

1日でわかるWebサービス 制作実践講座

- マーケジン・アカデミー
- 2013年11月19日(火)
- by Yusuke Wada and Osamu Ise

小話

ディレクターとクリエイター
お互いの仕事に興味を持つ関係性

アンケート！

皆さんの職種は？

- * 可能ならばやっているサービスについて
- * その規模について

アンケートその2

今日得たいもの... ?

軽い自己紹介

オモロキでボケてを開発・運営

イセオサム

- おっさむ
- ディレクター

和田裕介

- ゆっけさん
- デベロッパー

ディレクターとクリエーターの関係

そこで... セルフアンケート

2人はいい関係ですか？

今回のテーマ

- ・そもそもディクターとは何か？
- ・僕らの実例を詳しく！
- ・共通言語になるべくWeb技術について
- ・ディレクターとしてのレベルアップ

つまり
ディレクターとクリエーター
2人が新しいプロダクトを生み出して運用していく心構えとコツ
を紹介します！

1時間目

ディレクターとはなにか？

まず、自己紹介

イセオサムです。

株式会社HALO COO/Co-Founder

株式会社オモロキ CSO

【経歴】

慶應義塾大学経済学部卒業後、日本テレビ放送網に就職、ズームインス
ーパーの制作に関わる。

その後、株式会社オプトにてモバイルのメディアプランニングを統括。
2008年にスマホアプリを企画・制作する株式会社ハロを共同創業。

2013年から株式会社オモロキにCSOとして参画。
手がけたアプリは累計500万インストールを突破。

【代表的なアプリ】



ポケて



ロックジョイ



カオコレ

ディレクターとはなんぞ！？

- いったい、何をする人？
- 1年生でも大丈夫？
- プロデューサーとの違いは？

Q：ディレクターの仕事とは？

進行管理だけではなさそうだ

Director

ディレクター または ダイレクタ (英: *director*)
とは、監督全般を指す言葉

*Wikipedia*より

監督らしいけど、1年生でも大丈夫？

ちょっと深掘ってみる

制作物の作品としての質に責任を持つ者のこと。
その責務を全うするために、企画・立案・制作に
関与して業務全般をつかさどる場合もある
(この場合は「ディレクター職の者がプランナー
職・プロデューサー職も兼任することが多い」)
*Wikipedia*より

んん！！

一筋縄ではいかなそうだ

ちなみに

テレビや映画の制作現場では、段階があります

- AD : Assistant Director (アシスタント)
- D : Director (コーナーを任せる担当)
- CD : Chief Director (演出、監督的な)

プロデューサーとの違いは？

ディレクター本来の担当範囲は企画・製作に及ぶものではない。制作物の経済的責任を持つプロデューサーと対をなすポジション。

つまり、プロダクトへの責任をもつ人のこと、ですね

ディレクターの仕事で大事なこと

プロダクトの演出、味付け。世界観をつくり、方向性を定め、チームを導くこと

チーム構成と役割分担

Producer : お金周りを含む企画 자체、製作に対して責任を持つ

Director : プロダクトの作品としての質に責任を持つ

Designer : プロダクトのデザインを行う

Engineer : プロダクトの制作を行う

ディレクターが身に付けるべきスキル

（このスライドは、ディレクターとして必要なスキルをリストするためのフレームです。）

新人ディレクターあるある

進行管理に追われる



- ディレクションとは、作品の方向付けをすること
- また、チームの方向づけをすること

追われるのではない、道を示し、調整するのがディレクター

自分がエラいと勘違いする



- 自分だけでは作れないことを認識し、チームの力を借りる

デザイナーからみた
イケてないディレクター、イケてるディレク
ター

<イケてない>

- 指示がコロコロ変わる
- コミュニケーションが一方的
- ジャッジに時間がかかる

<イケてる>

- 作りたいもの、ゴールがしっかりと見えていて、軸がブレない
- 客観的にデザインを見れる
- 納得感のある指示ができる。理由付けができる
- ボキャブラリーが豊富
- 人の扱いがうまい

エンジニアからみた
イケてないディレクター、よいディレクター

<イケてない>

- 指示の意図が分からぬ
- 自分が何やっているかわかっていない。ゴールを見失っている
- 知ったかぶりな人

<イケてる>

- 指示に理由と理屈がある
- 軸がしっかりしていてゴールが見えている
- マイルストーンが共有できる
- ざっくりとした工数や、実装で大変なポイントが分かっている
- 代替案を持っている

