



SaaSって何？

- またバズワード？
- この単語って知ってる？？

- ASPみたいなもんでしょ？
- まあ正解！
- でも流行りものなので



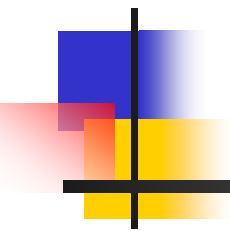
SaaSって良いの？

- しょせん道具。馬鹿と鉄は使いよう



使ってみたよ

- なかなか良かった！
- ということで発表



SaaS(Software as a Service)
- サービスとしてのソフトウェア -
~ MCEA における Salesforce 導入を例として ~

2006年10月度
MCEA技術者交流会資料
by 佐々木 健



自己紹介

- 本職はネットワークエンジニア
- 3年近く前に、ふと組合に来てしまった。
- 「エンジニアの幸せのために働いてみないか」などという口車に乗せられた
- なぜかそのまま事務局で働くことに。。。
- 良くわからないうちにシステムの面倒を見ることになり、、、、
- 以下、涙、涙、涙

- まあそんなことはどうでも良くて本題にGo!



自己紹介(余談)

- 5月にもネットワークに関する発表をしたので良かったらそっちも見てね！
- ブログも書いてるよ。
- 毎日2000人程度は見に来てて500人ぐらいの常連客がいるらしい。
- 良かったた見てね。



今日のアジェンダ

- システムの話、なぜ今SaaS？
- MCEAにおけるセールスフォース導入の背景
- デモ！デモ！デモ！

- 時間が余ったら適当に～
 - LL RING とか JANOG の話、できるかな？



配布資料

- ComputerWorldの記事。SaaS大研究
 - MCEAが取材されてます
- 日経コンピュータ記事。SaaSの衝撃
- 沢山あるので欲しいだけ持ってってね



なぜ今SaaS?

- なんだか流行ってるよねえ ~
 - Salesforceだけでなく、SAPもOracleも参入
- 流行ってるには理由がある？



そもそもシステムって何のために使うの？

- **ビジネスにおいては以下の2点のため！**
 - **業務効率化によるコスト削減**
 - **商品に付加価値を付ける**



システムはお金がかかる

ライフサイクル毎にお金がかかる

- 企画
- 設計、構築
- 運用
- 保守
- 廃棄



システムはお金がかかる(企画、設計、構築)

- 企画
 - ここをきちんとしないと場当りのにお金が出血してしまう
 - 経営レベルとの連携が必須
- 設計、構築
 - 何もしなければお金がかからない



システムはお金がかかる(運用、保守、廃棄)

- 運用
 - システムとしての価値はこのフェーズではじめて生まれる
 - 何もしなくてもお金がかかる、基本的には変動しない
 - 稼働しているシステムすべてについて必ず発生する
- 保守
 - リスクによってかける金額を考える必要がある
 - システムの品質により金額が変動する
- 廃棄
 - 見落とされがち
 - 競争力維持の肝
 - コスト削減の肝



コストを下げるにはどうしたら良いか？

- そもそもシステムを使わない！
 - それができれば苦勞しない
 - そもそもシステムってコストを下げるものだし、、、
 - 費用対効果を考えて、不要ならシステムを使わない
 - 業務を整理してシステムの数減らす
- ライフサイクル毎の費用を削減する
 - それぞれのコストを把握するところから始める
 - 費用対効果も考える
- 指針はいろいろとある



コスト削減への道、その1 パッケージ (メリット)

- システムをフルスクラッチから作ると時間も費用もかかる。
- みんなやりたいことは同じ。
- みんなで同じものを作ろう。
- 価格が安くなる
- ソースツリーも一個
 - 保守も楽になる
- 要望があったらバージョンアップ時に追加
 - 他の人が使わなくても、ソースは一本のほうが楽
- 出来合いのものを買ってくればすぐ使える
- 作るための時間とお金を節約できる



コスト削減への道、その1 パッケージ (問題点)

- みんなと違うことがやりたいときに使えない
- 道具は使い方を覚えなければいけない。
- 高機能なものはカスタマイズもそれなりに大変。



コスト削減への道、その2 ASPの活用(メリット)

