

# Web2.0::これからのIT事業の姿

アマゾン ジャパン株式会社  
ウェブサービス日本担当  
Emerson Mills

emills@amazon.com  
<http://www.amazon.com/aws>

2007年4月

Amazon, Amazon.com, Amazon.co.jp, Amazon.co.uk,  
Mechanical Turk ならびにこの資料上に表示されているその他の商標は、  
Amazon.com, Inc. またはその子会社の商標であり、米国または日本、あるいは  
その他の国において登録されている場合があります。  
©2006 Amazon.com, Inc. or its affiliates

# Web2.0

## 技術でもある考え方でもある

- ユーザに主導権を戻す
- 出版という概念を脱却
- コミュニティとの共存

## Web2.0は新しい発想？

- 技術よりユーザに対する概念
- 考えとしては新しい
- 最新技術により認知度が高まった

# Cluetrain Manifesto

- 1999年にサイトとして公開
  - <http://www.cluetrain.com/>
- 2001年に出版、ビジネス雑誌に取り上げられた
- それ以降はどうしたんでしょうか？

# Web2.0の基礎概念

- コミュニティの重要性
- 「市場は対話である」
  - (<http://www2.gol.com/users/jheine/cluetrainj.html>)
- ユーザによるコンテンツ

## 主導権はユーザがにぎる

- ユーザの動作によるコンテンツ作成および制御
- 事業者と開発者が枠を提供する

## Web2.0の技術側面

- クライアント端末の性能を利用
- インタプリタ型言語
- 標準技術利用による互換性

# デスクトップアプリケーション

アプリケーション

ライブラリやAPI

オペレーティングシステム

# ネットワークアプリケーションでは

アプリケーション

ライブラリやAPI

ネットワーク

オペレーティングシステム

# ウェブはプラットフォームである

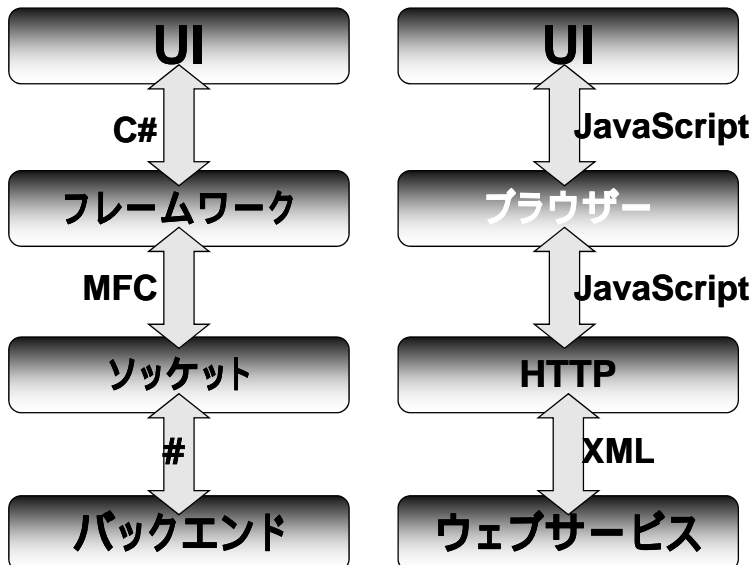
アプリケーション

HTTP

ウェブサービスやAPI

インターネット

## AJAXの基本構造



## ユーザによるコンテンツ作成

- AJAXを利用し、ブラウザにクライアントの役割を持たせる
- ユーザからのコンテンツをすぐに反映
- サイト及びアプリケーションはユーザにより更新

## ウェブサービスとは？

- ウェブを通じてアクセスする
- ウェブの標準技術を利用する
  - HTTP
  - XML
  - SOAP
  - REST
- オープンプロトコル
- 企業も含めて情報APIが多発

# Web2.0と事業の新体制

## WEB2.0の新ビジネスモデル

- ニッチ市場も開拓できる
- 専門分野に集中する
- ウェブサービスによる再利用
  - 利用する
  - 利用される



## パッケージ販売のコスト解析

- サービス及び製品の開発
- 印刷物やメディアの制作
- 流通
- 販売店舗または商社契約
- サポート

## ウェブサービスのコスト解析

- サービス及び製品の開発
- サポート

## ウェブサービスで業務改善

- インテグレーションの改善
- 間接業務の削減
- セルフサービス化
  - コミュニティ・サポート

## ロングテールとニッチ市場

- パッケージ販売に伴うコストがかからない
  - 損益分点が下がる
- ニッチ市場も開拓可能

## 「差別化」の本当のところ

- 競争で力を消耗してしまう
- お客様を囲むために力を消耗してしまう

## 競争ではなく協力

- 専門領域に集中する
- 取り組む際の簡易性
  - 自然に利用される
  - 競合製品からのユーザ移行

# Web2.0とスケーリング

## スケーリングとは

- 負荷に応じて処理能力、ストレージ容量、ネットワークを増加する

# 予測方法



## サーバーの増加プロセス

- 場所の確保
- サーバーの契約または購入
- サーバーの納入及び設置
- サーバー個別のインストール
- ネットワーク設定
- サービス・イン

# 計算表でみると

	1月	2月	3月	4月	5月	6月	7月	8月
平均ヒット数(日)	2000000	2200000	2750000	3437500	4125000	4950000	5940000	7128000
伸び率	10%	25%	25%	20%	20%	20%	20%	20%
サーバーの最大TPS	8	8	8	8	8	8	8	8
最低必要サーバー台数	3	4	4	4	5	6	8	11
スパイク対策台数	1	2	2	2	2	3	3	4
負荷分散制御	3	3	3	3	4	4	4	5
合計題台数	7	9	9	10	12	15	16	20
最大TPS	32	48	48	56	64	88	96	120
契約台数	10	12	15	16	20	22	24	25

