



# ユーザ企業から見たIT産業・ 基盤への問題提起

---

ゼリア新薬工業  
熊野

2009/6/17



# ユーザ企業から見た問題意識

---

- ◆ 複雑で鈍重なIT基盤
- ◆ バージョンアップ地獄
  - ▶ 当社のBIツールの事例
- ◆ 多層化・縦割りの組織
- ◆ ユーザ側も問題
- ◆ 管理主義～官僚主義
- ◆ 再構築地獄



# 複雑で鈍重なIT基盤

- コンピュータは年々遅くなっていると感じる。  
20年前のスーパーコンピュータの計算力が、数万円で購入できる。  
しかし、毎年コンピュータは遅くなっていくと感じる。  
(少なくとも体感は・・・)
- ユーザの使えるリソース  
CPUの処理能力の80%が、OS・ミドルウェアが消費している。  
企業のBiz Applicationは、20%程度しかリソースを使えていない。  
特に、ネットワーク上のパケットにおいて、Bizに必要な情報は  
何%程度なのだろうか?
- 多層・隠蔽構造  
ハードウェアを隠蔽するためにOS  
OSの差を吸収するために仮想マシン  
その上に、ミドルウェア・Application Server・Frameworkなどがあり、  
その上に、Biz Applicationが乗る。
- 多機能化  
使わない機能も多い。  
また、異なるソフトウェアで機能重複 → コスト増  
どのソフトでも同じことができる → どこで実施するかを考えて、開発ガバナンス



# 複雑で鈍重なIT基盤

- そもそも、多層化・隠蔽化は何のために行われたのだろう  
**変化に柔軟に対応するため、だと思われる。**

オブジェクト指向、データモデリング、正規化、MVCモデル、DI(Dependency Injection)コンテナ・SOA・疎結合・・・

- しかし、**現状のIT基盤は鈍重で変化に弱い。**

簡単にバージョンアップできない。

簡単に捨てることができない。

一度走り出すと止まらない。

**本末転倒**であると感じる。

- Black box化

しかも、このように多層化・隠蔽化されたIT基盤は、ブラックボックス化している。

そもそも、コンピュータは論理を積み重ねて構築されているはず。しかし、いつのころからか現象的にとらえるしかなくなってしまうている。



# バージョンアップ地獄

- ユーザ企業のIT部門の仕事の多くは、バージョンアップの仕事
  - ☞ ソフトウェアのバージョンアップ
    - 非互換の調査・テストに手間がかかる
  - ☞ サーバー(ハードウェア)のリプレース
    - リプレース地獄(OS・ミドルのバージョンにも気を使う)
  - ☞ Windows Update
    - 通信インフラに負担。メディア対応することも。
- 特に基幹系ERP・フロー制御基盤などのバージョンアップが難しい。
  - ☞ あるソフトウェアをバージョンアップすると他が・・・



# 当社のBIツールの事例

## ➤ 障害多発

- ☞ 原因不明のサーバハングアップが多発
  - ▶ 多いときには一日に数回サービス停止～ユーザから苦情電話の嵐
  - ▶ 週に2～3回の深夜の処理異常
- ☞ そして、その障害は再現性が無い

## ➤ 開発元にサポートを依頼

- ☞ ほとんどとり合ってもらえない。
- ☞ 「再現させてください」が口癖 → 再現テスト地獄
- ☞ あの手、この手でアプローチ → ほとんど結果出ず。

## ➤ その後

- ☞ BIT社カスタマーサポート部と七転八倒
- ☞ 内部がわからないものだけで現象を追いかける
- ☞ ようやく、いくつか突破口を見つける
- ☞ いくつかの回避策(抜本的でない)を得て、運用でカバー ← いまココ



# 当社のBIツールの事例

障害が多いのでバージョンアップを検討したのだが・・・

➤バージョンアップの非互換調査を実施

☞既存レポートが壊れる現象

▶結果 修正にかかる工数は、**30人月**

➤バージョンアップ後の機能の検証

☞旧バージョンに存在していた機能で、

**新バージョンで利用できないもの**がいくつもあった。

➤バージョンアップ後の負荷の検証

☞**約1.5倍重くなる**

▶コテコテしたユーザインタフェース

▶機能が増えたのだが、使わないものも多い

➤サーバ構成の検討

☞重くなるのでサーバ増設を検討した（バージョンアップするとすれば）

☞ライセンス体系の変更を伴い、**数千万円の追加投資が必要**と判明  
（サーバーライセンスを、もう1つ追加）



# 当社のBIツールの事例

## ➤ ライセンス問題

- ☞ 企業の合併・新バージョンに伴ってライセンス体系を変更  
～なんだかんだで、支出増
- ☞ CPUのCoreライセンスが問題
  - ▶ スタンダードなハードウェアは、ほとんどMulti-core。
  - ▶ ハードウェアリプレイス時に問題