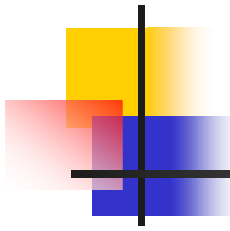
- 短期間でサービスを利用できるようになる。
 - 時間とお金を節約できる
- 運用、保守を自前でやらなくて良い
 - 自前でやる場合より安価な場合が多い
- ウェブベースのシステムはクライアントのコストを下げられる



コスト削減への道、その2 ASPの活用(問題点)

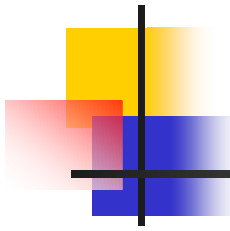
- 比較的簡単な業務にしか適用できない
- 回線速度に影響される
- Webベースは操作性が悪い

- 今までのイメージが悪い
 - **ここ重要！！！！**




コスト削減への道、その3 オープンソースの活用(オープンソースを使う理由)

- みんなが使っていてフリーで良いものがある
 - じゃあそれを使おう!!
 - みんなが使ってるなら充分実用的に違いない
 - 無料だし!
 - 実際に良いものなことが多い
 - 気がつけばデファクトスタンダード、事実上の標準
- 標準になったものを使うメリットは大きい
 - 拡張性
 - 他のシステムの連携
 - 運用、保守面



コスト削減への道、その3 オープンソースの活用 (問題点)

- カスタマイズ等にはやっぱりコストがかかる
- 保守をする人はやっぱり必要、それにはコストがかかる
- 運用も無料じゃない
- 業務システム等を構築する場合には、それほどコスト削減にならない場合が多い



コスト削減への道、その4 EUC(それって何?)

- Extended UNIX Code --> 違う！
- End User Computing
- 「@ITの情報マネジメント用語事典」によると
 - 情報システム部などのシステム管理者ではなく、企業経営者や業務部門などのシステムサービス利用者が直接的・主体的にコンピュータを操作したり、システムの構築に関与し、経営や業務に役立てることをいう。
 - 各人が勝手にシステム化する、という話。
 - メインフレーム時代からの移行時に、みんなでパソコンを使って業務改革！、と言われた時代の用語。
- 古い考え方だけど、思想は今も有効。
 - 各人が業務の効率化、高度化していける、というのは競争力を高める上ではかなり有効
 - 昔の EUC は、Word、Excel など
 - 今の EUC は gmail、Messenger、checkpad など



コスト削減への道、その4 EUC(問題点)

- データ連携ができなくなる
- 各人がばらばらなデータ形式を勝手に作っちゃう
- セキュリティの問題
- データの管理が面倒
- ユーザによる差が出てしまう
- 教育にコストがかかる
- トータルのシステムコストが上がることもある
- 各グループや個人が勝手にシステムを導入する
- クライアントPCの運用コスト、セキュリティリスクが上がる



コスト削減への道、その5 EA(エンタープライズアーキテクチャ)(EAって何? その1)

- 組織、業務的視点からシステムを分析・再構築する手法。
- 組織 (enterprise) の構造と機能のある一定の考え方・方法で包括的に体系化
- 全体と構成要素の相互関係を明確化
- 構造の背景にある基本理念・設計思想を含めて企業活動の全体最適を実現を指向
- あるべきモデル (To Be) を目指して、長期的かつ体系的に行われる



コスト削減への道、その5 EA(エンタープライズアーキテクチャ)(EAって何? その2)


- 4層の視点から業務を分析し整理を行なう
 - 政策・業務体系
 - データ体系
 - 適用処理体系
 - 技術体系
- 各層で使われる技術の標準化を行なうことで全体としてのコスト削減を行なう
 - 重要なのは全体最適



コスト削減への道、その5 EA(エンタープライズアーキテクチャ)(問題点)

