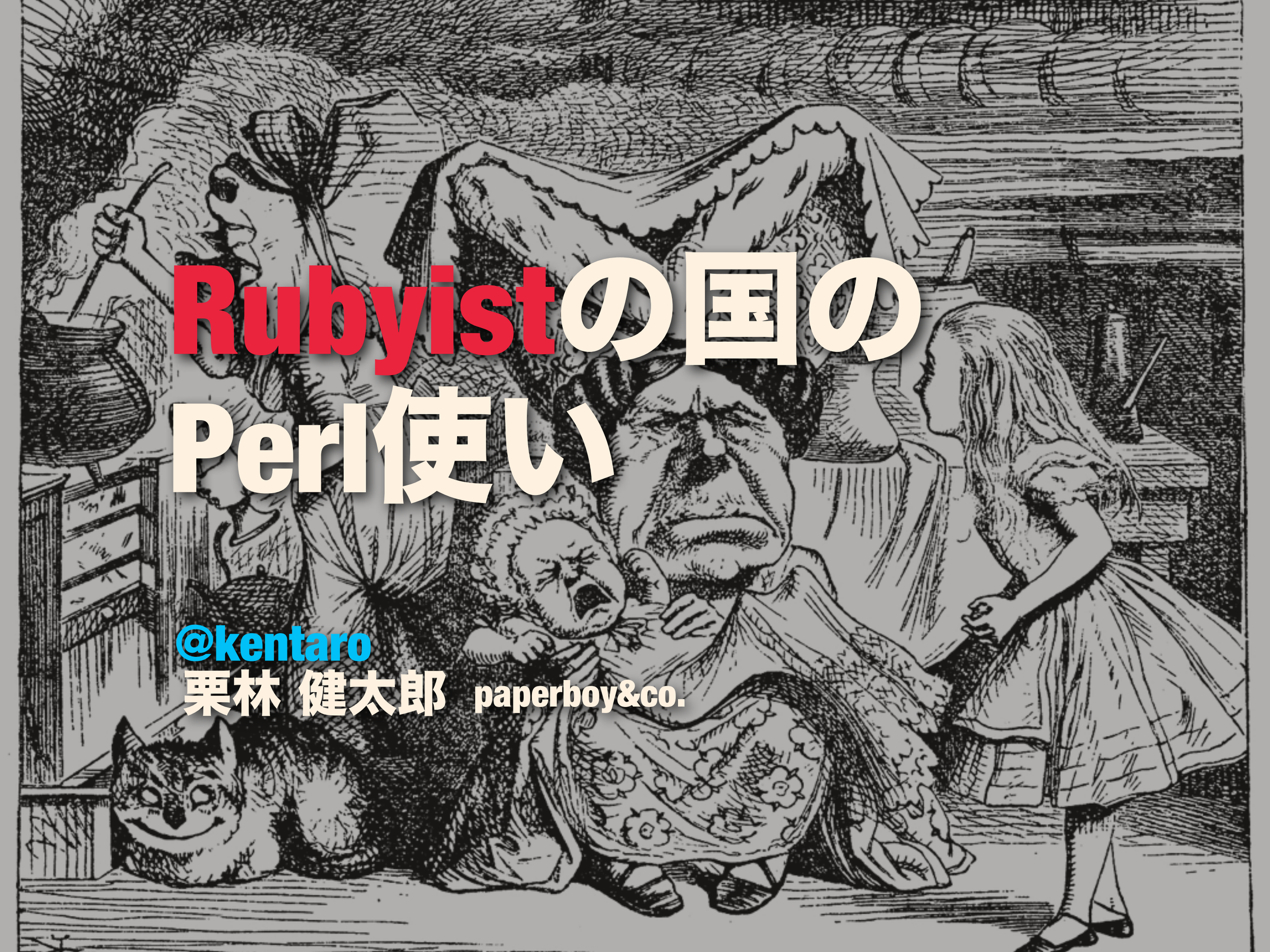


Rubyistの国のPerl使い

@kentaro
栗林 健太郎 paperboy&co.

@kentaro
栗林 健太郎

Kyoto.pm所属
エンジニア
Perler, ルビースト
弊社公式コンテンツ
(New!)

paperboy&co.

paperboy&co.

【採用情報】スマートフォン向けゲームデザイナーの募集を開始しました。

京都
おこしやす!プロジェクト
～そうだ、ペパボ、行こう～

メディア公式
アカウント一覧

サービス一覧

- SERVER ロリポップ!
- SERVER ヘテムル
- DOMAIN ムームードメイン
- BLOG JUGEM

- 年間580円からの格安ドメイン取得
- 月額875円から、気軽にお店開店！
- 楽しいショッピングモール！カラメル
- 携帯完全対応！無料ブログサービス
- インタビューコミュニケーション
- クリエイティブなレンタルサーバ heteml
- パブーでだれでも電子書籍出版・販売
- たったの105円から 簡単＆気軽にHP作成
- 【容量無制限】大切な写真を保管・共有！
- 格安ホームページ作成はグーペ！

スタッフインタビュー #02
「モノづくりの原動力は、自分自身や身近な人たち」 →

NEWS ペパボ最新ニュース　RSSを購読する

2012年09月14日（金）［ プレスリリース ］
ゲームカンパニー「ペーパーボーヤ」プロダクト第3弾 転がるスイカや爆弾を取り除き、花火をたくさん打ち上げろ！シンプルアクションゲームiPhoneアプリ「ハナビショット！」提供開始

2012年09月07日（金）［ 講演・出演情報 ］
『Rails Girls Tokyo』にエンジニアの柴田博志がコーチとして参加します

2012年09月07日（金）［ 講演・出演情報 , ブクログ ］

paperboy&co.

募集職種一覧

現在募集中の職種一覧です。

paperboy&co.で働く

株式会社paperboy&co.では、一緒に働く仲間に大切にしてほしい３つのことがあります。
これら３つのことを大切にしていただける方、持ち合わせている方からのご応募をお待ちしております。

◎みんなと仲良くすること
一日のうち、家族よりも長い時間接する職場の仲間とは、良好な関係を築きましょう。

◎ファンを増やすこと
スタッフ全員がpaperboy&co.という会社やpaperboy&co.の提供するサービスのファンであり、
ファンを増やすための努力を惜しまないようにしましょう。

◎アウトプットすること
[illegible]積極的に発言しましょう。

技術基盤整備エンジニア

▼エンジニア ▼デザイナー ▼ディレクター ▼カスタマーサービス ▼バックオフィス ▼ペーパーボーヤ

▸ エンジニア

職種名	勤務地
JUGEM インフラエンジニア（正社員） NEW !	東京本社
社内インフラエンジニア（正社員） NEW !	東京本社
ヘテムル　ソフトウェアエンジニア（正社員） NEW !	東京本社
ECサービス　スマートフォンエンジニア（正社員） NEW !	東京本社
カラメル　ソフトウェアエンジニア（正社員） NEW !	東京本社
iPhoneアプリエンジニア（正社員） NEW !	東京本社
Androidアプリエンジニア（正社員） NEW !	東京本社
カラーミーショップ　インフラエンジニア（正社員）	東京本社
カラーミーショップ　ソフトウェアエンジニア（正社員）	東京本社
技術基盤整備エンジニア（正社員）	東京本社
[redacted]インフラエンジニア（正社員）	福岡支社
技術基盤整備エンジニア（正社員）	福岡支社

手っ取り早く。
やりたいことだけ。

Quick deploy & Flexible resources.

