

# モバゲータウンをこうして作った

DeNA 川崎 修平

2008/9/5

ITpro Challenge! 2008

# 自己紹介

- 1975 0歳 千葉県に生まれる
- 1983 小学2年 ファミコン入手(1983/7 発売)
- 1985 小学4年 ファミリーベーシック入手(1984/6 発売)
- 1988 中学1年 X68000 入手(1987/3 発売)
- 1990 中学3年 Machintosh SE/30 入手
- 1993 高校3年 国産パソコンに失望し、パソコン離れ
- 1994 浪人 ゲーセン:バーチャ1・2にハマる
- 1995 学部1年 DOS/V マシン購入。大学で unix, mosaic に出会う
- 1997 学部3年 linux を知る。プログラムバイト開始。
- 2000 修士2年 ネットオークションに出会う
- 2001 博士1年 オークション統計ページ(仮)開始(5月)
- 2002 博士2年 DeNA にバイトとして入社
- 2004 社員1年 モバオク(3月)、ポケロト(?月)、ポケットアフィリエイト(7月)
- 2006 社員3年 モバゲータウン(2月)

# パソコン少年時代(小学～中学)

- ファミコンとゲームパソコンが大好き
- パソコンオタクでマイナー志向
- 愛読書はベーマガとOh!X
- 将来の夢はゲームを創る人
- アセンブラでハードの隅々まで触れるハードが楽しくて仕方なかった
- Mac も買ったが Pascal とかさっぱりわからなかった

高校時代は、X68000 の後継となる国産ゲームパソコンを  
心待ちにしていたが、一向に出る気配がないのでパソコン離れ。

- ビジネスパソコンなんて・・・
- ハードを直接いじれないパソコンなんて・・・
- x86 なんて・・・

# 大学時代(1/2)

- 大学に入って unix と web の世界を知り、「これは楽しい」と感じてプログラミング復帰。
- マニアックな国産パソコンを待ったせいで時代に取り残されたのを教訓にメジャー志向に転向。
- フリープログラマとして(バイト)
  - 複数社から仕事を受けていて、依頼を断れずに仕事が重なって死にそうに。
  - 開発スピードと危機回避の本能は身についた。

# 大学時代(2/2)

## ● 個人サイト運営

- オークション統計ページ(仮)  
(現:aucfan.com)
- 自分の作ったサービスで喜んでもらえるのが嬉しい。
- 企業サイトと違って、思いついた機能をすぐに入れていけるのが楽しい。
- ひとりで貧乏仕様なサービス運営をする能力が身についた。
  - 自宅(?)サーバ4台で運営

## ● 当時の悩み

- 掲示板の管理とか大変
- 営利目的でないのでサービスが拡大しにくい
- 自分の理想のオークションサイト作りたいけど個人じゃ・・・

# その後

- ビッダーズ社員からメールで呼び出された。
  - ファミレスで話をして、そのままバイトとして手伝うことに。
- 最初は分析とかパフォーマンスチューニングとかの仕事。
- 入社を決め、2004年にとりあえず年内にモバイルサービスを3つ立ち上げようという大雑把な目標で動き出す。  
↑「10個つくって1つあたればいい」くらいのノリで。
- モバオクやモバゲーなど作る

# モバオク

## (ケータイ専用オークション)

- 開発期間2ヶ月くらい、2004年3月サービス開始

- はじまり

守「携帯用のオークションサービスをやろうと思うんだけど、予算も人もあまり割けないからひとりで作れる？」

川「ビiddersと切り離してゼロから作っていいんですか？」

守「それでいいよ。」

川「ならやります。」(やった！)

守「当てる前提だからデイリー1000万PVくらいでも動くようにして。」

川「わかりました。」(そんなに来るか？)

# モバオク～方針

- 自分の夢に描いたオークションサイトを創る。
- PC用オークションのBIDDERSとは敢えて切り離し、モバイル専用として使い易いサービスを作る。
- パケット定額制に合わせたサービス設計。
- サービスが成功してもヤフオクに流れないような文化作り。
- 当たるかわからないので、できるだけ安上がりに。