まとめると、

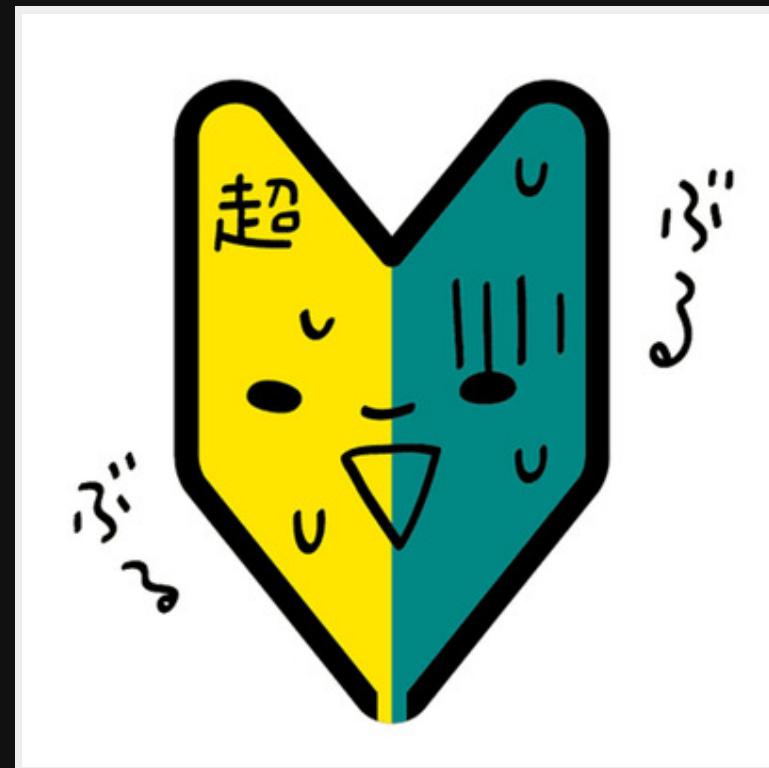
1. 成功イメージと、それに向かう責任感がある
2. 客観的にサービスを見れる
3. コミュニケーション力が高い

ディレクターとして、僕が心がけていること。

- 責任を持って決める人になる
- 誰よりも、先を見て、道を示す
- 作品が成功するように、嫌なことを率先してやる

ディレクター1年生はどうすればいい？

- 実績なし
- デザイン、プログラム知識無し
- みんな先輩・・・



できることは、

- サービスについて、とことんリサーチして、道筋を示す
- チームの皆が快適に仕事をできるよう、下っ端コミュニケーションをとる
- 弟子として、先輩からワザを盗む

よいディレクターとクリエイターの関係

お互いのリスペクト！

スキルを身につけるために、日々やっていること

- 誰よりもサービスに詳しく
- デザインパターンを得る
- プログラミングを楽しむ

Webの仕組みを学びながら、簡単なデザイン、コードを書いてみよう

以降、ゆっけさんへ！

2時間目

僕たちのWebサービス作成記

改めて自己紹介

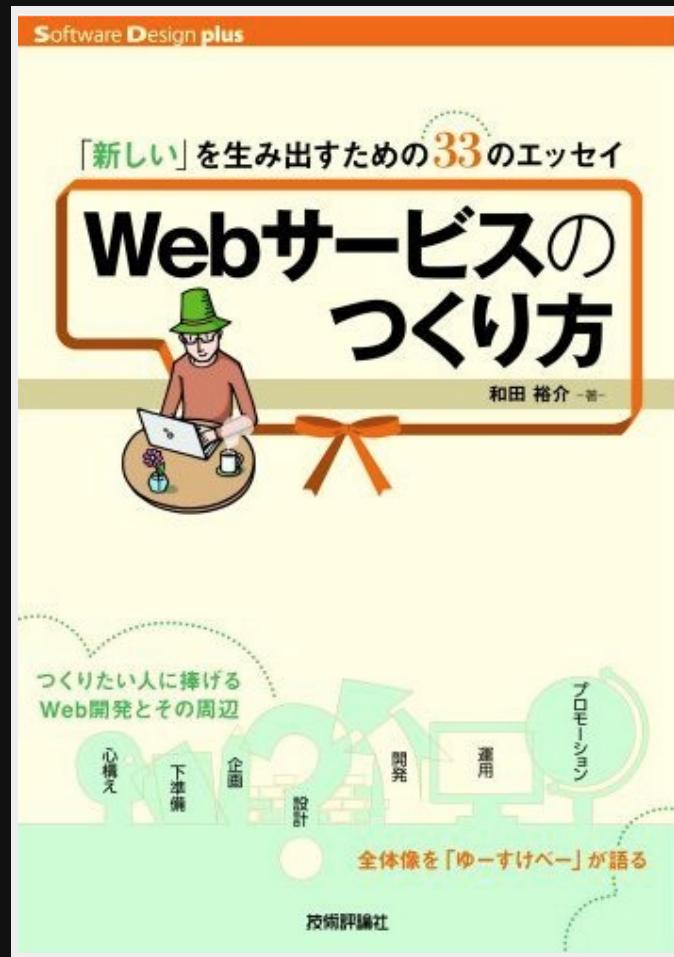
和田裕介

- 1981年12月23日生 神奈川育ち
- (株) ワディット代表取締役
- (株) 才モロキ取締役兼最高技術責任者
- 一般社団法人Japan Perl Association理事
- a.k.a. **yusukebe**

日々のお仕事

- ボケての開発・システム運用
- 執筆・イベント企画・コンサル

Webサービスのつくり方



この時間では...

