

I -1. 現状認識（歴史・ベンダー）

◆ システム化の歴史 …… パッケージ・ソフト

1. 仕事の改善手段としてコンピュータ利用 → 半世紀以上(60年代後半)
2. 当初は自社の特定の部署の業務改善手段 → 業務システムの始まり
3. 専門部隊による開発が本格化 & 問題も発生 → 高額な開発費
4. 「費用」対「効果」の発想が生まれた → ユーザ負担
5. 専門部隊も採算性を問われる時代の到来（'80年前後） → 情報部門の独立
6. 特定ユーザ向けソフトの有効利用（開発リスクも回避） → パッケージの出現

* パッケージ製品の原点は、特定ユーザ向けの個別システムをレベル・アップで汎用化

◆ ベンダー(開発会社)事情

1. 開発のプロ集団 → プログラムはデキルが“業務知識は乏しい”
2. 業務系に弱い、高額な開発コスト → 自社固有のパッケージ開発は困難

* 開発時の対応で問題が発生しやすい → 表面上の会話以外”バグ“の確率大

I-2. 現状認識（ユーザ）

◆ 社会環境の変化

1. 高齢化社会・少子化・終身雇用の崩壊 → 次世代へのノウハウの継承
2. 企業の商品化 → 人材育成の減速化
3. 分業化の加速 → 企業体質の弱体化
4. 便利なweb社会 → 情報過多、信頼できる情報は？
5. 人事制度に変化 → 個人単位の仕事の加速（勤務時間・場所）など

◆ ユーザ事情

1. 自社開発からの撤退 → パッケージの利用
 2. 企業への定着率・愛着が低下 → 個人主義の加速・優秀な人材の減少
 3. 分業化の加速 → 全社的な業務のわかる人材が皆無化
 4. 閉鎖的な業務分担（実務優先、管理責任の範囲） → 牽制・監査的機能の衰退
- * 人を育つ・育てる環境がなくなっている → 優秀な業務リーダーの皆無化傾向

I-3. 現状認識（ソフト・開発）

- パッケージ .. 単一システム(限られた範囲の業務に対応)
 1. 製品化 .. ユーザ向け開発ソフトをカスタマイズ(会計・顧客・在庫・人事給与ほか)
 2. 業種 .. 特定分野（開発したユーザの業界・分野）
 3. 品質 .. 通常、カスタマイズしない範囲で安定化(カスタマイズにより不安定確立大)
 4. パッケージを独自に開発できない(理由) → 受注型対応以外の開発は困難
高額な開発費用が必要、独自の業務知識に乏しいなど

- 統合システム .. 単一システムを繋ぎ合わせて総合的な仕組み
 1. 概要 .. 上流・下流システムのデータを連動して(カスタマイズ)総合化
 2. 課題(品質面) .. 組合せによるトラブルの多発(運用面の不具合・矛盾など)
 3. 課題(費用・納期) .. 高額なカスタマイズ費用、長期に渡る安定化までの期間

- 開発現場 .. “言った事”しか開発できない現状 ⇒ 『 結果に不満 』
 1. ユーザ側 .. ニーズ(業務の詳細)を全て言えない → 追加費用の要請
 2. ベンダー側 .. 業務知識がない(理解できない) → 不具合の原因・品質不良