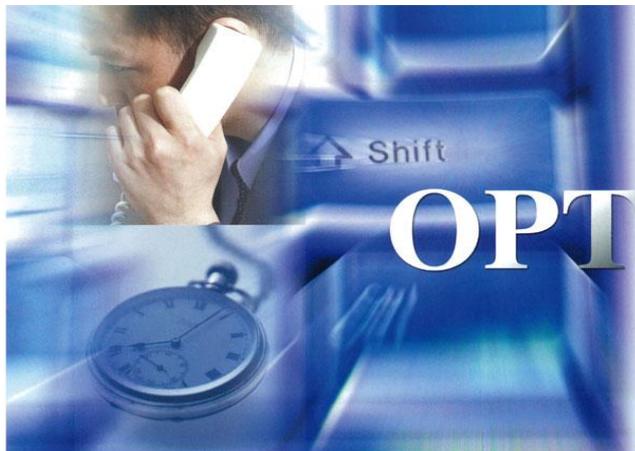


# WhyRuby? GoRuby ～Rubyへ取り組んで～



**OPTIMA**



**株式会社オプティマ**

2014/10/23

梶谷伸二

# 目次

## 会社紹介、自己紹介

### I Rubyって？

- ・Rubyとは
- ・Ruby紹介(環境、事例、特徴)

### II 他の言語と比較して

- ・Rubyの学習
- ・生産性、保守性
- ・Ruby On Railsの取捨選択
- ・環境面

### III 事例紹介

- ・事例その1
- ・事例その2

### IV 会社としての取組み

- ・コミュニティーへの参加
- ・技術者育成

# 会社紹介

- 株式会社オプティマ
- 1972年8月創業 43期
- 資本金 1億円
- 社員数 317名
- 大崎本社、千葉支店、大阪支店、九州支店、島根開発センター(予定)
- グループ会社
  - HCS(物流PKG)、ライズシステム(波情報配信)
- JISA、JIPDEC、LASDEC、FSA

# 会社紹介

- 元々公共、金融系の受託開発とパッケージ開発・販売を中心
- 二代目社長を軸に更なる発展を計画、2014年度より新たな中期計画に基づき事業展開中
- 第三統括部ではインフラ構築含めたトータルソリューションとして民需向けWEB系やSNS系の受託案件を推進
- Java、C#、VB、Ruby、PHP、Android、iOSなど
- 更に新規ソリューションの構築・展開を企画中

# 自己紹介

梶谷伸二(かじたにしんじ) 1965年12月生まれ

- 1988年4月 オプティマムシステムズ(現オプティマ)入社
- 公共配属 自治体システム開発従事
- 組込制御部門 立上げ
- NEWビジネス推進室 新規事業推進
- 新規社内パッケージの開発販売、波情報サービスの構築、中国向け食品トレサビリティシステムの提案など
- Web系主体部門の推進
- Rubyを軸に、島根開発センター構築の準備に従事

【キャリア】 PG → SE → PLPM → 営業企画 → 現在

最近好きになったこと 魚、百人一首

---

# I Rubyって？

# Rubyとは

---

- オブジェクト指向スクリプト言語
- まつもとゆきひろ氏
- フリーソフトウェア / OSS

# 歷史

---

- 1993年開發開始
- 1995年公開
- 2004年Ruby On Rails登場
- 2013年2月 Ruby2.0.0
- 2013年6月 Rails4.0
- 2014年4月 Rails4.1
- 2014年9月 Ruby2.1.3

# 開発方法

---

## ■ Rubyの開発環境 (IDE)

- テキストエディタ (Vim、Emacsなど)
- Eclipse (プラグイン追加)
- NetBeans
- 3rdRail
- RubyMine

# PaaSや開発事例

---

- Heroku
- Engine Yard
- MOGOK
- Sqale
- COOKPAD
- 東京ガス
- SmartStage
- DTSクラウド連携

# Rubyの特徴

## ■ Rubyの特長

- シンプルな文法
- 普通のオブジェクト指向機能(クラス、メソッドコールなど)
- 特殊なオブジェクト指向機能(Mixin、特異メソッドなど)
- 演算子オーバーロード
- 例外処理機能
- イテレータとクロージャ
- ガーベージコレクタ
- ダイナミックローディング(アーキテクチャによる)
- 移植性が高い。多くのUNIX上で動くだけでなく、DOSやWindows、Mac、BeOSなどの上でも動く