ダッシュボード

OR

Log in with Twitter

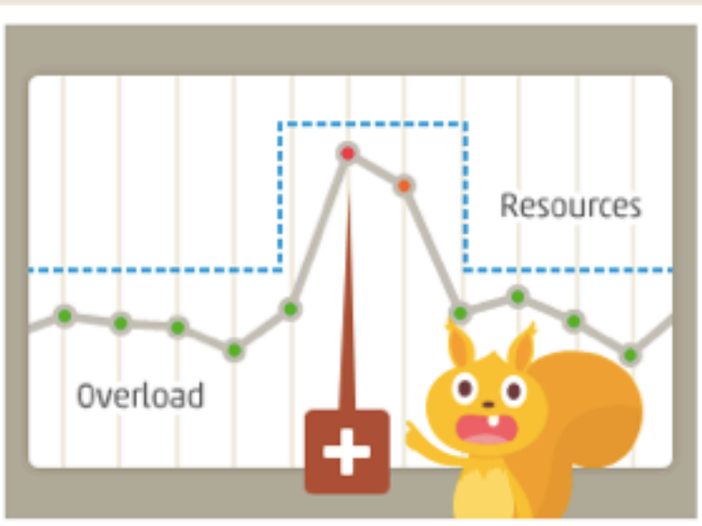
Log in with Facebook

Log in with GitHub

コードを書く時間を1秒でも長く。
Sqale（スケール）は、開発者のためのホスティングサービスです。

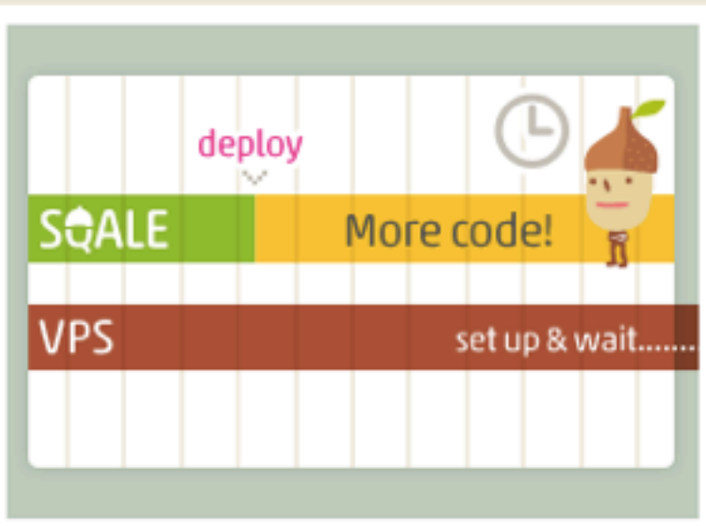
Ruby on Rails 対応。月額 940円から。

Sqale なら、あなたが作った Webアプリケーションを手



必要になったら、そのときに考えよう。

あなたのアプリケーションがいつヒットするか、事前に



コード！コード！もっとコード！

VPS のようにサーバーの構築や運用に時間を費やすの

＿人人人人人人人＿

＞　突然のRuby化　＜

￣^Y^Y^Y^Y^Y^Y^Y^Y^Y￣

のんたん

mizzy

まつけん

終了

paperboy&co.

Kyoto Perl Mongers
perl community in kyoto

2012.07.29

Kyoto.pm Tech Talks 02を開催します！

こんにちは、@shiba_yu36 です。

次回のKyoto.pm開催のお知らせです！

次回は8/18(土) にTech Talksの形式でKyoto.pmを開催しようと思います。Perlに興味があれば歓迎なので、是非気軽にお越しください！

詳細としては

- 日時 : 2012/8/18 (土) 15:30-18:00
- 場所 : 株式会社はてな(http://www.hatena.ne.jp/company/location)
- 参加費 : 100円〜500円 学生は無料(当日のお菓子に使います。人数によって変動)

としています。

Kyoto.pmから来ました

Kyoto.pm Tech Talks 02 懇親会

また同時に**発表者も募集しています。**今回も気軽に発表しやすいように、通常の発表は10〜40分、LTは3もしくは5分とある程度自由に発表時間を決められるようにしています。内容はPerlに関わればなんでも大丈夫です*1。YAPC::Asia 2012の練習がてら、是非応募お願いします！

プロフィール

id:kyotopm PRO
読者になる 6
ともだち申請

検索

Search

最新記事

Kyoto.pm Tech Talks 02を開催します！
Kyoto.pm 町家ハッカソン 01を開催します！
Kyoto.pm Tech Talks #01 レポート

PrePAN
Social Reviewing for Perl Modules

Submit Module

0 kentaro@github

Requests for Reviews

Class::LOP - The Lightweight Object Protocol

I write a lot of modules that manipulate classes. Two popular ones are Sub::Mage and Class::Monkey (monkey patching). I was sick of writing the same boilerplate code, but didn't need a lot of the features current OOP frameworks offer. I built Class::LOP to help me write faster code, and build Class/OOP framework modules based on my needs. It takes care of all the dirty work for me, so I don't have to.

bradhaywood@github 2012-09-01 09:47:56 3 comments

Proc::Child - Controll external program running in background

This module controlls external programs running in background. It's inspired by childprocess (Ruby).

I know some modules such as IO::Async::Process or AnyEvent::Util::run_cmd, but I want to controll processes with * not * using callbacks.

Cside@github 2012-08-30 00:07:31 3 comments

Dist::Zilla::Plugin::PrePAN - Set PrePAN URLs in your meta files

This is based on Module::Install::PrePAN We'd like to have PrePAN links on MetaCPAN and something like this would make it easier for authors using Dist::Zilla to automatically include links back to PrePAN for module discussion.

In your META.json, for example, you'd see something like this:

```
"resources" : {
  "X_prepan_author" : "http://prepan.org/user/3Yz7PYrBzQ",
  "X_prepan_module" : "http://prepan.org/module/429En4oFdi",
  "bugtracker" : {
     "web" : "http://github.com/oalders/HTTP-CookieMon
  },
  "homepage" : "https://github.com/oalders/http-cookiemonst
  "repository" : {
```

