

Railsでアジャイルに ビジネスSNS開発事例

川端光義
アジャイルウェア

堂端翔
若干ハタチの天才プログラマ

自己紹介

川端光義 (かわばた みつよし)

Ruby暦 1年、Java暦 6年、他忘れました。

アジャイル(特にeXtreme Programming)、オブジェクト指向、TDD(テスト駆動開発)が好き。

『バグがないプログラムの作り方 - JavaとEclipseで学ぶTDDテスト駆動開発』(翔泳社、2004年9月出版)

昨年まではXPJUG関西などで活動。XP祭り関西2006

お客様と開発者が満足する仕事がしたい。

松江でRuby認定資格GET!

楽しく仕事をするがモットー

好きな言葉:「人間は得ることで生計を立て、与えることで人生を築き上げる」



愛娘。春から小学一年生!

自己紹介

名前: 堂端 翔(どうはし しょう)

年齢: 20歳

プログラミング歴: 8年

Ruby歴: 1年

「iQube」ってなによ？

グループウェアとナレッジマネジメント・ツールが融合したEnterprise2.0 Suite。企業ポータル、各種コミュニティ機能は、Web2.0の発想がフルに活かされ、かつ優れた内部統制が施されています。「誰でも楽しく使いこなせる」をテーマに、集合知、暗黙知を様々なアルゴリズムにより表示する他、外部連携、携帯機能、各種インセンティブ機能などを装備しています。

良くわからないので・・・

DEMO デモ

「iQube」の
フリートライアル
ができる
ポータルサイト

17日(月) 公開リリース！

17日の日経産業新聞でプレスリリース！

私が担当しました。

With加藤照雄さんと
ペアプロで。はーと

www.iqube.net

~~へ今すぐアクセス!~~

ごめんなさい、今はBASIC認証かかっています。

どうぞ会社(組織)で
使ってみてください。

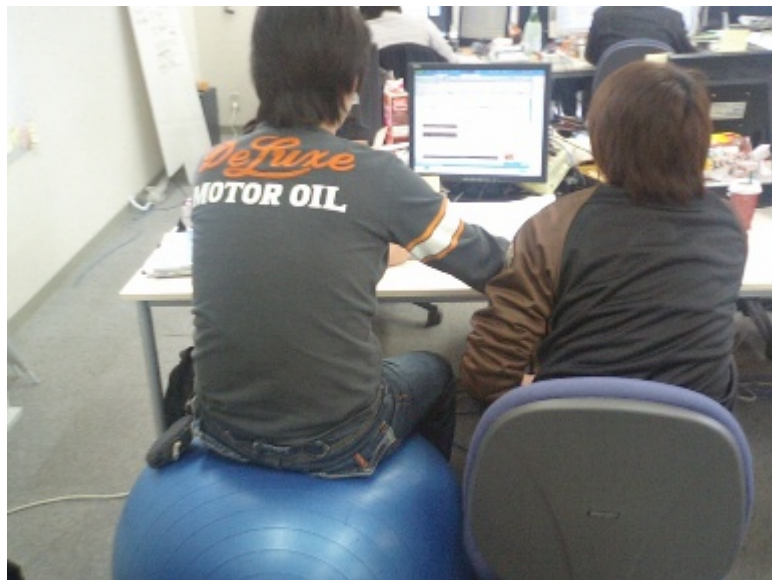
ハックしないでね。

フィードバック大歓迎。

初期不良はご了承くださいm(__)m

開発
現場

開発分室



バランスボール乗ってます

環境

サーバ
CentOS5
Apache
MySQL

Ruby 1.8.6

Rails 1.2.6

Myクライアント開発環境

Windows上のUbuntu7.10
MySQL
Mongrel
Aptana

Ubuntu Onlyの人もあるし

Macの人もあるし

いろんなOS混在だから

MS Office 使わない。
使えない。

設計書なんて誰も作れない作らない。

設計書なしで要件聞いたら、すぐ実装だから

とりあえず アジャイル

仕様書の前になるのは

パワポのみ

画面イメージなど

296ページ

チーム構成

- プログラマ10人
- 仕様担当3人
- テスター3人

表立ったマネージャ、リーダーはいない
自己組織化されたチーム

優秀なプログラマたち

Ruby暦

斉藤宏	…	8年
加藤龍一	…	2年半
稲光隆史	…	2年
森田真悟	…	2年
数又拓	…	2年
加藤照雄	…	7年
堂端翔	…	1年
山内竜太	…	1年
高瀬真宏	…	1年
川端光義	…	1年