## ➤ さらに、保守料・技術サポート料も高騰

- ☞ 保守もサポートも不満なのに・・・
- ☞ なぜ、保守料値上げ？ → 不景気だから？

## ➤ 結果、バージョンアップを断念

- ☞ 30人月かけて、機能ダウンして、重くなって、コストは倍増で・・・無理

## ➤ しばらくすると、旧バージョンは保守対象外に。

- ☞ 旧バージョンで障害が発生しても、一切受け付けてもらえない。

## ➤ 旧バージョンを動作させるOSにも寿命が。

## ➤ 出口が見えない ← いまココ





# 多層化・縦割りの組織

- IT基盤だけが、多層化・構造化しているわけではない。
- 人も多層化している
  - 還元主義→分業化→接着役としてのプロマネだらけ。  
ラピッドな意思決定できるわけが無い。伝言ゲームだらけ。
- 大手ベンダーにプロジェクトをゆだねると……
  - プロマネが5人も10人も
  - プログラマやモデラーよりも給料高い傾向
- 下請け・孫請け・オフショア……ヒエラルキー構造を作っている
  - 👉 官僚主義的になる傾向がある。



# ユーザ側も問題

---

➤ 技術力の低下・弱体化

ドッグイヤーで、変化の早い技術についていけない。

➤ 高齢化・空洞化

➤ ポリシーなき、アウトソーシング

➤ **丸投げ体質**

思考停止してしまっていて、自分たちで考える必要のあることを、**ベンダーに任してしまう。**

- ▶ 要件を語ることができずに、ベンダー任せ
- ▶ 下流工程で手戻り多発

➤ ベンダーと、半癒着関係

困い込まれると楽？

➤ 管理主義

ソフトウェアって、もっと自由で、ボーダーレスな存在なんだが・・・  
管理・管理・管理・・・で、そのポテンシャルを奪っている気がする



# 管理主義～官僚主義

- 管理主義・官僚主義は、ITの導入・活用に抑制的に働く。
  - ☞ ヒエラルキー文化は、人間を機械のように扱う傾向がある。
  - ☞ ITは人間の創造力・思考力を倍増させる利器。
  - ☞ 機械のように人間が働く環境では、ITは単なる便利な道具
  - ☞ 昨今、様々な法規制が、管理主義を加速させている
  
- 企業と情報力
  - ☞ 20年くらい前は、大企業が情報に恵まれていた。
    - ▶ メインフレーム・専用回線に投資
  - ☞ 今は、小規模企業のほうが有利
    - ▶ SaaS・HaaS・PaaS・Cloudを軽快に使いこなす。
    - ▶ 大企業はこの軽快さが無い
  
- Google phobia(グーグル恐怖症)
  - ☞ セキュリティ・内部統制・個人情報保護・・・
  - ☞ 大企業は、Google phobiaを克服できないだろうといわれている。
  - ☞ なぜ、安価で高機能で大容量で便利なツールを大企業は使わないのか



# 再構築地獄・・・

---

➤ 基幹系システムの再構築・・・

☞ 5億・10億当たり前にかかる。

➤ 10年使って、5億で再構築としたら、

☞ 年間、5000万の換算

➤ サブシステムが、10個存在し、それぞれ10年使うとすると

☞ 毎年再構築プロジェクト

➤ なぜ、再構築しなければならないのか？

☞ 現実の業務と乖離

☞ インフラ・サポートにおいて、“梯子を外される”

➤ それでも、“価値”を生み出せばいいんだが・・・

☞ 単なる道具だったりする

☞ 単なる道具の再構築に、5億とかかかるのはいかなものか



# ユーザサイドから見たシステムのあるべき姿

---

◆ **基幹系**

◆ **情報系**

◆ **コミュニケーション系**



# ユーザサイドから見たシステムのあるべき姿 ～基幹系～

- **変化に即応**できること
  - これが最も重要。
  - 現実世界は常に変化する。コンピュータシステム然り。
  
- そして、変化に上手に対応できること
  - ここでいう変化とは、経営要求と同義。
    - ▶ 変化に対応っていうが、今までも経営要求にソフトウェアは対応してきた!
    - ▶ ソフトウェアを汚さないで、が前提である。
  
- ソフトウェアを汚さないってどういうこと？
  - ▶ 属人化しないことだったり、他人が容易に認知できること。
  - ▶ 『不良債権プログラム』にならないこと
  