# Rubyの特徴

- 楽しくプログラミングできること
- インスタンス変数の操作を行うために、getterとsetterを変数の数だけ定義……  
しません
- 純粋なオブジェクト指向  
「オブジェクト指向の学習に適した言語」  
Martin Fowler氏（リファクタリング著）

# Rubyの特徴

- 派閥

map派？ collect派？

単純に名前が違うだけのメソッドですが、  
Rubistが集まると派閥ができる・・・

とはいえどちらかに統一して欲しいというエンジニアもいます

# まつもとゆきひろ氏

- Ruby開発は趣味、気分良く開発できる言語にしたい
- Rubyの「これまで」と「これから」  
Web、アプリケーション開発言語としての地位を確立しつつあるが、今後は科学技術計算、組込ソフトなど多くの領域に適用可能、そのあたりを支援するライブラリが増えていくのでは

# まつもとゆきひろ氏

- 内部実装

遅いと言われ続けてきたが、ようやく誤解を  
払しょくできそう

Cruby、Jruby、Rubinius 今後も高速化が継続  
されることを期待します

---

## II 他の言語と比較して

# Rubyの学習

## ■ 言語初心者

覚えやすい

自由に書ける

## ・ちなみに

Java8をターゲットに考えると、RubyからスタートすればJava8に入りやすい  
(ラムダ式など)

# Rubyの学習

## ■言語経験者

- ・オブジェクト指向がわかっているならば簡単  
Javaがしっかり書ければ簡単にできる
- ・ルールが緩やか  
なぜコンパイルエラー？という悩みがない  
JavaやCの書き方でもOK  
いろいろ省略が可能  
→ 馴染みやすい

# 生産性

## ■高い

- ・コーディング量は少ない
- ・知っている人が書けば生産性は高い  
不十分な知識で書くと、さほどでもない
- ・IDEも有効

# 保守性

---

## ■高い

- ・当社事例ではまだ十分ではない
- ・但し、バグ修正対応は短時間でできている

# 自由度

- 型宣言がわずらわしくない  
自由を感じる
  - 全てがオブジェクト int、+・・・  
何でも上書きできる
  - メタプログラミング オープンクラス  
自由に書き替え(破壊?)可能  
Javaではやらないメソッド自体の書換が可能
- ※ 自由度が高いのは諸刃の剣

# Mix-in

---

モジュール機能をクラスに取り込める

include モジュール名

これだけでモジュールのメソッドが使えます

Java出身者絶賛！（メンバーにより賛否あり）

# Ruby On Railsの取捨選択

■とても便利なので是非活用を

Active Record

セキュリティケア

ログ出力 ほか

■SQLほか自力でやりたい方

独自フレームワークを作っても良い

# 環境面

- 欲しい部品がまずある  
困ってる人は既にいて、しかも解決済
- 上記GemやRailsなど更新頻度が高い

※進化をキャッチアップする大変さはある

※お堅いお客様では保障が微妙な部品で  
構築できない

# 環境面

- フレームワークの取捨で悩まない  
sinatra (軽量) などあるが、まず Ruby On Rails  
Java の場合は多すぎて悩ましい
- Java の Pray ほか Rails の模倣？  
が広がっている  
じゃあ大元使えばいいのでは？

# とりまく環境

- Ironruby  
    コマンドでお試しに便利
- RubyMotion  
    ネイティブiOSアプリ／Android対応も発表
- Jruby  
    RubyクラスのようにJava部品を活用  
    JNIは大変
- mruby  
    組込み向け軽量ruby

# Ruby増えてませんか？

---

- REDMINE: PRJ管理
- Capistrano: デプロイ
- Chef: インフラ構築

Rubyで自由に改修

# 参加しやすい

## ■開発へ参加できる

- ・コミッターへのハードルが高くない  
気軽に参加できる
- ・GitHubなどもあり、ソーシャルコーディング  
しやすい
- ・コミッターが協業しやすい
- ・コミッターへの負担を下げる仕組みに  
なっている(と思われる)

# 他の言語にないもの

- 日本発の言語 応援したい！
- 国を挙げて！？応援している  
島根、福岡・・・
- 根底に「楽しさ」がある
- RubyとRails覚えればOK
  - 全エンジニアの第二言語になれる  
もちろん第一でも。。

