上級情報処理技術者育成指針

各　　論

第4部　情報システムの開発

財団法人 日本情報処理開発協会
情報処理研修センター

008294

I
36
~~2—B4~~ 5

ま　え　が　き

当財団では，昭和43年以来，６年にわたり，情報処理育成事業の一環として，初級，中級，上級の「情報処理技術者育成指針」（ただし，上級のみ標題は「情報処理技術者研修ガイドブック」となっている）を作成してきた。このうち上級情報処理技術者育成のための指針については，情報処理技術の著しい進展により，全般的な見直しが必要となってきた。

このため，昭和51年度から３カ年計画で，これまでに作成した初級，中級，両指針との関連を考慮し，同時に，現在および今後数年間に想定される環境にも調和できるようにとの考え方にもとづいて，新しく「上級情報処理技術者育成指針」（以下「本育成指針」とする）の開発を進めてきたが，これが完成のはこびとなった。

本育成指針は，個々の企業内や，一般の企業人向け教育機関などにおいて，さまざまな形で行なわれている上級情報処理技術者の教育に対して，標準的なカリキュラムを示し，情報処理技術者の養成を効果的に推進することを，そのねらいとしている。

本育成指針の内容は，「総論」および，分野別に編集された各論，すなわち「組織システムの分析」，「システム開発運用の背景」，「コンピュータおよび情報処理技術」，「情報システムの開発」の，合計５分冊から構成されている。この利用に当っては，まず総論を参照し，必要に応じて各論の関連部分を参照されるよう希望したい。

なお，この事業実施にご尽力いただいた関係各位に心から感謝の意を表わすとともに，この指針が各方面で利用され，情報処理技術者育成の一助となることを念願する次第である。

昭和 54 年３月

財団法人　日本情報処理開発協会
情報処理研修センター

所　　長　吉野元之助

上級情報処理技術者育成指針委員会

（敬称略，50音順）

委員長	松田武彦	東京工業大学
委員	魚木五夫	広島修道大学
〃	江村潤朗	日本アイ・ビー・エム株式会社
〃	岡本行二	東京芝浦電気株式会社
〃	小佐井純正	株式会社三菱銀行
〃	台　文彦	新日本製鉄株式会社
〃	土居範久	慶応義塾大学
〃	中村　薫	通商産業省
〃	西川禎一	通商産業省
〃	西村恕彦	東京農工大学
〃	服部幸英	株式会社東洋情報システム
〃	原田睦明	株式会社ソーシャル・サイエンス・ラボラトリー
〃	前川良博	横浜商科大学
〃	間野浩太郎	青山学院大学
〃	三浦大亮	東レ株式会社
〃	吉野元之助	(財)日本情報処理開発協会

上級情報処理技術者育成指針作成小委員会

（敬称略，50音順）

委員長	魚木五夫	広島修道大学
委員	岡本行二	東京芝浦電気株式会社
〃	小佐井純正	株式会社三菱銀行
〃	土居範久	慶応義塾大学
〃	服部幸英	株式会社東洋情報システム
〃	三浦大亮	東レ株式会社

序

本書は，上級情報処理技術者育成指針（以下「本育成指針」という）のうち，「情報システムの開発」に関連した分野の教育に有用と考えられる資料類を，整理・編集したものである。本書の内容は，次の４つの単元に分かれている。

単元Ｄ１：情報分析

単元Ｄ２：システム設計

単元Ｄ３：開発プロジェクト

単元Ｄ４：コンピュータ部門の運営管理

これらの単元は，教育の計画や実施などにおける一つの作業単位であり，本書ではこれをさらに幾つかの章に分けて解説している。

上級情報処理技術者の育成を，実際に計画し，また実施しようとする場合には，本育成指針のうち「総論」をまず参照されたい。総論は，本育成指針利用のための案内書であり，技術者育成の体系的な計画，教育の目標，それに至るための教育内容や教育方法，などに関して方向づけを与えるものである。

本書は，下記の方々の執筆または協力と，本育成指針作成小委員会の監修を得て作成されたものである。

（敬称略，50音順）

主査　服部　幸英（株式会社東洋情報システム）

委員　鶏内成太郎（野村コンピュータシステム株式会社）

〃　須賀　雅夫（株式会社三菱総合研究所）

〃　台　　文彦（新日本製鉄株式会社）

第 4 部

情報システムの開発

総 目 次

まえがき

序

単元D1　情報分析

1章　システム・ライフサイクル概論 …… 1

2章　システム・ライフサイクル管理 …… 16

3章　基本的な分析手法 …… 35

4章　システム代替案の検討 …… 53

5章　システムの経済性評価 …… 67

6章　システム仕様書 …… 94

単元D2　システム設計

1章　基本的な設計手法 …… 101

2章　システム資源の選択 …… 123

3章　ソフトウェアの設計 …… 143

4章　データベース …… 170

5章　プログラムの作成と検査 …… 196

6章　システムの評価と改善 …… 217

単元D3　開発プロジェクト

1章　テーマの選択 …… 231

2章　チームの編成 …… 242

3章　実習の進め方と教師の役割 …… 245

4章　開発の実習 …… 251

5章　報告書作成と発表 …… 306

単元Ｄ４　コンピュータ部門の運営管理
１章　コンピュータ部門の組織 …… 311
２章　コンピュータ部門の要員管理 …… 340
３章　システム開発管理 …… 359
４章　コンピュータの導入 …… 379
５章　コンピュータ室の運用管理 …… 397
６章　情報処理サービス業の利用 …… 425

単元Ｄ１

情　報　分　析

単元Ｄ１

情　報　分　析

目　　次

第１章　システム・ライフサイクル概論 …… 1

1.1　システム・ライフサイクル …… 1

1.2　システム・ライフサイクルの各段階 …… 4

1.3　システム・ライフサイクルとシステム分析 …… 6

第２章　システム・ライフサイクル管理 …… 16

2.1　プロジェクト管理 …… 17

2.2　プロジェクトの計画と管理 …… 20

2.3　プロジェクト管理と文書 …… 23

2.4　プロジェクト・チーム …… 26

第３章　基本的な分析手法 …… 35

3.1　システム分析 …… 35

3.2　システム分析の要素 …… 37

3.3　各種の分析手法 …… 39

第４章　システム代替案の検討 …… 53

4.1　システムの目的と代替案 …… 53

4.2　代替案の開発と選択 …… 54

4.3　代替案の類型 …… 55

4.4　ＰＰＢＳとシステム分析 …… 56

第５章　システムの経済性評価 …… 67

5.1　情報の価値 …… 67

5.2　システムの評価基準 …… 69

5.3　システムの効果 …… 74

5.4　情報システムの費用 …… 80

5.5　システムの費用効果分析 …… 83

第6章　システムの仕様書 …… 94

6.1　システム仕様書 …… 94

6.2　ロジカル設計とフィジカル設計の機能分担について …… 96

6.3　要求仕様技術 …… 97

第1章　システムライフサイクル概論

キーワード

システム・ライフサイクル（system life cycle），情報分析（information analysis），システム設計（system design），ロジカル設計，機能設計（logical system design），フィジカル設計，機構設計（physical system design），システムズ・アプローチ（systems approach），システム分析（systems analysis）

目　　標

本章は，システム開発の学習の全体的骨格を把握するための導入部である。

システム開発は，一貫性のある全体像でとらえられるべきではあるが，学習の便宜上，これをその過程にそって，前半と後半に分けて，Ｄ１とＤ２で学ぶ。即ち前半では情報分析（ロジカル設計）に重点をおき，後半ではシステム設計（フィジカル設計）に重点をおく。したがってこの章では，これらの学習の全体の見通しをまずはじめにつけておくことを主眼とする。

システム開発のアプローチには，システム分析という考え方が出発点となる。これについては，既にＡ１及びＡ３などで学習しているが，ここではそれらの知識が，システム開発の学習にどのように結びつくものであるかを再び説明し，考え方の足場をはっきりさせる。そのためには，システム開発の過程が，システム・ライフサイクルという概念に，まとめられるようになった由来などを説明して，システム分析の考え方の大略を理解させるのも一方法である。

分析的アプローチをシステム開発の過程に適用したものが，システム・ライフサイクルの概念である。この章ではシステム・ライフサイクルの概念の大要を説明する。Ｄ１，Ｄ２の章・節の組立が，システム・ライフサイクルに準拠して構成されていることを指摘して，学習体系の概要を，あらかじめ学習者に把握させることが，本章の主目標である。

システム開発費用の増大などの動向との関連で，システム・ライフサイクルの概念が最近重視されつつある理由にふれることも必要である。

内　　容

1.1　システム・ライフサイクル

システム・ライフサイクルとは，システム開発作業を全体として把握する概念である。単元Ｄ１及びＤ２にわたって，システム開発について学ぶが，そのための各章節の分割は便宜的に

システム・ライフサイクルの段階に準拠している。第1章ではその導入部として，システム・ライフサイクルの概念を理解させる。

システム開発の過程には，まずその要望を感知し，新たにシステムを開発することの可否やその可能性を検討し，さらに具体化の作業を進め，その作業の完了に続いてシステムのリプレイスをおこなうという一連の作業の段階があるが，この開発作業を全体として把握する概念が，システム・ライフサイクルである。この考え方は，米国におけるシステム開発の経験と研究が蓄積されて，整理されたものである。

この考え方の中心になる思想の一つはヨーロッパ社会の近代化の基底にある解析原理である。即ち複雑な現象も細分化すれば単純な法則が見出されるとする思想である。システム開発の出発点として，システム分析を重視することは，このためである。さらにもう一つとして，目的原理がある。即ちシステム設計をおこなうには，はじめにその意図と目的を明らかにして，開発に着手する前に，環境や技術に対する予測を十分におこない，必要な機能の設計と，それらの相互の関係を構造化し，これを共通の目的にむけて結合してゆくという方向づけである。

前者は帰納的アプローチであり，後者は演繹的アプローチである。単元D1では前者に重点をおき，単元D2では後者に重点をおいて学習する。

システム分析とは，要約していえば，とらえどころのない現象を把握するには，まずはじめにその現象を，詳細に具体的に記述することから出発しようとする態度である。

システム設計の考え方の中には，産業革命以来，ヨーロッパの生産社会でつちかわれた標準化の思想が重要な位置をしめている。

以上二つの重要な思想傾向，即ち解析原理と目的原理に焦点をあててシステム開発の全体像を把握させる。

まずはじめに，カリキュラムの基底になっているシステム・ライフサイクルという概念を理解させる。この概念を解説しているものは，参考文献にあるように，各種のものがあるが，その要点をあげれば，次のようになる。

- システム設計の段階分け
- 中心円をなすシステム・ライフサイクル概念による各作業ステップの統制
- システム・ライフサイクル概念を中心として動く衛星群としての各作業ステップとその相互関係
- 最終ステップ（実施と運用）から再び第一ステップ（問題の提示とニーズの把握）への回帰

システム・ライフサイクル概念の理解をたすける概念図の一例として Rubin のものを，図1−1に示す。

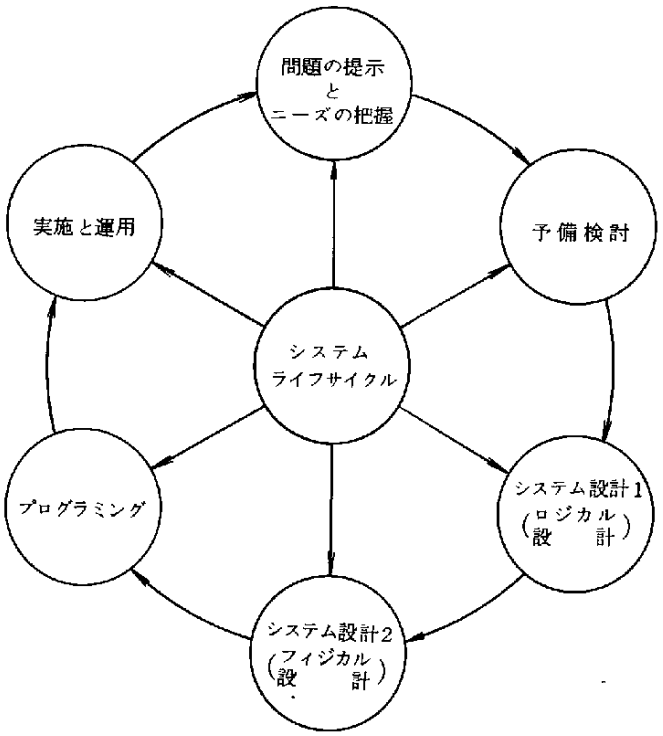


図1－1　Rubinのシステム・ライフサイクルの概念図

システム・ライフサイクルの説明については，単なる開発ステップ論ではなく，この概念が必要とされるようになった背景についても説明する。次にあげるような事項である。

・システム開発コストの増大

ハードウエア・コストに対して，システム開発コストを含めたソフトウエア関連コストが増大の一途をたどっている事実。

・システム開発コストに対する資金制約

コンピュータ関連費用のうち，新規開発に向けられる資金は限られていること。既存システムの操業，メインテナンス関連費用は削減困難なこと。

・長期間多人数のシステム開発要員を作業に拘束する必要のあること。

・異質のメンバーの協同作業であるため，その間の情報交換や管理統制が困難になること。

・以上の諸点からシステム開発のリスクが増大していること。

これらの状況を説明するために使われる図表として次のようなものがある。

(Basili, R. and Baker, T. の Structured Programming Tutorial の図表に準拠)

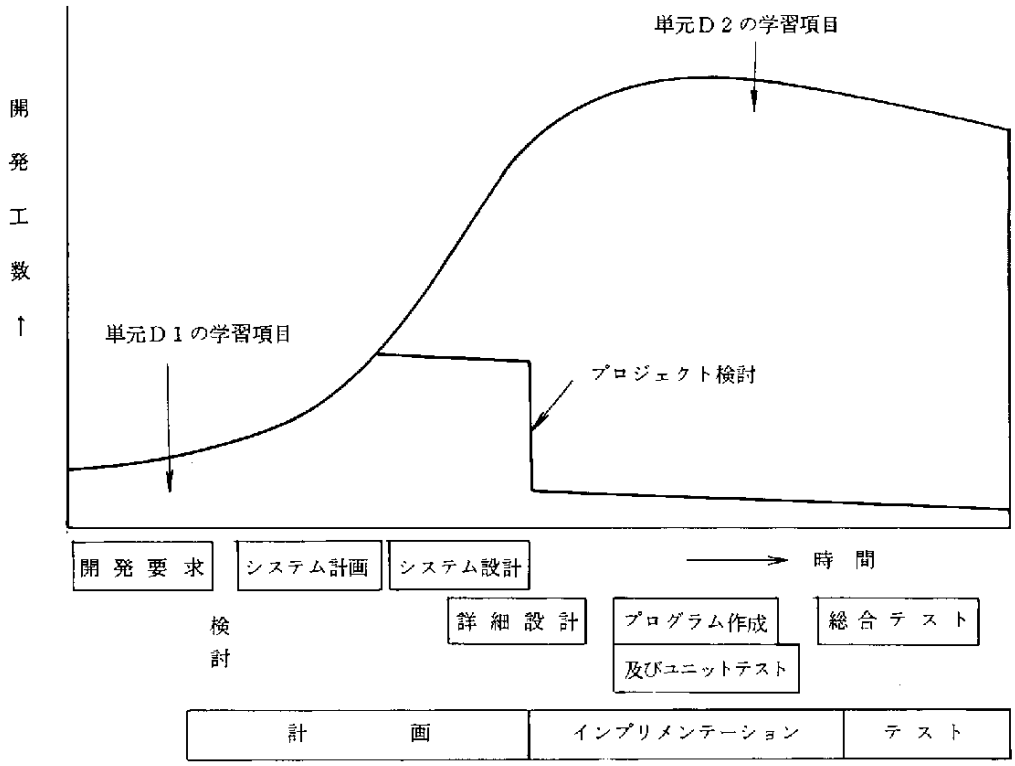


図１－２　システム開発工数の内訳

1.2　システム・ライフサイクルの各段階

システム・ライフサイクルの段階分けについては，各種の説があるが，前掲の図１－１によって，そのあらましを述べれば次のようになる。

(1)　問題の提示とニーズの把握

システムズ・アプローチにおける，ニーズの明確化の重要性を説明する。問題の提示はコンピュータ部門外から行なわれる場合と，コンピュータ部門内から発生する場合があること，提示された問題解決には，新規システム開発に待たねばならない場合と既存システムの改善で解決出来る場合があることなど，実務に則して説明する。問題が提示された時，本質的な問題が隠されている場合もあり，又利害相反するユーザ間の調整を待たねばならない場合もある。出来れば実例をもって説明するのが良い。

(2)　予　備　検　討

予備検討は，システム化の可能性や実用性を探るための予備的検討である。この段階が承認されると具体的なシステム設計に入る。この段階でシステム上の欠陥が指摘されずに，以下の工程に入ってしまうとリスクは極めて大きい。

予備検討の目的は以下の点であることを説明する。

① 解決すべき問題点の摘出

② 摘出した問題をどう解決するか

③ システムの可能性や実用性に照して，コンピュータ化が適切か否かの検討

④ ユーザに対するプロポーザルの提示

予備検討の内容としては，

① 現行システムの現状と特徴

② 新規又は改善システムの構想

③ 実施上の問題点

④ 新システムの開発実施計画

であることを説明する。

(3) ロジカル設計（機能設計）

予備検討が承認されると機能設計に入る。機能設計においては，

・提案された解決案を，どのようにコンピュータシステム上に機能設計すべきか。

・次工程のフィジカル設計（機構設計）に耐えうるシステム仕様書の作成

が目的となる。

ロジカル設計（機能設計）とフィジカル設計（機構設計）との区別は，前者が主としてシステム化の対象との関連におけるシステムが持つべき機能の設計であるのに対し，後者はコンピュータ・システムとの関連におけるハードウエア，ソフトウエア等の機構の設計である。

