

# エンドユーザ主導によるシステム開発手法

京都情報大学院大学 教授

今井 恒雄

我々は業務担当者が自らの手でシステム開発を行う方式、即ち“エンドユーザ主導によるシステム開発”の方式を開発し実践に使用して成果を得ているので、それについて述べる。

我々のシステム開発法は、業務担当者が仕事の合間(3ヶ月以内)に自習方式でCOBOLの初級プログラマの力をつけることから始まる。これが終わると手作業で使用している帳票類を参考にして本物の帳票を出力するプログラムの作成に入る。帳票出力プログラムは納得のいくまで修正をするが、この繰り返しを通して、業務担当者は“エンドユーザ主導によるシステム開発”のシステム開発法を身につけていく。そして、それと並行して自然な形でOA化が進んでいく。システム開発で特にネックとなるプログラマ不足に焦点を当てたこの開発法を実現するために、我々はCOBOL言語教育とシステム設計、特にファイル設計、そして双方の関連性に着目し研究を進めた。業務プログラムは確実に・数多く・早く・簡単に作る必要がある。コンピュータに関しては全くの素人を、短期間の言語教育(自習)だけでこのようなことのできるプログラマに育てるには無理がある。しかし、ファイル、即ちデータの持ち方に独自の工夫をし、その考え方に沿った教育方法を考えることによって解決することができた。この考えに基づいたシステム開発法を5冊の訓練書にまとめて出版し実際の業務の開発や大学での教育に使用し成果を上げている。

キーワード: システム開発, ユーザ主導, 帳票印刷, 容易なプログラム開発, ファイル項目の構成

## 1. はじめに

システム開発での大きな問題の1つに、システムを開発する主体が実際の業務担当者ではなく、その組織のコンピュータ部門や外部のソフト開発の専門家になっていることがある。システム開発での業務担当者の役割は、開発すべき業務内容を開発者に伝え、実際に仕事をする立場から画面や帳票を決めることと、出来上がったシステムのテストをすることであり、業務プログラムの開発に直接参画することはない。業務担当者がシステム開発後の維持管理を担当できないためにシステムが実際の業務に合わなくなることがしばしば起こる。そのために十分に活用されないとか、せっかく導入したシステムが使われずに埃をかぶってしまうことが往々にしてある。こうした現実の開発の仕組みが現状のままである限り解決しない。

筆者は長年メーカーの立場で大学・高校等のシステム開発を経験し、従来のメーカーやソフトハウス中心のやり方では業務担当者によるシステム開発は無論のこと、出来上がったシステムの維持管理さえもできないことに問題意識を持ち、その解決法を試みていた。

システム開発の中で特にネックとなるプログラマ不足に焦点を当てたこの開発法を実現するために、我々はCOBOL言語教育とシステム設計、特にファイル設計そして双方の関連性に着眼し研究を進めた。業務プログラムは確実に・数多く・早く・簡単に作る必要がある。コンピュータに関しては全くの素人を短期間の言語教育(自習)だけでこのようなことのできるプログラマに育てるには無理がある。しかし、ファイル即ちデータの持ち方に独自の工夫をし、その考え方に沿った教育方法を考えることによって解決することができた。このシステム開発法(言語教育を含む)を5冊の訓練書<sup>1)~5)</sup>にまとめて出版し、業務開発や大学の教育、メーカーの新入社員教育等に使い実績を上げている。

本論文では、その方式論の概要について述べる。

## 2. 我々のシステム開発法

我々が行うシステム開発では一般に行われている業務分析や基本設計等の段階を踏んだ手法とはならない。我々の場合は、業務担当者が仕事の合間に自習方式でCOBOLの初級プログラマの力を1~3ヶ月以内につけることから始まる。これが終わると、手作業で使用している帳票類を参考にして、本物の帳票を出力するプログラムの作成に入る。帳票出力プログラムは納得いくまで修正をするが、この繰り返しを通して業務担当者は“エンドユーザ主導によるシステム開発”のシステム開発法を身につけていく。なお、データ入力担当者は、エディタの使用法を覚えたら、入力画面の設計と実際のデータ入力を始めることができる。このような形でOA化を進めるが、その裏付けとなる開発法について述べる。