# モバオク～開発内容

- システム基本構成の評価・選定
- ケータイブラウザのクセの調査
- 絵文字の使い方の調査
- フレームワークから作成
  - テンプレートエンジン、ディスパッチャ、絵文字変換、daemon 処理など
- SHM を使った高速リスティングの実装
- 検索エンジンの作成

# モバオク～困ったこと

- サービス開始当初、客が全然こなかった
- 流行りはじめたら、トラフィック増加にサーバ増設が追いつかなかった
- 携帯特有のトラフィックでネットワーク機器の限界が
  - CDN 導入により回避
- メール送信能力が足りない
  - qmail ⇒ postfix ⇒ MobaMail
- 画像サーバが不安定に。とか。
  - 即時の実装変更

# モバオク～驚いたこと

- ケータイユーザはものすごくページを見る
- PC とコミュニケーション方法が全然違う
  - 即レスが普通。短い文面でのやりとりの回数が多い。
- ユーザと運営者の距離が近く、反応もすごく早い
- 障害や不具合に寛大

次は開発スタイルについて・・・

# 新規サービス開発の進め方

- 基本的に自宅開発で深夜作業  
(作業中は基本的に連絡もとらない)
- 自宅サーバ上で開発中のサービスを公開してるので、他のメンバーも進捗はいつでも確認できる。
- 基本的に現物が仕様。ドキュメントはない。ソース読めば分かるように。
- たまに会社に行って打合せ。
- 細部の詰めはいきつけの飲み屋で、みんなで携帯を開いて相談。
- ほぼできたら会社の環境に移設する。

# 開発スタイル

- やる気が出るまで横になる
  - やる気がでるまで漫画とか読んでる
  - 疲れると頭が働かないので、設計は寝ながら
- 設計のイメージができたなら、力尽きるまで作って調整して寝る。
  - 開発中は食事はあまりとらず、午後の紅茶ミルクティー。
- 全体ができたなら使い倒して気に食わないところを修正。
  - 機能がイマイチだったらぱっさり捨てる。
- 内容が大体固まったらコード整理。
- サービスイン後は、使われ方をみながら適宜修正。

# 守備範囲

- 開発作業(設計・コーディング・デザイン・ワーディング...)
- 障害対応(監視メール ⇒ PC常時携帯で対応)
- 負荷対策
- サーバ発注・増設・設定
- データ分析

負荷対策しながら新規開発もしないといけないので、  
楽に運用できる工夫が必要

# 運用の工夫

- 各所にリミッターをつけておく
- 意図的に弱い部分を作っておく
- シンプルな設計に
- 構成要素は少なめに

以降、モバオク以外のサービスについて...

# ポケロト(ケータイ専用懸賞サービス)

- 開発期間1週間弱、サービス開始時期不明
- はじまり
  - 守「とりあえず懸賞サイトでも作ろうか？」
  - 川「じゃ、とりあえず作ってみます。」
- 基本方針
  - とりあえず懸賞サイトがどんな客層・規模感なのか試す
  - 工数をかけずに無難につくる



# ポケットアフィリエイト

(モバイルアフィリエイトネットワーク)

- 開発期間2週間、2004年7月サービス開始
- はじまり(昼飯に行く途中の道端での会話)  
守「アフィリエイトやろうかと思うんだけど。」  
川「うちがですか？いいんですか？」  
(中略)  
川「じゃ、それっぽく作ってきます」

# ポケアフィ〜基本方針

- 自分がメインで使えるアフィリエイトサービスを作る。
- 当時のアフィリエイトネットワークは稼げる広告が少なかったなので、モバオクという、いい案件を自社で持っている強みを活かす
- 支払いと誤魔化しに関して不信感のないサービスに
- 軽い管理画面
- 絶妙な緩さ
- 携帯だけからでも使える

# ポケアフィ～開発内容

- 特に凝ったことはしてない。
- フレームワークをPC対応に修正。
- 安定稼動する設計を職人的に。
- サーバはバックアップを入れて2台。  
(先に作ったポケロトとサーバ同居。)

