パネル討論
高度なソフトウェア技術者の育成

ソフトウェアアーキテクトの育成経験から

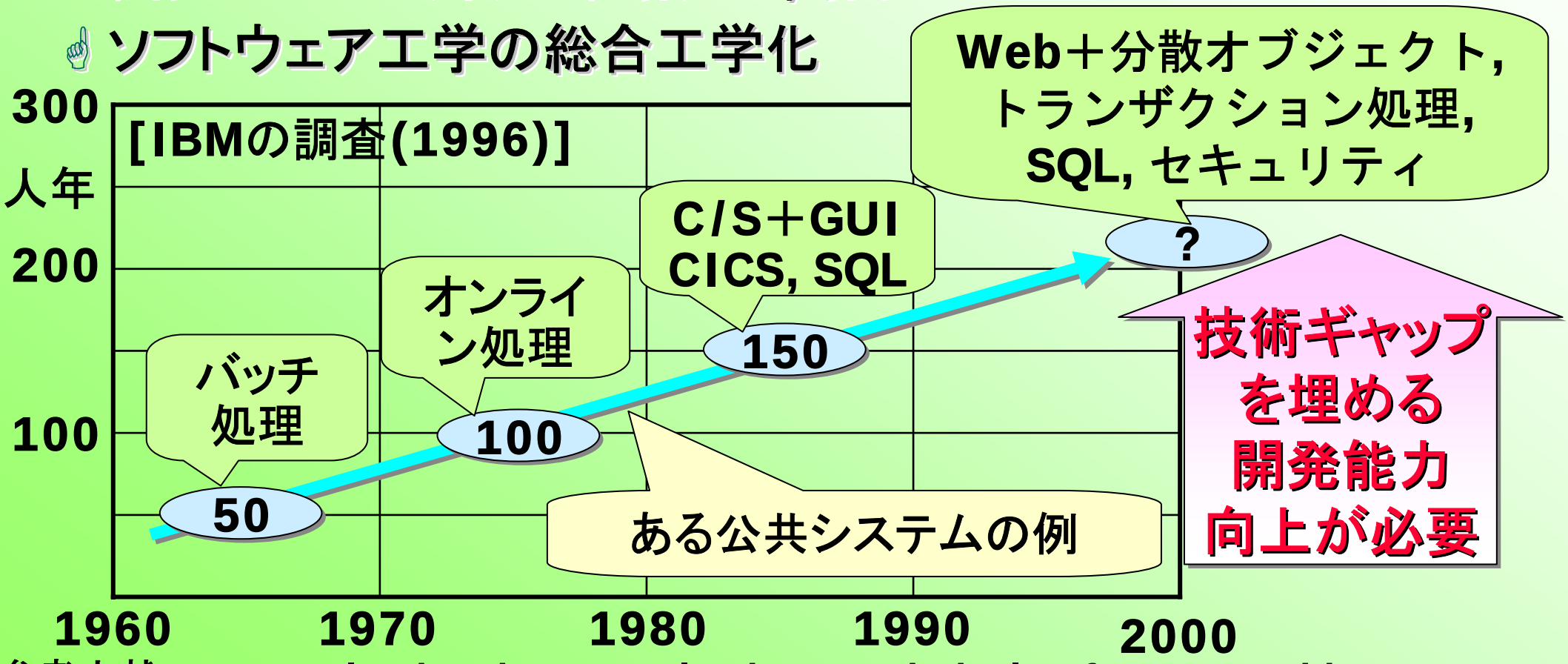
青山 幹雄
南山大学 数理情報学部 情報通信学科
mikio.aoyama@nifty.com
http://www.seto.nanzan-u.ac.jp/~amikio/NISE/

2004年3月10日

問題: ソフトウェア開発リスクの増大

技術障壁の高まり

- ソフトウェア開発の高度化・ハイリスク化
  - 同一「機能」のシステム ⇒ 高度な技術の必要性
  - 開発力による淘汰・市場分化, 寡占化
  - ソフトウェア工学の総合工学化



参考文献: P. T. L. Lloyd and G. M. Galambos, Technical Reference Architectures, IBM Systems Journal, Vol. 38, No. 1, 1999, pp. 51-75.