(4) フィジカル設計（機構設計）

システム仕様書をうけて，コンピュータ処理のためのハードウエア，ソフトウエア面での機構設計に入る。フィジカル設計の主たる業務は

・ハードウエア，ソフトウエアの選択と評価

・入出力の設計

・ファイルの設計

・ソフトウェアの設計

である。

(5) プログラミング

フィジカル設計をうけて，プログラム開発に入る。主たる業務は，

・システム設計の，プログラム開発の側面より見たチェックと内容の確認

・プログラム・ロジックの開発

・プログラム作成

・プログラム・テストとプログラムの手直し

である。

(6) 実施と運用

主たる業務は以下の通りである。

・システムの最終テスト

・システム・インストール

・システム切換え

・ユーザ・オペレータ訓練

・処理の開始

・運用

・運用後の評価と修正

以上で1サイクルが完結する。経営環境の変化，状況の変化，システムの陳腐化などの要因で，新たなニーズが発生すると，システムも再び（1）からの新しいサイクルから検討されることになる。

1.3 システム・ライフサイクルとシステム分析

システム分析は，現代社会の複雑な問題を解決するための体系的な方法である。その内容は，システムの目的を的確に記述し，複数の解決案を目的に照合して比較検討し，最適な案を選択するという方法である。この考え方は，米国のThe RAND Corporationなどの思考グループによって研究され整備されたものであって，システム・ライフサイクル概念の成立の過程でも，その考え方の源流となって，システム開発のための方法の一部として吸収されている。

分析的な考え方はサイクルのフェーズ分けの中にも，開発のための方法の中にも含まれている。RAND, SDC, GEのTEMPOセンターなどによってなされたシステム分析の方法論の整備と，システム・ライフサイクル概念の成立の経過を歴史的にふり返って見ることによって，システム開発に必要な考え方の基礎事項を認識させることができる。

システム・ライフサイクル概念の特徴は，かなり明示的にシステムズ・アプローチを導入していることである。以下の各章の進め方のベースになる考え方であるので，その基本を理解させる。

システムズ・アプローチの特徴を以下の点で説明する。説明にあたってはクェードの定義が参考になる。

・可能な限り分析的な概念の枠組を利用する。

・目的と代替案を探求する。

・適切なフレームワークを使い，予想される結果と比較分析する。

・行動選択を援助する体系的なアプローチである。

又，分析する場合の要素としては，

・目的

・代替案

・費用

・モデル

・評価基準

を上げている。模式化して図示すると図１－３になる。

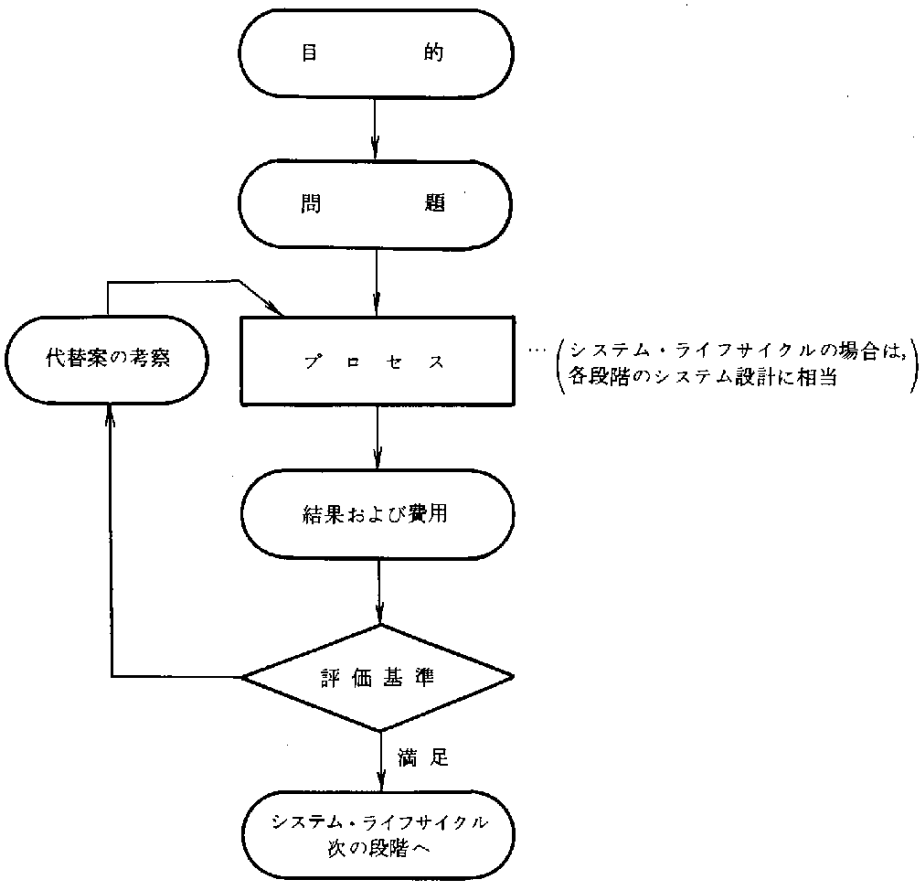


図１－３　システム・ライフサイクルにおけるシステムズ・アプローチ

指導上の留意点

(1) 単元Ｄ１と単元Ｄ２の構成を全体としてあらかじめ理解させることに焦点をあてる。そ

のための下敷がシステム・ライフサイクルの概念である。

(2) システム・ライフサイクルの段階分けの議論を詳しくする必要はない。むしろ学習のポイントは，

情報分析 → ロジカル設計

システム設計 → フィジカル設計

と対比される分析と設計の性格の差をはっきりと認識させ，単元Ｄ１では前者に重点があり，単元Ｄ２では後者に重点があることなどを示す程度でよい。

(3) 第１章は単元Ｄ１の序章としての位置づけでもあるので，これから学ぶことのうちの重要な項目について概説する。例えばシステム開発過程の各段階の特徴・システム分析の必要性などの要点を説明する。

(4) システム・ライフサイクル概念とかシステム分析のような考え方を理解させるには，それを完成された定義とか規範として説明するよりも，歴史的な成立過程をたどって，次第に形をなしてきた状況とともに説明する方が望ましい。

参　　考

1．システム・ライフサイクルの段階分けについて

(1) Rubinの段階分け

- コンセプション
- プレリミナリ分析
- システム設計
- システム・プログラム
- システムの文書化
- システム設置
- システム運用
- システムの終結

(2) Benjaminの段階分け

- フィージビリティ・スタディ
- システム・スペシフィケーション
- システム・エンジニアリング
- システム開発
- インプリメンテーション
- オペレーション

(3) 通産省情報処理技術者用の段階分け

・問題の提示とニーズの提示

・対象システムの目的の解明

問題の認識，確定

実情の把握

・新システムの基本設計

電子計算の役割り

基本的手段の設定

システム内での機械化分担の決定

・EDP システム設計書の作成

入出力の決定

ファイルの設計等

他システムとのインターフェイス

・プログラムのモジュール化

プログラム間のインターフェイス

入出力の設計

ファイルの確定

・電子計算機処理のためのプログラム仕様の決定（プログラム設計書の作成）

・プログラム設計書に基づくプログラミング

フローチャート及びコーディング

その他の文書化

・プログラムのデバッグ

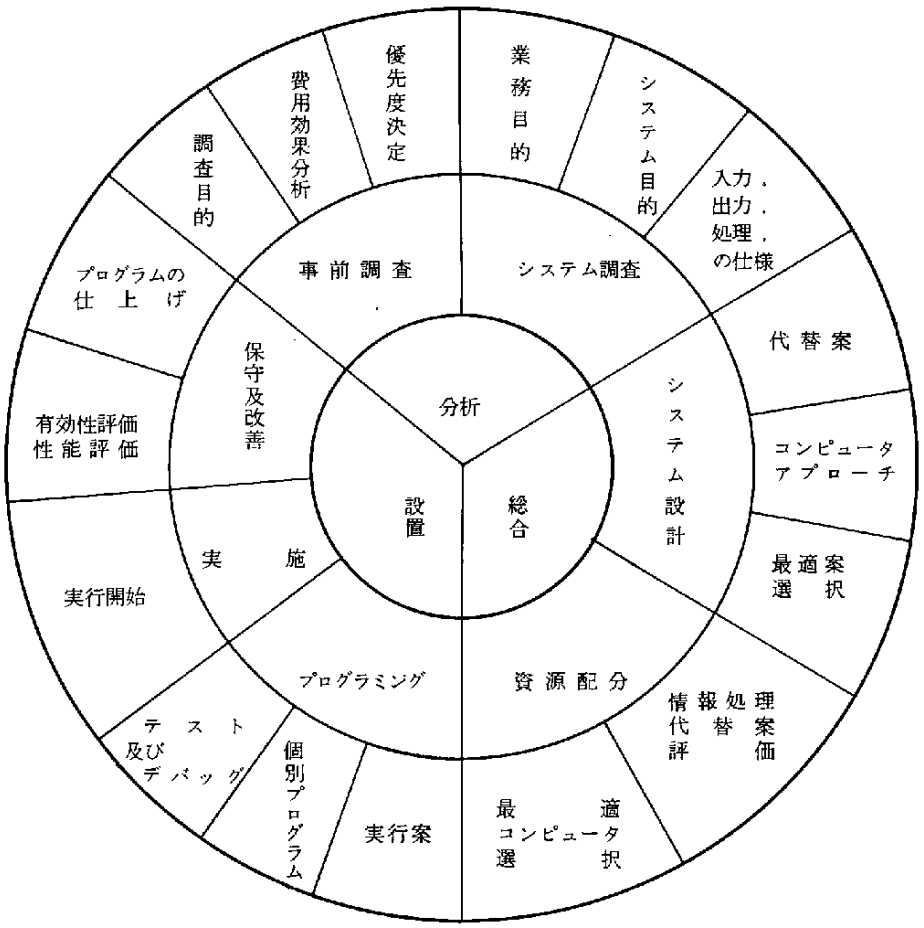
テストラン

・プロダクション（実際運用）

2. システム・ライフサイクルの詳細の一例

J. Kanter,"Management-oriented Management Information Systems"より

開発のサイクル



3. 解析原理に基づく方法の二三の例

(1) デカルトの方法序説

古典的アプローチとしてデカルトの方法論があげられる。（デカルトの「方法序説」第2部）

① 私が明証的に真であるとみとめた上でなくてはいかなるものも真として受け入れないこと〔明証の規則〕

② 私が吟味する問題のおのおのをできるかぎり多くのしかもその問題を最もよく解くために必要なだけ数の小部分に分つこと〔分析の規則〕

③ 私の思想を順序に従って導くこと，最も単純で最も認識しやすいものからはじめて，少しずついわば段階をふんで最も複雑なものの認識にまでのぼってゆくこと〔総合の規則〕

④ 何ものをも見落すことがなかったと確信しうるほど完全な枚挙と，全体にわたる見直しとをあらゆる場合におこなうこと〔枚挙の規則〕

(2) ケプナー・トリゴーの問題分析法

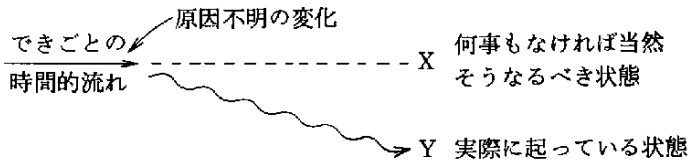
典型的な帰納的方法を定式化した一例として，ケプナー，トリゴーの問題分析法の要約を紹介しておく。

（C.H.ケープナー，B.B.トリゴー「管理者の判断力」産業能率短大出版部）より

＜問題分析の過程＞

状況 → 問題 → 重要な問題 → 考えられる原因 → 最もそれらしい問題 → 具体的改善

＜問題とは＞



逸脱 ＝ 問題

＜問題分析の手法＞

① あるべき姿と実際の姿の比較

② あるものとないものの比較

③ 実際と推論の比較

最もそれらしい逸脱の原因は問題の明細書にあらわれる事実をすべて説明するものでなければならない。

ケプナー，トリゴーのワークシート

逸脱		ある	ない	ある固有の差異	その中の変化
何が	何が逸脱か				
	逸脱の対象物				
どこで	対象のどこで				
	みられた場所				
いつ	何時				
	期間				
どの程度	量				
	数				
テストすべきもの 原因らしきもの		メモのためのスペース			

このワークシートがデカルトの方法序説の中の４つの規則に合致している点は興味あるところである。

(3) ナドラーのワークデザイン

方法としては，帰納的方法というよりはむしろ演繹的設計法ではあるが，考え方の中核にやはりヨーロッパ社会の解析原理がある点に注目すべきであろう。

（ジェラルド・ナドラー：「理想システム設計」東洋経済新聞社）より

ナドラーのワークデザインの中核部分は機能展開のプロセスである。その思想と手法を要約してみると，

① 小さなシステムをSoとし，その機能をFoとした場合，下記のような対応がある。

$So < S_1 < S_2 \cdots\cdots < Sn$

$Fo < F_1 < F_2 \cdots\cdots < Fn$

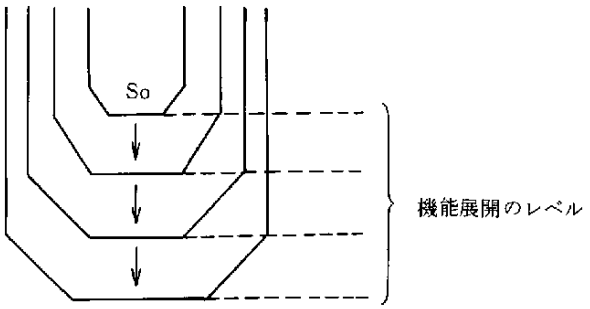
② 機能の系列を先にきめれば，システムの系列は自動的にきまる。（この機能の系列を決めて行くことを機能展開という。）

③ 機能展開の手続きは，手掛りになる小さなサブシステムを一つ選ぶ。次にそのシステ

ムの直接的な機能を一つ決める。次にこれに続くその次の機能を決める。繰返して次の機能を決める。

注意事項

機能展開のガイドとしてホッパー表現を提案している。



機能展開で次のレベルをひき出すのには大きくジャンプしないよう心掛ける。

次のレベルを引き出すための試みのための問いかけ

(i) ～ というのはどういうことか?

(ii) ～ を言い換えてみるとどうなるか?

(iii) ～ は何のためにあるのか?

(iv) ～ をする目的は何なのか?

機能表現は特定的(specific)にすること。しかし限定的(limiting)であっては,いけない。

参考文献

〔1〕 Benjamin, R. I.,"Control of the Information System Development Cycle," Wiley-Interscience, 1971

〔2〕 Hartman, W., Matthes, H. and Proeme, A.,"Management Information Systems Handbook," McGraw-Hill, 1968

〔3〕 Glans, T.B., Grad, B., Holstein, D., Meyers, W.E. and Schmidt, R.N.,"Management Systems," Rinehart and Winston, 1968

〔4〕 Rubin, M.,"Introduction to the System Life Cycle," Auerbach, 1970

〔5〕 Rubin, M.,"System Life Cycle Standards", Auerbach, 1970

〔6〕 Gunther, R. C.,"Management Methodology for Software Product Engineering," Wiley-Interscience, 1978

〔1〕はシステム・ライフサイクルを簡単に解説している。〔2〕はフィリップス社でま

とめられたシステム開発手順であって，各段階にわたって詳細に説明がある。〔3〕はシステム開発の予備調査段階に詳しいＩＢＭ社のＳＯＰ（Study Organization Plan）の概念を含む。〔4〕,〔5〕システム開発の標準形として解説してある。〔6〕最近のソフトウェア工学の成果等も含んで概説している。

〔7〕通産省情報処理振興課「昭和51年度情報処理技術者試験案内書」, 1976

〔8〕日本経営情報開発協会「システム・ライフサイクルのマネジメント」, 1973

〔9〕西谷直二，疋田英幸，秋葉二郎，八柳祐久　「リアルタイム・システムズ・マネジメント」　企画センター，1972

〔10〕高浜節夫編　「システム設計」　産報, 1972

〔11〕Ackoff, R.L.,"Redesigning the Future," John Wiley & Sons, 1974

〔12〕Quade, E.S., and Boucher, W.I.,"System Analysis and Policy Planning," American Elsvier, 1968

〔13〕Hare, V.,"Systems：A Diagnostic Approach," Harcourt Brace and Jovanovich, 1967

〔14〕宮川公男編　「システム分析概論」　有斐閣，1973

〔15〕クエイド，ブッチャー編　（香山健一，公文俊平監訳）　「システム分析」（Ⅰ.Ⅱ）竹内書店，1972　（〔12〕の訳書）

〔16〕古農文雄　「システム・アナリシス」　日本経営出版会，1969

〔17〕吉原賢治　「システム開発計画」　日本経営出版会，1969

〔18〕鵜沢昌和編　「システム分析，設計講座」　電子計算機通信学院，1971

〔19〕S.L.オプトナー著　（植木　繁訳）　「システム分析入門」　日本能率協会，1969

〔20〕Ackoff, R.L.,"A Concept of Corporate Planning," John Wiley & Sons, 1970

〔21〕Blumenthal, S.C.,"Management Information Systmes：A Framework for Planning and Development," Prentice-Hall, 1969

〔22〕Burch, J.G.Jr. and Strater, F.R.Jr.,"Information Systems：Theory and Practice," Hamilton, 1974

〔23〕Churchman, C.W.,"The Systems Approach," Delacorts Press, 1968

〔24〕Cleland, D.I. and King, W.R.,"Management：A Systems Approach," McGraw-Hill, 1972

〔25〕Condon, R.J.,"Data Processing Systems：Analysis and Design", Reston, 1975

〔26〕Optner, S.L.,"Systems Analysis for Business Management," Prentice-Hall,

1975

〔27〕 Thierauf, R. I.,"Systems Analysis and Design of Real-time Management Information Systems;" Prentice-Hall, 1975