ボケてをつくっているお話をします



ボケてとは？

- 国内最大級のお笑いWebサービス...らしい
- 3秒で笑えるがコセプト
- オモロキがコアとなって開発・運営

こういうの



HOT入れてじゃない、ほっと
てくれって言ったんだよ。

ボケ by らぶひげ お題 by 太鼓の素人
Photo by 64182011@N08
<http://www.flickr.com/photos/64182011@N08/>



ちょい古めの参考データ

- ほげほげページビュー
- 月間お題投稿数3万件
- 月間ボケ投稿数120万
- モバイルアプリは200インストール
- 合計ボケ投稿数1,100万ボケ

デバイス展開

- PC向けWeb
- スマホ向けWeb
- iPhoneアプリ
- Androidアプリ

iPhoneアプリ



歴史

- 2008年8月 - ベータ版リリース
- 2009年2月 - ボケて本第1弾発売
- 2009年5月 - ガンマ版リリース
- 2012年5月 - まとめサイトをキッカケにアクセス急増
- 2012年7月 - デルタ版リリース
- 2012年10月 - iPhoneアプリリリース
- 2012年12月 - Androidアプリリリース
- 2013年7月 - Yahoo!と連動開始
- 2013年1月 - コラボスタート
- 現在 - イプシロン版を開発中

オモロキ初期メンバー

- 鎌田武俊 - 団子さん
- 和田裕介 - ゆっけさん

いわゆる*co-founder*は2人

最初は2人で開発



役割分担

- 2人
 - アイデアを考える
- 団子さん
 - デザイン
 - 軽いディレクション
- ゆっけさん
 - システム開発

マネタイズのことは考えなかったけど...
2人で2ヶ月を費やしつくれた！

「2」という数字はキーワード？

2人組はいいかも！

- 今でも要所要所のコアは2人組
 - Web : 団子さん + ゆっけさん
 - スマホ向け : おっさむ + ゆっけさん
 - コラボの案件 : しんちゃん + おっさむ

意思決定が早い
2人とも賛成 or 妥協するか

大抵のスタートアップなサービス
は2ヶ月でつくれる

かも...

2つの「2」についてでした

話変わって...

ボケてがスケールした瞬間

2012年5月からのアクセス急増のタイミングで
おっさむが声をかけてくれた

*iPhone*アプリつくらない？

僕ら2人は「クリエーター」だった

つくることは出来るけど...

- マネタイズ
- プロモーション
- マーケティング

戦略的にスケールさせる

おっさむがジョイン

- iPhone/Androidアプリをつくる
- 大きくするためのマネタイズ
- プロモーションの施策
- マーケティング的な視点を入れる

バージョン「デルタ」の開発

2012年7月

大きなトラフィックに耐えれるようにコードを修正
モバイルアプリと連動するためのWeb APIを作成

広告の掲載

2012年8月

PCサイトに広告掲載を開始

モバイルアプリリリース

2012年10月 iPhoneアプリ

2012年12月 Androidアプリ

企業とのタイアップを開始

2013年1月



Yahoo! JAPANと連動

2013年7月



イプシロン版開発中

2013年11月

オモロキメンバー

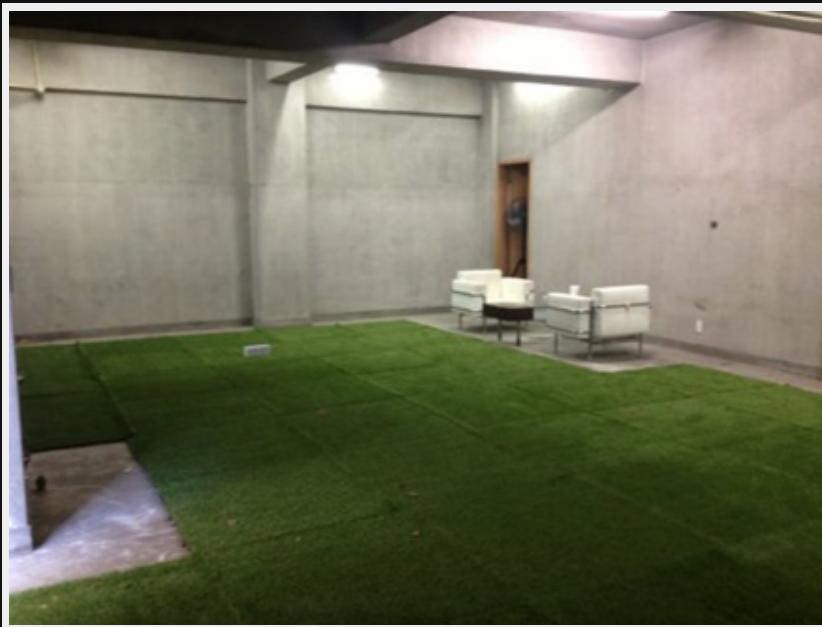
5人だけ



会社所在地は熱海だけど

- みな自分の会社を持っている
- それぞれの場所で作業
- 連絡はメール等オンラインで
- たまーに集まる嬉しさ

熱海ガレージ



「つくる人ではない」メンツのありがたさ

- おっさむ
 - マーケ、PR、広告、ディレクション
- しんちゃん
 - ツンデレ営業
- つよしさん
 - ユーティリティ法務

パートナー体制でスケールする

会社を大きくしない

モバイルアプリの場合



ここ最近ボケてから学んだこと

キーワード「2」で進行

- 2人組で意思決定
- 開発は2ヶ月を単位に

つくれない人の視点が重要

- 自社サービスといえどもクリエーターだけでは事業化が難しい

しんちゃんの言葉

自分はつくれないからつくれない人になる

まとめにかえて

お互いを尊重しえると楽しい

- 自分に出来ないことをそれぞれ持っているメンバー
- 素直に尊敬し、そして、褒める
- そしてメンバーから吸収する

3時間目

Web技術について

技術にまつわる注意

- 生半可は危険
- 言葉に踊らされない
- 技術を強要しない

「そこHTML5でつくって」

「？？」

(HTML5と言われても... その定義は広いし...)