NAME

Cinnamon - A minimalistic deploy tool

SYNOPSIS

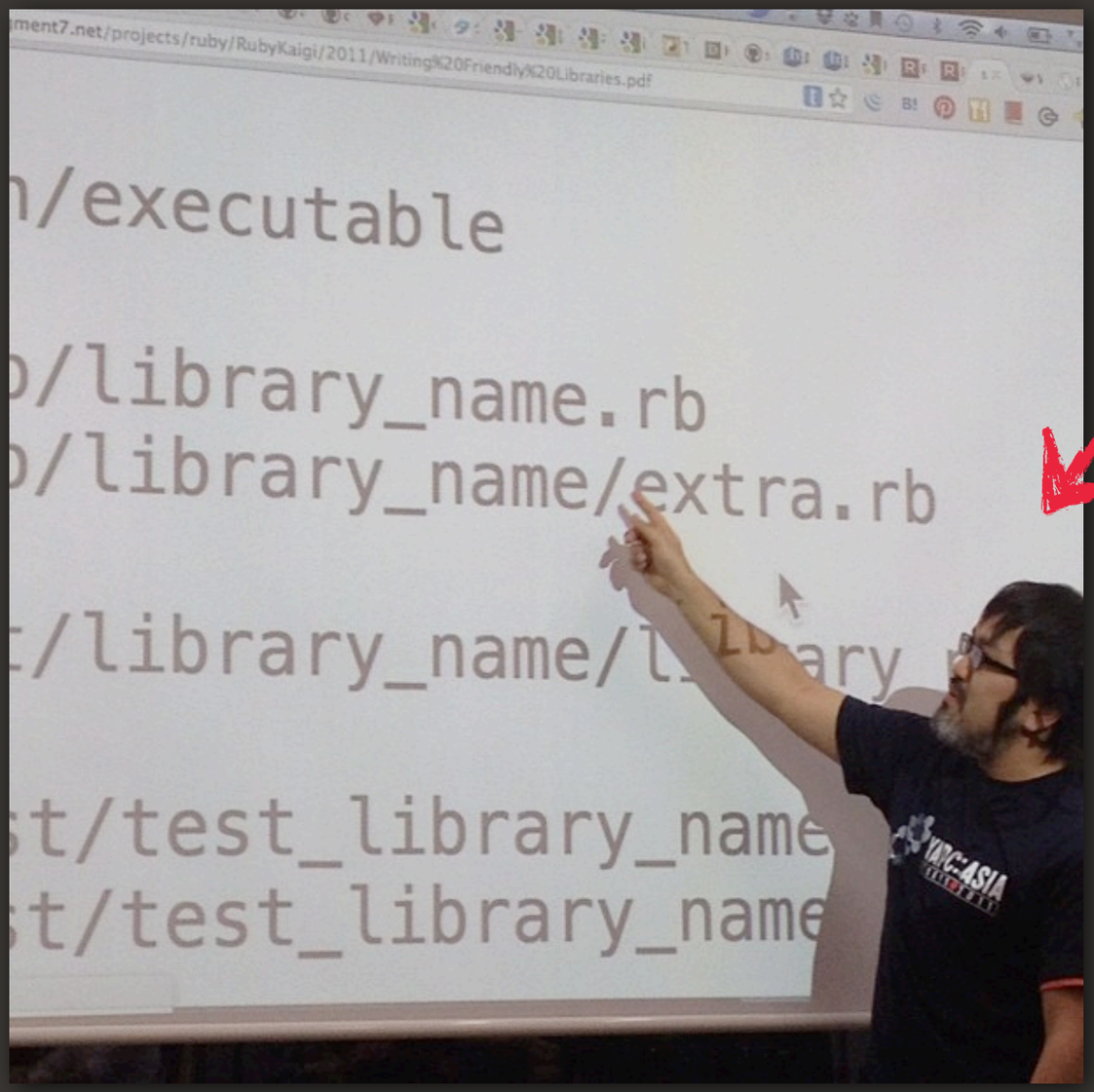
```
use strict;
use warnings;

# Exports some commands
use Cinnamon::DSL;

my $application = 'My::App';

# It's required if you want to login to remote host
set user => 'johndoe';

# User defined params to use later
set application => $application;
set repository  => "git://git.example.com/projects/$application";

# Lazily evaluated if passed as a code
set lazy_value  => sub {
    #...
};

# Roles
role development => 'development.example.com', {
    deploy_to => "/home/app/www/$application-devel",
    branch    => "develop",
};

# Lazily evaluated if passed as a code
role production  => sub {
    my $res   = LWP::UserAgent->get('http://servers.example.com/api/hosts');
    my $hosts = decode_json $res->content;
       $hosts;
}, {
    deploy_to => "/home/app/www/$application",
    branch    => "master",
};

# Tasks
task update => sub {
    my ($host, @args) = @_;
    my $deploy_to = get('deploy_to');
    my $branch = 'origin/' . get('branch');

    # Executed on localhost
    run 'some', 'command';
```

https://speakerdeck.com/u/kentaro/p/datamapper

```
sub new {
    my $class  = shift;
    my $config = pit_get('tanpaku.grouptube.jp' => r
equire => {
        username => 'username required',
        password => 'password required',
    });

    ble
}

package
use utf
use str
use war

use URI
use WWW
use Cla
    new
    rw
);

use Readonly;
Readonly my $domain => 'http://tanpaku.grouptube.jp'
;

sub new {
    my ($class, $config) = @_;
    bless {
        ua      => WWW::Mechanize->new,
```

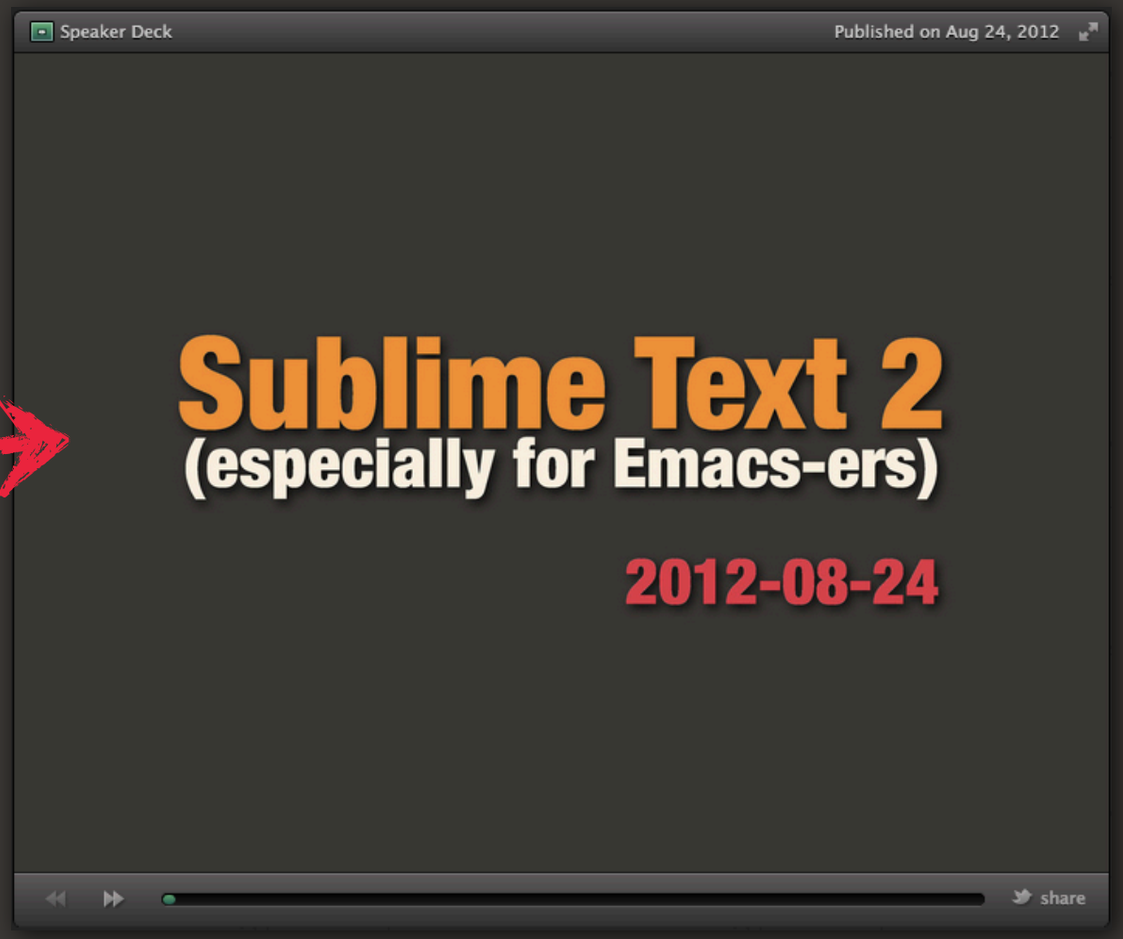
```
package Nippo::Entry;
use utf8;
use strict;
use warnings;

use autodie;
use FindBin;
use File::Temp;
use Time::Piece;

mplate t
;
args{fil
en my $f
;
, <main:
    }

    bless {
        %args,
        template => $template,
    }, $class;
}
```

最近はPerl全然書いてない(たまにコードレビューするぐらい)