### 2.1 担当者の選定とその役割

“エンドユーザ主導によるシステム開発”では、システム開発は業務担当者が行うこと、メンバの年齢、学歴、学力、経験等は全く問わないことを基本にしている。これまでの経験では学歴、学力、経験等のない方がむしろ成功率が高い。役割としては次の3種類があるが1人で複数の役割を受け持つこともある。

●リーダー: 我々の開発方式を身につけた経験者。

メンバ1, メンバ2の指導, ファイル設計, 難しいプログラムの作成, サブルーチンの作成等を行う。

●メンバ1(初級プログラマ): コンピュータに関しては未経験者で、なるべく若い人。1~3ヶ月以内にCOBOLの初級プログラマのコースを終えると、直ちに、1つの入力ファイルから帳票を出力するプログラム

の作成を担当する。

●メンバ2(データ入力者): 手作業での仕事をよく理解している人。年配者でもよい。1週間でエディタの使用法に慣れたら、次はエディタを使って入力画面の設計と実際のデータ入力を担当する。

## 2.2 独自のファイル設計<sup>6)</sup>

初心者が実際の業務プログラムを作ったり、データ入力画面を設計したりするためには、ファイル設計のあり方が重要な鍵となる。このため我々は“殆どのプログラムで使用する主ファイル”と“入力画面を作るための原ファイル”の2種類の設計法を決めている。

### (1) 初級プログラマが扱えるファイル(主ファイル)

1つの業務内で主として使用するファイルを主ファイル<sup>19)20)</sup>と呼んでいる。主ファイルは特別な場合を除いて順編成の形をとり、通常的设计ではマッチング処理で取り込むデータ類も含めて1件が1レコードに入るように設計する。この主ファイル1つを入力し、その内容を出力するプログラムであれば初心者でも作ることができる。即ち、初級プログラマは主ファイルから帳票を作成するプログラムが確実に組めれば充分なので、それに合わせた言語教育を行っている。前述した“言語教育とファイル設計を関連づける”とはこのことを言う。我々の設計では、このような簡単なプログラムがプログラム全体の8割以上を占めるので、1~3ヶ月の訓練を終えた初級プログラマの活躍する場は大きい。

### (2) データ入力者が扱えるファイル(原ファイル)

データ入力は、初心者にとって使用法が簡単で、変更が楽にできるようにエディタを使う。入力画面の形式がそのまま業務の原簿になるような設計にするので非常に分かりやすい入力ができ、印刷すればそのまま重要な原簿となる。このファイルを原ファイル<sup>21)</sup>と呼んでいる。図1に職員データの入力画面の例を示す。なお、エディタを使用する際には、エディタの持つプロテクトコマンド(1カラムからあるカラムまでを入力禁止にする機能)を使用すれば、初心者でも安全な入力ができる。図1の例では、「S\*\* \*\*\*\*=>」までを入力禁止と指定してから入力をする。

## 2.3 ファイルごとに基本プログラムを準備

プログラムを簡単に作れるようにファイルごとに基本プログラム<sup>22)</sup>を用意する。図2が基本プログラムの一覧表である。これらの1つをコピーして内容を追加すればプログラムは出来上がる。例えば図2の「ファイルの内容変更」プログラムは、“全員について学年を1年上げる処理”とか“成績が不可のレコードを削除する処理”を行う際に使用する。重要なことは基本プログラム中には、サブルーチンと呼ぶ領域と、デバッグに必須の16進ダンプサブルーチン等があらかじめ組み込まれていることである。

## 3. 我々の教育法

初心者は、コンピュータは難しく、専門家の扱ひものだという先入観を持っている。不安とコンプレックスを持っている業務担当者に、コンピュータは難しいものではないということを実感として理解させ、達成感と自信を持たせ、早い時期に実際のシステム開発の仕事を担当させる教育を行っている。<sup>14)16)17)23)24)</sup>