〔28〕 シャーマン・C・ブルメンタール著 （菊地和聖訳） 「経営情報システムの設計」東洋経済新報社，1971 （〔21〕の訳書）

〔29〕 A.M. マクドノウ，L.J. ギャレット著 （今居譲吾訳） 「経営システムの設計」産業能率短大出版部，1966

〔30〕 D.I. クリーランド，W.R. キング著 （上田惇生訳） 「システム・マネジメント」ダイヤモンド社，1969

〔31〕 S.L. オプトナー著 （石田武雄訳） 「経営問題解決のためのシステム論」 ダイヤモンド社，1968

〔32〕 A.M. マクドノウ著 （松田武彦監訳） 「情報の経済学と経営システム」 好学社，1966

〔33〕 R.N. アンソニー著 （高橋吉之助訳） 「経営管理システムの基礎」 ダイヤモンド社，1968

〔34〕 J. Kanter "Management-oriented Management Information Systems" Ludedition, 1977

〔1〕~〔10〕： システム・ライフサイクルについての参考

〔11〕~〔19〕： システム分析についての参考

〔20〕~〔33〕： 教材としての参考

第2章　システム・ライフサイクル管理

キーワード

システム開発（system development），プロジェクト管理（project management，project control），組織の役割（organizational behavior effects），要員管理（managing personnel management），文書化（documentation）

目　　標

前章で学んだことは，システム開発の段階と手順であるが，それに対応して，本章で学習することは，そのステップに従って，システム開発を実行にうつす方法である。

システム開発の実行には，通常プロジェクト管理という形式が採用される。ここでは主として，そのプロジェクト管理の概要を説明する。

プロジェクト管理という新しい方法が発達して来た由来を，システム・ライフサイクル概念の形成過程と結びつけて説明すれば，システムの本質がさらに明確に理解されるようになるであろう。

従来の機能的定常的な管理方式と，新しい形式としてのプロジェクト管理方式との対比なども説明して，システム開発の性格をあきらかにする。

プロジェクト管理は，システム開発を実行する際の人間の役割に光をあてた，新しい視点である。このような視点は行動科学の研究等に触発された点も多い。この意味でＢ２の学習との関連がある。且つまた，Ｄ３の演習に入るための予備知識の整理としての位置を占めている。

システム開発のための業務管理の面を明示したものが，プロジェクト管理であって，それをさらに具体的に形にあらわしたものが，文書である。この章ではシステム・ライフサイクル管理のための文書化の問題を特に重点的に学ぶ。

本章は，第１章とともにＤ１，Ｄ２の全体に対する導入部である。さらに第３章に対する導入として，システム分析の必要性についても若干触れておく。

なお，文書化については，その技術的側面の考察は，Ｄ２の１章でもとりあげているので，本章では，その管理的側面に焦点をあててとりあつかうものとする。

内　　容

2.1　プロジェクト管理

システム開発をプロジェクトとしてとらえ，そのプロジェクトの生成より完成に至る過程を一貫して管理するのは，コンピュータ部門管理者の必要な基本業務である。ここでは，プロジェクト管理の目的，プロジェクト管理の対象，プロジェクト管理のための手法を理解させる。

(1)　プロジェクトとは

プロジェクト・マネジメント或いはプロジェクト・コントロールという形態は，わが国では比較的新しい分野である。システム開発に対して，この形態がとられるようになった経緯とその必要性を説明して，伝統的な「タテ」型のマネジメントと異なるその性格を明らかにする。

プロジェクト概念の発生の背景には，米国におけるＴＶＡ，マンハッタン計画，ＳＡＧＥ計画，アポロ計画と続いた一連のビッグプロジェクトの経験と，F.W.テイラーの「科学的管理の諸原則」に始まる管理論の発展の二つの系譜があることを指摘し，その発展に伴って形成されたシステム概念が，それ以前の社会の常識であった成り行き（ordinary course of things）にまかせて自然発生で出来あがったものと質的に相異する点があることに注意をうながし，システムというものがもっている人工的性格がプロジェクト管理を必然的に要求することを認識させる。

プロジェクト管理が，方法としてさらに明確な形態をもつに至った経緯も，システム・ライフサイクル概念の成立と平行するものとして説明する。即ち，OR, QC, IE 等の分野における各種技法の発見とその精密化や，コンピュータの発達などが技術的側面となり，石油精製，石油化学，原子力発電，天然ガス液化などの巨大プロジェクトの発生が需要面からの刺激となり，プロジェクト管理は現代を代表する組織形態の一分野となった。特にコンピュータ利用のシステムを開発するためにはかかすことの出来ない管理形式として定着しつつあり，その方法論としての細目が検討されつつある。第２章ではこれらの発展の概要をふりかえることによって，プロジェクト管理の目的の本質的なところを把握し，次にシステム・ライフサイクルの各段階と対応させて，管理対象の細目を把握させる。さらに現在までに開発された技法の二三の例を学習する。

(2)　プロジェクトの特徴

まずはじめにシステムのもつ性格をあきらかにする必要がある。そしてそのシステムを開発する方法としてのプロジェクト管理はどうあらねばならないかを考察する。

システムにはいくつかの特徴が考えられる。例えば

・未知の要素が多い。

新しい技術分野に対する技術予測が必要になる。

開発に要する資源，経費，要員の見積りが困難である。

・時間経過とともに変化する。

開発の進展につれて要求される組織や要員等が質量ともに変化する。

・ひとつの目標がある。

新しいシステムを作るということは，何をやりたいかということがはっきりしていることを意味する。この一つの目標にむかって各種の機能を結合させたものがシステムであって，これを実現させる方法がプロジェクトである。

・制約がある。

プロジェクトとは，限られた資源，手段，要員の制約のもとで，決められた期限内に目標のシステムを作りあげることである。

別の表現をすれば，システムとは人工的で，不確実で，やり直しがきかない，新しいものといった性格をもっている。このためにこれを開発するためのプロジェクト管理には，目的を明らかにして，着手する前にあらかじめ考えられることを十分に予測し，計画を綿密にたてて，その計画に沿って開発作業の管理をし，協同作業者同志のヨコの連絡をとり，計画路線からはずれそうになる時は直ちに軌道修正をおこない，計画通りの期限内に，目的とするものを完成させる，等々の注意が必要になる。これらの多くの事項を体系的にまとめたものがプロジェクト管理である。

制度としてプロジェクト管理を見た時には次のような特徴があげられる。

① 職能別管理制（functional organization）と異なり，課題中心の一貫担当となり組織としてヨコ割り型の臨時編成の性格をもつ。この組織はタスク・フォースとかプロジェクト・チームと呼ばれる。

② 人間中心，能力中心の創造主導型となる。したがって従来の組織中心の人間歯車型と性格が相反するものとなる。

③ 期間，資源，要員の配分に制約が生ずる。したがってこれらの配分問題の検討や決定が重要な技術問題として注目される。

④ 開発管理の手順として工程の段階区分を明確にすることが重要になる。このことはシステム・ライフサイクルの概念と一致するところである。且つその進捗管理の手段として文書の役割を重視する。表面的に見れば文書管理がプロジェクト管理であるかの観さえある。

(3) プロジェクト管理の内容

プロジェクト管理の内容はそれ自身で多くの学習すべき項目をもっているが，ここでは末梢的な部分にとらわれることなく，システム開発に不可決なものとしての本質的な部分のみを把握すればよい。

学習項目として整理すると

- 計画
- 組織及び要員
- 管理
- コミュニケーション

等の区分が考えられる。

次にその細目の例をあげる。

① 計画

開発目的

システム仕様書，その他の仕様書

システム分析，機能設計

開発スケジュール

各種見積，技術予測

② 組織と要員

タテ組織とヨコ組織（プロジェクトの性格）

専門家集団，委員会

プロジェクト・マネージャー

プロジェクト・チーム

組織の流動性

作業分担

③ 管理

狭義のプロジェクト・コントロール

工程管理，進捗管理

原価管理，開発費用管理

テストと品質管理

文書管理

新システムへの移行

評価システム

④　コミュニケーション

ヨコ組織におけるコミュニケーション

開発管理に必要な情報

文書の役割

2.2　プロジェクトの計画と管理

プロジェクトの具体化には，まずそのプロジェクトのスコープを明確にすることが必要である。開発目的の定義が不明確であったり，方針が設定されていなかったり確認されていなかったりすると，プロジェクトがその開発の途上で空中分解にさらされる危険がある。プロジェクト計画で最も重要な事項は開発目的の明確化である。

プロジェクトを実行する組織は，従来のピラミッド型の組織ではなく，ヨコの関係を重視する能力中心の遊撃隊になることが多い。このために人間関係や情報交換が重要な役割をはたさなければならないことになる。管理の確実さを期するためには，情報交換は記述された文書によらなければならない。

以上述べたようなプロジェクトの計画と管理での重要事項（目的の明確化，文書による情報交換）は，わが国での実状ではいずれも苦手とする習慣であるので，特に強調して教育する必要がある。

(1)　プロジェクトの計画

プロジェクト計画について学ぶべきことを重要さの順に列記すると次のようになる。

①　プロジェクトの採りあげ方について

・能力の限界を考える

コンピュータで何が出来るか？

実体のシステムとそれを支える情報システムとの間の関係を明らかにする。

・システムの改廃について考える

開発のあとの保全はどうか？

前進は易しいが退却は難しい。

・技術的知識と環境の知識のバランスをとる

習慣性から人工性への変革には抵抗がある。

無理に作りあげたものはくずれやすい。

十分すぎるほどの技術的知識が必要である。

②　目的の設定について

・システム計画の必然性があるか？

・経済的に妥当であるか？

・目的が仕様書の形に記述できるか？

・システム開発をすることの意味，可能性，副次効果，反作用等を多面的に評価する。

③ 計画の作成について

・開発計画，導入計画を記述する。

・主要マイルストーンを決定する。

・ワーク・ロードを推定する。

各種の見積方法，推定のための基準値を整備する。

・システム要件を決定する。

アプリケーション
オペレーション
ハードウエア
ソフトウエア
} → プロジェクト・マニュアルとしてまとめる。

・プロジェクト・チームを編成する

作業分担を決める。

情報ルートを定める。

ドキュメンテーション・プランをたてる。

・スケジュールを決定する

実施計画の日程表を作る。

(2) プロジェクトの実行

システム分析・システム設計の作業とプロジェクト管理とは表裏一体のものである。プロジェクト管理を実務手順として分解すると次のような項目がある。

① 開発要求書検討

② 開発プライオリテイの決定

③ 開発所要時間・必要資源の推定

④ タスクリストの作成

⑤ 要員配置

⑥ 開発スケジュールの作成

⑦ テストスケジュールの作成

⑧ 報告制度と文書管理の制定

⑨ 進捗管理

⑩ テスト管理

⑪ 新システムへの移行

⑫ 評価・監査

⑬ 開発費用の配賦

以上はプロジェクトの進行に沿って抽出した項目であるが，視点を変えて横断的にみると，

・要員管理

・要員教育

・作業管理

・品質管理

・文書管理

・開発支援活動

などのさまざまな情報網が必要になる。 これらの活動をささえるものには，ミーティングのような無形の情報も重要であるが， プロジェクト管理として最も重視されるものは，文書である。

さらにプロジェクト管理を実施する人間の側の問題として，その組織についても検討しておかなければならない。即ち次の諸点である。

・開発部門の位置づけ

・開発部門の内部編成

・要求部署との関係（受託・協同・相談）

・委員会組織との関係

(3) プロジェクト管理の手法

管理技術の面からみて，プロジェクト管理を実行するための手段が，手法として各種開発され試用されている。例えば，

① ネットワーク表現

PERT，CPM

② 予算配分

PPBS

③ 予測とコントロール

FAME

④ スケジューリング

ガントチャート

山積み，山崩しのシミュレーション

⑤ 最適化計画

⑥ 報告制度の自動化

⑦ 文書作成の自動化

⑧ 評価制度の自動化

プロジェクト管理の性格を明らかにするために，これらの手法と先に述べた計画と管理の実務とを結びつけて説明することが望ましい。

2.3 プロジェクト管理と文書

システム・ライフサイクルを具体的に示すものは，その各段階の区切を表現する文書である。プロジェクト管理は，プロジェクト・メンバーの能力に依存する面が大きいため，何らかの方策を講じない限り，開発内容の実体が把握しにくいという弱点がある。このためコミュニケーションのルートを十分に整備する必要がある。即ち次のような情報交換網を固めなければならない。

① 各個人の作業継続のための情報

② プロジェクト・メンバー相互間の情報

③ プロジェクト・メンバーと管理者間の情報

④ プロジェクトとプロジェクト外部間の情報

プロジェクトは分業と統制によってシステム開発を進行させるものである。この役割をはたすために必要な位置を占めるものは，③のルートである。且つまたシステムの完成のために重要な情報は，プロジェクト・チームとその外部との交流である。このために必要なルートは④である。これらは文書のもつ役割のうちの管理的側面である。これに対して，前の二者①②は技術的側面というべきものである。文書の役割のうちの技術的側面については単元Ｄ２で学習する予定であるので，ここでは文書の役割の管理的側面に焦点をあてて学習する。即ち指示書，報告書，仕様書，計画書等について検討する。夫々の文書の様式の具体例等については単元Ｄ４で目にふれるであろう。

システム・ライフサイクルの各段階を区切るものとしての文書は即ちその各段階の性格を表現するものでもある。そこに記述されている項目が何を表現し，なぜその項目が必要であるかを，システム・ライフサイクル管理に照らして研究するように指導する。