チーム構成の  

テスターチームの充実

プログラマは開発とテスト
コードに集中できる

テストの自動化  

テストコードの指針

Unit

Functional

Integration

← 一番多く書いたところ

画面は手動テスト

グレーボックスの感覚でテスト

厳格なテストはテストチーム
が行う

バグ管理

Mantis



主にテストチームや仕様担
当が登録

プログラマもすぐに取り掛から
ない、担当不在のものを登録

バグ管理

Mantis



担当が割り振られてもスケ
ジュール次第で他の人にたらい
回しできる

担当不在のバグもチームで手
の空いた人が勝手に直していく

ポイント

誰でもいつでも
どのソースでも
修正できる

ソースの共同所有

アジャイルプラクティス

Subversion

Subversion

- ・それぞれ一日に平均5コミット以上
- ・コミットコメント

プロジェクトiQube

Subversionで管理されている
ソースの数

827

app、lib、testフォルダの総計

コミット総数(3月15日現在)

約5,000

昨年よりポータルサイト構築始まる

プロジェクトPortal

でも、プロジェクトiQubeのデータベースと共有。
iQubeのModel、lib、プラグインなどの資産を利用したい。

Subversionのexternal

プロジェクトPortal

静的HTMLファイルが多く存在する

デザインを外部委託

Railsプロジェクトに取り込むとログインや申込画面などViewに変換

プロジェクトPortal

デザイナーからのHTMLファイル郡
をSubversion管理

HTMLファイルが更新されたら
Subversionのヒストリーを比較し
て更新箇所を取り込む

プロジェクトPortal

ログインや申込画面では個人情報
を扱うため、https接続SSLで暗号化

ssl_requiredプラグイン

プロジェクトPortal

これだけ。。。

```
class AccountController < ApplicationController
  ssl_required :login
  .
  .
  .
```

プロジェクトiQube

論理削除を自動的にやってくれる

acts_as_paranoidプラグイン

これだけ。。。

```
class User < ActiveRecord::Base
  acts_as_paranoid
  .
  .
  .
```

プロジェクトPortal

メインは決済処理

外部の決済システムを利用

クレジット決済時、待ち時間が発生

プロジェクトPortal

決済確定ボタン押した時、
画面をモーダル制御してみる

プロジェクトPortal

JavaScriptライブラリ Control.Modal

プロジェクトPortal

公式サイトでのsampleでは
リンクでモーダル処理

決済ボタンはsubmitではまる

Control.Modal

```
<input class="btn" name="commit" type="submit" value="申し込みを確定する"  
"id="modal_link_one" />  
-----  
<style>  
modal_container {  
background-color:#fff;  
border:1px solid #666;  
overflow:hidden;  
font-size:12px;  
color:#333;  
text-align:center; }  
modal_overlay { background: #000; }  
</style>  
<script type="text/javascript">  
  new Control.Modal('modal_link_one',{  
    overlayCloseOnClick: false,  
    opacity: 0.5,  
    width: 250,  
    contents: function(){return '<p class="al_center"><table border="0"  
      cellpadding="0" cellspacing="0"><tr><td align="center">ただ今決済処理を  
      行っております。<br />そのままお待ちください。</td></tr></table></p>'},  
    afterOpen: function(){this.form.submit();}  
  });  
</script>
```

RESTful

iQubeは(半分)RESTful

routesのほとんどは map.resources

後々の拡張性が高い

いつかは
「日本のビッグAPI」

A j a x

iQubeはAjaxバリバリ

- ・Web2.0
- ・ユーザビリティを考慮した適材適所のAjax

使用javascriptライブラリ

- ・おなじみprototype.js & script.aculo.us
- ・Ext.js
- ・オリジナルjavascriptライブラリ

prototype.js

- ・基本的なjavascript構文を拡張
- ・Classの定義を簡略化して、javascriptをオブジェクト指向言語に近づけた
- ・ブラウザ間の差分吸収をしてくれる
- ・Rails標準搭載

script.aculo.us

- ・prototype.jsを用いて、D&Dなどいろいろな拡張機能を提供するライブラリ
- ・Rails標準搭載

Ext.js

- ・Yahoo UI(YUI)をより汎用的に改変したライブラリ
- ・prototype.jsなどをwrapして機能拡張できる
- ・リッチなUIを提供してくれる

Ajaxの難問

- ・デバッグが難しい
- ・ブラウザ間の差分の吸収
- ・使用ライブラリの選定

解決策

- ・デバッグが難しい
 - デバッグツールの導入
Debug bar、Firebugなど
- ・テストツールを導入
Selenium、JUnitなど

解決策

- ・ブラウザ間の差分の吸収
- ・なるべくprototype.jsなど、差分を吸収してくれるライブラリを利用する
(javascript標準の機能はなるべくそのまま使わない)

解決策

- ・使用ライブラリの選定
 - ・調査はしっかりする
 - ・同じような機能のライブラリがあれば、負荷のかからないもの(処理的にも実装的にも)
 - ・複数のライブラリを読み込む必要がある場合は、バージョンを考慮する

Ajaxバグワード No.1

「IEだと動かないんだけど・・・」

SaaS

iQubeはSaaSアプリ

広告配信・企業と企業のマッチング
などSaaSならではの展開を計画

SaaS

SaaSならではの問題も...