### 3.1 システム開発に関する一連の訓練書

我々は、教育はシステム開発の一部であるという観点から、言語教育を含めたシステム開発に関する5冊の訓練書を作成し、実際に使用し成果を挙げている。この5冊のタイトル、簡単な内容、そこに含まれる問題数は次の通りである。

#### (1) COBOLによる業務プログラムの作成Ⅰ

内容:初級プログラムの問題と簡単な文法。

問題数:92

本書を終了すれば、初級プログラマとして実際の仕事での帳票出力を担当することができるようになる。ここで言う帳票出力とは、前述した主ファイルを使用しその内容を印刷する簡単なプログラムである。

#### (2) COBOLによる業務プログラムの作成Ⅱ

内容:中・上級プログラムの問題と簡単な文法。

問題数:79

```
S 01 番号 0123 ---*---1---*---2---*
S 02 ふりがな =>お外 知内
S 03 氏名      =>京都 太郎
S 04 性別      =>1
S 05 生年月日 =>19720510
S 07 所属      =>3
S 10 電話番号 =>075-XXX-XXXX
S 15 住所      =>京都市○○区○○
```

図1 職員データ入力画面(職員原ファイル)  
fig.1: A sample of data entry (original file)

- ・ファイルの内容DISPLAY
- ・ファイルの内容変更
- ・ファイルの項目別DISPLAY
- ・ファイルのソート
- ・ファイルのソートと内容DISPLAY
- ・ファイルのまとめ
- ・ファイルのまとめとソート
- ・ファイルのまとめとソートと内容DISPLAY
- ・ファイルのソートと印刷
- ・ファイルの印刷
- ・ファイルのCOBOLでの定義部分
- ・ソートファイルのCOBOLでの定義部分

図2 基本プログラム一覧表 fig.2: The list of program parts

中・上級プログラマとして、(1)ではまだできなかったマッチングを行うプログラムや、処理的に難しいプログラムを作れるようになる。

### (3) ファイル処理入門

内容:COBOLの文法と初級システム設計問題。

問題数:187

プログラマにとって非常に面倒で間違えやすいファイル処理をきちんとこなすこと、ファイル処理と切っても切れない関係にあるシステム設計の初歩を理解することを主目的としている。

### (4) システム設計入門

内容:“誰にでもできるOA化”のシステム開発法。

問題数:107

ファイル設計、データ入力、業務プログラムの作成方法等を覚えることを主目的としている。

### (5) PFD 入門

内容:FACOM 汎用機及び EDITOR (PFD)の使い方。

問題数:39

コンピュータの使用法ではあるが、覚えやすくするために問題集の形にしている。

## 3.2 訓練書を作成するに当たり工夫したこと

5冊の訓練書の作成に当たり最も工夫を凝らしたのが次の2点である。

(1) コンピュータ言語(COBOL)教育はシステム開発の一部として考えないと“エンドユーザ主導によるシステム開発”は実現しない。前述したように、システム設計、特にファイル設計と言語教育には密接な関係を持たせている。

(2) 5冊の訓練書は自習書としても使用できるように解答例つきの問題集の形にしているが、問題作成に当たり注意したのが心理的効果考えた問題の作成である。どのような心理的効果なのかは後述するが、問題は楽しく、少しずつ目先が変わるものであり、それをこなすうちにプログラムの作成やシステム設計に熱中し、自然に実力がついていくような配慮をした。

## 3.3 言語としてCOBOLを使う根拠

第4世代言語が盛んに開発され使用に供され始めている。それなのにいまだにCOBOLを使用する根拠は次の3点である。

(1) 企業によっては過去の膨大なCOBOLプログラムを抱え、それを今でも使用している場合が多い。そのようなところではCOBOLを使いこなす必要がある。

(2) 第4世代言語と言われているソフトでも、内部的にはCOBOLに展開されることが多い。このようなソフトを使用した場合、重度のエラーが発生した際には、COBOLを知らないと解決できないことも出てくる。