システム・ライフサイクルのフェイズ区分との対照で，各文書の役割を図示した例を次にあげる。各文書に記述すべき項目は，夫々のフェイズの設定理由と密接に関係する。

ベンジャミン（Benjamin）

R.I.Benjamin,

"Control of the Information System Developmont Cycle"から

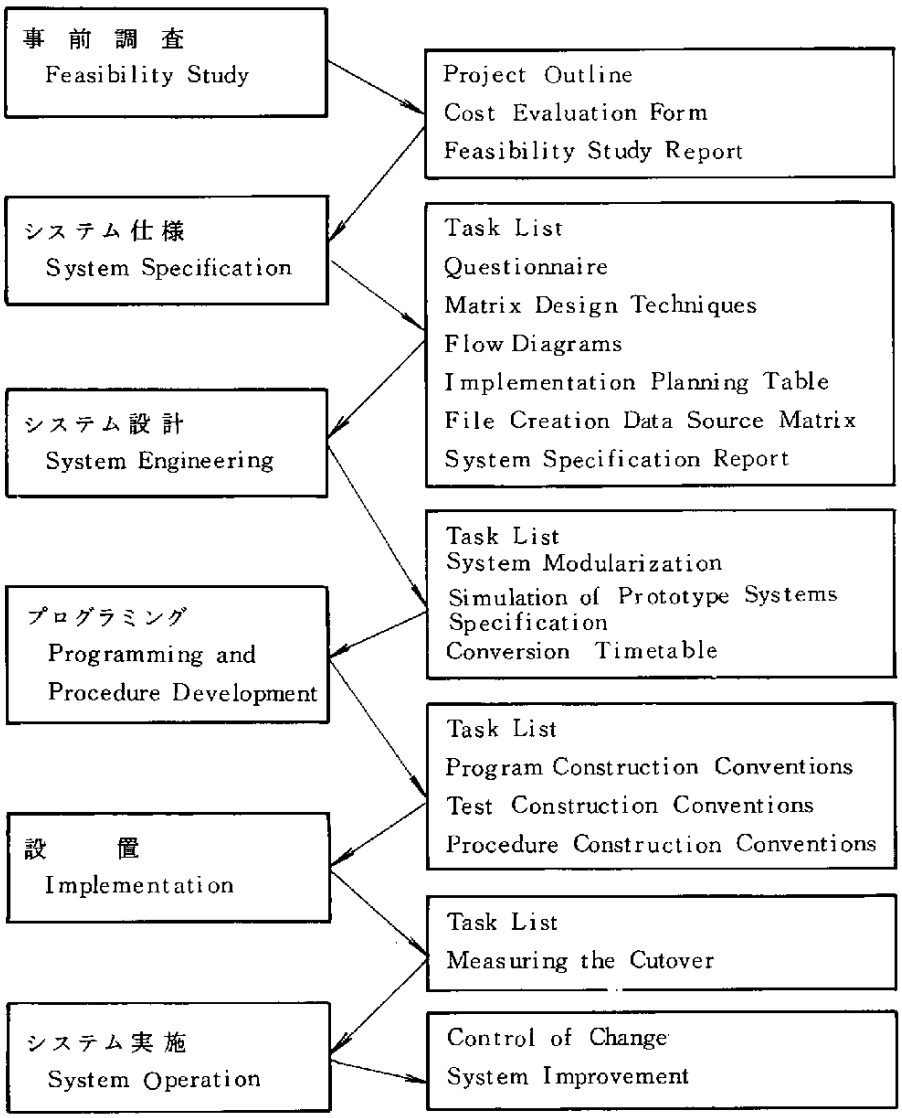


図２－１　局面間のインターフェスとしての文書

クラウス（Krauss）

L. I. Krauss,"The Company Data Processing Function"から

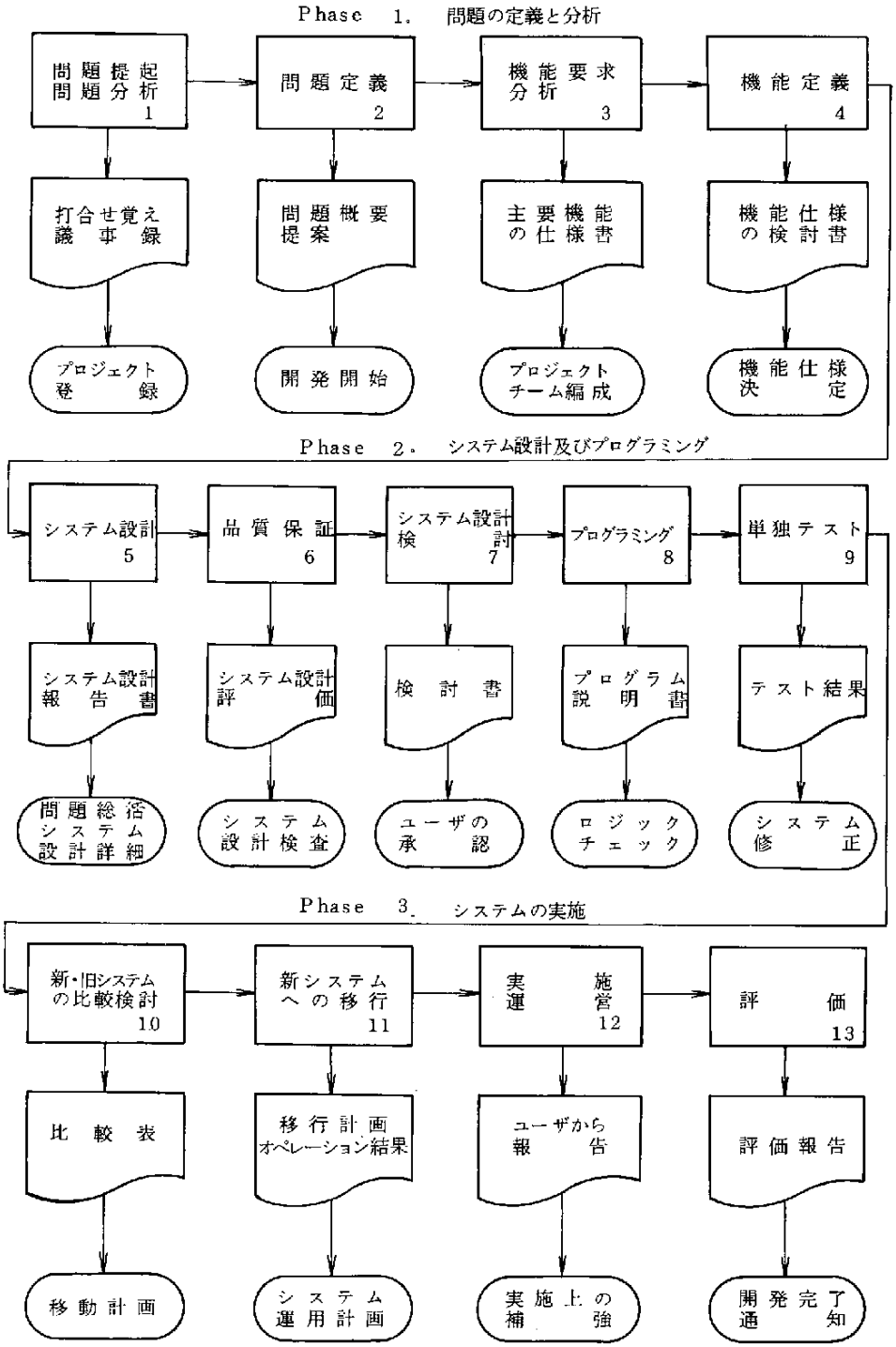


図２－２　報告書としての文書

2.4 プロジェクト・チーム

プロジェクト・チームによるシステム開発の場合は，システム・ライフサイクルの各局面において，各々の局面に適応したメンバーに編成しなおして開発に従事することが理想である。その連続性を保証するためには，前に述べた文書による管理体系が確立されていなければならない。且つまた局面に即応して編成されるチームメンバー同志の意気があっていなければならない。コンピュータ・システムの開発の場合は，通常，システム分析，システム設計，インプリメンテーション，テストの4段階程度にわかれていることが普通の形であろう。この種の段階わけに基づいて各グループの作業分担及びその構成メンバーのあるべき姿等を考察してみる。

(1) システム分析グループ

（システム仕様書を作成するグループ）

わが国の実状ではこのグループは明確には存在せず，システム設計者がこれを兼ねることが多い。このようにシステム仕様書が独立に存在しないためにおこる開発途中の問題点を指摘し，システム分析の局面が分離独立することの必要性を認識させる必要がある。あわせてシステム分析とは何であるかを教育する。

(2) システム設計グループ

（プログラム設計書を作成するグループ）

(3) プログラム開発グループ

（プログラム作成をするグループ）

(4) オペレーション・グループ

（システムテストをするグループ）

通常オペレーション・グループとは，新システム移行後の実施をおこなうグループと解されているが，ここではむしろ，プログラムとシステムテストとを異なるグループで行うことの利点とその必要性を強調することをねらいとする。

以上のような考察のほかに，プロジェクト・マネージャーの資質と役割，プロジェクト・メンバーのあり方などについても考察をおこない，異質のメンバーの協同作業というプロジェクト制によるシステム開発のもっている人間的側面の重要性について認識させることも重要である。

このことは単元D3による実習に対する準備でもある。

指導上の留意点

(1) システム・ライフサイクル概念とプロジェクト管理とは表裏一体の思想であることを理解させる。

(2) プロジェクト管理技法については，Ｄ１，Ｄ２を通じて以降の章で詳細にふれることがないので，この章で実例によってやや詳しく学習してもよい。特にネットワーク表現によるスケジューリングとコントロールなどは，単元Ｄ３に対する予備知識として必要である。

(3) 文書を通じてのプロジェクト管理の体系を教育の縦糸として全体像を示す。即ちシステム・ライフサイクルにおけるフェーズ分割の重要性などに重点をおく。

(4) プロジェクト管理の実務的な詳細には立ち入らなくてよい。

(5) システム分析の思想としての要点を説明し，第３章に対する導入をおこなう。

参　　考

1. プロジェクト管理のための報告制度

プロジェクト管理を円滑に行うためには，業務の進捗状況の把握がその基本になる。その他，ユーザ部門・コンピュータ部門間の意思伝達方法の確保，作業計画改訂にともなう問題処理等，日常管理業務の遂行には文書による報告制度の整備が不可欠となる。ここでは，代表的な管理のための帳票について説明する。

(a) **業務依頼書**

ユーザ部門からコンピュータ部門に対する業務依頼であり下記項目を含む。

- 業務の概要
- 業務ニーズの説明
- 希望する業務処理内容
- 期限その他要望事項

(b) **業務受託書**

コンピュータ部門からユーザ部門へ返送される。通常見積書を兼ね，ユーザの希望通りの条件で受託できない場合は，その旨記載される。

- 開発コスト推計（職種別所要要員のマン・アワー，単価，労務費，ハードウエア使用費，単価，機械使用費，総開発コスト）
- 運用コスト推計（運用要員の労務費，使用ハードウエア費，通信費，保守費その他の固定費割掛分，総運転コスト）
- 依頼業務の諾否
- その他特記事項

(c) **システム開発計画表**

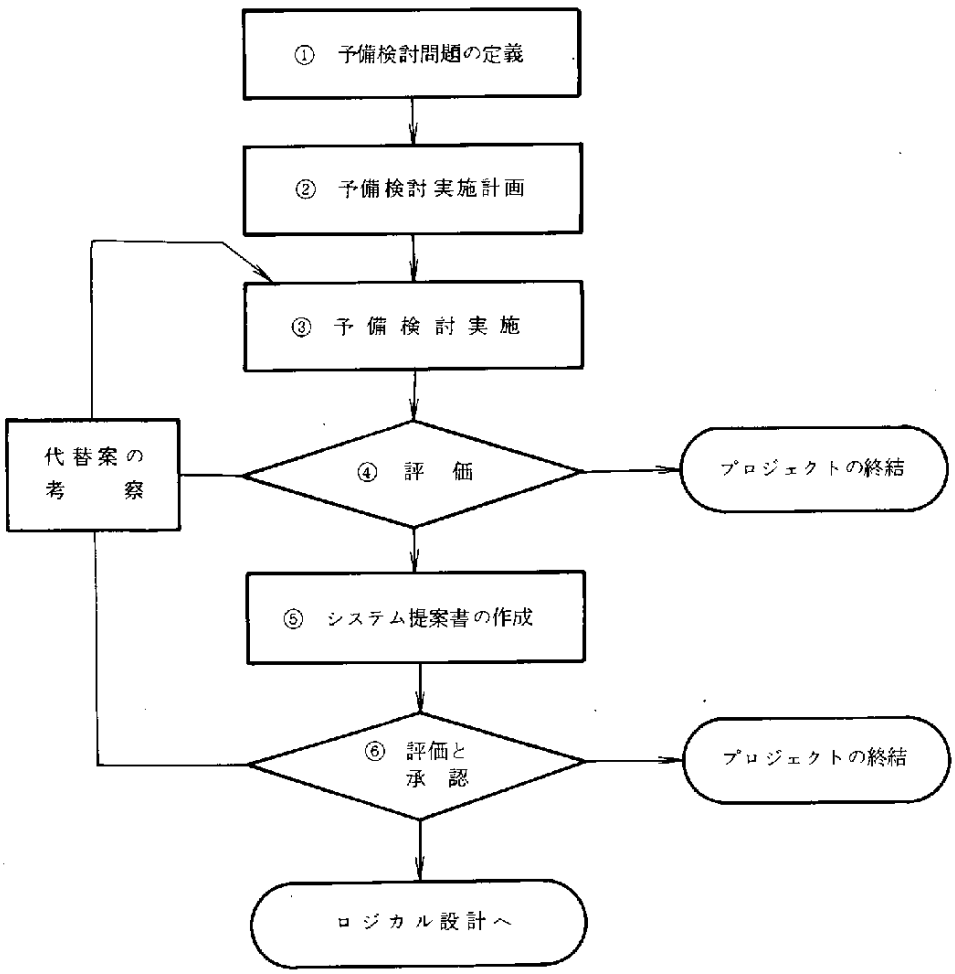
- 要員名別業務分担表
- 業務推進計画表（業務名称，担当，所要人数，目標達成日，関連業務との相互関係）

(d) **プロジェクト進捗状況表**

- (月，週間)業務進捗報告書(当初目標，実績進捗状況，遅延・トラブル等あればその事由，担当者名)
- システム設計進捗状況表
- プログラム開発進捗状況表(プログラム名，プログラマ，プログラム状況)

2. システム・ライフサイクル管理のポイント

(1) 予備検討



システム提案書

1．予備検討の要約

a．問題の概要

b．問題に対する対策

c．今回の提案

2．提案するシステムの説明

a．提案するシステムの目的と，解決される問題

b．提案するシステムの概要

システム・フローダイヤグラム

インプット／処理／アウトプット

3．提案するシステムのコストと評価

a．開発コスト

b．運用コスト

c．費用，効果分析

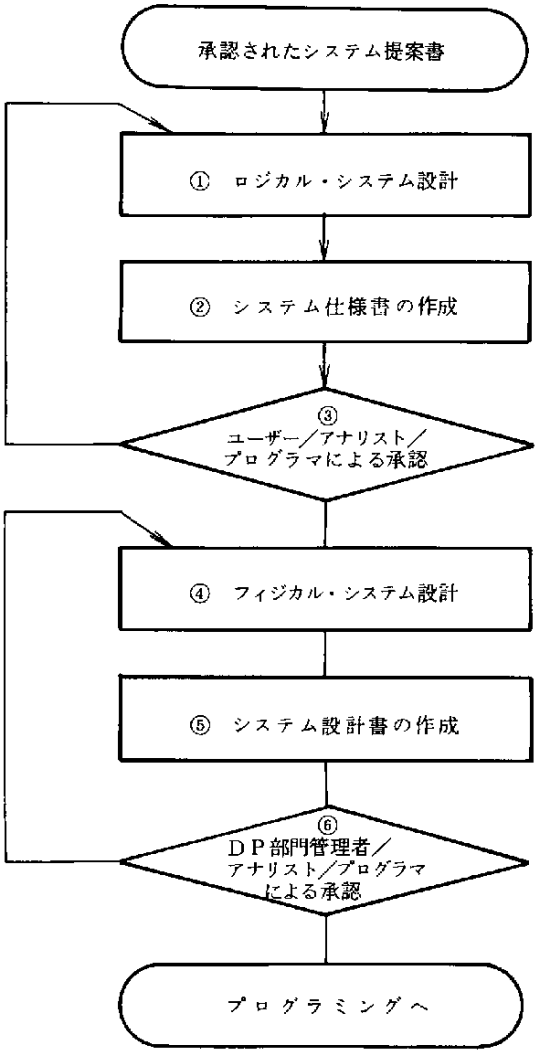
4．システム開発計画

a．システム開発スケジュール

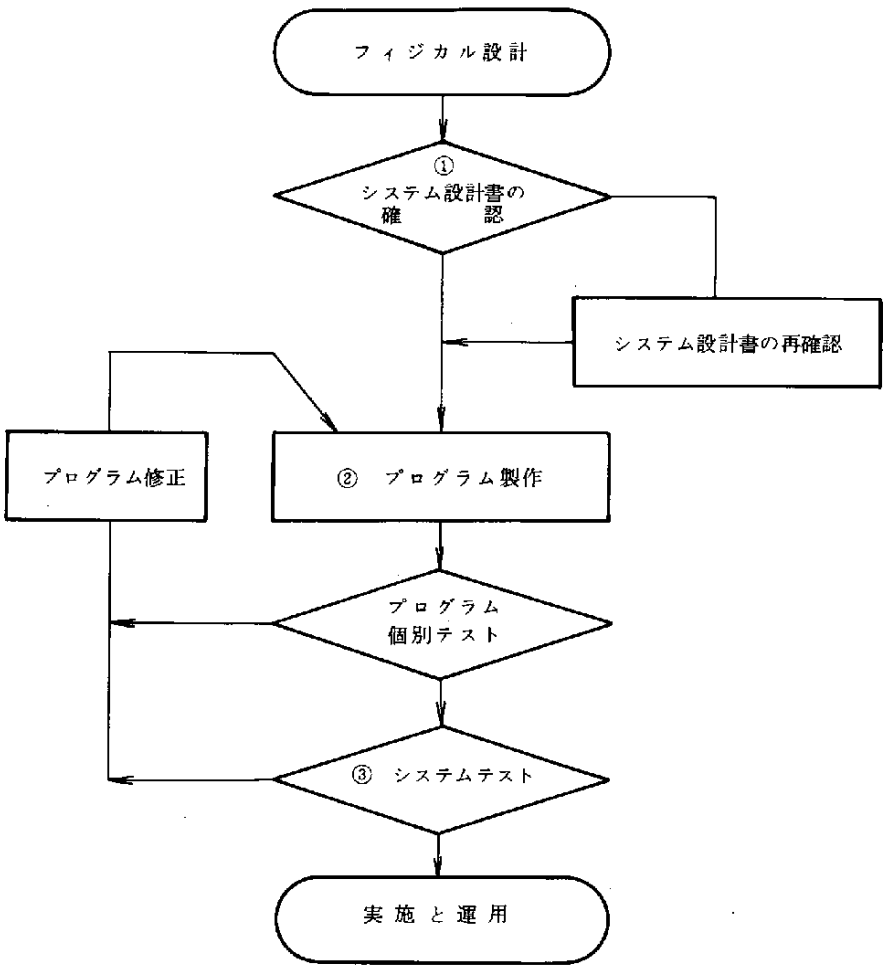
b．要員及びその他資源投入計画

c．今後の推進体制

(2) システム設計


承認されたシステム提案書
① ロジカル・システム設計
② システム仕様書の作成
③
ユーザー／アナリスト／
プログラマによる承認
④ フィジカル・システム設計
⑤ システム設計書の作成
⑥
DP部門管理者／
アナリスト／プログラマ
による承認
プログラミングへ

(3) プログラム作成



参考文献

〔1〕 宮川公男 「意思決定論」 丸善，1975

〔2〕 占部都美編 「経営学のすすめ」 筑摩書房，1970

〔3〕 後藤幸男，小林靖雄，土屋守章，宮川公男 「経営学を学ぶ」 有斐閣，1971

〔4〕 アーネスト・デール，L.C.ミシュロン著 （北野利信訳） 「現代経営の方法」 評論社， 1968

〔5〕 荒川幾男 「管理社会」 講談社，1970

プロジェクト管理を学ぶための前提として，一般に上記のような組織論の常識を概観しておく。

〔6〕 Lecht, C.P.,"The Management of Computer Programming Projects", American

Management Association, 1967

〔7〕 Benjamin, R. I.,"Control of the Information System Development Cycle", John Wiley & Sons, 1971

〔8〕 Hartman, W. et al.,"Management Information Systems Handbook," McGraw-Hill, 1968

〔9〕 Krauss, L. I.,"Administering and Controlling The Company Date Processing Function," Prentice-Hall, 1969

〔10〕 Brandon, D. H.,"Management Standards for Data Processing,"Van Nostrand Reinhold, 1963

〔11〕 Gunther, R. C.," Management Methodology for Software Product Engineering," John Wiley & Sons, 1978

〔12〕 Gray, M. and London, K.,"Documentation Standard,"Auerbach publishers,1969

〔13〕 Kuehne, R. S., Lindberg, H. W. and Baron, W. F.,"Manual of Computer Documentation Standards with Forms", Prentice-Hall, 1972

〔14〕 Walsh, D. A.,"A Guide for Software Documentation, Advanced Computer Techniques", 1969

〔15〕 Katzan, H.,"Systems Design and Documentation," Van Nostrand Reinhold,1976

〔16〕 Mckeever, J. M.,"Management Reporting Systems," John Wiley & Sons, 1971

〔17〕 Weinwurm, G. F.,"On the Management of Computer Programming," Auerbach Publishers, 1970

〔18〕 Szweda, R. A.,"Information Processing Management," Auerbach Publishers,1972

〔19〕 Albrecht, L. K.,"Organization and Management of information processing Systems," The Macmillan Company, 1973

〔20〕 Brandon, D. H. and Gray, M.,"Project Control Standards," Auerbach publishers, 1970

〔21〕 川原 裕 「オンライン・システム導入のためのドキュメンテーション」 企画センター, 1974

これらの参考文献をみれば，文書とプロジェクト管理の密接な関係がわかるであろう。ここには単元Ｄ２で学ぶべき，技術的に詳細な文書も含まれている。

〔22〕 Ackoff, R. L.,"A Concept of Corporate Planning",John Wiley & Sons, 1970

〔23〕 Brooks, F. P.,"The Mythical Man-Month,"Addison-Wesley, 1975

（山内 正訳 「ソフトウェア開発の神話」 企画センター）

〔24〕 Simon, H.A.,"The Science of the Artificial", MIT Press, 1969

〔25〕 Weinberg, G. M.,"An Introduction to General Systems Thinking," John Wiley & Sons, 1975

〔26〕 Matthies, L.H.,"The Management System : Systems Are for People," John Wiley & Sons, 1976

〔27〕 Mc Lean, E.R. and Soden, J. V.,"Strategic Planning for MIS", John Wiley & Sons, 1977