- ・純粋に負荷が高い
- ・各社間のデータをアプリ側で隔離
- ・汎用化

スケジューラ@機能

スケジュール管理機能を、個人、グループ、
設備の視点で参照、登録できる

スケジュール@表示

Ajaxバルーンでスケジュールを登録、
確認できる
(目指せGoogle Calendar)

スケジュール@繰り返し

繰り返し条件も柔軟に設定可能
毎日・毎週・隔週・毎月etc...

スケジュール@便利機能

- ・予定を登録するときに、参加者に選んだユーザや指定した設備が他の予定と重複してるかチェックできる

Wiki@概要

RubyistにはおなじみのHikiDocを採用
Wikiの基本機能はすべてそろえています

Wiki@iQubeならではの

「タグをつける」「GoodJob!」など、グループウェアならではの機能を追加
Wikiがより便利に快適に

スタートページ@機能

- ・Googleの「パーソナライズドホーム」のようなインターフェース
- ・「モジュール」と呼ばれる小窓の一覧
- ・モジュールのD & Dによる移動・位置記憶
- ・提供されているガジェットを自由に追加・削除可能

スタートページ@実装

iQubeのスタートページは2ラインの実装

- ・モジュールを表示するための標準的な機能
- ・各モジュールの中身の情報表示部分の実装

モジュールの汎用化

データをなるべくコードで持たず、DBに保存するようにしている

今後の拡張と、モジュールそれぞれを同じインターフェースで実装させるため

位置情報の保存・読み込み

各ユーザのモジュール配置の情報は、配列を各ユーザごとにYAMLでDB保存

プログラムで読み取るときはその情報をデシリアライズして処理

スタートページAPI (まだだけど)

iQubeスタートページ(トップページ)のモジュールは、APIを公開しユーザが自由に作成できる！(予定)

API化を踏まえた実装

Google Gadgetsを参考に、モジュールAPIの設計を考える

各モジュールの実装を設計とマッチするように実装していく

シングルサインオン

mixiなどの主要サイトのログイン情報を登録しておく、iQubeから外部サイトにログインできる！(予定)

Rubyのよさ

- ・ホントのプログラミング未経験者でも自然と身につけていく
- ・直感的でシンプルな構文が、コードの書き手、読み手の負担を軽減する
- ・語彙が多く、「こんな書き方もできた！」みたいな発見が多い

DRY

- ・Don't Repeat Yourself
- ・DRYなコードは変化に強い
- ・コード全体の見通しもよくなる
- ・汎用的なものは共通化し、拡張が必要になったら各々対応していく

簡単なところもDRYに

```
#a = "cat" or "dog"  
case a  
when "cat"  
  b = "にゃあ"  
when "dog"  
  b = "わん"  
end
```

```
#a = "cat" or "dog"  
b =  
case a  
when "cat"  
  "にゃあ"  
when "dog"  
  "わん"  
end
```

プロジェクトの進め方



イテレーション計画

タスクかんばん



Good Job!

タスク Done!

ペアプログラミング



ペアプログラミング

ペアプログラミングとは？

- 「2人」のプログラマーが横並びに「1台」のコンピュータに向かい、同じ設計、アルゴリズム、コード、テストについて、継続的に共同作業すること。
- 1つのディスプレイ、1つのキーボード、1つのマウスを、2つの頭脳、2つの口、4つの目、4つの耳、4つの手を使って開発を進める。

ドライバーとナビゲーター

- ドライバ
 - コーディングや設計の書き下ろしをする。
- ナビゲーター
 - 構文エラー、タイプミス等の欠陥を探す。
 - 実装しようとしているコードが、要求を満たしているかの確認
- 頻繁に役割を交代

多くの
メリット

実は開発
コスト削減

ペアプログラミングによって 開発コストを削減

ペアプログラミングによって、開発コストは15%増加



生産物であるコードの欠陥は、15%減少



結果的に修正コスト(テスト, 品質保証, 後続のリリースにかかるコスト)を15% ~ 60%削減

出展「ペアプログラミング」 by ピアソン・エデュケーション

モチベーションの 向上

統計の結果
「ペアの方が作業が楽しい」と答えた人の割合
85%を上回る

要求品質の向上

- 勘違いや誤解による実装が少なくなる.
- 仕様の疑問や問題点が表れやすくなる.