興味を持つことが大事

- エンジニアとしては嬉しい
- エンジニアは教えるのが好き
- 技術からの発想があるかも
- ちょっとした事が出来るかも
- 現にこの資料は...

改めてWeb技術の基礎

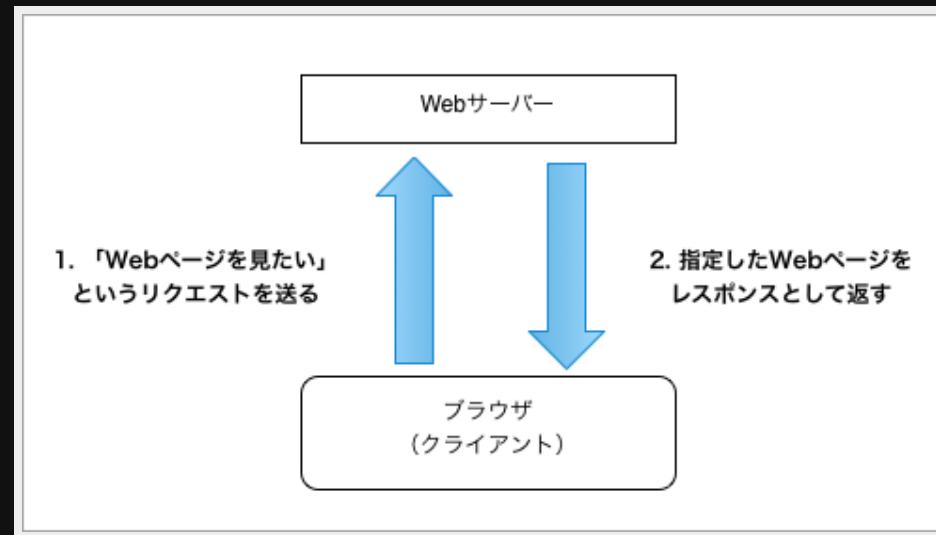
あたりまえに見ているWebページの裏側を知ろう

* 基本**Markezine**の連載を元にしてますがより噛みくだいて

サーバー・クライアントモデル

- サーバー
 - よく聞きますよね？
 - 例えば、Webページを配信する
- クライアント
 - Internet Explorer, Chrome, Firefox...
 - いわゆるブラウザもその一種
 - サーバーに対してリクエストをしてレスポンスを受け取る

リクエストとレスポンス



リクエスト「このWebページを見たい！」

レスポンス「Webページはこれだよ！」

Webにおける約束事

- サーバー・クライアントモデルを成り立たせるために
- 両者間のやりとりについて約束事がある
- URL、HTTP、HTML
- 世界中どこでもWebページが見れるために...

URL

"Uniform Resource Locator"

`http://example.com/about.html`

- どこのWebページが見たいか？を示すアドレス
- http - HTTPというプロトコルを使うことを宣言
- example.com - サーバーのホスト名
- /about.html - ホスト配下の具体的な場所

HTTP

Hypertext Transfer Protocol

- サーバーとクライアント間の通信のやりとりを規定
- RFC 2616という仕様にまとめられている
- GET - URLで指定されたリソースを取り出す
- POST - URLに対してクライアントがサーバーへデータを送信する

まずは

「そういう決まりごとがあるんだな」

程度の感覚で！

HTML

HyperText Markup Language

```
<html>
  <head>
    <title>Welcome Page</title>
  </head>
  <body>
    Hello, I am <a href="http://yusuke.be/">yusukebe</a>!
  </body>
</html>
```

- 皆さんご存知のコレです
- 他のページへのハイパーアリンク
- 画像等の埋め込み
- プレーンテキスト形式にタグを書く
- 要素、属性、値、内容
- ブラウザはHTMLテキストを解釈し描画している