〔28〕 Mc Rac, T.W.,"Computers and Accounting," John Wiley & Sons, 1977

〔29〕 Murdick, R. G.,"Introduction to Management Information Systems", Prentice-Hall, 1977

〔30〕 Newlland, A. and Simon, H. A.,"Human Problem Solving," Prentice-Hall, 1972

〔31〕 Orlicky, J.,"The Successful Computer Systems: It's Planning, Development and Management in a Business Enterprise", Mc Graw-Hill, 1969

〔32〕 "Business Enterprise", McGraw-Hill, 1969
Ramsgard, W. C.,"Making Systems Work : The Psychology of Business Systems", John Wiley & Sons, 1977

〔33〕 H. A. サイモン著 （高宮 晋監修） 「システムの科学」 ダイヤモンド社，1969
（〔24〕の訳書）

〔34〕 松田正一 「システムの話」 日本経済新聞社，1973

〔35〕 中根千枝 「タテ社会の人間関係」 講談社，1976

〔36〕 川喜田二郎 「パーティ学」 社会思想社，1956

〔37〕 川喜田二郎 「組織と人間」 日本放送出版協会，1966

〔38〕 川喜田二郎 「チーム ワーク」 光文社，1966

〔39〕 加藤秀俊 「人間関係」 中央公論社，1966

〔40〕 会田雄次 「合理主義」 講談社，1966

〔41〕 関 計夫 「人間関係の幾何学」，1970

〔42〕 江川 朗 「プロジェクトチーム」 日本能率協会，1970

〔43〕 小林 茂 「創造的経営」 マネージメント出版部，1967

〔44〕 三浦大亮編 「システムの分析と設計」 オーム社，1973

〔45〕 南條 優，保坂岩男 「システム開発と運用」 オーム社，1972

〔46〕 大日方 真 「プログラム開発の工程管理」 情報処理 Vol 12, Vol 13

ここにあげた文献は，プロジェクト管理に関連して，システムの一般的性格を論じてい

るものと，プロジェクト開発を人間的側面からとりあげているものとに焦点をあてた。

第3章　基本的な分析手法

キーワード

分析と総合（analysis and synthesis），帰納と演繹（induction and deduction），面接（interview），予測（forecasting），モデル化（modeling，model building），分析手法（analysis tools，techniques for analysis），予備調査（preliminary investigation），実現性検討（feasibility study），事務分析（office work analysis，job analysis），ネットワーク分析（net work analysis），システム提案（system proposal），ディシジョンテーブル（decision tables）。

目　　標

本章では，システム分析・設計の基本的な考え方と，その具体的な方法の全体的な把握を行なう。

システム分析とシステム設計とは，システム開発の方法としての一貫性から考えると，別々に分離して学習しにくい面もあるので，ここでとりあげる項目は，Ｄ２の第１章で再び学習する内容と重複する。ここでは，システム分析のうちで，目標志向的な面に重点をおく。問題解決的な面については，Ｄ２の第１章で採りあげる。

システム分析の方法の特色は，目標に対する接近法が反復的であることを指摘し，システム・ライフサイクルの性格をあきらかにする。この章の内容は，Ａ１の第３章の各論でもある。

システム分析の手法の詳細を，３章，４章，５章で学習するが，その全体に対する概論的な把握をこの章でまずとりあつかう。特にその中で，問題の発見，ニーズの把握，システムの目的の解明などの事前調査に必要な手法に重点をおいて学習する。システム分析の基本要素のうち，目的の考察と，モデル化の役割について研究する。

各種の分析手法を例示して，それらがどのような場面で使われるかを考察させる。これによってシステム分析の意味をより深く把握させる。

内　　容

3.1　システム分析

システム・ライフサイクルの出発点は，システム分析である。方法論としてのシステム分析は，米国における国防問題に研究の端を発し，その後一般に民間でも，複雑な問題を解決する

ために普及するようになった体系的な手法である。その形をなした歴史はまだ日が浅いために，人によって解釈はまちまちであるが，その考え方はヨーロッパの近代社会の歴史とともに育った解析原理を母体とする思想であるので，解釈に対して細かな差異はあっても根底では一致するものをもっている。システム分析の基本的な特徴をあげると次のような諸点である。

① システムをその背景との関係で構造のあるものとしてとらえる。（構造）

② 問題の本質を細分化することによって明らかにしようとする。（解析）

③ 問題の不確実性を認識して，解決法を唯一つだけあげようとせず，代替案を多くとりあげて検討する。（不確実性）（誤謬）

④ 解決の方向として，与えられた問題だけを考えるのではなく，その目標にまでさかのぼって考えてみて，より根本的な解決を見出そうとする。（目的）

システムを構造のある体系として見ようとする態度の根底には，欧米人の考え方の中にあるギリシヤ以来の思潮がある。即ち一見混み入っているように見えるものでも，よく分析してみれば，その中にはかならず整った原理とか体系が存在している筈であるとする信念がこれにあたる。システム分析について云えば，対象を階層をもった構造的な体系としてとらえ，その問題がそれをとりまく全体の中のどの階層にあるのか？　上下の結合関係はどうなっているのか？と考えてみる方法がこれにあたる。

分析の方法として，問題を細分化して，分割不可能な基本の要素に到達しようとする態度も，ヘレニズムの伝統に基づくヨーロッパ的思考プロセスの基本形である。問題の不確実性や人間のもつ錯覚を認識して，できるだけ多くの代替案を考えてみるという態度も，デカルトの「まず疑うことから始める」という方法であるし，フランシス・ベーコンの「ノブム・オルガヌム」の中で述べられている先入観に対する警告としてのイドラの思想である。

これらの考え方は，自然科学の方法としてその後洗練されたものでもあるが，より広く経験世界の問題に適用されて，帰納法として近代思潮の源流となった思考形式である。

特に注目しなければならないのは，これらの思考形式や分析手順は，単に頭脳の中でのプロセスではなく，かならず記述されたものとして客体化されていなければならないとする点である。

このようにふりかえってみるとシステム分析の特徴は，とりたてて新しい方法というべきものではなく，昔からものを考える人々の間でくり返し検討された問題である。

システム開発という新しい需要に呼びおこされて，昔からの思考方法が，現代的に整理されて，手法として再認識されたものがシステム分析である。目先の与えられた問題を，試験問題を解く学生のような受身の態度で即答するのがシステム分析ではない。なぜその問題を解かなければならないのか？　その問題の本質は何か？　などと問題の奥にある原因や，システムの

目的を追求するところから出発するのが，システム分析の方法である。

特徴の項で述べた①と③はシステムに対する認識に関する部分であり，②と④は方法としての分析に関する部分である。

システム分析の学習では，考え方の根底をまず身につけることが大切であって，「だれそれ」の唱える方法は「かくかく」であると諸説をひろいあげて，それらをおぼえるのは無意味である。

システム分析の方法としての性格をさらに述べると漸近的，反復的，改良的ということができる。とりあつかう問題は一般に自然科学の問題のように一義的に解答が得られるものとは異なる。解決案を得るための道筋は比較検討の積み重ねによる螺線状の軌跡になる。そのために比較検討の判断基準として目標が必要になる。厳密にいえば検証不可能な性質をもつものが多い。このような状況下での判断のための方法として整理されたものがシステム分析である。その方法を具体的に述べてみると次のようになる。

① 目的を的確に定義する。

② 代替案を比較検討する。

③ 代替案の中から目的に照らして最善と考えられる案を選択する。

即ち，システム分析の考え方の中の重要なキーワードは目的と代替案である。

システム開発の出発点又は準備として，問題点の理解を確実にしようとする方向が，この分析の作業であって，問題点へのアプローチは帰納的方法である。単元Ｄ１ではこの方向での研究態度を中心にして学習する。（分析が帰納的方法を中心とするのに対比すると，設計は演繹的であるということが出来る。単元Ｄ２の課題は，分析とは逆に，いかに総合し，一つの具体的システムを構成するかということである。）

システム分析を学習するには，まずはじめに上記のような考え方としての本質的なところを確実に把握しておくことが大切である。

3.2 システム分析の要素

システム分析の主要な構成要素は次のようなものである。

(1) 目　的

(2) 代替案

(3) モデル

(4) 費　用

(5) 効　果

(6) 評価基準

これらの構成要素がシステム分析に対してどんな役割を占めるかを，さきに述べたようなシステム分析の性格と照合して考察する。ここでの学習は，次章以降のＤ１の各論に対する序章でもある。

次にシステム分析のプロセスの標準的なパターンについて概略を把握し，基本的な分析手法が何であるかを学習する。分析のプロセスを模式化すると，次のようなステップになる。分析の反復的な性格から，これらのステップはループ状に何回も繰り返えされる。

①　問題の明確化（仮定の再吟味）
②　目的の選択
③　代替案の作成
④　データの収集
⑤　モデル化
⑥　費用効果分析（評価）

分析のプロセスの骨格を確実に把握した後に，個々のステップに適用される各種の分析手法を学習するべきである。

システム分析を進める上で，重要な役割を果たすものはモデル化の方法である。この分野の研究はシステム工学で早くから実用化されている方法である。システム分析におけるモデルの機能を図式化すると次のようになる。

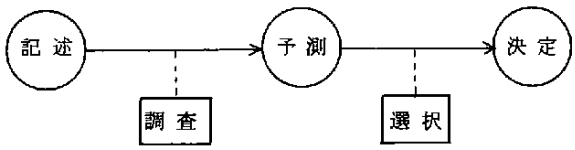


図３－１　モデルの機能と効用

調査のためのモデル化は，問題について理解を深めるために必要である。要因によって生じる変化をこれによって予測することが出来る。選択のためのモデル化は，現実の場での高価な試行錯誤をさけるために必要である。前者を規範モデル，後者を最適化モデルと呼ぶこともある。モデル化の技法はＯＲの分野で発達している。システム分析にも，問題の本質を把握するために，これらの技法は活用されなければならない。

以上の基礎事項を概説した後にシステム分析の具体的作業内容にも言及する。基本的分析手法が，これらの作業の局面でどう生かされるかの理解を助けるための程度でよい。

次のような作業内容がある。

１）システムの範囲の位置づけ

システム対象を定義する。

2）システムの構成要素の決定

設備，人間，組織，時間，空間，情報

3）開発の立場を明らかにする

現状改善か，新システムの開発か。

全体システムか，部分システムか。

4）業務分析，予備検討をおこなう

・マスター・プランを作る

要望と制約の調整

範囲，目標の設定

システム実現のための環境を保証する。

・予備検討

必要性，実現可能性，継続保守の保証

サブシステム，開発スケジュール

・承　認

現状分析 → モデル作成 → システム提案

3.3 各種の分析手法

(1) 面接の技術

システム・ライフサイクルの第一段階，予備検討（フィージビリティ・スタディ）での情報収集活動において，面接は効果的な方法である。又，システム・インプリメンテーションに至るまで，ユーザ部門との折衝は多い。

面接の技術はＩＢＭ社の ＳＯＰ（Study Organization Plan）において，取り入れられて以降，システム・アナリストのマスターすべき要件となりつつある。コミュニケーション理論の発展とも無縁ではない。又，その背景には，国際化の進展による異文化圏に属する人々の間のコミュニケーションの重要性の認識がある。この理論を応用すれば，システム・アナリストとユーザ間の意志疎通に大いに役立つわけである。

ここでは，コミュニケーションの理論を参照しながら，面接の基本を学ばせる。

次のような項目が考えられる。

・情報収集方法のなかの面接の位置づけ。

・面接の方法，面接の実際手順。

・コミュニケーション理論，例えば一般意味論。

(2) 情報整理の技術と発想の技術

現行システムに関する実態調査が進展した段階で，収集した情報の整理を行う。この整理の過程での工夫が，次の発想へと結びつく。ここでは情報整理と，整理した情報から提案を導きだす技術について説明する。

次のような人々の提案が参考になる。

・梅棹忠夫

・川喜田二郎

・渡部昇一

・中山正和

・オズボーン

・ウイリアム・ゴードン

・デ・ボノ

(3) 事務分析

ここでは，問題の発見と現状分析に役立つ事務分析フローチャートを中心に説明する。体系化された事務分析手法の例として，ＩＢＭ社のＳＯＰ（Study Organization Plan）などが教材として適切である。

事務分析のためのフローチャート作成は，コンピュータの利用以前から用いられている。そのため，旧来の事務分析フローチャートは，電子計算機時代の，電子計算機主体の事務の流れを描くのは不便な点もあるが，事務の基本を理解し，手作業部分の抽出と実態把握には不可欠の手法といえる。

フローチャートの書き方には，ＮＯＭＡ式，日能式，能大式等がある。

(4) ＩＥ的分析手法

生産工程での生産性向上と管理のレベルアップに，情報処理技術が適用される度合が増大するにつれ，情報処理技術者が生産能率向上のための技術，ＩＥの基礎知識を必要とする場合が多くなっている。生産性向上運動は第二次大戦後急速に盛んになり今日におよんでいる。情報処理技術を用いて生産工程の合理化をはかろうとするものにとって，いわば古典に相当するＩＥの基礎理論は避けて通れない知識といえる。

ここでは，ＩＥスペシャリストとプロジェクトチームを構成し，生産工程での合理化を推進する立場の情報処理技術者が最低限要求されるであろう基礎知識を中心に，ＩＥの基本的なアプローチの仕方について理解させる。

次のような学習項目がある。

・特性要因図

・パレート図

・工程分析（作業分析，動作分析）

・価値分析（V.A.）

・ワーク・デザイン

(5) 予測の手法

システム分析の基礎知識として予測の手法が要求される場合が少なくない。例えば，経営環境の認識のための予測，技術進歩に関するサーベイ的な予測，需要予測，さらには予測的接近をしたいのだがどの様な分析手法を用いたら良いかと云った選択の場合とか，将来に予想されるデータ量増大の予測とか，システム・コンフィギュレーションが将来の業務量増大に耐えうるかといった具体的問題に至るまで，予測を要求される場合が少なくない。

この節では，予測の手法の概要を一般のシステム・アナリストが必要とする程度において学習させる。したがって手法の技術的側面のみの学習ではなく，どういう場合に利用されるか，その場合の前提は何か，その手法の限界は何か，といった側面を理解させることが肝要であろう。

次にあげるような各種の予測手法がある。

・客観的予測

客観的予測とは，統計的処理と，OR的処理ができる対象についていう。時系列データ，過去のデータを分析し，将来に外挿するといったアプローチを取ることなどを説明し，主要な手法の概要，特徴を理解させる。

グラフ化，指数平滑法，季節調整法，回帰分析を用いての予測，微分方程式による決定論的予測，確率モデルによるものなど。

・主観的予測

計量化できない現象に対して適用される場合が多いが，積極的に学識経験者，スペシャリストの意見を取り入れて予測を行う場合も多い。

アンケート調査，デルファイ法，関連樹木図による予測など。

・システム的予測

種々の要因が複雑にからみあっている問題に対する予測でコンピュータの利用が可能になってから特に発達した技法である。

次のような例があげられる。

① 計量経済学モデル

対象は経済，地域経済，産業，個別企業など。

② 産業関連モデル

③ システムダイナミックス

④ クロス・インパクト・マトリックス法

⑤ シナリオ・ライテング

⑥ 各種シミュレーション

(6) システム開発技法

内容的には単元Ｄ２でとりあつかう「設計」に関する技術が，分析の段階と不可分に一体化されて，システム開発技法として提案されている。これらはソフトウェア工学と呼ばれ，最近注目をあびている分野である。この中には，多くの基本的分析手法が含まれている。

次のようなものの中に参考になるものが多い。

・ＩＢＭ社の ＩＰＴ

・ミシガン大学の ＩＳＤＯＳ

・ＴＲＷ社の ＳＲＥＭ

・Ｓof Ｔech社の ＳＡＤＴ

・ＢＯＥＩＮＧ社の ＳＳＤＭ

これらはシステム分析結果の集約としてのシステム仕様を，どのように表現すべきかという点に焦点をあてている。

(7) 分析結果の整理及び表現

分析結果の整理表現には次のような方法が有効なヒントを提供する。

① ディシジョン・テーブル

ディシジョン・テーブルは，複雑なロジックを記述するのに有効な手段である。流れ図だけでは一読して理解が困難なロジックでも，ディシジョン・テーブルを併用することにより，容易に理解することが可能となる。

② マトリックス・デザイン技法

マトリックス・デザイン技法の簡単な例は縦軸にインプット・エレメントを記述し，横軸にアウトプットやファイル名を記述したマトリックスを用いてインフォメーションの整理を行うものである。こうすることにより，システム設計の完全性を保証するわけである。ＩＢＭの ＴＡＧ（Time Automated Grid system）はこのマトリックス・デザイン技法を活用した１例である。

③ ＨＩＰＯ

ＨＩＰＯは図形表示を多用した文書化であり，文章表示を少なくし，図的一覧表示により，個々の機能を明確にする。あわせて階層構造を導入している。

④ 川喜田二郎のＫＪ法

Ａ型図解，Ｂ型文章化

⑤ ナドラーのワークデザイン

ホッパー図によるシステムの機能階層の説明は，システムのもつ階層構造に着目したものである。

指導上の留意点

(1) 良きシステム分析者は彼が分析しようとする対象について学ばねばならないだろう。又その対象をシステム化するに要する手段，手法について知らねばならない。両者が相まち完全なシステムが可能となるのである。第3章でとり上げた手法はこの辺の事情を反映している。すなわち断片的な技術教育にしない。分析対象物との関係を出来るだけ具体的に取り入れる努力をするということである。

(2) したがって個々の項目についての取り上げ方は，許容される時間配分の中で工夫が必要とされる。例えば，

- 技術的な堀り下げはどこまで必要か。
- 全体系をどう説明するか。
- 重点的な取り上げ方をするとすればどこを中心とするか。
- 具体的事象として何を与えたら適切か。