設計品質の向上

- ペアの方が品質の高いコードを生産する.
- 同じ機能の場合, 単独のプログラマよりも少ない行数で実装される.

コード品質の向上

- ペアの方が、早く間違いに気がつく。
- コーディング規約が自然と守られるようになる。

問題解決の 効率化

- ペアは、ブレインストーミングとペアレビュー(どちらかが行き詰ったときのタッグチーム)を組み合わせ、問題解決の効率を最大化する。
- デバッグ時間の短縮 => ペアデバッグ

大きな 学習効果

- ペアリングは、基本的に「交代従弟制」です。教える側と教わる側がすぐに入れ替わる。
- プログラミング、開発ツールの使用方法、または仕事の進め方など継続的に学習。
- 「言葉にならない技術や習慣でさえも、パートナーには伝わる。」

信頼とチームワークの 向上 コミュニケーションの 強化

- ペアはいつもコミュニケーションを取る。そのため、ペアの入れ替えは情報の流れを太くする。
- チームとしての効率をも向上させる。

キーパーソン損失の リスク削減

- 誰もがシステムの中枢部に詳しくなり、プロジェクトからキープログラマが離脱するリスクを下げることができる。
- ペアをローテーションすることで、トラックナンバーを増加させることができる。

集中力の向上

- ペアリングは、お互いに良いプレッシャーを与え続けるので、仕事に対する集中力は非常に高い。
- ただ、一日中会議をしているようなものなので、適度の休憩が必要である。

責任感の向上

- ドライバーは絶えずナビゲーターにコーディングを見られ、手抜きすることは出来なくなる。
- 自然とプログラムに責任を持つようになり、良いものを作ろうと心掛ける。

ペアを組むべきでないとき

- 新しいことを探索するとき
- バグの原因になりかねないアイデアが複数あるとき
- 犬猿の仲の二人
- どちらかが風邪かな？と思ったとき

デメリット

- 個人の開発スタイルが崩される
- 開発ツールの強制
- 二人が座れる広さと近さが必要

より大事なのは

個人？ Or チーム？

とにかくペアプロ
は楽しいです

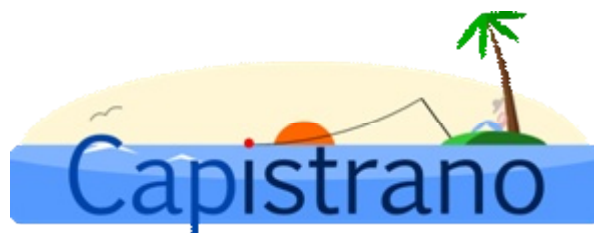
デプロイ 自動化

Capistrano

アプリケーションの配布・再起動などを簡単に行えるツール

Rubyで作られていて、簡単に拡張できる

Ruby on Rails向けに作られているため、サーバ名などを設定するだけで使用することができる



Capistrano

```
default_environment["LC_CTYPE"] = "ja_JP.UTF-8"
set :application, "xxxxx"
set :repository do
  case
  when ENV['TAG']
    "http://xxxx:12345/svn/#{application}/tags/#{ENV['TAG']}"
  when ENV['BRANCH']
    "http://xxxx:12345/svn/#{application}/branches/#{ENV['BRANCH']}"
  when ENV['TRUNK']
    "http://xxxx:12345/svn/#{application}/trunk"
  when ENV['URL']
    ENV['URL']
  else
    raise "TAG={name}, BRANCH={name}, TRUNK=, URL={url} のいずれかを必ず指定してください"
  end
end

set :deploy_to, "/var/www/#{application}"
set :user, "apache"
set :use_sudo, false
set :rails_env, "production"

desc "target を www.iqube.net に設定"
task "www.iqube.net" do

  set :target, task_call_frames.first.taskname
  set :pink_1, "1.1.1.1" # web
  set :dark_master_1, "1.1.1.2" # db
  set :dark_master_2, "1.1.1.3" # db
  set :dark_slave_21, "1.1.1.4" # db
  role :app, pink_1
  role :web, pink_1
  role :db, pink_1, :primary => true
end
```

いろいろな技術
を紹介してきました

様々な自動化
でとても効率的
になっています

でも、一番
大事なものは

効率的にし
ようとする気
持ち





ご清聴ありがとうございました

ご協力ありがとうございました

株式会社ループス・コミュニケーションズ

敬称略

齊藤 宏、加藤 龍一、稲光 隆史、
森田 真悟、数又 拓、林 伸哉、
大迫 正治、金子 憲太郎

株式会社エスディー（Ruby技術者集団）

敬称略

堂端翔、山内竜太、高瀬真宏、加藤照雄

ご質問をどうぞ

お仕事はこちらまで

川端光義 kawabata@agileware.jp

堂端翔 douhashi@sdpack.net