# 参考：島根県の取組み紹介

- 島根県ではRubyで町おこしを実現中
- まつもとゆきひろ氏在住、Nacl社フェロー
- RubyWorld Conference
- Ruby合宿
- 学校でRubyを学習
  - 松江市中学校Ruby教室
  - 2016年度から松江市・全市立中でRuby授業予定
- しまねOSS協議会、松江オープンソースラボ
- Ruby技術者認定資格取得助成金

---

# III 当社のRuby事例紹介



# 事例紹介 その1

## UM(うるとらミート)

食肉卸加工業向けの統合管理パッケージ

- ・トレーサビリティ確保
- ・計量器連携
- ・歩留、原価、粗利管理

入荷、出荷、売上、加工、在庫管理ほか



# 事例紹介 その1

## 【特徴】

- ・牛 個体識別番号
- ・豚 ロット管理

いずれも対応

各々機能レベルで3段階での提供



# 事例紹介 その1

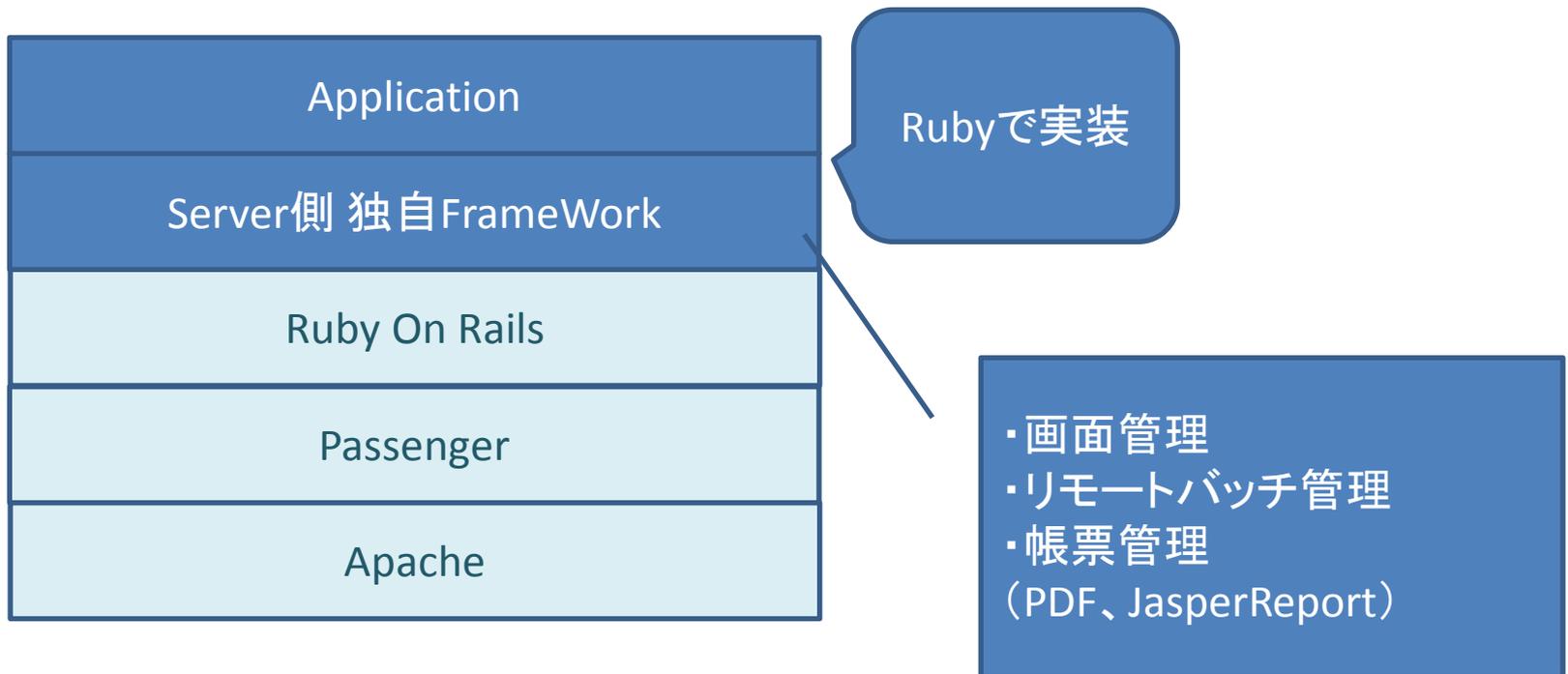
■開発にRubyを使用

独自フレームワーク、サーバ側

→ Rubyで実装

# 事例紹介 その1

構成 Ruby1.9.1 / Rails2.3.5 / MySQL /Linux





# 事例紹介 その1

## ■きっかけ

担当エンジニアの一言

「Rubyっておもしろそうなので、

やってみませんか？」



# 事例紹介 その1

## ■反省点

参画エンジニアRuby習得度のレベル差があり  
一部のエンジニアのコードを読めない、  
同じように書けない という状況に

→ 事前教育、準備不足・・・



# 事例紹介 その1

- Windowsサーバへの取組み(当時)  
お客様のニーズも多少あったので  
前向きに検討
  - passengerが動作しない
    - thinでチャレンジ
      - オンラインはOK  
帳票とバッチがNG
        - 断念

# 事例紹介 その2

## 物品検索サイト

- バッチやオンラインで登録したデータを検索
- 全文検索あり
- 約70画面(機能)
- アクティブユーザ1万~2万
- 1秒あたり15~80トランザクション

# 事例紹介 その2

---

## ■きっかけ

Q:「JavaかRubyかどちらにする？」

担当エンジニア:

「Rubyの方が楽しそうなのでRubyで」

# 事例紹介 その2

---

規模が大きめなのが気になったが、何とかなる  
でしょう……

ということでRuby/Railsを選択

# 事例紹介 その2

## ■できるだけRailsの良さを活用しよう

- ・コーディング規約、変数名、メソッド名の規約はRails遵守
- ・文字コード、改行コード、インデントなど最低限のこと以外はあえて定義しない