などである。

(3) 又，指導上ある程度は避け得ない偏りについても考慮する必要がある。

例えば，いわゆる

- 事務分析系統
- ＩＥ的分析系統
- ＯＲ的分析系統

などである。理想的に言えば，これらの狭い系統を脱したところの「システムズ・アプローチ」で構成されることが期待される。

参　　考

1．帰納法と演繹法

システム分析，システム設計を進めるための方法として，帰納法と演繹法とがあることが一般にみとめられる。帰納法は現状の調査分析から出発する設計方法である。演繹法は現状にとらわれることなく，対象システムの本質的な機能や役割を追求して，合目的な理想システムをえがいてその実現を求める設計方法である。この二つの態度を比較すると，問題発見，問題形成の段階では，帰納法（分析）に重点があり，問題解決の段階では演繹法（設計）に重点が

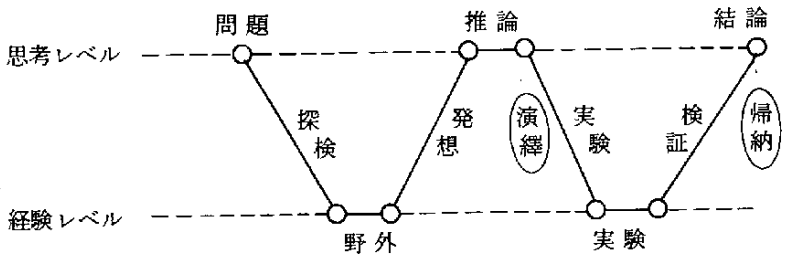
あることが多い。

帰納法的設計は，リサーチ・アプローチと呼ばれる経験的方法でもあり，対象が経験的に安定化，ルーチン化している場合に有効である。事務処理システムの設計などの分野にその例が多い。演繹的設計は，デザイン・アプローチと呼ばれる理想システムを想定する一つの方法であり，技術開発分野にその例が多い。

新しいプログラム開発技法の考え方に対する傾向の中には演繹法的に設計を進める思想が含まれている。例えば top-down integration and testing の考え方はシステム設計の進め方における演繹法的方法に共通する性格をもっている。これらの新しい傾向は各種のシステム開発技法の中に標準化されて採用されつつある。例えば，ナドラーのワークデザイン，IBM社のIPT，Sof Tech 社の SADT，BOEING社のSSDMなど。

帰納法と演繹法の両者を夫々の性格にあわせて適所に使用することを提唱している例として川喜田二郎のKJ法がある。

W型問題解決ステップ



KJ法の特色はここに示す後半部分の帰納，演繹の使いわけの提唱部分ではなくて，むしろそれより前の段階の現状把握，本質追求のラウンドにある。これに「探検」「発想」という独得の性格づけをした点である。

川喜田二郎の文化人類学分野での探検の体験にもとづく各種の提案は，示唆に富むものが多く，システム分析に対するヒントとしても有益な考え方を提供している。「発想法」その他の多くの著作に発表された方法は，KJ法と呼ばれる体系的な方法論として整理され，企業内での問題発見やシステム分析に活用されている。

2．IE的分析手法

IEの源流はティラーの時間研究であるが，この方向の分析手段を詳細化した例として次のような例があげられる。

・ギルブレス夫妻が考察したサーブリック分析作業を構成する動作要素（search，find，

select 等）に夫々名称と記号を与えてこれによって，分析対象の作業を記述することにによって分析を進める。

・ワーク・ファクター分析

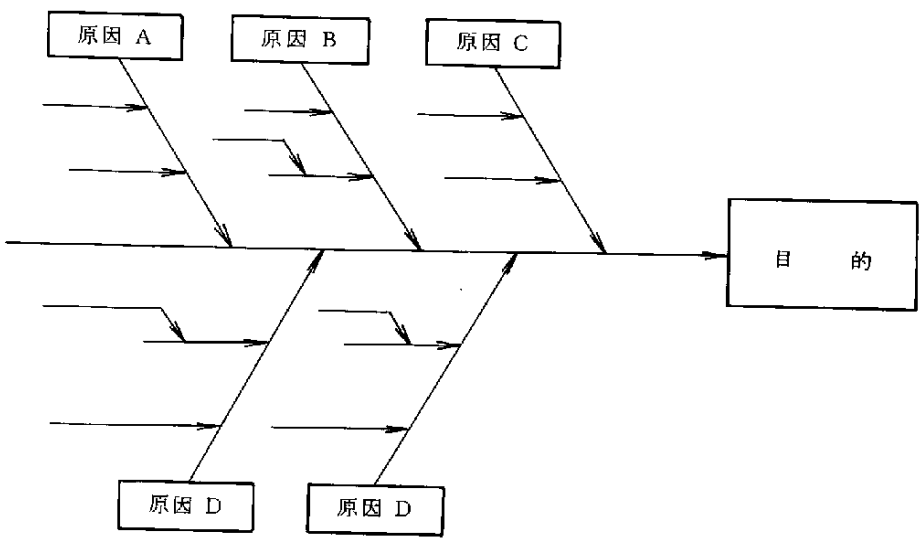
・マンデルのメモ・モーション分析などの分析もこの系統の延長である。

・マクドノウの「情報の経済学と経営システム」（第1章参考文献）の中に述べられている経営システムの記述用述語体系の考え方も，ギルブレスの考え方をさらに大きな範囲に拡大したものと見なすことができる。

3．分析の図式表示

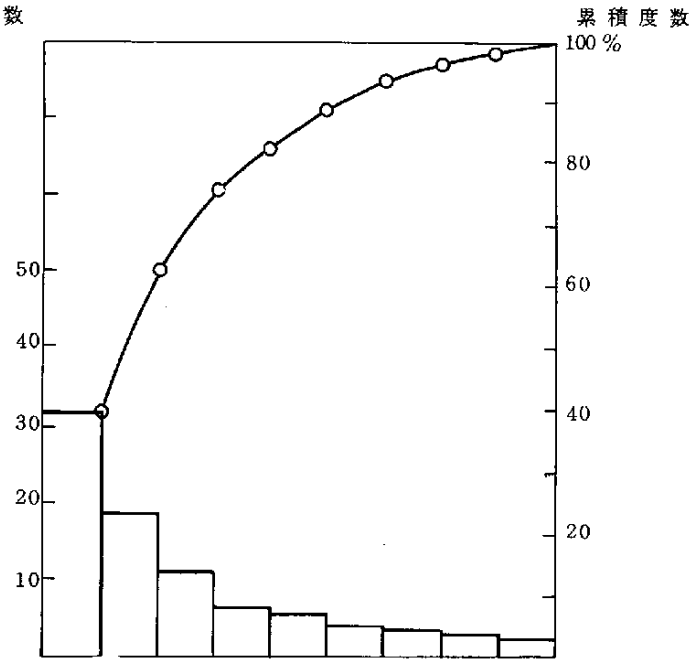
システム分析の表現には概念の明確化のために各種の図式表示が用いられる。次にその2，3の例を示す。

・特性要因図

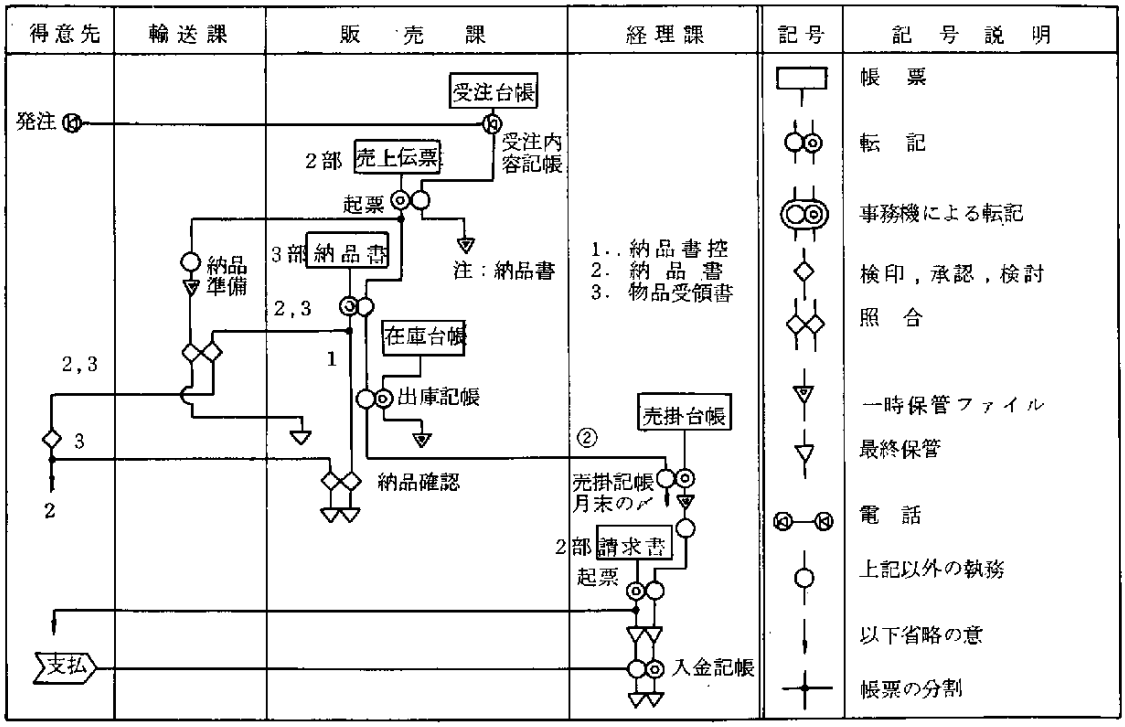


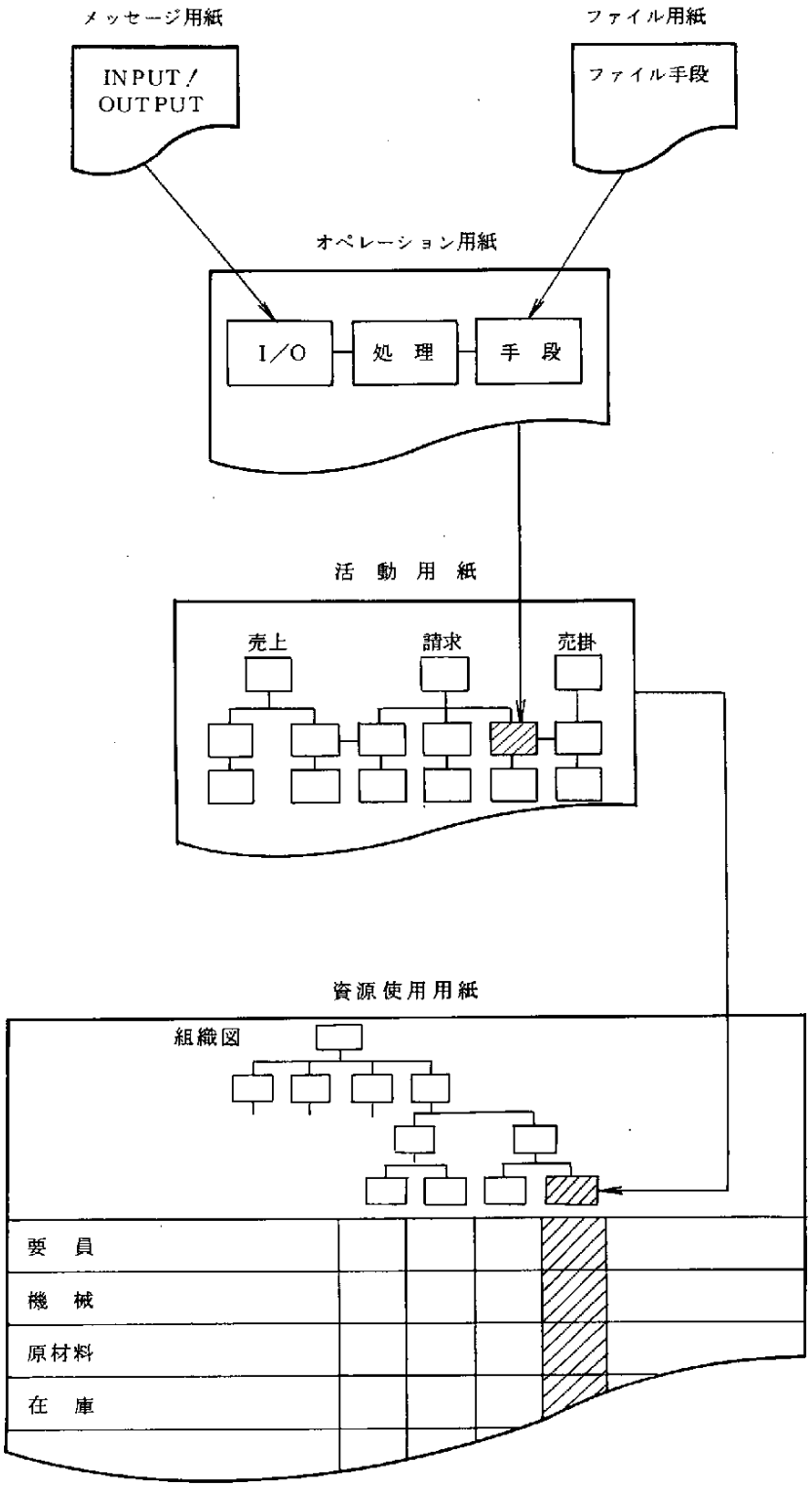
・パレート図

問題が発見されると，次はどこから改善作業に着手するかということになる。とかく枝葉末節にとらわれがちになる点を，パレート図を描くことにより正しい着手順序を知ることが出来る。



・事務分析の図式例




メッセージ用紙
INPUT／
OUTPUT
ファイル用紙
ファイル手段
オペレーション用紙
I／O
処　理
手　段
活　動　用　紙
売上
請求
売掛
資源使用用紙
組織図
要　員
機　械
原材料
在　庫

SOP文書化の模式図

4. 価 値 分 析

価値分析VA（Value Analysis）は，L.D.Milesを中心にして，アメリカのゼネラル・エレクトリック社で始められた価値分析活動を緒にしてひろまった考え方である。

不必要なコストを分析的に明らかにすることによって企業活動が欲するコストダウンを達成する技法である。価値を使用価値（use value），貴重価値（esteem value），コスト価値（cost value），交換価値（exchange value），などと分析的にとらえて，より安いコストで同等の性能を追求する。「何か他にそれに代るものがないか？」と質問をすることによって目的を達しようとする手法である。その手順として，

① 機能を明確にする。

② 比較による機能評価。

③ 価値の代替品を考える。

という段階によって分析作業をおこなうIE的手法である。

マイルズは，次の13の着眼点をあげている。

ⅰ）一般性を排除する

ⅱ）すべての利用できるコストを考える。

ⅲ）最善の情報源のみをたよる。

ⅳ）まず一切を破壊してから創造し洗練する。

ⅴ）真の独創力による。

ⅵ）障害物を明らかにしてそれを取除く。

ⅶ）専門家の知識を活用する。

ⅷ）重要な公差を金額換算する。

ⅸ）業者の機能製品を利用する。

ⅹ）業者の熟練や知識を利用する。

ⅺ）専門的生産工程を利用する。

ⅻ）利用する標準品を活用する。

ⅹⅲ）自分の金だったらどうするかを考える。

価値工学VE（Value Engineering）は上記のVAを発展させた技術である。

参 考 文 献

〔1〕 Couger, J.D. and Knapp, R.W.,"Systems Analysis Techniques," Wiley, 1972

〔2〕 Davis, G.B. and Eyerest, G.C.,"Readings in Management Information Systems," McGraw-Hill, 1976

〔 3 〕Grindley, K.,"Systematics : A.New Approach to Systems Analysis"., Mc-Graw-Hill, 1975

〔 4 〕Hare, V.,"Systems Analysis : A. Diagnostic Approach," Harcourt Brace and Janovich, 1976

〔 5 〕IBM Corporation,"Business Systems Planning : Information Planning Guide," Application Manual GE 20－0527－1, IBM, 1975

〔 6 〕Pollack, S.L., Hicks, H.T.Jr. and Harrison, W.J.,"Decision Tables : Theory and Practice," Wiley, 1971

〔 7 〕マクミラン・ゴンザレス著 （野々口格三・守谷栄一・丹羽明訳） 「システム分析」 鹿島出版，1975

〔 8 〕C.W.チャーチマン著 （竹内靖雄訳） 「システム科学への挑戦」 竹内書店，1970

〔 9 〕企業会計臨時増刊「システム科学と会計」 中央経済社，1970，22巻13号

〔10〕深尾毅 「システムの数理」 筑摩書房，1975

〔11〕松田正一 「システム理論序説」 オーム社，1971

〔12〕日本IBM社 「SOPの考え方」「SOPの手法」「SOPの手法と文書化の手引き」，1972年翻訳

以上の参考文献は1章，2章の参考文献に対する追加として引用したものである。システム分析に対しては，1章，2章の基本的な文献を同じく参考にする必要がある。

〔13〕Rogers, C.,"Client Centered Therapy," Houghton-Mifflin, 1951

〔14〕産業訓練特別資料№9「Active Listening」 日本産業訓練協会，1972

〔15〕S.I.ハヤカワ 著 （大久保忠利 訳） 「思考と行動における言語」 岩波書店，1951

〔16〕S.I.ハヤカワ 著 （四宮満 訳） 「言語と思考」 南雲堂，1972

〔17〕井上・福沢・平栗 著 「一般意味論」 河野研究所出版部，1974

〔18〕斉藤美津子 「話しことばの科学」 サイマル出版社，1968

〔19〕斉藤美津子 「きき方の理論」 サイマル出版社，1972

〔20〕NHK 「アナウンサー読本」 日本放送出版協会，1955

〔21〕松野善弘 「正しい話し方」 明治書院，1970

〔22〕D.C.バーンランド著 （西山千訳） 「日本人の表現構造」 サイマル出版社，1973

〔23〕ジョン・コンドン 「ことばの世界」 サイマル出版社，1972

〔24〕内川・岡部・竹内・辻村編 「現代の社会とコミュニケーション」 東京大学出版会，1973

〔25〕鮑戸弘　「コミュニケーション」　筑摩書房，1972
〔26〕入谷敏男　「ことばの生態」　日本放送出版協会，1968
〔27〕芳賀純・安本美典　「ことば，文章」　大日本図書，1968
〔28〕荻昌朗　「耳学問」　日本経済新聞社，1972
〔29〕野田孝　「会議術」　日本経済新聞社，1971
〔30〕ロバート・L．ホワイトサイド著　（石川弘義訳）　「フェイス・ランゲージ」　読売新聞社，1975
〔31〕ジョージ・バード著　（桑名一央訳）　「絵と図形による伝達技術」　産業能率短大出版部，1973
〔32〕G.I.ニーレンバーグ・H.H.カレロ著　（上田敏晶訳）　「ひとの心を読む技術」　日本生産性本部，1973
〔33〕J.L.アラングレン著　（藤竹暁訳）　「人間コミュニケーション」　鹿島出版，1971
〔34〕加藤秀俊　「自己表現」　中央公論社，1970
〔35〕加藤秀俊　「取材学」　中央公論社，1975
〔36〕川喜田二郎　「野外科学の方法」　中央公論社，1973
〔37〕板坂元　「日本人の論理構造」　講談社，1971
〔38〕入谷敏男　「ことばと人間関係」　講談社，1971
〔39〕入谷敏男　「ことばの生態」　日本放送出版協会，1968
〔40〕鈴木孝夫　「ことばと文化」　岩波書店，1973
〔41〕川本茂雄　「ことばとこころ」　岩波書店，1976