(3) どのような言語にしる、それだけを知っていれば事足りるのなら、それを使用するのもよいであろうが、結局はCOBOLも知る必要があるのなら初めからCOBOLだけを徹底的に覚えた方が楽で確実であり、しかも業務システム全体から見ても効率がよい。

上記のような理由から、我々のシステム開発では、コンピュータ言語としてCOBOLを使用している。

## 3.4 “COBOLによる業務プログラムの作成”での教育内容

訓練書の中で最初に学ばせる“COBOLによる業務プログラムの作成”をシステム開発の中での教育の例として取り上げる理由であるが、何事であれ物事を覚えるのに重要な時期は始めてからの3ヶ月ぐらいである。この時期を上手く乗り越えれば後は1人で伸びていく。このためにも初級用の訓練書は、この時期に訓練を受ける側の心理をわきまえた内容及び問題配分であるべきと考えて作成したので、例として最適であると考えている。特にこの訓練書は初級用であるので内容的に4段階に分けており、それぞれの段階で覚えることを決めている。なお、第4段階に入ると、主ファイルを使って、実際の業務で使用する帳票出力プログラムの作成を担当させる。早い段階なら身につくことも、その段階を過ぎたら他のことを覚えるのに精一杯で身につかなくなる。幼児の躰ではないが“時期を逸しないこと”が、言語教育では特に大切である。

### (1) 第1段階: “コンピュータ大好き人間”を誕生させる。

第1段階では、約3週間で25本のプログラムを作成させる。内容としては次のようなものがある。

●DISPLAY 命令を使って端末に文字列を表示するプログラム。(英字、漢字ひらがな、花文字等を表示)

(問題 A0101A) 端末にローマ字氏名IMAI TSUNEOを出力せよ。

(プログラム例)

```
00100 IDENTIFICATION DIVISION.  
00200 PROGRAM-ID. A0101A.  
00300 ENVIRONMENT DIVISION.  
00400 DATA DIVISION.  
00500 PROCEDURE DIVISION.  
00600 DISPLAY "IMAI TSUNEO".  
00700 STOP RUN.
```

(問題 A0101B) (問題 A0101A)のプログラムのシーケンス00600の IMAI TSUNEO を自分のローマ字氏名に変え、端末に自分のローマ字氏名を出力せよ。

図3 DISPLAY命令を使った文字列の出力  
fig.3: Output using display instruction

第1段階では図3、図4のような短いプログラムを、見本を見ながら作ることや、簡単な変更をすることを徹底的に練習させる。これにより“コンピュータを動かす→エディタを使ってプログラムを作る→プログラムを実行する→コンパイルエラーがあれば修正し再度実行する”等々の一連の流れを身体で理解するようになり、その結果、コンピュータを動かしたり、プログラムを作ったりするのは決して難しくないことが分かり“コンピュータ大好き人間”が誕生する。なお、第1段階ではコンピュータやCOBOL言語に関する説明は一切せず、マニュアル等も一切読ませないようにする。この段階では、コンピュータの操作やプログラムの作成方法を覚えるのに精一杯で説明は全く理解できず、嫌気を起こさせるだけだからである。

## (2) 第2段階：“コンピュータは大馬鹿だ!”を理解させる。

約3週間で13本のプログラムを作成させる。内容としては次のようなものがある。

- コンピュータの日付を取り込み、DISPLAY命令で表示するプログラム。
- ACCEPT、DISPLAY命令を使ってコンピュータとのやりとりを行うプログラム。(例えば電卓、占い等)
- DISPLAY命令を使って自分のローマ字氏名を10行表示するプログラムを5種類の書き方で作らせる。第2段階での特徴としては次のようなことがある。
- 第1段階で多くのプログラムを作り、散々エラーを出したり直したりした結果、コンピュータは1とI、0とOの区別もできない“人間には遙かに及ばない馬鹿な機械である”“だからプログラマは注意深くなる

