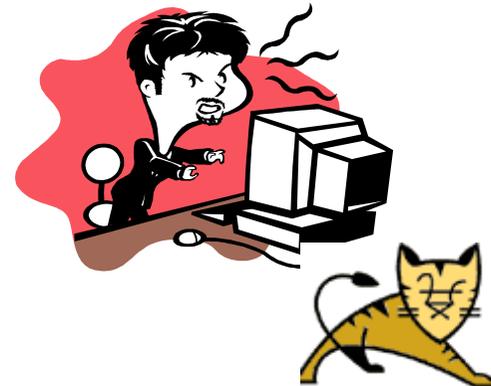
} XML設定ファイル

### 【Rails 版システム：設定ファイル】

- database.yml

## 利点2. デバッグしやすくてラク

- Railsでは、更新がすぐに反映される。
- 再起動待ちがないため、ストレスを感じない。



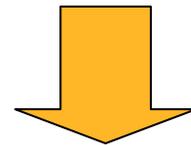
サクサク開発できている気がする！

## 評価: Railsを使ってみて困った点

- 1. 簡潔すぎるコード
- 2. 実行効率

# 簡潔すぎるコード

- Railsは、“設定より規約” (CoC:Convention over Configuration)という哲学によりコードを書く量が減る。

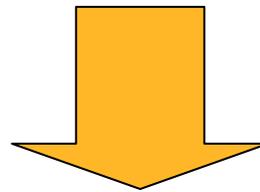


言い換えると

- 前提となる概念や知識を理解していなければ、デバッグすることが難しい。
  - 命名規則
  - MVCモデル
  - 暗黙変数, etc...

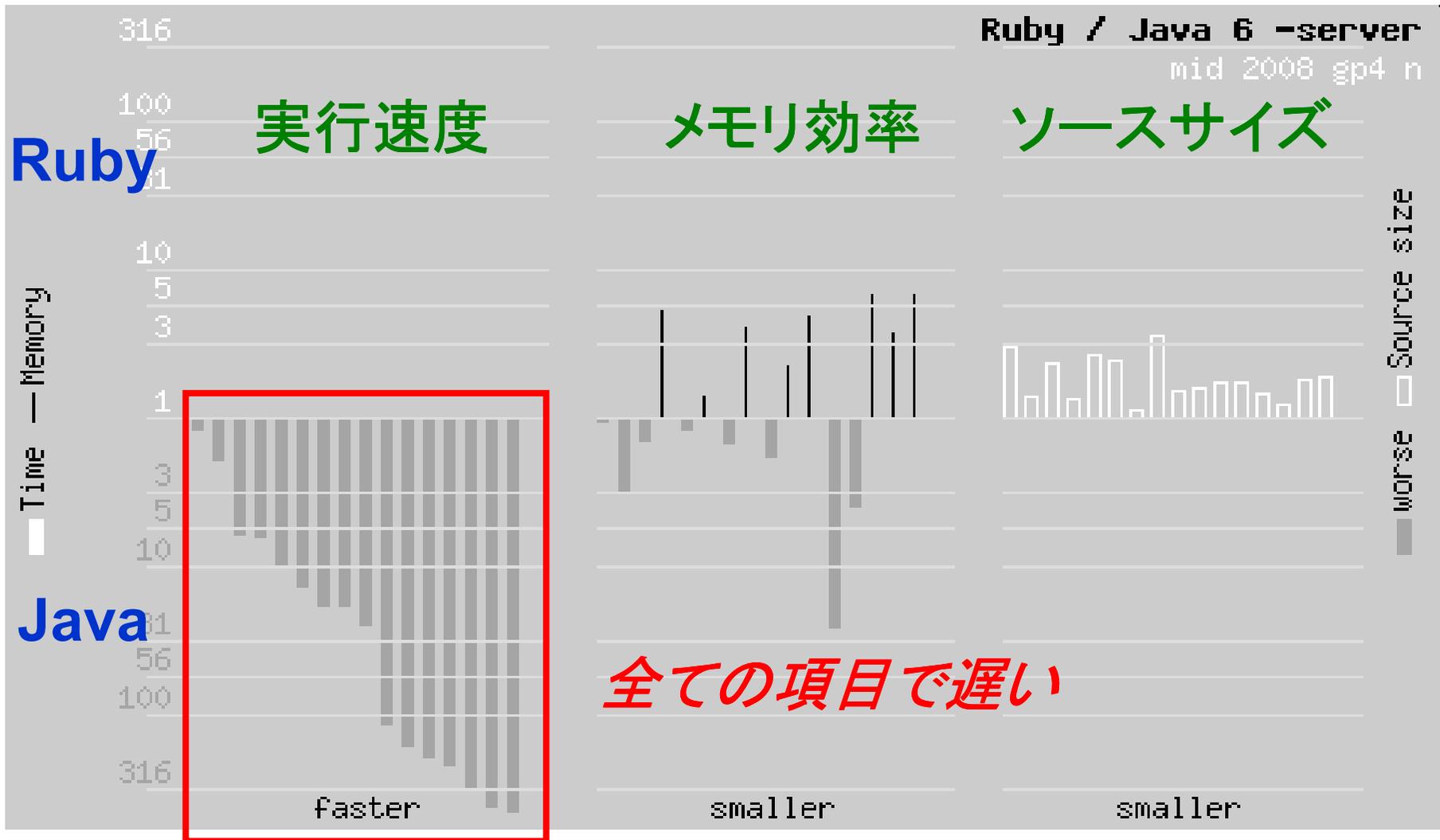
# 実行効率

- **実行速度の遅さが、各方面から指摘されている。**
  - 高負荷に対するスケーラビリティの低さ
  - Rubyの実行速度の遅さ



- **今後のバージョンアップを注視する必要アリ！**
  - 現行Rubyの5倍速い、Ruby version 1.9
  - Ruby on Rails 2.x or 3.x

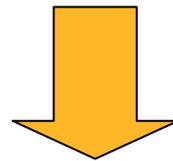
# Ruby vs. Java



## 5. おわりに

結論

本当にラクに開発できるの？



確かにラクして開発ができた！！

## まとめ

- Ruby On Railsを使った開発の利点・欠点をまとめる。

【利点】	【欠点】
<ul style="list-style-type: none"><li>- 環境設定がラク</li><li>- 設定ファイルがラク</li><li>- 実行がしやすくてラク</li></ul>	<ul style="list-style-type: none"><li>- 簡潔すぎるコード</li><li>- 実行効率</li></ul>

- Railsは開発効率を高める！
- Railsを効果的に使うためには、知識・経験が必要！！

## おわりに

- Railsの登場以降、様々なWebアプリケーションフレームワークが登場している。

Ruby On Rails

Grails

Seasar2

JRuby on Rails

Django

etc...

- 技術動向を注視する必要がある！！