YAPCのTシャツを着てRubyの会でrubygemsをDISってる私の勇姿です

Emacsの会なのにSubilme Text2の話のみのスライド



というわけで、今日はPerlには特に関係のない話をします。

ペパボでは最近なにやってるんですか？

う〜ん…なんというか、いろいろですね…。

http://www.flickr.com/photos/timjoyfamily/532089575/

アウトプットの大切さ

アウトプットすること

意見やアイデアは、ミーティング、社内SNS、メールなどで積極的に発言しましょう。不採用かもしれないと思っても、他のアイデアと合わさることで新しいものになることがあります。そのために、すべてのアイデアに耳を傾けると同時に、頭に浮かんだことをどんどん外に出しましょう。
また、インターネットで表現し続ける、コミュニケーションし続けることを楽しんで、自分たちが一番のユーザーであることを心掛けましょう。

http://www.paperboy.co.jp/recruit/important/

アウトプット

一般化

外に出すことで内部事情を一般化できる

検証可

アウトプットされないことには検証しようがない

創発性

アウトプットの重なり合いが新しいものを生む

内向き

開発環境整備
開発コミュニケーション
アジャイル開発

外向き

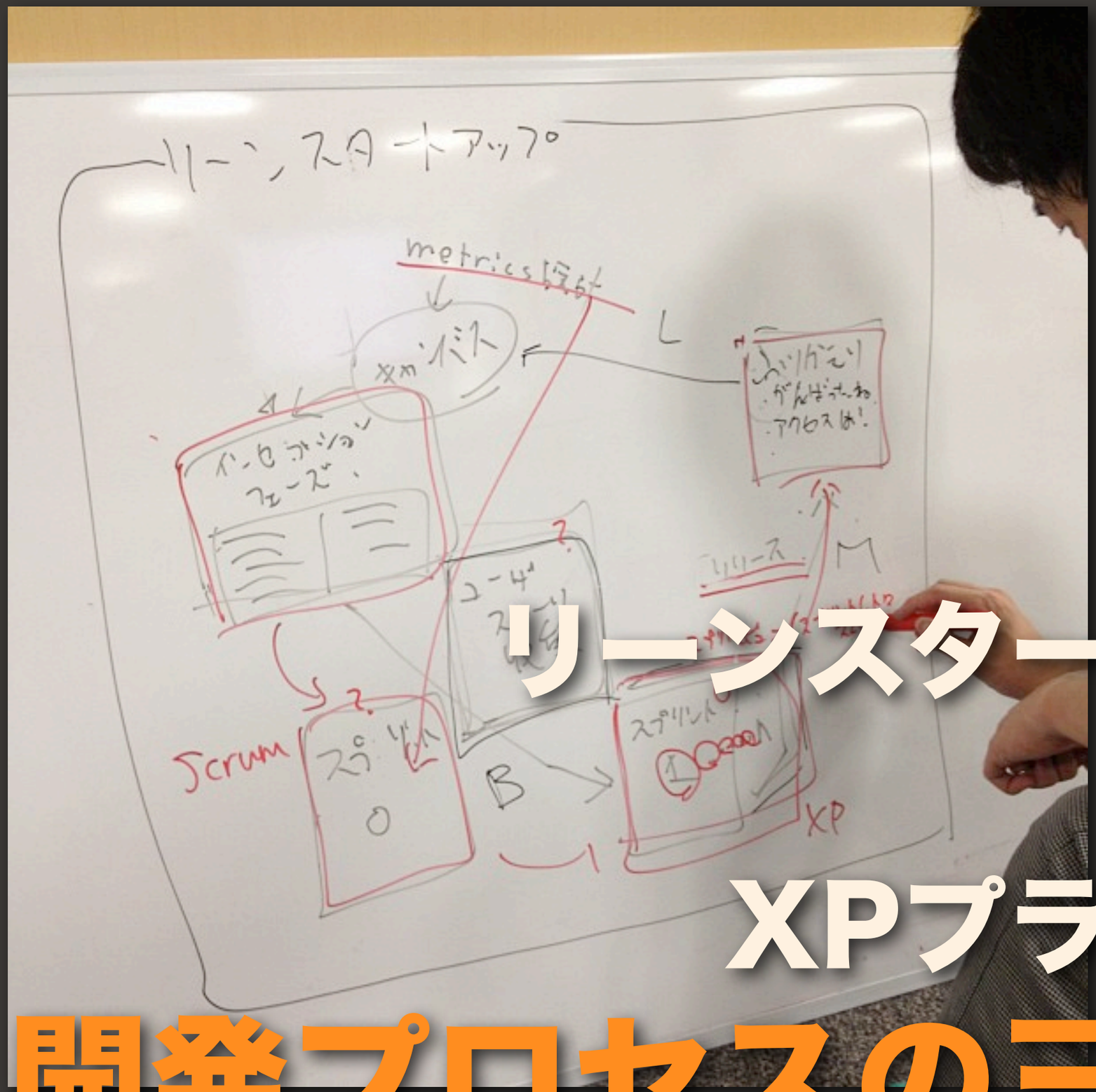
サービスリリース
リーンスタートアップ
技術的アウトプット

開発プロセスの改善

http://www.flickr.com/photos/betsyweber/6719452305/

w/@hsbt

隣席のるびりすと氏



リーンスタートアップ
スクラム
XPプラクティス
開発プロセスの三層構造

外部環境

お客さん

リーンキャンバス

リーン

PO

マネージャ

社長

インセプションデッキ

Scrum

チーム

広報

デザイナ

法務

スプリント計画・レビュー

XP

開発者

コードレビュー
ペアプロ

コンセプトを固める

http://www.flickr.com/photos/goodoldsparky/2716165693/

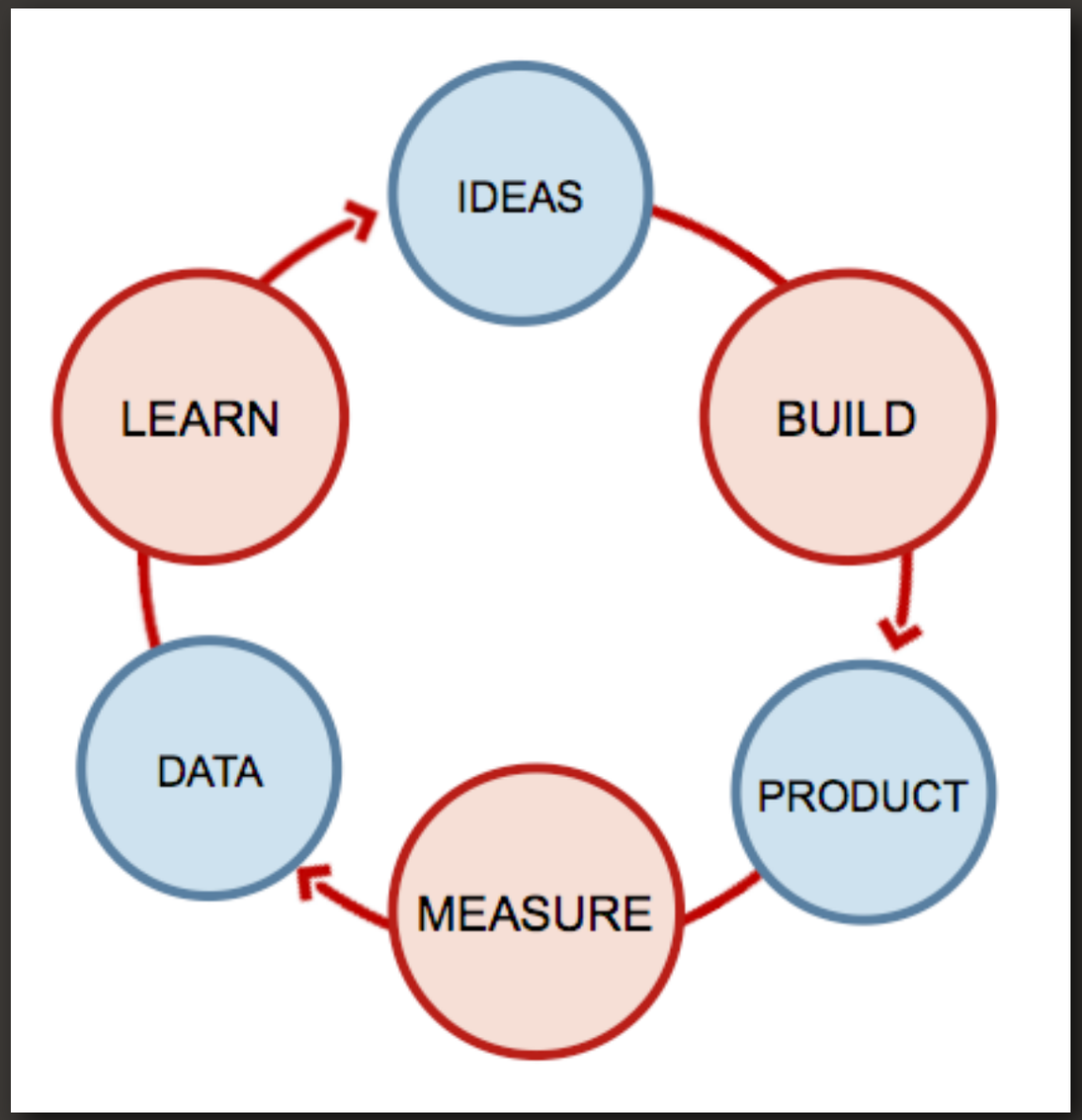
開発者のためのリーンスタートアップ

id:antipop
paperboy&co.