必要がある”ことを理解するようになる。

- プログラムの作成等に馴れて余裕が出る結果、訓練を受けている者から「これはどういう意味か?」等の文法上の質問が自然に出るようになる。このように、自分から覚えたいという気持ちにさせることは非常に大切であると考える。
- “コンピュータとのやりとり”のプログラムを作ることにより、業務で重要なデータ入力のイメージをつかむようになる。
- 簡単なプログラムでも多くの書き方があることを理解するようになる。
- ピリオドのつけ忘れ。不要箇所にピリオドをつける等をしたら、コンパイルエラーだけではなく、論理エラーも出るような条件文を含むプログラムを数多く作らせ、論理エラーの意味を分からせる。図5がその例である。このプログラムのシーケンス1200行の最後に間違っピリオドをつけてしまうと自分のローマ字氏名を入力した時にコントロールはシーケンス1300行に移ってしまい、“ZIBUN”と出力して終わるべきなのに何も出力しないでもう一度①に戻ってしまう。即ち論理エラーを起こすことになる。このようなプログラムを作らせることにより、コンパイルエラーと論理エラーの違いを身体で理解できるようになる。なお、この段階では条件文等はまだ教えない。

## (3) 第3段階：ファイル処理プログラムの基礎を覚えさせる。

約1週間で5本のプログラムを作成する。内容としては次のようなものがある。第2段階で作成したプログラムを、ファイル処理命令に変える。第3段階では、これまでに作成したDISPLAY、ACCEPT命令を使ったプログラムを、OPEN、READ、WRITE、CLOSE命令を使ったファイル処理プログラムに変えることで少しずつファイル処理のプロ

(問題 A0120A) 端末にS.Hの花文字を出力せよ。  
(プログラム例)

```
00100 IDENTIFICATION DIVISION.
00200 PROGRAM-ID. A0120A.
00300 ENVIRONMENT DIVISION.
00400 DATA DIVISION.
00500 PROCEDURE DIVISION.
00600 DISPLAY "SSS H H".
00700 DISPLAY "S S H H".
00800 DISPLAY "S H H".
00900 DISPLAY "SSS HHH".
01000 DISPLAY " S H H".
01100 DISPLAY "S S H H".
01200 DISPLAY "SSS * H H".
01300 STOP RUN.
```

(問題 A0120B) 端末に自分のローマ字氏名の名前の1  
字目と姓の1字目を(問題 A0120A)のような花文字で出力  
せよ。(IMAI TSUNEOならI.T)

(問題 A0204) 下記のプログラムを作成せよ。

- ・端末に“NAME O INPUT SEYO”と出力する。
- ・端末からローマ字氏名を入力する。
- ・その氏名が自分の氏名以外の場合には、“TANIN”と出力しYOMUに戻る。
- ・その氏名が自分の氏名の場合には“ZIBUN”と出力し終了する。

(プログラム例)

```
00100 IDENTIFICATION DIVISION.
00200 PROGRAM-ID. A0204.
00300 ENVIRONMENT DIVISION.
00400 DATA DIVISION.
00500 WORKING-STORAGE SECTION.
00600 01 W-NAME PIC X(40).
00700 PROCEDURE DIVISION.
00800 YOMU.
00900 DISPLAY "NAME O INPUT SEYO".      ①
01000 ACCEPT W-NAME.                    ②
01100 IF W-NAME NOT = "IMAI TSUNEO"    ③
01200 DISPLAY "TANIN"
01300 GO TO YOMU.
01400 DISPLAY "ZIBUN".                  ④
01500 STOP RUN.
```

(註:1200行のIMAI TSUNEOを自分のローマ字氏名に  
変えよ。)

図4 DISPLAY命令を使った花文字の出力  
fig.4: Output a figure of characters

図5 “自分”か“他人”か fig.5:Is this your name or not?