- 概念等が難しい
- 理想論を現実の問題解決まで持っていくのは実は大変
- 政府は笛を吹いてるけど現実は。。。

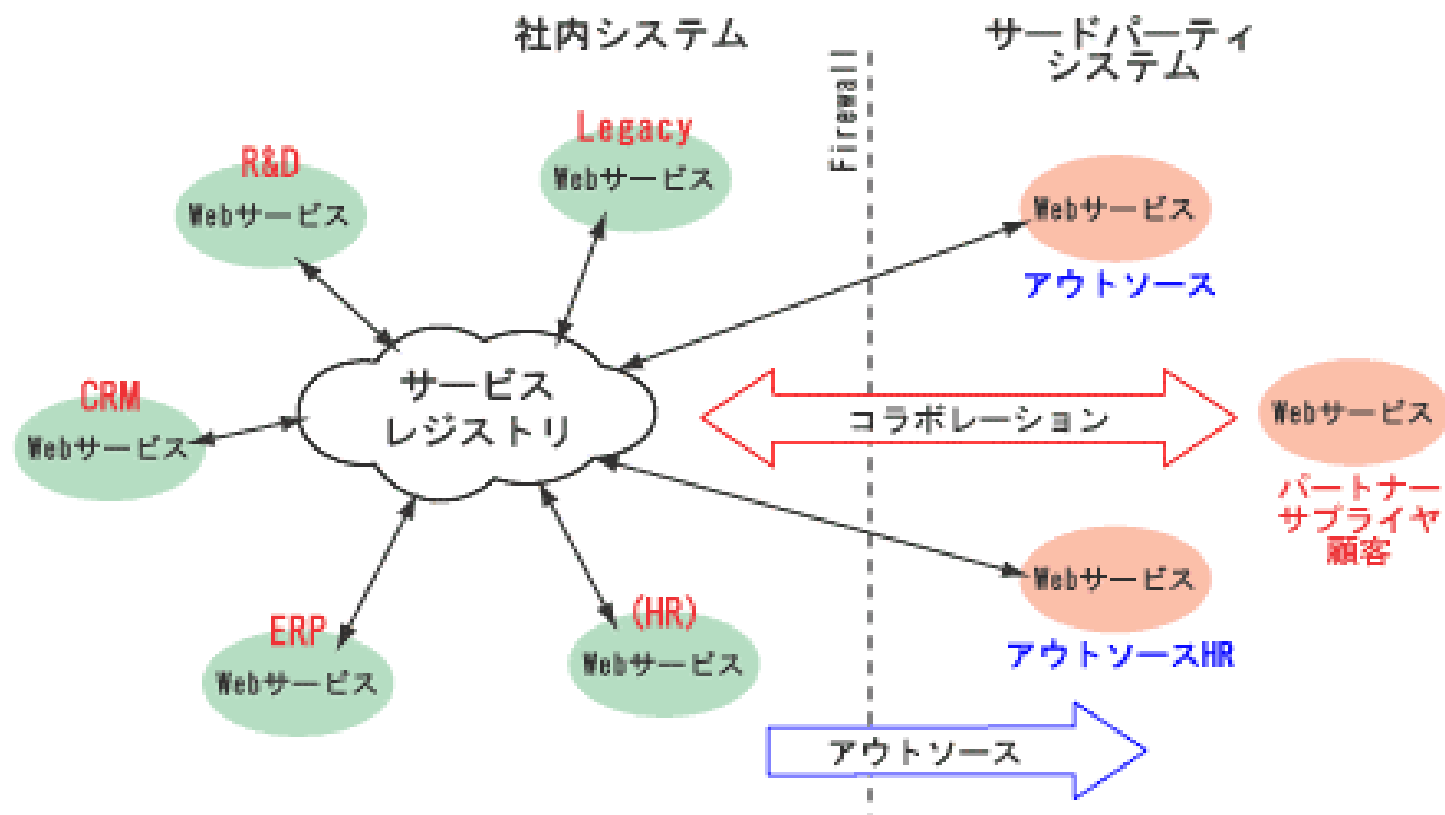
- でも重要なのは考え方！！！！
 - レイヤモデル
 - 全体最適！



コスト削減への道、その6 SOA(サービス指向アーキテクチャ) (SOAって何?)