- サービス・インの3ヶ月前から契約
- 最大TPSを超えるとサービス障害
- 3ヶ月先の予測が非常に慎重

# 成功に基づく失敗

- Slashdot Effect
- 予想以上に注目が集まる、または利用者の急増
- サイトやサービスに悪影響を与えてしまう

# スパイク現象

- 急激に負荷が跳ね上がる事
- 主要因 – 利用者予測以上に増える
- ほかの原因もある
  - DBのクエリーの変更
  - 新しいサービス・イン

# 成功を逃がさないためには

- キャパシティーと利用の予測
  - ネット事業の最重要ポイント
  - インフラ計画の原点

# アジャイル事業体制

- アジャイル開発手法から
- 予測は参考になるが現状に適応すべき

# 先週の実績は当てにならない

- すべての計画の基は推測
- ある時点を境に誤差が急増



# インフラはリスクである

- 従来はインフラを強化したら
  - 利用されなくてもコストが発生
  - 契約期間終了まで運用する必要
  - 購入した場合
  - 補修、入れ替え

# 柔軟性のあるインフラ

- 従来はScale Up と Scale Out
- これからはScale With と Scale Down
- インフラ型ウェブサービスで実現
  - Amazon Elastic Compute Cloud
    - 仮想サーバー
  - Amazon Simple Storage Service
    - ストレージとコンテンツ配信

## インフラ型ウェブサービス

- 提供する企業が出始めている
- ユーティリティコンピューティングのAPI
- 低コスト
- 優れたパフォーマンス

## ウェブスケール

- ユーザが増えてもパフォーマンスは影響されない
- 必要なだけ使うことが可能
- 初期及び追加投資の回避

## Scale With

- 負荷が増えたときにおうじてインフラを増加
  - 予測から測定
- 今必要なだけインフラ用意して使い切る

## Scale Down

- 使用されていないインフラの最小限に抑える
- 利用率の低い時間帯にインフラ縮小

# Web2.0と技術者

## 技術者とは

- 専門知識
  - ツール
  - プラットフォーム
  - 作業手法
- 経験

## 従来

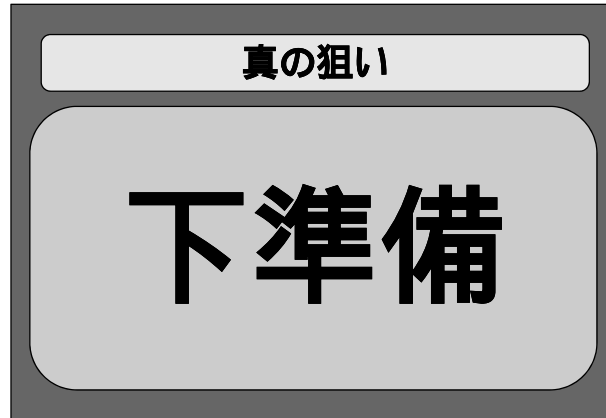
- ツールはCやJava
- プラットフォームはOS
- 作業手法はオブジェクト指向やデザインパターン

## これからは

- ツールはインタプリタ型言語とウェブサービス
- プラットフォームはインターネットとブラウザー
- 作業手法は分散処理法
- ただし従来方法と共存

# 80/20の法則

ある現象にて結果の8割  
が2割の要因から発生する



## 8割の作業は

- サーバー仕入れ
- サーバー運用
- ファイルシステム
- 分散キュー
- データーセンター

## 残りの2割

- サービスと製品の開発
- アイデアを実現

## 再利用

- オブジェクト指向から
  - Don't repeat yourself!
    - Or anyone else!
- スクリプトまたはインタプリタ系言語
- 低層開発はしない
  - メモリ制御
  - 基礎データタイプの開発

# ウェブサービスの有効利用

- データ
  - E Commerce Service
- 機能
  - Simple Queue Service
- インフラ
  - Elastic Compute Cloud
  - Simple Storage Service

# アプリケーション新概念

- 多様なサービスやデータを取りこむ
- 組み合わせからも付加価値を生む
- REMIX概念
  - オープンソース情報
    - AlexaやGoogle Trendsなど



# REMIX事例

- <http://connexus.jp/>
- <http://pulpsite.net/zontube/jp/>

# 技術者に必要な資質

- 数多くの技術の理解
- 多種のリソースの結合
- 分散設計の基礎知識
- 再利用及びREMIXを活かす

# 専門知識の商品化

- 優れているものを製品化する
- 物がよければ単独でもよい
- サービス開拓概念