- 汚れたソフトウェアって？
  - ▶ 再構築が付きまとう。



## ユーザサイドから見たシステムのあるべき姿 ～基幹系～

- しかも、問題解決しない、DeadCopy的な再構築事例多い。
  - ▶ 基幹系システムのあるべき姿をイメージできず、思考停止状態になると、こうなる
  - ▶ 億のオーダーの再構築で、問題解決しないのでは再構築の価値無い
- 経営・事業に合わせて、ソフトウェアを開発せよ…  
至極、当然のことを言っている？
  - ▶ 細かい点で、結構、「足を靴に合わせる行為」をしている。
  - ▶ これが、積もり積もって、億のオーダーで再構築
- **変化に即応する基幹系システムは、再構築しないものでありたい。**
  - ▶ 10年後に、フロントもバックエンドも全然変わってしまっても良い。
- 基幹系に必要なアーキテクチャ
  - ▶ オブジェクト指向設計が重要
  - ▶ または、アプリケーションそのものを、簡単に作り直す手法
    - ▶ Another solution ➡ ユニケージ手法



## ユーザサイドから見たシステムのあるべき姿 ～基幹系～

---

- 基幹系に必要なガバナンス
  - ▶ 開発ガバナンス・IT基盤アーキテクトの存在
  - ▶ マスター・データ・マネージメント
  - ▶ IT Capability (ITを使いこなす組織的能力)
  
- 作るべきか？ 買うべきか？
  - ▶ 買う、ということはパッケージソフトやSaaSのこと。
  
  - ▶ 基幹系システムが、顧客に対して差別化を生み出すものなら、自社開発すべき (Core or Context)





# ユーザサイドから見たシステムのあるべき姿 ～基幹系～

## ◆パッケージ導入編

➤パッケージソフトウェア・SaaSなどを選定する場合

➤クラスモデルや、データモデルなどが提示されるものを選ぶ

☞ 『モデルベースのFit&Gap』

➤モデルの提示

内部構造や物理を提示して欲しいと言っているわけではない。  
概念レベルでの、モデルを提示して欲しい。(これは必須だ)

➤クラス図、ER図などの、モデルで会話ができないベンダーが多い

☞ 手続き型、構造化手法等の従来のアーキテクチャで作られているかも?

☞ 変化に柔軟でない

☞ 結果、数年後に再構築の話が上がる

➤判断基準

▶ 簡単にやめられるかどうか

▶ 実績があるか?

パッケージやサービスを提供するベンダーが、なくなってしまうかも?

▶ バージョンアップが簡単にできるか? ☞ バージョンアップ地獄にならないこと



## ユーザサイドから見たシステムのあるべき姿 ～情報系～

---

- 定型的な計数をキレイに表示するだけではダメだ  
売上高、値引高、利益高、粗利率……  
昔の帳票でも見ていた内容。  
これらは、仕事の結果である。
- 情報系は、**局地戦で、(現場の)暗黙知を活用**せよ！
- どうも、BIベンダーも定型的計数を、キレイにグラフ化する  
プレゼンに終始している気がする。
  - ▶ そして、ある意味どのソフトも同じに見える。

# 現場の暗黙知って何？

- 病院ごとの患者の数 → 顧客のCapa分析に必須  
その中の、胃炎の患者さんの数  
→ このような情報は、営業マンは(なんとなく)知っている。
- 薬局で、どんなお客さんが、当社の製品を購入？  
女性？老人？固定客？一元さん？サラリーマン？本人？代理？
- お客さんは、どういう方式でご購入される？  
拾い買い？指名買い？薬局で推奨？
- 売場の棚割りは、どんな感じ？  
レジ横？コーナーの吊るし？フェイスの幅は？
- 小売店の立地は？  
ファミリー街？若い学生？出張のサラリーマン多い？飲み屋街？
- 薬は、どんな服用のされ方をしているのだろう？  
家で服用？外出先？お酒を飲む前？

# 局地戦で、暗黙知を活用する情報系システム

- **このような、マーケティングに最も大事な、顧客や商品に関する事態が、Central-DWHに無い**
  - ▶ あるのは結果としての、売上高とか利益額
- これらの情報は、営業マンが現場を回って、なんとなく知っている
  - ▶ 基本は頭の中。
- そして、このような情報をCentral-DWHで管理してはダメ！
  - ▶ このような情報は、中央集権的に管理することはなじまない。
  - ▶ 暗黙知を、“管理”してはだめ。
- 現場に入り込んで、小規模な環境で展開する
- このような**暗黙的な情報をデータ化**して、仮説立案～実施～検証のPDCAサイクルを回す。
- 逆だ。
  - ▶ 一般的なBI : data → information → intelligence → profit
  - ▶ こうあるべき : (Tacit) information or intelligence → data

# ツール・ソフトウェアはどうあるべきか

- ツールはあまり重要ではない
  - ▶ しかし、BIツールにはあまり無い、エントリー機能は必要。
- 様々な集合、属性が簡単に(テキスト入力感覚で)登録できること
- 属性の設定は、メインのリソースに対して、“関係”で表現する  
属性はカラムで持たない
- または、このような、ボトムアップで現場を支援するコンサルティングは、  
1つのビジネスモデルになるかもしれない
  - ▶ 現場に入るコンサル。
  - ▶ 顧客(この場合ユーザ)と、一緒になって悩み、喜ぶ。
    - ▶ トップダウンなコンサルではない