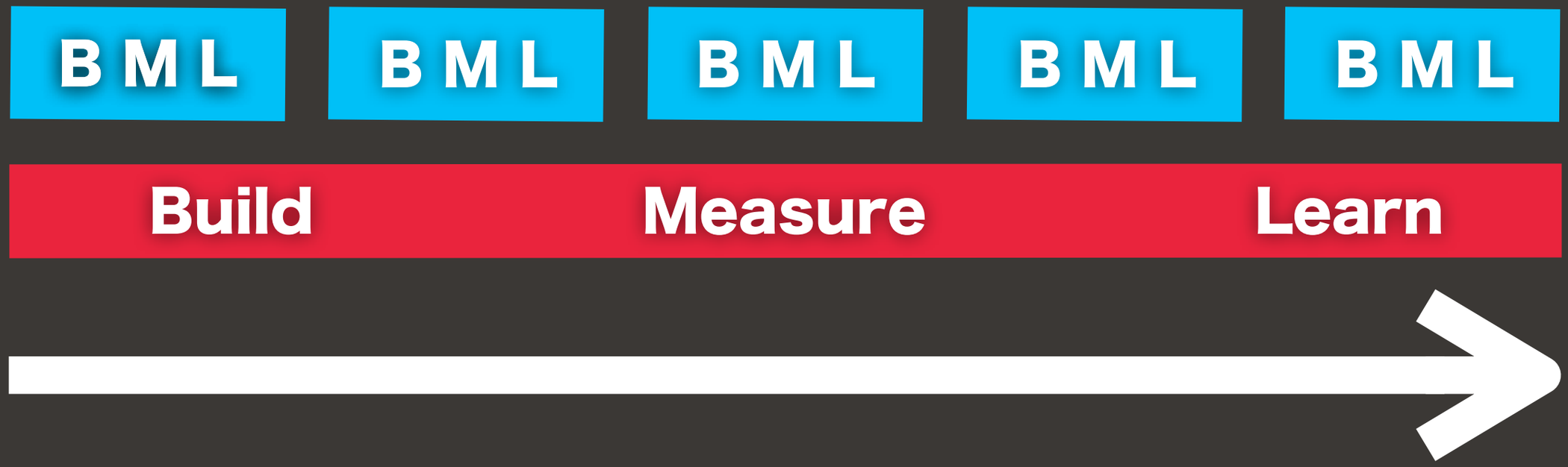
2012-06-28
ペパボ会議室

http://blog.kentarok.org/entry/2012/07/10/185408

Build
Measure
Learn



バッチサイズ



(1)サービス開発を科学実験のように仮説とその検証を通じて行うこと
と
(2)まずは仮説検証のための必要最小限の製品を作ること
(3)仮説検証ループをできるだけ速く回すこと

Problem	Solution	Unique Value Proposition	Unfair Advantage	Customer Segments
Top 3 problems	Top 3 features	Single, clear, compelling message that st[illegible] and [illegible]	Can't be easily copied or bought	Target customers
	Key [illegible]		Ch[illegible]	
	Key activities you [illegible]		Path to [illegible]	
Cost Structure		Revenue Streams		
Customer Acquisition Costs Distribution Costs Hosting People, etc.		Revenue Model Life Time Value Revenue Gross Margin		
PRODUCT		MARKET		

Introduction to Lean Canvas

id:antipop
paperboy&co.
2012-07-03

http://blog.kentarok.org/entry/2012/07/10/185408

http://www.imi.ie/news-and-events/wp-content/uploads/2011/11/IMI-Ash-Maurya-Running-Lean-Presentation.pdf

Problem

Top 3 problems

問題

1

Solution

解決

Top 3 features

4

Key Metrics

検証

Key activities you measure

8

Unique Value Proposition

Single, clear, compelling message that states why you are different and worth buying

価値

3

Unfair Advantage

優位

Can't be easily copied or bought

9

Channels

広報

Path to customers

5

Customer Segments

Target customers

顧客

2

Cost Structure

支出

Customer Acquisition Costs
Distribution Costs
Hosting
People, etc.