問題: 技術の高度化へ対応できる技術者

- Capacity(量)からCapability(能力=質)へ
  - 技術の高度化に対応できる高度な技術者の育成
    - ソフトウェアアーキテクト, CIOの育成
  - ITスキル標準など

問題: アーキテクチャとアーキテクト

- 「アーキテクチャ(architecture)」の語源
  - 古代ギリシャ語: architecton techneの簡略化
  - architectonice=arche(原理)+tecton(職人)
- 「アーキテクト(architecton)」とは，ものごとの原理や根本的な知識を備え，職人を指導し技術を統合して制作を企画しうる能力を持つ人
  - 「アーキテクチャ」とは，建物(building)ではなく，建物を創造するための技術，すなわち，ソフトウェアである

参考文献: 建築学教育研究会(編), 建築を知る, 鹿島出版会, 2002.

問題: アーキテクトに求められる資質

総合能力

- アーキテクトに必要な資質・能力
  - 技術力: 高い技術力・経験と新しい分野への探究心
  - 見識力: 大局的な見方・考え方と抽象化・モデル化の力
  - 実践力: 理想と現実・技術とビジネス等のバランスを取る感性
  - ビジネス力: 顧客要求を技術に翻訳する力
  - リーダシップ: 自分の意思を伝え, 理解, 協力, 参画を得る力
- ソフトウェア開発におけるアーキテクトの役割
  - 将来に向けて製品の方向性を設計
  - 製品の全体構造(ソフトウェアアーキテクチャ)の設計
  - 自己のアーキテクチャの優位性を社内外に説明し賛同を得る
  - 実装のガイドやコンサルテーション

参考文献: 内永ゆか子, IBMにおけるソフトウェア開発のプロセスとマネジメント, オブジェクト指向2001シンポジウム資料集, 情報処理学会ソフトウェア工学研究会, 2001, pp. 7-11.

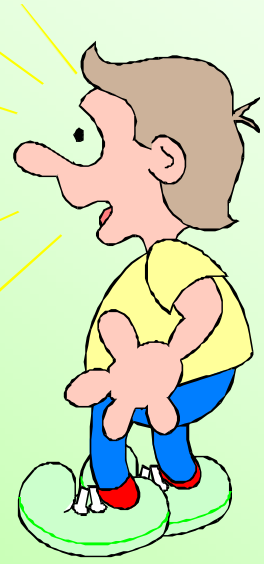
取組み: アーキテクト育成: 枠組み

- 実施: 2001年～[1~2回/年]
  - 実施までに約1年間準備
- 対象: グループサブリーダ, リーダ(課長前)
- 期間: 3/6か月(終了後3/6か月フォローアップ)
  - 業務と並行
- 内容:実際の問題解決(GEのWork-Outスタイル)
  - 現実の問題解決こそがリーダ育成の最高の手段
  - 講義と実習を並行
- 人数: 10名/回(所属部門長の推薦要)
- 講師: 社内アーキテクト＋社外(第一線の専門家)

参考文献: D. Ulrich, et al., The GE Work-Out, McGraw-Hill, 2002.

学会の役割: アーキテクト育成の評価から

- そもそもアーキテクトは育成すべき/できるか?
  - すべき: 個人任せから組織的取組みへ(競争力の源泉)
  - できる: 建築家, [トヨタのCE (Chief Engineer)制度?]
- アーキテクトは育成できたか?
  - Yes(評価はポジティブだが)/No(本来, 促成栽培は?)
    - 個人の方向付け
- 何が問題だったか? ⇒ 学会の役割
  - 教科書と教材の欠如, 技術の成熟度(半ばアート)
    - 青山ほか(編), ソフトウェアテクノロジーシリーズ(全12巻)
  - ソフトウェア工学知識の不足, アンバランス
  - アーキテクト認定: 受講者は強く要望
    - 基準の統一と強制力: ソフトウェア1級建築士(?)



学会の役割: 大学・社会へのフィードバック

- ソフトウェア工学教育振興: カリキュラムデザイン, 教員育成
  - 設計(方法論)教育と技術の深化・総合化への対応
    - プログラムからアーキテクチャ(マクロ構造)への視点
    - 小規模プログラミング(犬小屋プログラマ育成?) から脱却
- ソフトウェア/技術者の地位向上
  - アーキテクトは尊敬される目標
    - 米国のソフトウェアアーキテクト
  - ソフトウェアは3K産業?
    - 情報系学生の減少

まとめにかえて

今，ソフトウェア開発が工業プロセスに脱皮すべき時だ R. Troy(CEO, Verilog SA)

ソフトウェア開発の根本的な前提を問い直す必要がある.**今日のソフトウェア技術者は, 煉瓦積み職人に等しく,建築家ではない.**
ソフトウェア開発がエンジニアリングになる時が来ている.
この変化は、産業革命に比肩できるほどだ.
IEEE Spectrum, Vol. 31, No. 1,
Jan. 1994, pp. 40-41.