グラムに馴らしていく。図6がそのプログラム例である。このプログラムは第2段階で作った図5のプログラムをファイル処理に変えたものである。ファイル処理プログラムでは、レコード長やブロック数、項目の位置、項目のタイプ等々を間違えないように記載する必要があるので、初心者が戸惑うことが多い。これらをはっきり理解させるためには、ファイル処理プログラムに少しずつ馴らしていくことが大切である。いきなり“ディスクファイルを入力し、帳票出力を行う”というような問題に入らず、図6のように前に作ったプログラムの出力だけをファイル扱いにすることから始めると混乱を招かない。この時期にファイル処理の記述法を徹底的に身につけておかないと、ベテランと言われるようになってからでも基礎的なエラーを起こすことが往々にしてある。

(4) 第4段階：帳票出力プログラムを数多く作らせる。

約5週間で49本のプログラムを作成させる。内容としては次のようなものがある。

●ファイル処理命令 (OPEN, READ, WRITE, CLOSE及びSORT) を使って入力、出力を行うプログラム。(特定データの出力、改行と

改頁、複数の出力形式、統計表、編集、項目のレベル分け等)

第4段階では、初級プログラムの養成の最終段階であるので、色々なファイルを使って色々な処理を含むプログラムを作らせる。図7は項目のレベル分けのプログラム例である。このプログラムは“学生マスタファイルを入力し、学生番号の2桁目が“3”で4桁目が“1”の学生の学生番号と氏名を出力する”ものである。このようなプログラムをベテランプログラマに作らせるとテーブル処理を使った難しい書き方となることが多いが、レベル分けを使えば、初心者でも作れる簡単なプログラムとなる。なるべく簡単な書き方で、エラーを起こさないプログラムを作成するための訓練である。

なお、第1～第3段階の訓練を経てこの第4段階に入ると、ファイル処理プログラムを確実に作れるようになるので、開発のメンバとして“主ファイルからの帳票出力プログラムの作成”を完全に任せることができるようになる。メーカーや企業等で行うCOBOL言語講習会では、通常この第4段階から開始することが多い。このようなやり方では、業務担当者のようなコンピュータに関しての素人には理解できないであろう。

```
(問題 A0403A) (問題 A0120A) で作成したプログラムの
出力をファイル扱いに直し,S.Hの花文字を出力せよ。
00100 IDENTIFICATION DIVISION.
00200 PROGRAM-ID. A0403.
00300 ENVIRONMENT DIVISION.
00400 INPUT-OUTPUT SECTION.
00500 FILE-CONTROL.
00600 SELECT O-FILE ASSIGN TO OOO.
00700 DATA DIVISION.
00800 FILE SECTION.
00900 FD O-FILE
01000 BLOCK CONTAINS 40 RECORDS
01100 DATA RECORD O-REC.
01200 01 O-REC.
01300 02 FILLER PIC X(30).
01400 PROCEDURE DIVISION.
01500 OPEN OUTPUT O-FILE.
01600 MOVE "SSS H H" TO O-REC.
01700 WRITE O-REC.
01800 MOVE "S S H H" TO O-REC.
01900 WRITE O-REC.
02000 MOVE "S H H" TO O-REC.
02100 WRITE O-REC.
02200 MOVE "SSS HHH" TO O-REC.
02300 WRITE O-REC.
02400 MOVE " S H H" TO O-REC.
02500 WRITE O-REC.
02600 MOVE "S S H H" TO O-REC.
02700 WRITE O-REC.
02800 MOVE "SSS * H H" TO O-REC.
02900 WRITE O-REC.
03000 CLOSE O-FILE.
03100 STOP RUN.
```

図6 ファイル扱いで花文字を出力する fig.6: Output characters to the file

```
(問題 A0612) 学生マスタファイルを入力し,学生番号の2桁目
が“3”4桁目が“1”の学生の,学生番号と氏名を出力せよ。
(プログラム例)
00100 IDENTIFICATION DIVISION.
00200 PROGRAM-ID. A0612.
00300 ENVIRONMENT DIVISION.
00400 INPUT-OUTPUT SECTION.
00500 FILE-CONTROL.
00600 SELECT I-FILE ASSIGN TO III.
00700 DATA DIVISION.
00800 FILE SECTION.
00900 FD I-FILE
01000 BLOCK CONTAINS 40 RECORDS
01100 DATA RECORD I-REC.
01200 01 I-REC.
01300 02 I-GNO.
01400 03 I-GNO1 PIC X(1).
01500 03 I-GNO2 PIC X(1).←2桁目
01600 03 I-GNO3 PIC X(1).
01700 03 I-GNO4 PIC X(1).←4桁目
01800 02 FILLER PIC X(30).
01900 02 FILLER PIC X(30).
02000 .....
02100 PROCEDURE DIVISION.
02200 OPEN INPUT I-FILE.
02300 OPEN OUTPUT O-FILE.
02400 YOMU.
02500 READ I-FILE AT END GO TO OWARI.
02600 IF I-GNO2 NOT = "3" GO TO YOMU.
02700 IF I-GNO4 NOT = "1" GO TO YOMU.
02800*
02900* 2桁目が“3”4桁目が“1”の処理
03000* .....
```