面接の技術に関連する文献は無数にある。米国の出版物の中には，カーネギーの（Public Speaking and Influencing Men in Business ）に代表されるような口頭コミュニケーションをとりあつかった本が非常に多いが，ここではそれらは除いて，コミュニケーションを考える上で興味ある資料をあげた。

〔42〕梅棹忠夫　「知的生産の技術」　岩波書店，1969
〔43〕渡部昇一　「知的生活の方法」　講談社，1976
〔44〕川喜田二郎　「発想法」　中央公論社，1967
〔45〕川喜田二郎　「続発想法」　中央公論社，1970
〔46〕オズボーン著　（上野一郎訳）　「独創力を伸ばせ」　ダイヤモンド社，1958
〔47〕W.J.J.ゴードン著　（大鹿譲・今野正訳）　「シネテックス」　ラテイス社，1964

〔48〕ヴァンファンジェ著　（加藤八千代訳）　「創造性の開発」　岩波書店，1967
〔49〕中山正和　「発想の論理」　中央公論社，1970
〔50〕中山正和　「カンの構造」　中央公論社，1968
〔51〕多湖輝　「企画力」　光文社，1969
〔52〕北川敏男編　「創造工学」　中央公論社，1971
〔53〕小林茂　「組織蘇生学」　ごま書房，1971
〔54〕小林一作　「情報選択の技術」　日本経済新聞社，1969
〔55〕井上如　「情報のよみ方」　日本経済新聞社，1972
〔56〕加藤秀俊　「整理学」　中央公論社，1963
〔57〕加藤秀俊　「情報行動」　中央公論社，1972

情報を的確に把握するための手法は，発想段階から整理段階までにわたって，数多くの工夫がある。文化人類学の分野では，探検資料の整理に対する必要からこの問題に対する研究も盛んであって，多くの重要な資料が得られている。ＫＪ法の開発も，このような活動の副産物といえる。直接的な参考文献ではないが，２，３の探検記の例を引用しておく。

〔58〕梅棹忠夫　「モゴール族探検記」　岩波書店，1956
〔59〕中根千枝　「未開の顔・文明の顔」　中央公論社，1959
〔60〕泉靖一　「フィールド・ノート」　新潮社，1967
〔61〕川喜田二郎　「ネパール王国探検記」　光文社，1956
〔62〕グレアム・クリバリー著　（斉藤仁夫・亀谷悟郎訳）　「重役室の中の呪術」　講談社，1972
〔63〕本多勝一　「極限の民族」，朝日新聞，　1967
〔64〕板坂元　「考える技術・書く技術」　講談社，1973
〔65〕大出晃　「日本語の論理」　講談社，1965
〔66〕清水幾太郎　「論文の書き方」　岩波書店，1959
〔67〕八杉龍一　「論文・レポートの書き方」　明治書院，1971
〔68〕鈴木孝夫　「閉ざされた言語・日本語の世界」　新潮社，1975
〔69〕本多勝一　「日本語の作文技術」　朝日新聞社，1976
〔70〕山中秀男　「技術文書のまとめ方」　東洋経済新報社，1971
〔71〕丸谷才一　「文章読本」　中央公論社，1977

書く技術に関する文献を追加した理由は，システム分析とは，記述することであるとい

う着眼点からである。

〔72〕L.D．ミルズ著　（産業能率短大訳）　「価値分析の進め方」　日刊工業新聞社，1962

〔73〕三浦宏文　「システムと評価」　共立出版，1969

〔74〕エンリッヒ・ヤンツ著　（マネジメントセンター訳）　「技術予測」　マネジメント・センター，1968

〔75〕牧野昇編　「技術予測入門」　日刊工業新聞社，1970

〔76〕斉藤嘉博　「予測」　日科技連，1970

〔77〕ＮＡＳＡアポロ計画部編　（渡辺茂訳）　「情報管理システムＦＡＭＥ」　東洋経済新報社，1970

その他の個別の分析手法に関しては，各種のハンドブックを参考にすればよい。次のようなハンドブックがある。

・情報工学ハンドブック	森北出版
・電子計算機ハンドブック	オーム社
・ＭＩＳハンドブック	日本経営出版
・共立総合コンピュータ辞典	共立出版
・情報処理ハンドブック	オーム社
・コンピュータ・マネジメント・サイエンス・ハンドブック	オーム社
・システム・エンジニア・ハンドブック	オーム社
・ＥＤＰシステム設計ハンドブック	日刊工業新聞社
・ドキュメンテーション・ハンドブック	東京電機大学出版部
・ＯＲ事典	日科技連
・研究開発ガイドブック	日科技連

第4章　システム代替案の検討

キーワード

代替案 (alternatives)，目的 (objectives)，要望 (needs)，制約 (constraints)，PPBS (planning programming budgeting system)，評価 (evaluation)，手動と自動(manual versus antomated)，汎用と専用 (generalized versus tailored)，分散と集中 (disaggregation versus aggregation, decentralized versus centralized)

目　　標

本章では，システム分析の基本要素のうち，システムの目的と代替案の考え方について，さらにくわしく学習する。

システム分析の中心的な方法に，代替案の比較検討というアプローチがあるが，この方法がなぜ重要であるのかという点をさらにつっこんで，システムのもつ不確実性という本質を認識させる。

代替案という考え方が意識的に採用され，さらに明示的な方法にまで作りあげられた経緯として，PPBSやランド・コーポレーションの研究のはたした役割が大きいが，これらの概要もあわせて学習し，これによって，システム分析の考え方は，完成されたものではなく，いままさに作られつつある進行形のものであることを認識させる必要がある。即ち，学習のあり方としては，出来あがったものを受動的にうけ入れる態度でなく，なぜ代替案を考えなければならないかという研究的態度がまず要求される。

各種の代替案を，システム開発の各段階や各要素に対応させて例示し，これらの検討を通じて，システム評価の視点の次元が多面的であることを認識させる。システムの採用に際しては，代替案の評価が，仮定や判断の立場に依存して大きく変ることを理解させ，科学上の問題の解のように一義的に決定できるものと異なる面があることを学習させる。これによって，システム分析における問題構造の理解が重要であることを学ぶことが，本章のねらいである。

内　　容

4.1　システムの目的と代替案

問題の本質は分析のプロセスで次第にあきらかにされてゆく。評価基準についても，はじめから明らかなのではなく，分析によってそれが発見される。目的と手段とは相互作用関係にあ

る。一定の目的に対して最善の手段を選択するといった単純なプロセスではないのが普通である。表面的に代替案を選ぶように見える問題も，その実は目的の選択の方が本質的であることが多い。システム分析とは，このような目的と手段の関係をあきらかにすることである。

ランド・コーポレーションでシステム分析をまとめたクエードは，システム分析を次のように定義している。

"適切なフレーム・ワークを使い，意思決定者の直面するあらゆる問題を調査し，目的と代替案を探究し，代替案とそのもたらす結果を比較することにより熟慮した判断と直感的洞察をもたらし，行動進路の選択を援助する体系的なアプローチである。"

目的の選択は決定的な重要性をもっているが，代替案の設計もその分析と同程度の重要性をもっている。

4.2 代替案の開発と選択

システムの目的が設定され，それが分析的に記述されると，代替案を列挙することになる。代替案の列挙は種々の異なる観点から考えなければならない。機能の代替案，活動の代替案とより細分化された代替案を並行的に開発する。各種の代替案を列挙することは，システムの不確実性に対する弾力性を確保するためにも必要である。

代替案の開発は，はじめから最適なものが得られるとは限らない。その代替案を分析して考察することによって，さらによい代替案に近づいてゆくというプロセスが必要になる。これは開発の不確実性によるものである。

評価基準についても不確実性が存在する。測定尺度の異なる有効度を出力するような代替案について判定しなければならない時に，このような状況になる。

評価基準には，

- 最適化基準
- 満足化基準

がある。最適化基準の場合には，あらゆる代替案を比較することが可能な一組の最適基準があること，そして問題の代替案がこの最適基準によって他の代替案に対して優れていることが要求される。

これに対し，満足化基準は，最低満足な代替案を記述する１組の満足基準があり，問題の代替案がその満足基準を満たすか，それを越える場合に採択される。

費用と効果の視点から見た代替案の選択モデルを図４－１に示す。

代替案の選択問題をオプトナーはシステム変換のプロセスとしてとらえて次のように３種類の代替案があると分析している。

(1) 現行の方法を続ける

現行システムが，すべての出発点となる。又，種々の代替案を考慮した後，あらためて現行へ回帰し，現行システムの利点，問題点を再考する。

(2) 1つずつ変換する

システムを構成する要素を1つずつ変換していく方法は，最も良くとられる代替案の例である。あたかも，故障の部品を1つずつ取り替えていくアプローチに似ている。

(3) 新システムを設計する

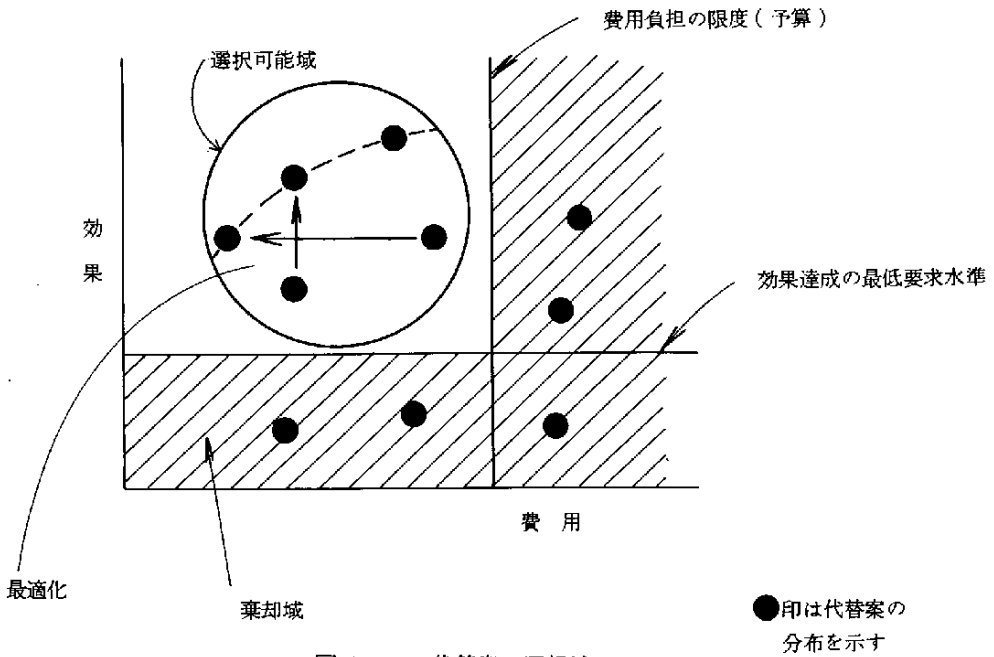


図 4－1　代替案の選択域

4.3 代替案の類型

代替案とは何かを，より具体的に把握するために，代替案に関する各種の類型を列挙し，その内容をさらに細分して，代替案の比較検討をいろいろな視点からおこない，システム分析における代替案の重要性を認識させることも必要である。

代替案の類型とは例えば次のようなものである。

- 自動と手動
- 管理者の要求に関して，リスポンス効率など
- 言語について，自然言語と形式言語
- 汎用と専用
- 図形と文字と音声

・要求報と配布報

・情報検索と問題解析

・集中化と分散化

・データの取扱いの各種

・機能要求と経済的制約

・人間による判断と自動判定

さらに細分化して検討すると，システム・ライフサイクルの各局面に対応して，さらに多様な代替案がある。

・目標の差異によるもの（性能，費用）

・開発上の制約の差異によるもの（要員，期間）

・資源制約の差異によるもの（ハードウェア，ソフトウェア）

・インプットの内容，方法，サイクルの差異によるもの

・ファイルの形式，媒体の差異によるもの

・アウトプットの内容，方法，サイクルの差異によるもの

・ハードウェアの結合，効率の差異によるもの

・ソフトウェアの言語，形式の差異によるもの

・回線の方式の差異によるもの

・端末の種類の差異によるもの

代替案の類型の分類は，上記の各要素の組合せで，多様な複合的なものとなる。個々の代替案を局所的に比較検討していては，判断が不可能になることが多い。このような選択問題は一義的に定式化出来ないことが多い。システム分析に対する需要は，このために発達したと考えられる。小規模な問題について代替案の比較をおこなって，選択問題の困難さを実習させることは有益である。

4.4 ＰＰＢＳとシステム分析

代替案の選択問題は，システムの規模が大きくなると，その困難さが幾何級数的に増加する。それに対する解決案として整備されたシステムがＰＰＢＳである。ＰＰＢＳとはPlanning-Programming-Budgeting Systemの頭文字で，直訳すれば，基本計画・任務別計画・予算編成システムということになる。ミサイルの出現によって戦略概念に大きな変化がおこったため，米国の国防省が予算改革のために採用したシステムである。従来の陸・海・空の三軍に対する別個の予算ワクのわりつけ方式を改め，長期計画の下で，予算や資源を合目的，効率的に配分する方式である。このシステムの採用には，マクナマラ長官の依頼によって，ランド・

コーポレーションのチャールズ・J．ヒッチがその案の作成をおこなった。この経緯からも推測されるように，PPBSには，ORの手法やシステム分析がその基本思想として存在している。換言すれば，システム分析の手法を予算編成の運用形式の中にもち込んだものがPPBSである。

PPBSの基本計画のプロセスを図式化すると次のようになる。

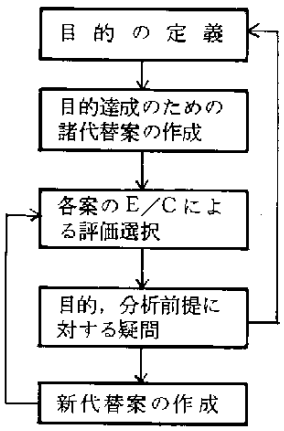


図4－2　PPBSの基本計画のプロセス

この図によってわかるように，有効性と費用（E／C）という尺度による評価，目的の再検討というループによって，さらによい代替案に接近するというプロセスは，まさにシステム分析そのものといえる。

PPBSの任務別計画は，基本計画の重点が目標設定におかれていたのに対して，方策手段の決定に重点がおかれている。即ち代替案の開発そのものである。そのプロセスとしては，計画要素の細分化による費用分析と，その再積上げによる計画費用の見積りが骨子になっている。任務別計画に採用されている諸案はこの章での学習項目に合致するものが多い。

予算計画に使用される文書等も参考になるものが多い。代替案のモデル化，モデル選択の評価基準等にさまざまな工夫がこらされている。

PPBSは国防省で採用された後，行政全部門に実験的に導入されるにいたっている。行政機構に組込まれるにともなって，システムとしての硬直化も発生するであろうし，中央集権化によっておこる欠陥も予想されるが，ここではそれらの現象面の問題ではなく，代替案の列挙と選択というPPBSの特質について技術的な検討をおこなうことを学習の焦点とする。特に費用効果分析の精密化はPPBSの普及とともに脚光をあびた課題である。

参　　考

1．代替案の具体例

(1)　目標と代替案

目標のたて方により，システム計画案の方向は大きく異なるものになる。

例えば次の様な例である。

（例　1）

マネジメントの要求が，収益の拡大であるときに，目標を，(a)販売の拡大，(b)製造コストの削減，(c)間接部門コストの削減，(d)販売直接費の削減，のどこに設定するか。システム化の対象をどこまで取り込むか。

(2)　与件条件と代替案

与件条件の認識の仕方や，資源制約条件の認識の違いの下での代替案の選択可能性に変化がおこる。例えば次の様な例である。

（例　2）

社内には one-for-one 変換ですまそうという雰囲気が強い。このとき，(a)社内経理制度の改変。(b)社内報告制度の変更，(c)コード体系の変更，など，どこまでとり込んで考えたら良いか。与件と思われたもの自体も，システム計画案の代替案となり得ること。すなわち，絶対的な意味での与件条件とは何か。

（例　3）

資源制約という与件条件のもつ絶対性と可変性の範囲の認識の違いにより，代替案は異なるものになる。

例えば，次の資源の利用可能性と代替案セットとの関係について考えてみよう。

- 利用可能なシステム開発要員数
- 利用可能な機益
- 開発工期

資源制約条件の絶対性とは何か。他の資源流用による代替案作成の可能性も考えてみるとよい。

(3)　インプット設計における代替案検討

①　インプット内容の代替案

- インプット項目／非インプット項目

　ファイル参照項目は何にするか。

- 固定長／可変長

・数字／英字／カナ文字

・意味ありコード／意味なしコード

② インプット方法の代替案

下記のインプット方式の長所と短所，コストの費用別概算見積り，処理能力，チェック方式とその精度は，代替案検討のきめ手となる。

・カード

・紙テープ

・マークカード，マークシート

・OCR，MI

・カセット・テープ

③ インプットの運営方法の代替案

・発生個所別個別インプットか，集中インプット方式か

・インプット・サイクル

・インプット時期

・インプット収集方法

人が運ぶ／社内便／郵送／テープ送付／テレックス／テレタイプ／伝送

・チェック方式

・原始データによるチェック

・送付時／受付時

・機械によるチェックとその程度

・訂正方法

・訂正伝票による方法

・再インプット／リラン

・訂正サイクル

(4) ファイル設計における代替案検討

① ファイルの種類決定における代替案

・マスター・ファイル／トランザクション・ファイル

・各ファイル種類別にファイルする項目

・前提となるデータ量，発生サイクル，ピーク時負荷の考え方

② ファイル項目

③ 専用ファイルか統合ファイルか

次の観点からの評価と選択

・処理効率

・オペレーションの難易度

・プログラムの難易度

・データ管理

・データ格納効率

④ ファイル処理方式の代替案

・シークエンシャル処理

・ランダム処理

ファイル活動率からの評価と選択

⑤ ファイル編成における代替案

下記ファイル編成方式の特徴，すなわちファイル領域の使用率，プログラムでの取扱いの難易度，データの追加削除方法，ファイルの処理時間などはファイル編成を決める要因となる。