# 成功事例を共有する方法

## ➤ 成功事例の共有方法の一例

☞ マネージャが、部下にヒアリング

▶ 「**今月のあなたのベスト・プラクティスは何だった？**」

☞ このベスト・プラクティスを、本部で集める。

☞ 本部では、「さらにその中で、ベスト10」を選出

☞ 「ベスト10」を現場にフィードバック。それを壁に貼る。

☞ 現場の営業は、この「ベスト10」から

▶ 「**今月は、この4番目の方法を試してみます**」と宣言させる。

☞ その結果(成否)を本部に返す。

☞ 本部で結果を集計

☞ Best of Bestが残る

☞ 最初に戻る

➤ もちろん、この方式を、戦略的ビジネス単位(SBU)毎に行う。

➤ この方式は、基本的にIT不要

☞ **紙だけ**でもできる。



# ユーザサイドから見たシステムのあるべき姿 ~コミュニケーション系

➤ コミュニケーション系とは、ポータルやグループウェア。

➤ 今後最も、重要なシステムになる可能性がある。

## ➤ 背景

☞ 今の日本の組織は、縦割り、官僚主義

☞ 縦割組織・官僚主義では、“知”が限られる

☞ “知”は、横断的な知識を重ね合わせるのがいい

▶ アイデア創出法：全く異なる単語で、Google検索してみる

(例：万葉集と酸性雨とか・・・)

☞ 世界的に有名な事例 ➡ GEのワークアウト・タウンミーティング

➤ GEのタウンミーティングのような、コミュニケーションを、SNS等を使って行う。

▶ 企業内SNS・ブログ ➡ 官僚主義的な企業文化を変える力がある？



## ユーザサイドから見たシステムのあるべき姿 ~コミュニケーション系

### ➤これも、**管理主義・官僚主義**で実施してはダメ

- ☞ 管理しては逆効果
- ☞ 自由闊達で、本音が出るコミュニケーション場（タバコ部屋に変わる場）
- ☞ 組織の言葉から、個人の言葉

### ➤**SNSファシリテータが重要**

- ☞ ファシリテータとは・・・参加者の議論を促進する役割
- ☞ 一般的な合意形成型会議でのファシリテーションとは一風異なる

### ➤あくまで、“人間中心”

- ☞ フェイス・トゥ・フェイスと併用

### ➤企業内文化の変革 ⇨ ビジネス創出へ

横断的な“知”の交流から、新たなBizアイデアが創出すれば大成功  
そのために、上流にアイデアをエスカレーションするプロセスを  
定義しておく必要がある





## ユーザサイドから見たシステムのあるべき姿 ~コミュニケーション系

- ツールはどうあるべきか？
- 自社開発するメリットは無い。
  - ☞ ASP・SaaSが向くだろう。
  - ☞ OpenPNEなどの、オープンソースを利用するのも手だろう。
- といっても、本質はソフトウェア論ではない
  - ☞ アクセスコントロールや、ロギングなどは、どんなソフトウェア、サービスでも、大抵同じようにできると思われる。
  - ☞ 大事なものは、前述のSNSファシリテーションなど



# ユーザサイドから見たシステムのあるべき姿

## ◆コンシューマユースと企業ユース

- キレイなだけのインターフェースは企業ユースでは不要
  - ▶ PCパワー・ネットワークに多大な投資がいる
- 大容量HDD
  - ▶ コンシューマユースでも、HDDは動画のための大容量といってよい
  - ▶ 企業USEでも動画が使うが、媒体ベースで持てばよろしい
- PC起動も兎に角遅い
  - ▶ モバイルPCの起動に数分かかる。
    - ▶ 暗号化されているHDDの復号に時間がかかる
- モバイルPCの利用におけるインフラ
  - ▶ 電源と電波



# ユーザサイドから見たシステムのあるべき姿 まとめ

## ◆基幹系

変化に即応し、再構築で多大なコストをかけない

## ◆情報系

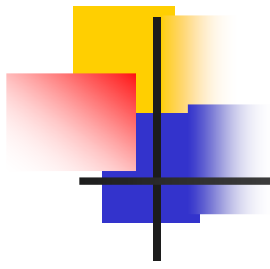
価値を創出することを念頭に、現場の暗黙知を活用する

## ◆コミュニケーション系

官僚的な企業文化を変革する、横断的コミュニティをファシリテートせよ

## ◆コンシューマユースと企業ユース

IT利用における本質的な違いを再認識し、企業ユースをデザインする必要がある



以上です。

ご清聴ありがとうございました