Webサイトの性質

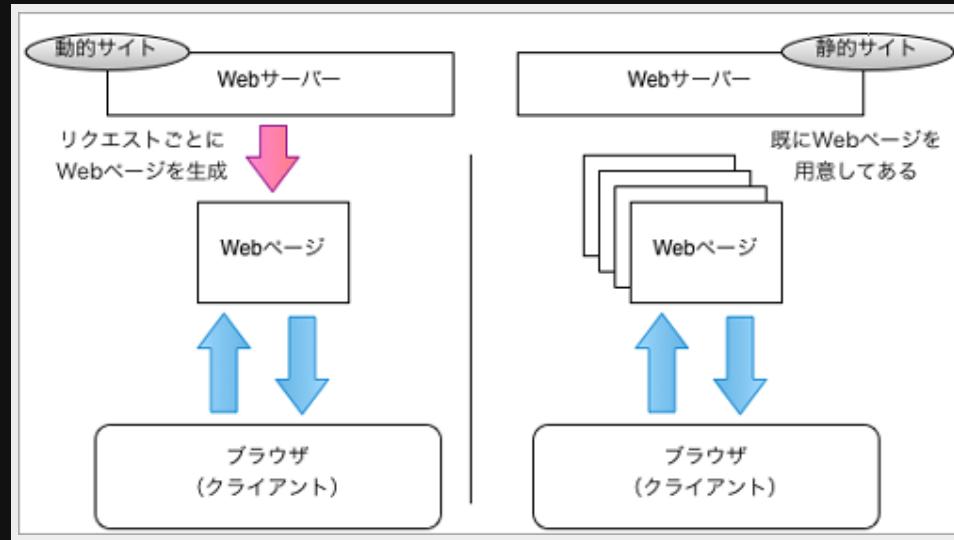
「あなたのWebは静的サイト？動的サイト？」

- 動的サイト
 - サーバーサイドの裏側で動的にページが生成される
- 静的サイト
 - HTMLや画像等を都度都度は変更せず、そのまま配信される

問題です

*WordPress、MovableType*で作られたサイト。
それぞれ動的？静的？どちら？

動的サイトと静的サイト



静的サイト

構成する要素

- HTML
- CSS
- JavaScript - ページに動的な動きを加える
- 画像、Flash、Webフォント…

CSS - HTMLを装飾する

CSSセレクタ

ヘッダー「h1」タグの「class」に属性「header」という値を記述

```
<h1 class="header">見出し</h1>
```

h1タグの内容の色を変更したい！とすると…

```
h1.header { color:red; }
```

どこを装飾するかを指示する「h1.header」がCSSセレクタ
どのように装飾するかはCSSプロパティで指定

JavaScript - HTMLを変化させる

クライアントサイドで動的な仕掛けを

- Webページの特定の部分を動かす
- フェードイン・フェードアウトなどもできる
- ボタンを押すと一番上までスクロールして戻る
- クライアントサイドだけでフォームの入力値検証
- スライドショーなど特殊な見た目の実装

ここをクリックしてデモ

静的サイトとしてWebページ

- 最低限はHTML
- 見た目を変えたければCSS
- 動きを与えたければJavaScript
- サーバーで配信をすればWebページとして機能

動的サイトについて

何気なく使っているサービス

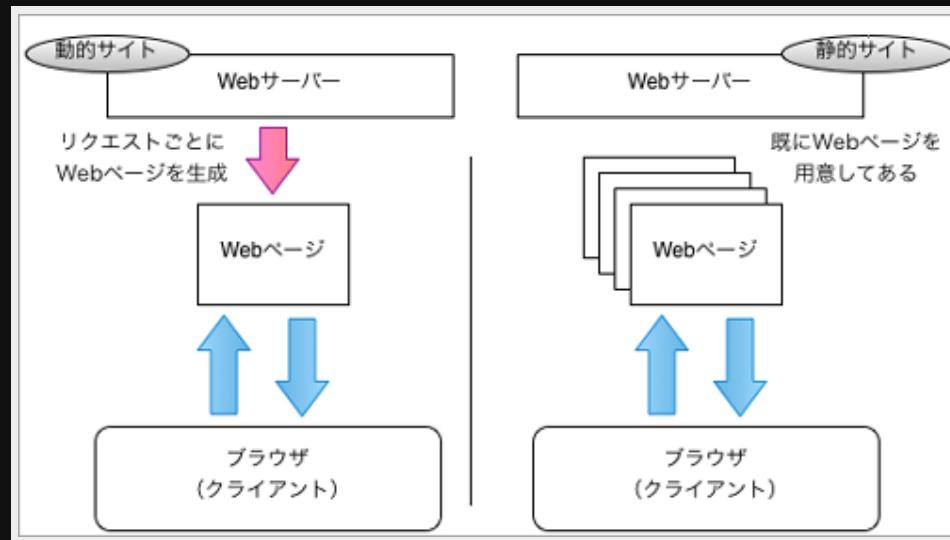
- Facebook
- Twitter
- クックパッド
- ソーシャルゲーム
- ポケて