※サービス開始後、1年近くシステムに手をかけていない。

ポケットアフィリエイトがモバゲーを成立させるキーに

# モバゲータウン

(モバイル専用ゲーム&SNS)

- 開発期間3ヶ月、2006年2月サービス開始

- はじまり

守「ハンゲームの無料ゲームとアバターのモデルがよさそうで、モバイルで展開したらいけそうなんだけど、ちょっと考えておいて。」

川「わかりました。とりあえず、あんまり見たことないんで調べときます。」

～ 後日打合せ

畑「こんな機能が必要かなと思うんだけど」(A4 1枚のリスト)

川「わかりました。とりあえずそんな感じで。あとはざっと作って見ながら考えましょう。」

# モバゲー～基本方針

- 自分でも使えるコミュニティーサービス
- 適当に理由をつけてゲーム開発もやる
- アバターの価値を最大化するサービス設計
- バーチャルかつオープンなコミュニティー
- ゲームを含め、可能な限りアバターを露出
- みんなが主役(便利さより絡みの楽しさを重視)
- モバゲー内の人格を大事にしてもらう
- 最初の文化作りを大事にする

# モバゲー～開発内容

- mixi,gree,ハンゲームの研究
- 基本システムはモバオクのものをベースに改良
- 高速なアバター描画ライブラリ
- アプリゲーム用フレームワークとAPI
- アプリゲーム開発
- 地図描画ライブラリなど(後日)

# モバゲー～困ったこと

- SNSを使わないので勘所がわからなかった。
- サービス成長期は、毎週のようにDB分割をして嫌になってくる。
- サーバ不足、ラック不足。
- 多人数開発前提じゃないつくりなので、現状結構大変。

意外とエピソードがないです・・・

# サービス作りの心がけ

- 自分が作りたいものを作る
- 自分が作りたいと思うサービスと、ユーザが使いたいと思うサービスが一致するよう訓練
- こんなサービスが目の前に現れたらワクワクするだろうという高揚感
- これができたら自分はすごいと思える問題設定
- サービス設計の大部分を任される責任
- 状況に応じてフットワークよくサービスをいじれる環境
- 定期的発生する修羅場



# 技術について

- 自分はハッカーとかじゃない。(なりたいけれど)
- 自分のものづくりのために技術を習得。
- 技術の習得と活用はバランスよく。
- 新技術は広く浅く動向だけは追い、必要に応じて習得。
- 何か思いついたら、すぐに設計が頭に浮かんで実装できるようにしておく。
- 枯れた技術で作れる部分は、枯れた技術で作る。
- モジュール・ライブラリはできるだけ自分で書く
  - 評価時間の削減
  - 問題発生時の対応が迅速にできる
  - カスタマイズが容易

# ひとり開発だと楽な点

- イメージ通りのものが作れる。
- 大部分の時間を本質的なことに割ける。
- 「こうしたい」と思った瞬間に修正できる。
- コミュニケーション能力が低くても、完成したサービスが語ってくれる。
- イマイチかもしれない機能でも自己責任でとりあえず実装してしまえる。
  - イマイチだったらなかったことにする。
- システム全体に統一感ができる。

# ひとり開発だと辛い点

- ひとりよがりになる危険
  - 定期的にメンバーと相談するのが大事
- 開発期間が2ヶ月くらいを越えると寂しくなり、飽きてくる。
- 失敗したときの精神的ダメージがでかい。
- 運用のシェアができていない部分が多く、24時間の障害対応が必要なので睡眠不良になる。
- 人依存が強すぎる。

# こんな環境で作れるのが理想

- 作家と編集者のような関係(?)
  - 作る側:実績・信頼・愛情・責任感
  - 任す側:任せられるかどうかを判断できる能力  
適切なインプットと判断力
- 自分は集客力のあるサービスを作ることに専念して、苦手なところは専門家に任せられる。
  - ビジネスチャンスの発見
  - 成長後のトラフィックの活用やマネタイズ
  - カスタマーサポート業務
  - などなど

以上。