7

Revenue Streams

収入

Revenue Model
Life Time Value
Revenue
Gross Margin

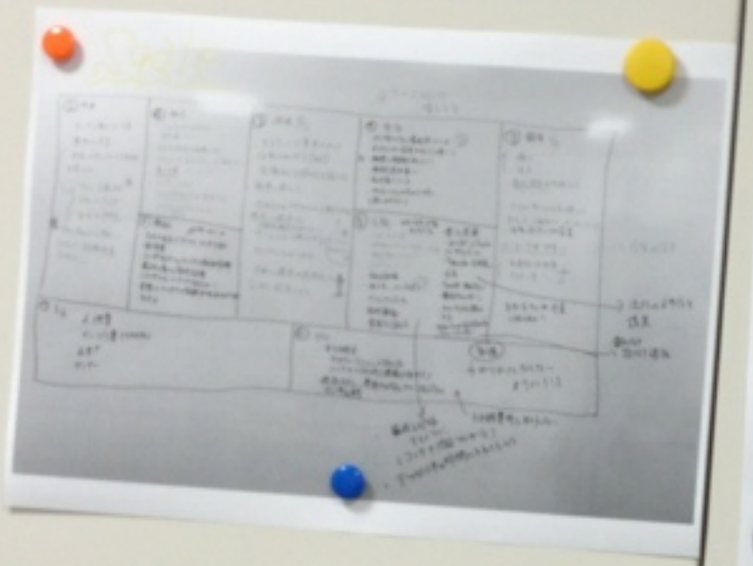
6

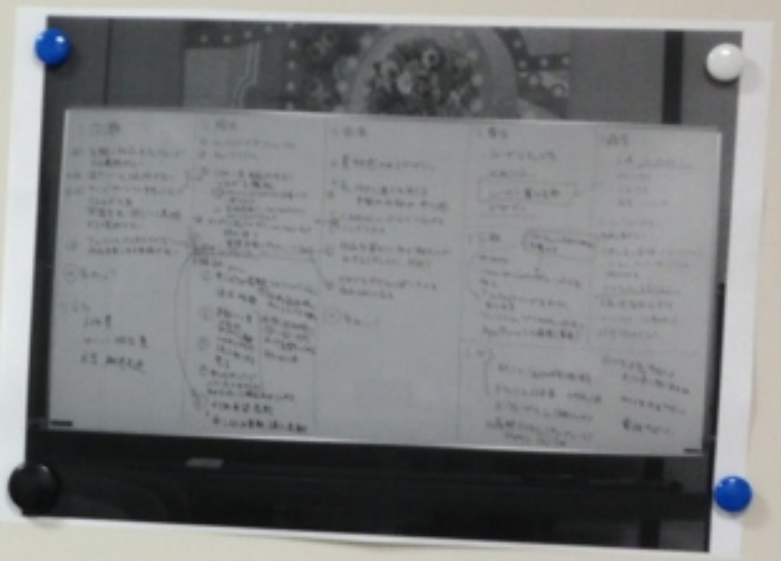
PRODUCT

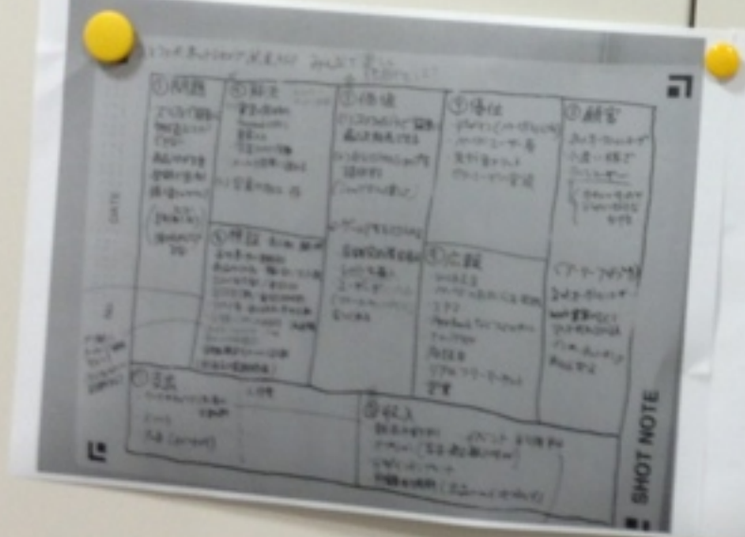
MARKET

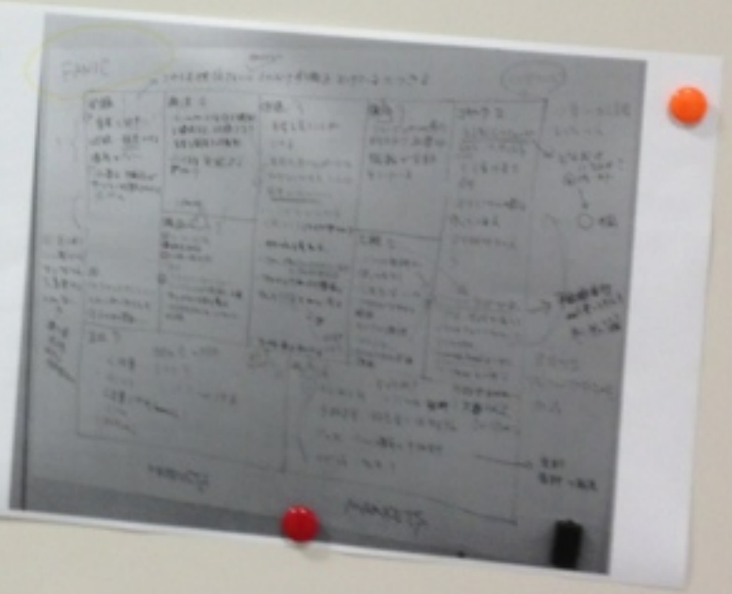
http://www.imi.ie/news-and-events/wp-content/uploads/2011/11/IMI-Ash-Maurya-Running-Lean-Presentation.pdf

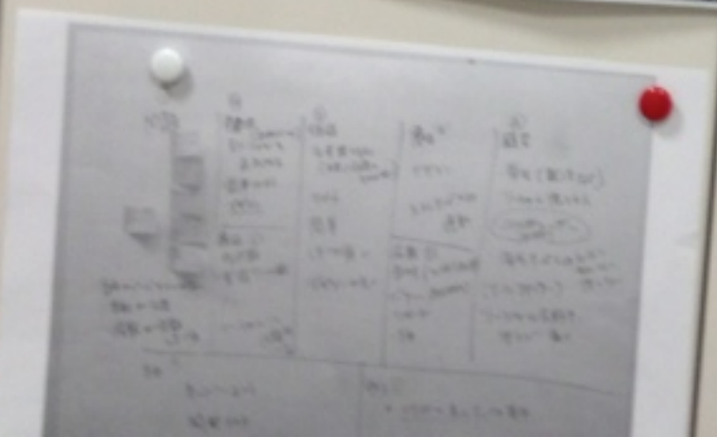
社内掲示板

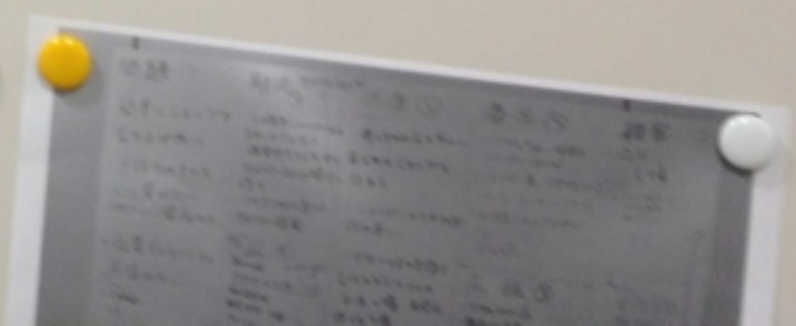


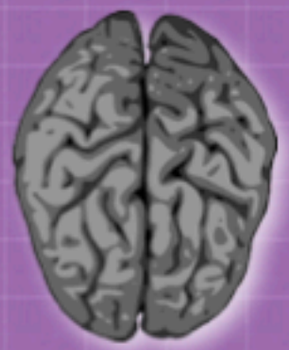












Head First

インセプションデッキ

頭とからだで覚えるデッキ作りの基本

agile tour Speaker 2011

西村直人
n-nishimura@esm.co.jp

http://d.hatena.ne.jp/nawoto/20111011/1318303619

インセプションデッキ

<プロジェクトの名前>

<スポンサーの名前>

我われはなぜここにいるのか

- 大事な理由その1
- 大事な理由その2
- 大事な理由その3

<このプロジェクトの根幹に
関わる理由を1つ、ここに書く

エレベーターピッチ

- [潜在的なニーズを満たしたり、潜在的な課題を解決したり] したい
- [対象顧客] 向けの、
- [プロダクト名] というプロダクトは、
- [プロダクトのカテゴリー] です。
- これは [重要な利点、対価に見合う説得力のある理由] ができ、
- [代替手段の最右翼] とは違って、
- [差別化の決定的な特徴] が備わっている。

パッケージデザイン

(プロダクトの名前)

(素敵な写真)

(最高のキャッチコピー)

(ユーザーへのアピールその1)

(ユーザーへのアピールその2)

(ユーザーへのアピールその3)

やらないことリスト

やる	やらない

あとで決める

プロジェクトコミュニティは...

(ほげほげ部門)

(他のチーム)

コアチーム

(○○グループ)

関係者全員を！

...思っているよりもずっと大きい！

技術的な解決策の概要

・<プログラミング言語>

・<ツール>

・<その他の要素技術>

←対象外

夜も眠れなくなるような不安は何だろう?

・もし起きたらこわーいこと、その1

・もし起きたらこわーいこと、その2

・もし起きたらこわーいこと、その3

俺たちの"チーム"

人数	役割	強みや
1	アー	ときに分析す る。 テストも書んで手伝える。 素早い繰り返し型の開発スタイルで働ける。
2	開発者	C#、MVC.NET、jQuery、SQL ユニットテスト、リファクタリング、TDD、 継続的インテグレーション
0.5	マネージャ	顧客と直接顔を合わせてのコミュニケーションを担当する。 状況報告、スコープ調整、予算管理、レポートラインへの報告