全て動的サイトの性質を持つ

Webアプリとして

リクエストごとにサーバ側のプログラムがHTMLなどのページ要素を動的に生成して、レスポンスとして返す

もう一度確認



Webアプリにおける情報の操作

CRUD

- CREATE - 作成
- READ - 参照
- UPDATE - 更新
- DELETE - 削除

サーバサイドのプログラミング

どんな言語でも出来るが最近多いもの

- PHP
- Ruby
- Perl
- Python

例. 問い合わせフォーム

入力値の妥当性をチェックする

- 郵便番号が7桁の数字で入力されているか
- 必須の項目が全て埋まっているか
- 連絡先のメールアドレスは正しいメールアドレスの形式であるか

バリデーション

最終的な実装すべき処理

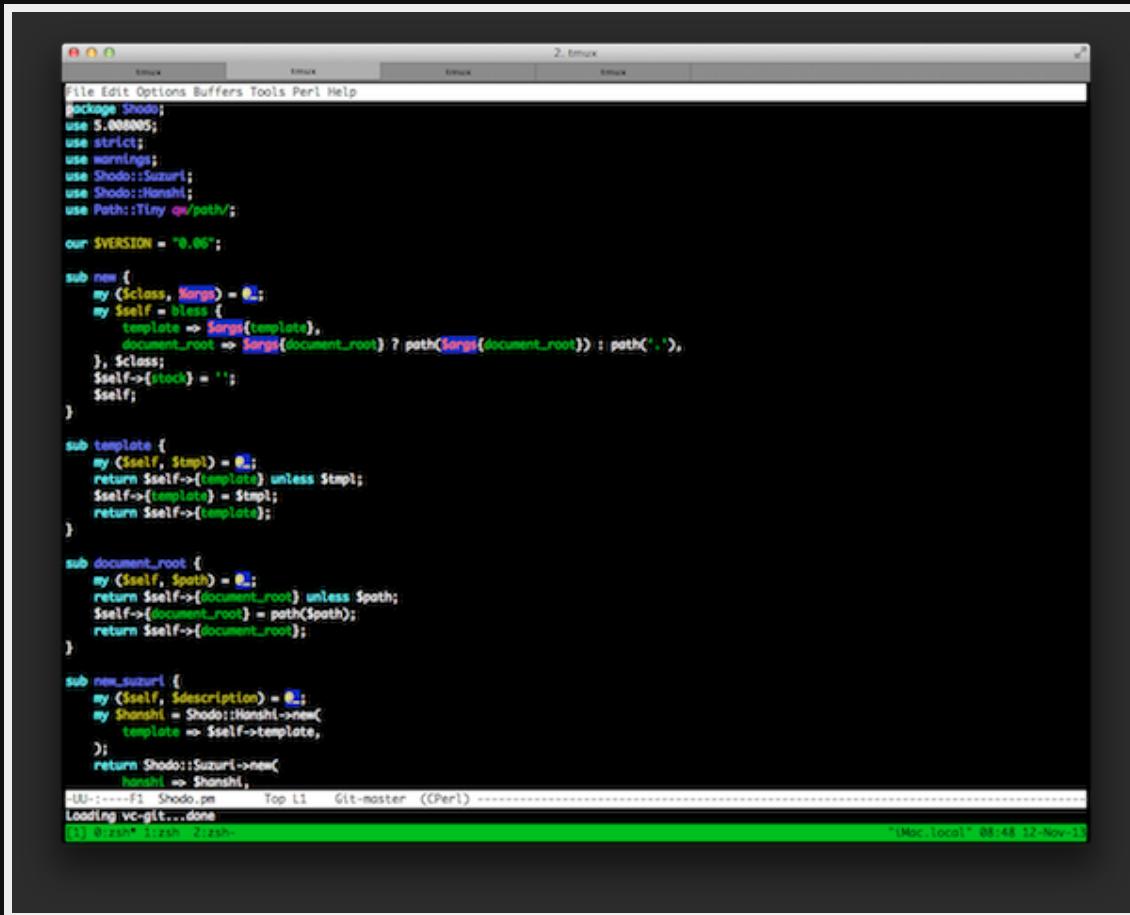
1. バリデーションつまりフォームの入力値検証
2. 適切なエラー表示
3. メール送信
4. データベースへの情報登録、及び閲覧

サイトの望むべき性質が静的か動的か？

エンジニアと相談しつつベストを探りましょう！

Webエンジニアの生態系

毎日黒い画面眺めている...



The screenshot shows a terminal window with four tabs labeled "Emacs", "Emacs", "Emacs", and "Emacs". The active tab displays Perl code for a module named "Shodo". The code includes imports for "Shodo", "5.000005", "strict", "warnings", "Shodo::Suzuri", "Shodo::Hoshi", and "Path::Tiny". It defines a class "Shodo" with methods "new", "template", "document_root", "new_suzuri", and "new_hoshi". The "new" method initializes the object with a template and document root. The "template" method returns the template or creates a new one if it's not defined. The "document_root" method returns the document root or creates a new path object. The "new_suzuri" and "new_hoshi" methods create instances of "Shodo::Suzuri" and "Shodo::Hoshi" respectively, passing the template and hoshi objects. The terminal window also shows the command "Loading vc-git...done" and the date "Sat Nov 13".

```
File Edit Options Buffers Tools Perl Help
package Shodo;
use 5.000005;
use strict;
use warnings;
use Shodo::Suzuri;
use Shodo::Hoshi;
use Path::Tiny;
use File::Path;

our $VERSION = "0.06";

sub new {
    my ($class, %args) = @_;
    my $self = bless {
        template => $args{template},
        document_root => $args{document_root} ? path($args{document_root}) : path('.'),
    }, $class;
    $self->{stack} = [];
    $self;
}

sub template {
    my ($self, $tmpl) = @_;
    return $self->{template} unless $tmpl;
    $self->{template} = $tmpl;
    return $self->{template};
}

sub document_root {
    my ($self, $path) = @_;
    return $self->{document_root} unless $path;
    $self->{document_root} = path($path);
    return $self->{document_root};
}

sub new_suzuri {
    my ($self, $description) = @_;
    my $shoshi = Shodo::Hoshi->new(
        template => $self->template,
    );
    return Shodo::Suzuri->new(
        hoshi => $shoshi,
    );
}

sub new_hoshi {
    my ($self, $description) = @_;
    my $shoshi = Shodo::Hoshi->new(
        template => $self->template,
    );
    return Shodo::Hoshi->new(
        description => $description,
        shoshi => $shoshi,
    );
}

1;

```

動くことに快感を覚える

1. プログラムを書く
2. 動かす
3. 出来たー！
4. 繰り返し…

勉強会なるモノがやたら多い



貢献したいと思ってる

- 勉強会での発表
- イベント主催
- 汎用ライブラリの作成と公開
- Blog記事でのノウハウ発信

エンジニアにとって怖いけど好きな言葉

Shut The Fuck Up Write Some Code!