図7 項目のレベル分け fig.7: Sorting by data items

#### 4. 適用事例

“エンドユーザ主導によるシステム開発”の初級プログラマ養成教育を受けただけで実際のOA化に参画し、殆ど1人でシステムの維持管理を行っている事例を紹介する。

関東地区のある大学では、1期生として卒業し、大学の事務職員として採用された女子1名が担当者となり、筆者の良きパートナーのリーダーのもとで教務事務のシステム化を行った。その後の維持管理はその女子職員が担当している。2年前には大きなカリキュラムの変更があったが、彼女が1人で約3ヶ月をかけて事務の仕事のかたわら変更を行った。彼女が学生時代に受けたシステム化への訓練は次の通りである。

1年前期: 授業として“COBOLによる業務プログラムの作成 I”及び“ファイル処理入門”の2冊の訓練書を終了した。

2年前期と後期: 卒業後は母校に就職することが決まったので、各種業務のシステム化の手伝いとして帳票出力プログラムの作成や資料の作成を行った。

2年後期: ゼミ活動として図書館の洋書目録のデータ入力を行った。これを通してデータ入力法を覚えた。このように、上記の2冊の訓練書を終了すれば、我々のシステム開発法での重要なメンバとなり得る。

#### 5. おわりに

ソフトウェア業界では、高い人件費と、従来の開発方法による生産性では、ハードウェアの価格破壊に追従できないことは明らかである。その理由の1つとしてSEが作る難しいシステムをSE自身がメンテナンスを行い、素人ユーザには手が出ないことがある。このような状況から、これまでの外注方式とは全く発想の違うシステム開発の方式が求められている。そうした新たな環境の中で、我々の“エンドユーザ主導によるシステム開発”の方式は普通に行われているシステム開発法とは違い、自分たちで開発できることや、生産性が高いこと等から、これからの開発方式として広まっていくことを確信している。

これからのメーカ等の主な役割は、ユーザが自分自身でシステム化を行い、それを維持管理できるような教育を含むシステム開発法を提供することであると考えている。

本論文で主張した内容は、一世代前のメインフレームで、COBOL言語を中心としたシステム開発であるが、情報処理のハードウェア、ソフトウェア、様々なツールがどのように開発され、進歩しても、その時代で形を変えて生きる考え方、思想である。システム開発を進めるに当たってどのような推進体制を組むかに関してある意味で挑戦的な方式である。現在のシステム開発の一般的な契約形態では実現は極めて困難であり、それは、メーカ、ユーザの両者にこのような考え方を持つ人材が必須である。

#### 6. 謝辞

本論文の内容は、筆者今井恒雄と、東京家政学院筑波女子大学の平木しげ子元教授との長年にわたる共同研究の成果を論文と

したものである。筆者は富士通株式会社のシステムエンジニアの立場であり、平木しげ子元教授は、京都産業大学、東京家政学院筑波女子大学で教授を務めたが、両大学の教務システムの開発も担当した。論文をまとめるに当たり、平木元教授ならびに、システム開発でお世話になった多くの方々に感謝の意を表します。