これで良かったのか？

# 事例紹介 その2

## ■開発環境

Ruby 2.1.2

Ruby On Rails 4.1.4

Linux

MySQL

Rubygems

passenger

RSPEC

Jenkins

Git

thinreports    etc

# 事例紹介 その2

## ■開発IDE

Ruby Mine

思い切って使ってみました  
今回はある程度の規模感があるので、  
費用対効果も出たのでは

※エンジニア数とコストが比例する

# 事例紹介 その2

---

## ■ここまでで苦労したこと

- ・決めることが多かった
- ・若手Javaエンジニア(1, 2年生)のRubyエンジニアとしての育成
- ・ソースコードの標準化

# 事例紹介 その2

## ■ここまでで良かったこと

- コーディング量が圧倒的に少ない
- 保守が効率的にできそう
- ソースの理解がしやすい  
(書き方、書き手による)

---

# IV 会社としての取組み

# 会社としての取組み

---

- 徐々にRuby実績が増えてきている
  - Ruby On Railsを使わない開発
  - Ruby On Railsを使った開発
  - Rubyにしては大規模な開発

# 会社としての取組み

■ 経験を生かし、より以上にRubyを打ち出して  
いきたい

- ・ 今後の市場性
- ・ 高い生産性
- ・ 言語と環境の進化が早い
  - お客様への付加価値になるのでは

ついていくのは大変ですが...

# 会社としての取組み

---

なんといっても

エンジニアが楽しく開発できそう

自由が悩ましいという意見もあるが、それをうまく使いこなすことができてこそ楽しいシステム開発になるのでは！？

# 会社としての取組み

---

## ■ 形式化

- ・ コミュニティへの参加  
Rubyアソシエーション
- ・ 島根開発センター開設(平成27年予定)

# 会社としての取組み

## ■ 技術者育成

- ・ 資格取得支援

社内制度の見直し中

RubySilver、Goldへの手当支給

(受験料、一時金など)

- ・ 社内独自ミニテーマ集

慣れるための課題をエンジニアが随時作成

# 人財募集

## ■人財募集中

- ・Ruby開発に興味がある方
- ・島根県松江市で働きたい方

随時採用募集しています。  
お気軽にお声掛け下さい。

Ruby以外でもWebやJavaやりたい方歓迎します

---

**ご清聴ありがとうございました。**

**参考:Ruby公式サイト** <https://www.ruby-lang.org/ja/>  
**参考文献:Ruby徹底攻略(技術評論社)**