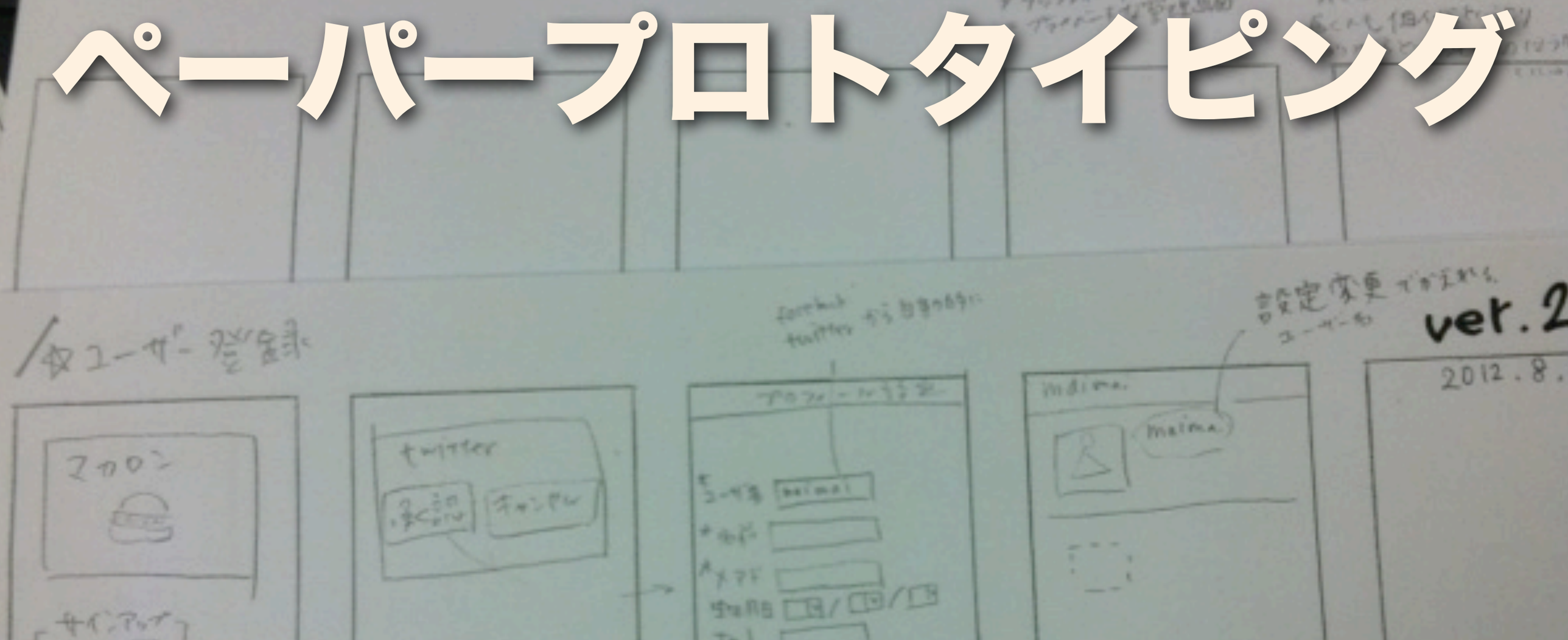
https://speakerdeck.com/u/nawoto/p/head-first-inception-deck?slide=11

チームで開発する

http://www.flickr.com/photos/cdm/2336025560/

ユーザストーリ収集

ペーパープロトタイピング



スプリント計画

プロダクトバックログ

スプリントバックログ

スプリントレビュー

44 spec/requests/static_pages_spec.rb

show inline notes

View file @ 56ae9b4

```
...  ...  @@ -1,12 +1,44 @@
1    1     # -*- coding: utf-8 -*-
2          -
3    2     require 'spec_helper'
4    3
5          -describe "StaticPages" do
6          -  describe "GET root_path" do
7          -    it "トップページが表示される" do
8          -      get root_path
9          -      response.status.should be(200)
     4     +describe 'Static Pages' do
     5     +
     6     +  subject { page }
     7     +
     8     +  describe 'home page(top page)' do
     9     +    before { visit root_path }
     10    +
     11    +    it { should have_selector('title', text: 'Macaron') }
```

コードレビュー

8

kentaro repo collab — 11 days ago

議論の別れるところだとは思いますが、shouldは(RSpec作者的には)オワコン認定されつつあるみたいなので、 `expect` の使用を検討してもいいかもです。@hsbt さんあたりにきいてみてください。

- https://www.relishapp.com/rspec/rspec-expectations/v/2-11/docs/changelog

hsbt repo collab — 11 days ago

未来は expect なので、こちらを使いましょう

inouetakuya repo collab — 10 days ago

今回は、タイムリミットが迫っているので、使い慣れた should のほうを使います（一部 expect）。expect 使えるようにしておきます

inouetakuya repo collab — 9 days ago

should を expect に置き換えました `d49c466` `c7b84e6`

inouetakuya repo collab — 9 days ago

このあたりのこと、Wiki に参考リンクを貼ったので、後で見ておいてくださいね > **@yuma300**

yuma300 repo collab — 8 days ago


ペアプログラミング

Rubyの世界の継続的デリバリ

Continuous Delivery in Ruby World

柴田 博志

SHIBATA Hiroshi

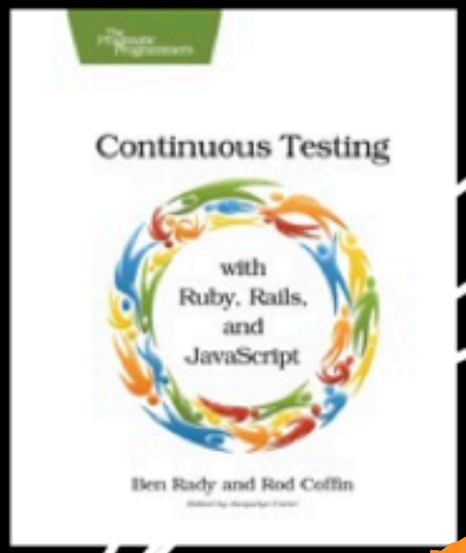
paperboy&co.
asakusa.rb

paperboy&co., Inc.

発表場所 Sapporo Ruby Kaigi 2012　2012-09-16(Sun)

https://speakerdeck.com/u/hsbt/p/20120916-sappororubykaigi2012

サイクルタイム
を短縮する



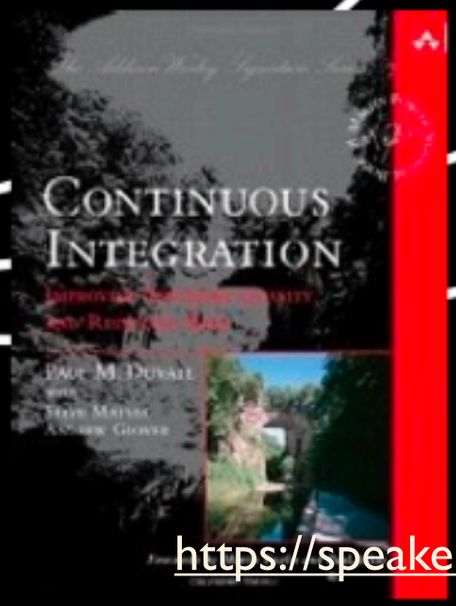
ユーザー

PL 他

QA

運用

開発者



https://speakerdeck.com/u/hsbt/p/20120916-sappororubykaigi2012?slide=59

There's much more...

赤線は開発者にとって特に重要な事項

http://www.slideshare.net/startuplessonslearned/2012-05-15-eric-ries-the-lean-startup-pwc-canada

非Rubyな会社で

仕事に  を

持ち込むための

5つの方法

@kentaro
栗林 健太郎 paperboy&co.