Webエンジニアは面白い人種

まとめ

- エンジニアに対する心構え
- Webの基礎技術
- 動的サイトと静的サイト
- その生態系

4時間目

戦略思考をもったディレクターになるための5
箇条

目次

1. プロダクトの向かう先を示す
2. 「つくる」ことを理解する
3. 「決める」覚悟を持つ
4. 「広げる」のもディレクターの仕事
5. チームに感謝しよう

1. プロダクトの向かう先を示す

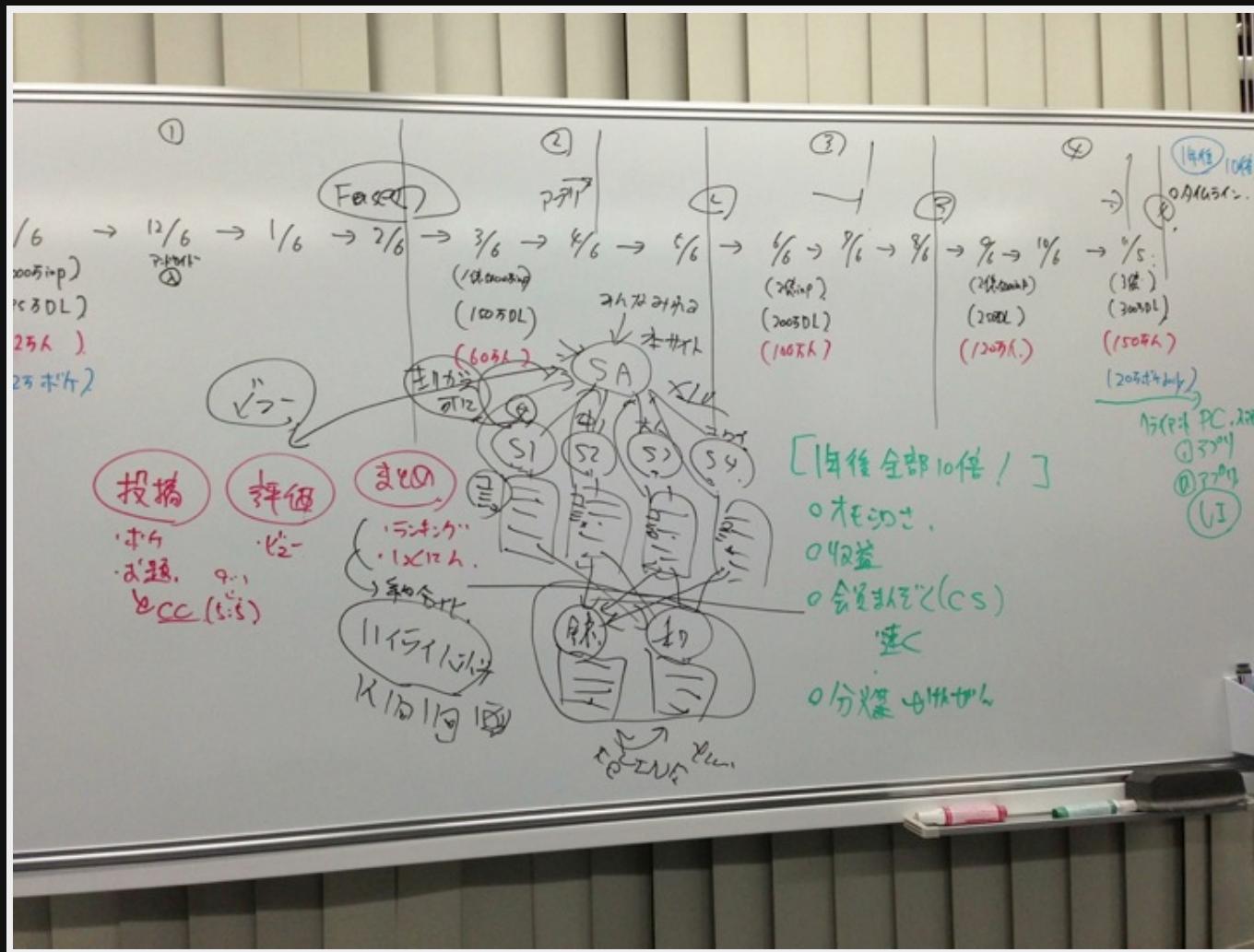
まず、チームでワクワクする目標を共有する



サービスが進化するためのマイルストーンを 設計する

- ローンチ後
- 3ヶ月後
- 1年後
- 3年後
- · · ·

ボケての場合



10倍にスケールさせるという目標を立て、 1年間 = 12ヶ月に具体的に落としこむ

その際に、サービスの機能増強や、マーケティング、業務提携などをあらゆる要素を織り込む

当然、最初に描いた計画通りには進まない

しかし、自分達でひいたマイルストーンとの差を
考えることで、対策を講じることができる

2. 「つくる」ことを理解する

ディレクターが、実際に手を動かすことは少ない。その分、作品の「世界」の構築に集中する

デザインパターンを持とう

- 具体的にイメージを伝えられるか？
- 他サービスをどれだけ知っているか？
- なぜ、このデザインがいいと考えるのか？

すべてを、デザイナーと共有し、「言葉で」説明しながら作業する

プログラムの特性を理解しよう

- この機能は、実装するのにどのくらいかかるのか？
- 本当に必要なのか？
- 別のやり方はないのか？

すべてを、エンジニアと話しながら、優先順位をつけていく

一番の仕事は、プロダクトを整えること。



ユーザーとクリエイターのインターフェイスとなり、世界観、ユーザビリティを調整する。