- サービス指向アーキテクチャ
 - ビジネスプロセスの構成単位に合わせて構築・整理されたソフトウェア部品や機能を、ネットワーク上に公開し、これらを相互に連携させることにより、柔軟なエンタープライズ・システム、企業間ビジネスプロセス実行システムを構築しようというシステムアーキテクチャのこと。
- サービスが連携し合って動く
- 通信主導的な考え方

コスト削減への道、その6 SOA(サービス指向アーキテクチャ) (概念図)





コスト削減への道、その6 SOA(サービス指向アーキテクチャ) (ポイント)

- 既存の技術を活用
- 通信はWeb技術をベースとするのが流行り(Webサービス)
- TCP/IP および HTTPを利用
 - データ形式はXMLを利用
 - XMLによるデータのカプセル化を行なう
- データの中身については各アプリケーションが実装を行なう。
- ある程度のデータ型はW3C等で定義が行なわれている



コスト削減への道、その6 SOA(サービス指向アーキテクチャ) (メリット)

- システム間での連携を取ることができる
- RPCなどのプロセス間通信と考え方は一緒
- ウェブ技術を使っているためインターネットを介しての利用が可能
- 広範囲な用途に利用可能
- 標準化された技術を元にしており、サービスを部品化が可能
 - 部品を交換していくことによりシステムの刷新が可能



コスト削減への道、その7 agile開発

- アジャイルマニフェスト (日本語訳)
 - <http://www.metabolics.co.jp/XP/AgilePrinciples.html>
- 以上！！
- 考え方が最重要！！！！



コスト削減手法のまとめ

- パッケージの活用
 - ASP
 - オープンソース
 - EUC
 - EA
 - SOA
 - Agile開発
-
- 先人はいろいろなことを考えた！！！！
 - 良いことは真似しないと！！！！
 - ということで、いよいよ SaaS



SaaSって何？

- 超高機能な ASP
 - 従来のASPより業務範囲、機能が大幅に拡大
- ASPって一度失敗してるので印象が悪い！
 - 名前を変えてみた！！
- でも昔のASPと比べておそろしく強い！！
 - まあ名前変えるぐらい大目に見るかあ、という風潮？



SaaS はどう強力？

- 業務適用範囲の拡大
 - ありとあらゆる業務に利用できる
 - 他のシステムとのシームレスな連携が可能
 - 開発APIの充実
 - AJAX等の活用による操作性の向上
 - ユーザレベルでのカスタマイズができる
 - 全体としての統一性を持つ
 - 自由に迅速なカスタマイズ
-
- 他の手法の良いところ取りをしている。



SaaS が他のシステムから学んでいる点

- パッケージ同様、ソースはこれ一本！
 - ユーザは自由にカスタマイズ可能
 - 定期的に最新機能が追加される
- ASPのメリットをそのまま受け継ぐ
- 通信、外部連携は標準規格、デファクトスタンダードをそのまま利用できる
- EUC的にユーザが自分のレポートや帳票を自由に開発可能
- EA的な全体アーキテクチャの統一性
- SOAの一部として他システムと連携が可能
- Agileマニフェストの実現



配布資料より

- 日経コンピュータ48ページ下
 - 短期導入！高い拡張性
- ComputerWorld 3ページ、図1
 - ソフトウェア実行モデルの選択肢
 - ユーザは自由にカスタマイズ可能
- ComputerWorld 9ページ、図6
 - SaaSのメリットとデメリット



SaaS の注意点

- 道具は使う人の問題が一番大きい
 - どう運用するか、が重要
- 導入してすぐ使えるわけではない
 - 短期間、なだけ。
 - 教育、訓練、カスタマイズ、は必要
- 標準的ではないビジネスロジックは実現できないことがある
 - 標準化できないところが実はその組織のビジネスコア！



Break!

- ここまでで質問ある？



MCEAにおけるSalesforce導入の話

- 話中心！
- 配布資料を見ましょう。
- 適当に業務改革の資料も見せるね。



Break!

- ここまでで質問ある？



Salesforceの考え方

- 業務をすべて記録する
- プロセスの流れにそって各人が記録する
- 記録は、営業、企業、商談、等に時系列順にまとめられる カルテ化される
- 見込み顧客から商談までのプロセスを可視化できる
- リアルタイムに営業の戦略を立てられるようになる



お待たせ！Salesforceのデモ！！

- ログイン
- トップ画面
- 情報検索
- ダッシュボード
- レポート
- 差し込み印刷
- Web to リード
- CRM機能
- SFA機能
- AppExchange
- 他のシステムとの連携
- API、開発ツール
- カスタマイズ



Salesforce その他

- 個人用の無料ライセンスがある
 - お試しに使ってみては？



おしまい

- ご静聴ありがとうございました。
- 資料は後でウェブに上げときます。

- さて、雑談タイム！？