速いサイクル

自動化

capistrano/webistrano
chef/puppet

確実性

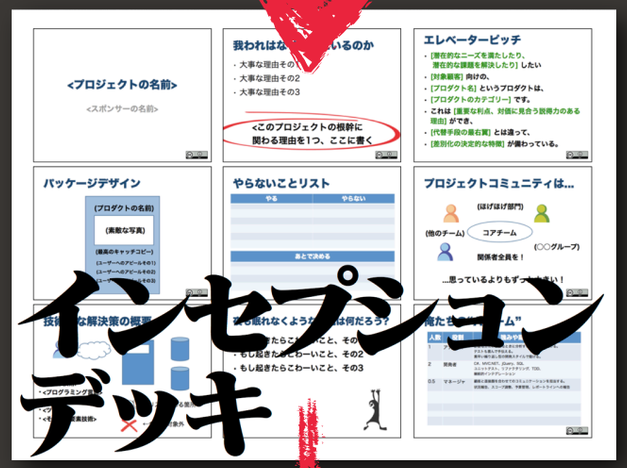
tests, more tests
continuous integration

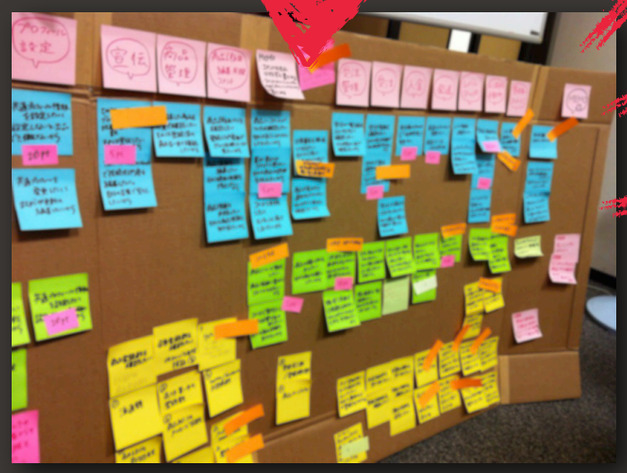
安全性

developer sandbox
incremental deployment

開発プロセスが設計を変える

http://www.flickr.com/photos/wolflawlibrary/4703808270/





(役割)として
(ゴール)を達成
したい。それは
(理由)のためだ

ユーザストーリ

(役割)として
(ゴール)を達成
したい。それは
(理由)のためだ

(役割)として
(ゴール)を達成
したい。それは
(理由)のためだ

(役割)として
(ゴール)を達成
したい。それは
(理由)のためだ

M

V

C

controllers

```
package Controller::Cart;
sub create {
    user->add_cart(params{book_id});
    # ...
}
```

```
package Controller::Follow;
sub create {
    user->add_follow(params{user_id});
    # ...
}
```

```
package Controller::Blog;
sub create {
    user->add_post(params{text});
    # ...
}
```

model

```
package Model::User;

sub add_cart {
    # ...
}

sub add_follow {
    # ...
}

sub add_post {
    # ...
}
```

Too many methods.

何が問題か？

機能と実装に関連性がない

ストーリの見えない設計

何が問題か？

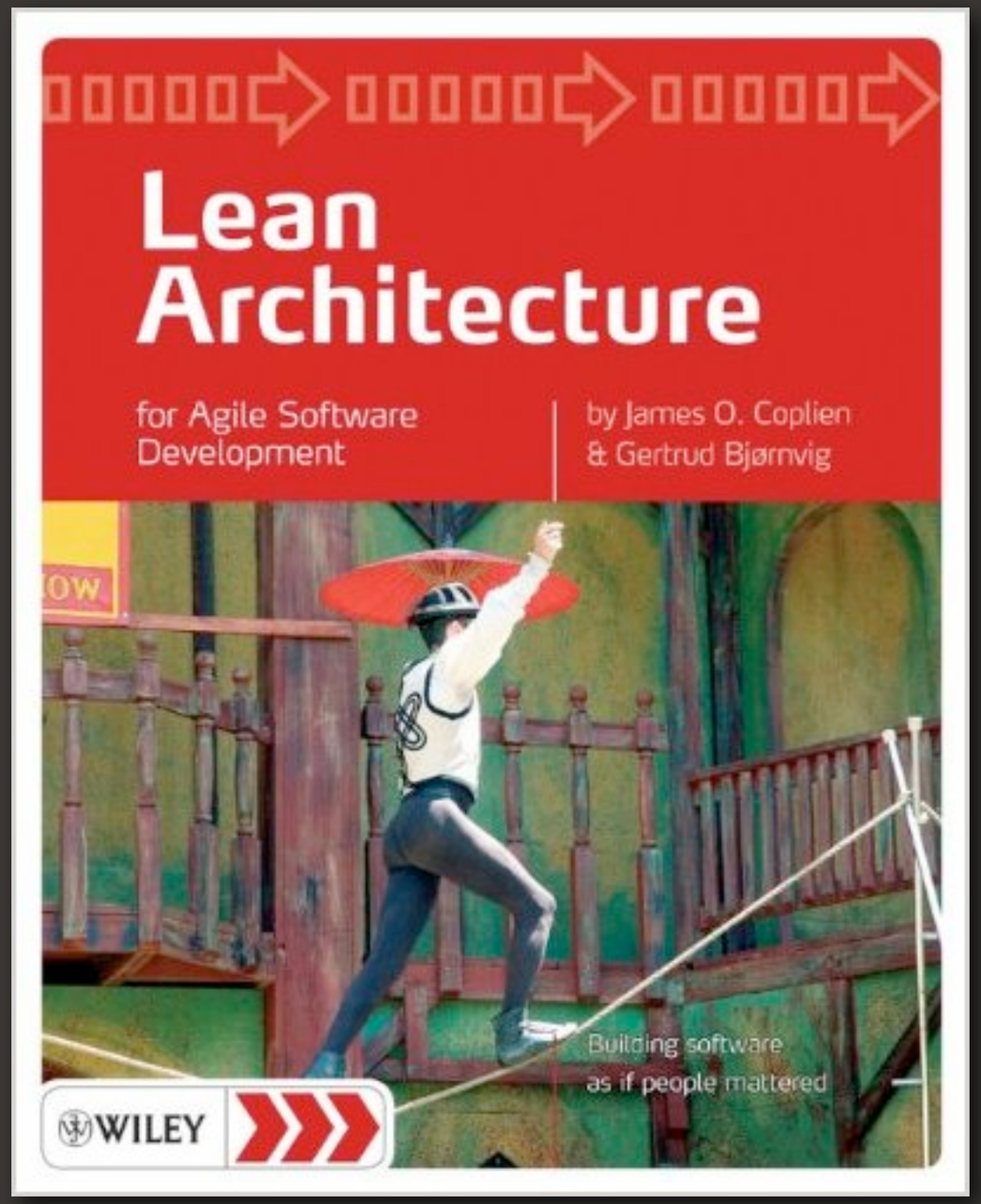
“Slim Controller, fat model”, but it’s too fat to be a model.

参考: http://lab.ursm.jp/blog/2012/09/17/sapporo-rubykaigi-2012/

DCI

Data Context Interaction

“We can translate use case scenarios into algorithms, just in time, as new senarios enter the business process. We encode these algorithms directly as *role methods*.”



コンテキスト　コール　データ

時間

会社

Employee

会社員
・仕事をする
・プログラム書く
・・・

Employee

User

実行時結合

・kentaro
・male
・1976.12.12

プライベート

Ikemen

イケメン
・飲みに行く
・本を読む
・・・・

Ikemen

controller

```
package Controller::Cart;
sub create {
    my $res = Context::Cart->new(
        user,
        params{item_id},
    )->call;

    # ...
}
```

context

```
package Context::Cart
sub new {
    my ($class, $user, $item_id) = @_;
    $user->extend(“Role::Customer”);
    bless {
        user    => $user,
        item_id => $item_id,
    }, $class;
}

sub call {
    my $self = shift;
    $self->{user}->add_cart(
        $self->{item_id}
    );
}

{
    package Role::Customer;
    sub add_cart { ... }
}
```

`$user->extend` is enabled by my experimental module, Class::Extendable.

https://github.com/kentaro/Class-Extendable

context

(役割)として
(ゴール)を達成
したい。それは
(理由)のためだ

1対1

Data
+
Role

(役割)として
(ゴール)を達成
したい。それは
(理由)のためだ

1対1

Data
+
Role

DCIによって

ストーリー指向の設計が可能

Too fat model問題も解決

DCIによって

開発プロセスと整合する設計・実装手法を入手

まとめ

まとめ

アウトプット重要

開発者のアウトプットを高める

開発プロセスを改善する

DCIもあるよ

ご清聴ありがとうございました