この部分には、経験に基づくセンスが必要になる

3. 「決める」覚悟を持つ

「決める」ことには責任がつきまとう。

- 失敗しても、彼or彼女と行ったジャッジなら納得がいく
- 一緒に軌道修正すればいい と思ってもらえるかが大事。

リーダー性が求められるのも、
ディレクターというポジションの宿命

※チームによります

4. 「広げる」のもディレクターの仕事

つくったサービスが使われるために

つくって終了

どう使われるかを考えぬく

WEBサービスは必ず誰かが使うために存在する

サービスが自然に拡がる仕組みを創るのが
ディレクターの仕事

- ・マーケッターだけでは、根本的にサービスを広げることはできない。

プロダクトにマーケティングを内包する

例：DropBox 友達を呼ぶと、容量が増える。



例：bokete
twitterやfacebook、LINEにカンタンにシェアして、友達を笑わせられる。

- 広告費を使わずに、ユーザーが自然に友達に広める仕組み。

どこまでサービスが完成してからのことを考えられるか。

パートナーシップを構築しよう

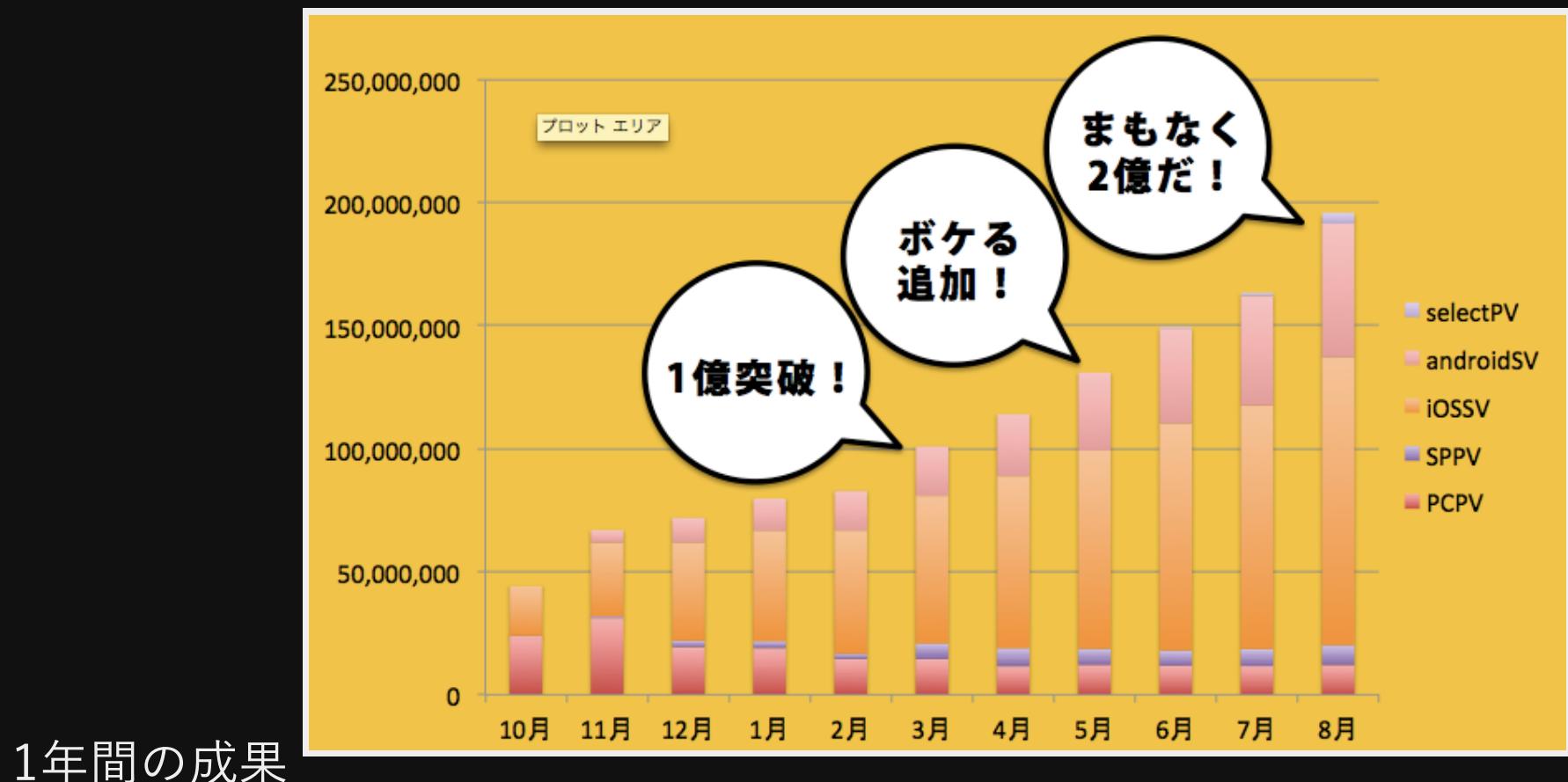
ディレクターは、外部との提携交渉などを任せられることが多い。

自分のサービスと、相手のサービスがお互いにメリットを出せる提携を探るのも仕事になりそうです。

5. チームに感謝しよう

繰り返しになりますが、ディレクター一人でモノはつくれない。
だから、チームに感謝を伝えよう。

そのために、チームで喜びをわかつあおう。



- ・自分たちが歩んだ道を振り返りながら、各自が活躍したところ、うまく行かなかったところを振り返る。
- ・飲み会を設定し、お互いをたたえ合う

Special Thanks

Yera Jin

Keiko Sasaki

Shohei Yamamoto

5時間目

ざっくばらんに懇親と質問タイム！