#### 参考文献

[著作]1)～5) はすべて恒星社厚生閣より出版

- 1) 平木: COBOLによる業務プログラムの作成 (1989)
- 2) 平木: COBOLによる業務プログラムの作成 I (1989)
- 3) 平木, 今井, 荒木: ファイル処理入門 (1990) II
- 4) 平木, 今井, 荒木: システム設計入門 (1993)
- 5) 今井, 平木, 荒木: PFD入門 (1990)

[情報処理学会全国大会] 発表者: 今井, 他

- 6) 第33回 プログラム開発をソロバン並にするファイル設計とその応用 2F-5,P607
- 7) 第34回 データエントリプログラムの自動作成について 2T-6,P1103
- 8) 第35回 順編成ファイルを対象にしたランダム処理サービス 7E-6,P2739
- 9) 第36回 システム開発の基礎教育について 6K-7,P2405
- 10) 第37回 システム開発の基礎教育について (その2) - ユーザプログラム作成の基本- 4S-8
- 11) 第38回 システム開発の基礎教育について (その3) - COBOL言語教育と利用環境-
- 12) 第38回 システム開発の基礎教育について (その4) - COBOLで起こしやすいエラーとデバッグ手法-
- 13) 第39回 システム開発の基礎教育について (その5) - 全員の素人を対象にしたCOBOLセミナー-
- 14) 第40回 システム開発の基礎教育について (その6) - 積み重ねと繰返しによるCOBOLの修得- 3K-2,P351
- 15) 第41回 システム開発の基礎教育について (その7) - システム開発の考え方- 1C-6,P1-11
- 16) 第41回 システム開発の基礎教育について (その8) - プログラムの即成訓練- 1C-7,P1-13
- 17) 第42回 システム開発の基礎教育について (その9) - 開発スタッフに入る前の準備- 1B-3,P1-5
- 18) 第43回 ソフトウェア設計の基本 (その1) - データエントリ- 3N-1,P5-63
- 19) 第44回 ソフトウェア設計の基本 (その2) - 現場担当者主導のシステム開発- 4J-1,P5-211
- 20) 第44回 ソフトウェア設計の基本 (その3) - 学生が行なった図書館業務のOA化- 4J-2,P5-213
- 21) 第45回 ソフトウェア設計の基本 (その4) - ファイル設計- 6S-04,P5-165
- 22) 第46回 ソフトウェア設計の基本 (その5) - 部品の考え方- 3J-01,P5-227
- 23) 第47回 誰にでもできないOA化から誰にでもできるOA化へ - 何故誰にでもできないのか- 3J-6,P5-213
- 24) 第48回 誰にでもできないOA化から誰にでもできるOA化へ (その2) - OA化の考え方とその教育- 4K-4,P5-369
- 25) 第48回 誰にでもできないOA化から誰にでもできるOA化へ (その3) - 大企業デモOA化教育- 4K-5,P5-371
- 26) 第49回 誰にでもできないOA化から誰にでもできるOA化へ (その4) - 誰にでもできるための訓練法- 1M-1,P5-99

# A system development method led by the end users

Tsuneo IMAI

Professor, The Kyoto College of Graduate Studies for Informatics

We would like to describe the method of an office automation system which can be developed by the personnel in charge of a particular project. Our system development method begins by instructing to the staff member in charge to self-study and finishes the assignments of basic COBOL programs and to improve his/her programming ability and knowledge in the intervals of work within three months.

After completing the first stage assignments, he/she begins to make programs to output the report results, by using as a model the documents which have been produced by hand. He/she corrects and improves the program until it is absolutely satisfactory.

Parallel to the programming and improving process, the office automation will be improved smoothly and naturally. We have paid attention to achieve this development method and made researches in the education of the COBOL language and file design, and the relationship between these two areas. The system development method based on this idea has been written into five training textbooks.

Since the publication, these books have been widely read and successfully used by businesses to develop actual business systems and universities to educate students.

今井 恒雄 Tsuneo Imai

---

京都大学工学部卒，同大学院工学研究科修士課程修了（数理工学専攻）。工学修士。富士通株式会社システム本部主席部長，株式会社富士通ラーニングメディア取締役を経て，現在，京都情報大学院大学教授，日本e-learning学会副会長。