

## 誰にでも出来ないOA化から誰にでも出来るOA化へ（その2）

4K-4

## — OA化の考え方とその教育 —

今井 恒雄、増田 秋俊（富士通） 平木 しげ子（東京家政学院筑波短大）  
荒木 雄豪（京産大） 古谷 千恵（東大生研）

## 1. はじめに

共同発表者の中の大学に属している3名は、大学の教員の立場でこれまで大学の事務処理システムを教員と職員、場合によっては学生も含めて開発してきた長い経験を持つ。

コンピュータの知識等の全くない普通の大学の事務職員をスタッフにしてシステム開発を実践してきたのである。その時々々の環境の中で、20年来独自の開発方法で取り組んできた。平成2年に開学した東京家政学院筑波短期大学の事務システムは平木とゼミの学生および、平木ゼミ卒業の職員という体制で開発し、現在も職員がシステムの保守作業の殆どを担当している。

こうした経験を元にシステム開発の教育を行ってきたが、大学を卒業した学生が、メーカ等に入社して社内教育を受けると、自分達の大学で受けた教育とはシステム開発の全てについて大きな違いがあることに驚くが、結果としてメーカの考え方を絶対的なものと思う会社人間ができあがる。

我々の考え方での教育と訓練で「自分でも出来る」と自信を持って卒業した学生が、落ちこぼれになるケースも当然出ている。

メーカを中心として大量のソフト要員を抱えたソフト業界が自分達の仕事が結果として少なくなるような開発方法を発想することは、置かれた立場を考えると難しいことかもしれないが、我々の目指すところは、システム開発を組織の中のメンバで、即ち自分達で開発することである。

こうしたことを背景にして、誰にでも出来るOA化という発想で、これまで実践してきた我々のOA化の考え方とその教育法について紹介する。

## 2. 開発方法の考え方

図表1に、やり方の違いを明確にするために一般的な開発方法と我々の方法との比較を試みた。言葉としては同じであるが、その目的と実態では大きな違いがある。繰り返して言うが、例えば大学の事務職員が、自分達でシステム開発とメンテナンスを可能にするノウハウは、徹底した再利用と易しく作ることに主眼を置くことである。

## 3. 教育

図表2に、同様に教育での比較を試みた。我々の教育方針は、自分達のシステム開発に必要な力をつけることに的を絞り、徹底した繰り返しによる訓練をすることにつきる。

---

A system development method that anyone can do (No.2)

Tuneo IMAI<sup>1</sup>, Akitosi MASUDA<sup>1</sup>, Sigeko HIRAKI<sup>2</sup>, Hugo ARAKI<sup>3</sup>, Tie HURUTANI<sup>4</sup>

1 Fujitsu Ltd. 2 Tokyo Kasei Gakuin Tukuba Junior College 3 Kyoto Sangyo Univ.

4 Institute of Industrial Science, University of Tokyo.

	一般的な開発方法	我々の開発方法
開発法	<p><b>開発の工程</b> 企画、設計、テストなど細かい工程に分けて、全工程で依頼者からの承認をもらって進み、工程の後戻りは原則として無い。</p> <p><b>標準化</b> 管理と各工程の担当者間での引渡しのため、設計、ドキュメントコーディング、インタフェースなどの標準を決める。</p> <p><b>部品化</b> 管理と作業分担のための、共通部品、ひな型を用意する。</p> <p><b>ファイル設計</b> アクセス効率、データの一元性、データ量などを重視する。</p>	<p><b>開発の工程</b> 工程という概念はなく、樹木が成長するように開発を進める。現場からみるといきなり帳票が出来始め、これを元に話し合いを進め必要帳票が次々と出来、システムが完成に近づいていく。初心者は、業務内容も殆ど理解出来ないところからプログラムを作りながらシステム全体の理解を深めていく。</p> <p><b>標準化</b> システム全体の再利用を目的とする。 ・インタフェースの統一 ・ファイルレイアウトの考え方(原ファイル、主ファイル)</p> <p><b>部品化</b> 業務プログラムの作り易さと再利用を目的とする。 ・共通部品 ・難しい部分の部品(サブルーチン) ・ファイル毎の13種類のひな型プログラム</p> <p><b>ファイル設計</b> 業務担当者のデータ入力のし易さと、初心者プログラマが業務プログラムを作りやすくすることを最も重要な設計指針とする。</p>
開発体制と役割の分担	<p>上級SE、SE、プログラマなどを、開発工程に合わせて必要な人材を投入、分業体制をとる。</p> <p>仕事の分担は、受託側の各担当者や顧客も、それぞれの役割を各工程ごとに詳細に取り決める。</p>	<p>リーダーと初心者を含めたメンバで協働体制をとる。 リーダー システムの再利用を最大限に活用し、開発の基盤を作る</p> <ul style="list-style-type: none"> <li>・ファイル設計</li> <li>・ファイル(テストデータ)の準備</li> <li>・部品の開発(ファイル毎に13種類のプログラムの原型、サブルーチン)</li> <li>・メンバの訓練とそれと並行してOJTでの開発指導</li> </ul> <p>メンバ リーダーの用意した環境で訓練を受け、業務プログラムを作る。並行して、システム設計の訓練書でこの開発法を学ぶ。</p>
問題点	<p>こうした開発方法は、不明確な受託条件の下で、決められた資金と工期でシステムを稼働させるための仕組みであり、引渡し後、現場担当者で維持できるシステムにならないことが殆どである。このため、システムの保守は例えば企業のシステム部門や開発を委託した業者が担当することが多い。</p>	<p>従来の方式と全く発想が違うため一般的な教育を受けてきた者にとっては受入れ難いので、この方法全体を理解するリーダークラスの育成が課題である。また、この方法で育成された初心者プログラマは従来の方式の開発法では難しすぎて自信を失うことがしばしば起こる。初心者でもできる易しい開発方法が望まれる。</p>

図表1：システムの開発方法の比較

	一般的な教育	我々の教育(訓練)
目的	SEとして必要となる幅広い知識が何かを知ること、その基礎を理解すること。	リーダーの下で自分達のシステム開発ができること。
目標	実際のプロジェクトでOJTによる開発まで経験させて、情報処理技術者試験に合格するレベルまでになること。	ファイル設計、データエントリ、プログラム作成等、トータルシステムが作れること。
方法	講義と実習、OJTとバランスをとりながら進めるが、何度か経験させる方法であり、訓練という考え方はない。	我々の共同の著作である5冊の訓練書に含まれるプログラムの作成や、システム設計に関する約500題の問題を出来るまで繰り返し解かせる。そして、あるレベルで業務ソフトの開発に参加させて、自信と感激を味合わせて育成する。
問題点	大きな企業では、新入社員に対して半年間程度の集合教育を実施しているケースが多いが、十分な教育効果が出ていないことが多い。これは講義、実習ともに詰め込み型になっており、繰り返し訓練して、体で覚えるような方式をとっていないことが大きな原因の一つと考えられる。	既に他の一般的な方法で教育を受けて育ったプログラマなどは、本当はまだ充分力がついていないのに、過去にやってきたという自信があるので、この訓練を受けると、初めの内こそは初心者よりも早く進むが、易しいプログラムを繰り返し作ることに抵抗を感じるのと手抜きをすること等で落伍することがある。

図表2：システム開発の教育面での比較

4. おわりに

これまでの学校事務の機械化を教員と職員、場合によっては学生で実践してきた経緯と経験を元にしてメーカーの発想とは大きく違うところがあるが、業界全体が目指す一つの目標である自分達で出来るシステム開発の実例を紹介した。