・シークエンシャル編成ファイル

・ランダム編成ファイル

・インデックス・シークエンシャル編成ファイル

⑥ ファイル媒体の代替案

下記ファイル媒体については，それぞれ代替案の候補となり得る。それぞれの媒体の長所と短所，コスト，容量，処理能力，ファイル編成の特徴，保管，オペレーション等は媒体選択に必要な項目である。

・磁気テープ

・磁気ディスク

・磁気ドラム

⑦ ファイルの運営に関する代替案

次の項目について代替的考え方がとれる。

・保存する日数

・世代管理

・保管方法

(5) アウトプット設計における代替案検討

① アウトプットの内容における代替案

種々の代替案が考えられる。

・使用目的別アウトプット

管理用か，事務担当者用か，操業用かその場合の

- 帳票種類
- 使用部署
- 時強，サイクル
- アウトプット方法

- 全数アウトプットか／サマリー使用か／例外管理用アウトプットか
- アウトプット帳票様式
  - フォーム印刷
  - 応用用紙

② アウトプット媒体に関する代替案

- 帳　票
- CRT
- COM
- プロッター
- 漢字プリンター

③ レポート作成方法における代替案

- 個別にプログラムをつくる。
- 汎用のレポートジェネレータ等を活用する。
- 情報検索する。

④ アウトプットの運営方法における代替案検討

- アウトプット発送方法

  人／社内便／郵送／テープ送付／伝送

- 保管責任区分

  秘扱／普通

- 保管期間

(6) リアルタイム設計上の代替案検討

① 信頼性確保に関する代替案

リアルタイム・システム設計にあたっては，信頼性の確保が極めて重要となる。適用する業務の重要度，マシンダウンした場合の損失の大きさ，中断が許される許容時間の大小により，障害対策に関する種々の代替案を考慮する必要がある。

(a) 機器構成

コア記憶装置，処理装置，チャネル制御装置，ファイル，ファイル制御装置などの多

角的構成(polymorphism)による信頼性向上対策。

- タンデム
- 衛星方式
- 共有ファイル・システム
- 並列方式
- 二重化
- ツイン

(b) フォールバック処理

- 予備ファイルの持ち方
- 予備機益の持ち方

(c) バイパス処理

- ハンド処理
- 支店処理
- その他のバイパス処理

② 人間と機械の処理方式に関する代替案

(a) 機益構成

人間の介入の程度に応じて種々の構成が考えられるが，その特徴，対象業務特性との関係などを考慮した代替案がある。

- DDC (Direct Digital Control)
- 制御機益設定
- シークエンス制御
- オペレータ・ガイド方式
- オンライン管理方式
- ハイラキー・システム
- TSS

(b) 適用業務

人間と機械の共存システムは単なるバッチシステムの代替案と考えられる。

- 座席予約システム
- 問題向け言語
- 情報検索
- プログラム・デバッグ

③ 通信回線に関する代替案検討

(a) 伝送速度

通信速度のちがいから，A規格～Z規格がある。

(b) 通信サービスの種類

・データ通信回線サービス

・特定通信回線

・直結方式

・分岐方式

・集線方式

・公衆通信回線

・加入電話回線

・加入電信回線

・データ通信設備サービス

・特定システム

・共同利用システム

(c) 通信回線の形態

・直線回線方式

直結でセンターと結び，データが多量で応答時間の即時性が要求される場合に適している。

・分岐回線方式

1回線に複数端末又は複数集信装置を結合する。通信回線網回線料金の経済化がはかれる。端末利用率が低いこと，即時性に劣る。

・集線方式

同一地域内の多端末が要求される場合，集線装置で集約することにより，回線料金の経済化がはかれる。

・加入電話回線

公衆電話回線ネットワークの利用，伝送速度，使用頻度上の問題，などがある。

・加入電信回線

テレックス・ネットワークの利用，伝送速度，コストなど。

(d) 各種コード

④ データ伝送端末装置

・データセット

・鍵盤だけの伝送端末装置

・鍵盤印刷機

・せん孔紙テープ伝送端末装置

・せん孔カード伝送端末装置

・CRT端末装置

・磁気テープ伝送端末装置

・電算機伝送端末装置

具体的には下記特性の差異

・トラフィックの量

・動作速度

・符号化

・チェックの方式

・画像，画面利用の効果

・インテリジェント機能の有無

・取扱いの容易性

・特殊端末での安全性

・バンキング・システム

・キャッシュ・ディスペンサー

・端末バッファーの有無

・ソフトコピー

・ハードコピー

更に，最近の事例として，端末装置の多様化がある。

・POS

・インテリジェント・ターミナル

・プッシュ・ホン

(7) データ・マネジメントの代替案

・データの構造

・線型構造，木構造，網目構造

・アクセス方法

・シングル・アクセス

・マルチ・アクセス

・データ・ベース言語

・パラメータ言語

・手続き言語

参考文献

〔1〕 Quade, E.S., and Boucher, W.I.,"System Analysis and Policy Planning," American Elsvier, 1968
(香山健一・公文俊平監訳 「システム分析1」 竹内書店, 1972)

〔2〕 Hitch and Mckean,"Economics of Defence in the Nuclear Age," Harvard University Press, 1960
(前田寿夫訳 「核時代の国防経済学」 東洋政治経済研究所, 1967)

〔3〕 Optner,S.,"Systems Analysis for Business Management", Prentice-Hall, 1968.
(植木繁訳,「システム分析入門」, 日本能率協会, 1969.)

〔4〕 Martin, J.,"Telecommunications and the Computer", Prentice-Hall, 1969.

〔5〕 Martin, J.,"Programming Real-time Computer Systems,"Prentice-Hall, 1965.
(北原安定訳 「電子計算機 リアルタイム・プログラミング」 日本経営出版会, 1967)

〔6〕 Martin, J.,"Design of Real-time Computer Systems", Prentice-Hall, 1967
(北原安定訳 「電子計算機 リアルタイム・システムの設計・管理・運用」 日本経営出版会, 1968)

〔7〕 Rubin, M.,"Advanced Technology : Input and Output", Auerbach, 1970

〔8〕 Rubin, M.,"Advanced Technology : Systems Concepts," Auerbach, 1970

〔9〕 Morton, S.,"Management Decision System", Harvard Business School, Division of Research, 1970

〔10〕 Thierauf, R.J.,"Systems Analysis and Design of Real-Time Management Systems", Prentice-Hall, 1975

〔11〕 H.ザックマン著 (竹内直文監訳) 「マン・マシン・デジタル・システム」 日本生産性本部, 1971

〔12〕 宮川公男編 「PPBSの原理と分析」 有斐閣, 1969

〔13〕 デービッド・ノービック編 (福島康人訳) 「PPBSの理論と手法」 日本経済新聞社, 1969

〔14〕 恵羅嘉男 「PPBSの話」 日本経済新聞社, 1968

〔15〕 チャールズ・L.シュルツ著 (大川政三・加藤隆司訳) 「PPBSと予算の意思決定」 日本経営出版会, 1971

〔16〕チャールズ・J.ヒッチ著　（福島康人訳）「戦略計画と意思決定」　日本経営出版会，1971

第5章　システムの経済性評価

キーワード

情報の価値（value of information），費用効果分析（cost-benefit analysis），システムの評価尺度（measures of system performance），システムのコスト（cost of system），直接効果と間接効果（direct and indirect benefits），コンピュータサービスの費用（costs and pricing of computer services），費用分析（cost analysis），VA（value analysis），システム有効度（system effectiveness），システム導入の採算（auditing system installation），開発費見積（costing of system development effort），運用費見積（costing of system performance），定性的評価（qualitative evaluation），定量的評価（quantitative evaluation），評価基準（criterion，criteria）

目　　標

本章では，システム評価の考え方を堀りさげて学習する。システムの費用・効果の分析をおこなうための足場として，まず，情報の価値というものの本質的な吟味を行い，情報を経済財と見た場合の特徴を調べる。次にシステムの評価尺度については，金額的尺度にのみとどまらず，システムの持つ多様な機能に対する尺度についても説明する。システムの効果については，直接効果，間接効果の他に，計量できない効果に対する接近の仕方についても研究する。

実務面について，システム評価の体系を全体的な視点からの説明をおこなう。システムの開発過程に沿った体系を考えると，事前評価と事後評価のステップがあるが，ここでは事前評価を詳細に検討する。この際，システム有効度の考え方などもあわせて説明する。事後評価については，D2の第6章で再び学習する。

本章では，特に費用の考え方を，体系的に全体にわたって説明することが望ましい。ここで学習した知識は，D2の第2章でのシステム設計のための資料となるものである。したがって実務的知識よりも，考え方や全体の構成を中心にして説明するものとする。実務的運用については，D4の第4章で具体例を検討する。

内　　容

5.1 情報の価値

システムの経済的評価について検討するに先立ち，情報の価値そのものについて理解する。

近年，社会における情報の役割りが増大している。この情報というものは，他の経済財とどう異なるのかという基本に立ち帰った理解をする必要がある。なぜならば，情報のサービス，情報の生産，加工，流通という供給サイドの問題は，増々コストの巨大化を伴なう情報システムの供給問題に対してどう対処したら良いかという基本問題であるからである。又，情報の受け手としての情報サービス購入者，自社内情報サービス利用者の立場からの"需要"の問題も，情報ニーズの増大，利用形態の多様化のなかで，便益最大を考える必要があり，情報の価値問題として体系的な理解が必要となっているからである。

情報の価値の検討に入る前に，情報に関する基礎的な知識を整理することも必要である。情報理論にもとづく情報量の概念，情報処理の基本構成，現代社会における情報の役割などについて考察し，情報の分類に関する諸説を研究することは，以降の学習のための足場を与えることになる。例えば，次のような分類がある。（野口による分類）

(a) 不確実性を減らす情報

(i) プログラム情報

体系的知識の増加で促進された技術進歩によりもたらされた産出高の増分が，この場合の情報の価値。

(ii) データ情報（資本財的データ情報）

不確実性の支配する環境での意思決定に，情報がどれだけその期待効果を高めるかによって決まる価値。

(b) 不確実性を減らさない情報

サービス財的情報

個人が消費財的に消費する情報で，その価値は個人の効用関数に依存するもの。

上記分類に従えば，我々が取扱うのは資本財的情報であることがわかる。

資本財的情報には次のような特徴がある。

・情報の生産では，同一情報を生産する場合においてはその限界生産コストがゼロである。一旦生産された情報は複製ができ，ファイル・コピーとか，ハード・コピーのようなわずかな費用で供給される。

・流通は不可逆的であること。

一旦取消された情報は取りもどすことはできない。この性格はサービス財と共通する。

・情報の生産構造は，その種類により投入産出構造が大きく異なる。

又，労働，資本の投入構造も異なること。当然のことながら投入産出構造の改善と，労働，資本などの経営資源節減が課題となる。

・情報の使用価値は外部効果が大きい。情報の使用者が既にどれだけ当該情報についての知

識をもっているか，情報の活用についてどれだけ強いインセンティブを持っているかにより使用価値が異なること。

・情報財の分割は出来ない。簡易に言えば半分の情報は，全体の価格の半分の価値を持ち得ぬこと。

資本財的情報の中には，財務的にも資産であるものがある。しかし，一般的には経費として扱われることが多い。

情報を経済問題としてとらえる必要がある場合には，当該情報を分類した後，その生産構造，需給構造，流通の仕方を基本にさかのぼり分析する姿勢を身につけさせる。

モートンは，情報の意思決定に及ぼす力に注目し，情報の仕様を表５－１のように整理している。

表５－１　情　報　の　要　件

情報の性格	オペレーショナル コントロール　マネジメント コントロール　戦略的計画
情報の源泉	内部情報 ⟷ 外部情報
情報の範囲	定義されており狭い ⟷ 非常に広い
情報集約の程度	ディティル一件別情報 ⟷ アグリゲイト集約情報
情報の時間的展開	履歴情報 ⟷ 未来情報
情報の新しさの必要性	最新情報 ⟷ 非常に古い
要求される情報の精度	高精度 ⟷ 低精度
情報使用の頻度	非常に高頻度 ⟷ 低頻度

必要とされる情報は，情報処理カテゴリーの属性と，その場合の情報の性格を認識することにより規定される。

情報の価値は，その受けとり手によって相対的に変化する。

5.2　システムの評価基準

システムの費用効果分析に入る前に，まず評価の基準となるものは何かについて説明する。

(1)　経済的評価尺度

費用効果分析では最終的に金額で計測された尺度で評価するのが望ましい。しかし多目的な「システム」というプロジェクトの評価をする場合，計測技術的に常に単一尺度が有効とは限らない。

評価尺度を分類すると次のようになる。

(a)　金額による評価

・市場価格があるもの

・市場価格はないが管理的な価格が設定できるもの

・なんらかの型で金額換算するもの

・金額換算が困難なもの

(b) その他尺度による評価

・数量化できるもの

・数量化が困難なもの

金額によっに評価できるものについての評価尺度は次のものによる。

・利　益　額

収入から支出を引いた利益額で表わす何らかの尺度

・利　益　率

投資額に対する収益の比率を表わす何らかの尺度

・回　収　期　間

投下した投資額を収益で回収するに要する期間の長さで表わす何らかの尺度

具体的な評価方法については，効果と費用を検討した後に扱う。又，金額換算が困難なものについての考え方についても，次節以降でやや詳しく扱うことにする。

(2) 性能評価尺度

① 時間的尺度

タイムリーであることを要求される業務とその時間的尺度について説明する。

業務の例としては，

・照会から応答までの時間の短縮によるサービスの向上

・同上待ち時間短縮によって得られる回転率の向上

・処理時間の短縮によって得られる処理効率の向上

など。

尺度の例としては，

・応答時間

・アクセス時間

・経過時間

・アーンアラウンド・タイム

・処理時間

時間の逆数を用いた尺度

・処理件数

・回転数

・待ち行列長

② 正確性に関する尺度

正確性の向上によって得られる効果とその尺度について説明する。

効果の例としては，

・正確性向上によって得られる金銭的ロスの減少

・顧客信用の維持

・訂正処理業務の減少等事務の合理化

尺度の例としては，

・間違い発生件数

・訂正処理件数

・誤インプット率

・誤差率

・有効ケタ数など

③ 信頼性尺度

業務処理の安定運用のための各種尺度について説明する。

効果の例としては，

・中断による経済的損失の減少

・中断により失なわれる信用の対策

尺度の例としては，

・故障率

・MTBFなど

④ 柔軟性尺度

・例外処理の容易さ

・中断時の切替え処理の容易さ

計数的尺度は作成困難であるが，業務の中断時間により表わされる。

・システム拡張時の切換作業の容易さ

コンバージョン・コスト

コンバージョン作業人

⑤ 安全性，機密性尺度

・事故に対する安全性

・機密に対する安全性

具体的尺度は社内標準という型でもうける場合が多い。

対象別にシステムの安全性が考慮される。

- 物的施設
- 組織と要員
- バックアップとリカバリー
- 磁気テープ等記憶媒体

⑥ 余裕に対する尺度

- 稼動率
- ピークロード時余裕率
- 容量，機器台数と平均負荷率，最大負荷率

(3) 業務目的別の評価尺度について

業務システムの評価にあたっては，当該業務を評価するのに適したレベルまでブレークダウンし，きめの細かい尺度を適用すべきである。ここではケース・スタディとしてＭＩＳを評価する場合の基準と，システム監査の立場からの評価基準につき説明する。

① ＭＩＳに関する評価基準

ＭＩＳを評価する基準としてAmstutzの方法が参考になる。AmstutzのMISに関する評価基準を以下に示す。

(a) MA (Management Access)

基準データを取り出すのに要する時間

(b) IR (Information Recency) 基準

事象が発生してからデータをコンピュータに格納するまでの時間

(c) IA (Information Aggregation) 基準

データの集約度に関する基準。例えば在庫ファイルでは最も低い集約度が要求される。片や，マーケット・シェア等を知りたい場合の検索では，月間で集約されたレベルで良いなど集約度に関する基準。

(d) AS (Analytic Sophistication) 基準

単なる統計処理しか出来ないシステムか。分析や，能動的なヒューリスティックな処理まで出来るシステムか。システムの高度化を評価する基準。

(e) ＳA (System Authority) 基準

経営者のコンピュータに対する信頼度やコンピュータ部門に対する権限の委譲の程度を評価するための基準。信頼度の程度は，低いものから高いものへ次の順序で示される。

ⅰ．データの検索

ⅱ．データの合計

ⅲ. 監　視

ⅳ. 監視結果にもとづく次の行動指示

ⅴ. 行　動

ⅵ. 将来事象の予測

ⅶ. 学習（過去に学び，将来を予測しつつ，前提条件を変更し次の行動を行なう）

Amstutzは，コンピュータ・システムとしてSA基準を重視している。特にSA基準のⅵ，ⅶⅰの予測，学習レベルへの到達を望ましい水準としている。

② システム監査の立場からの評価尺度

システム監査の立場からの評価基準である。システム監査は，完成されたシステムの品質を対象とするのみでなく，システム製作過程をもその対象とし，監査範囲は広い。

監査の対象とその場合の監査の観点を以下に示す。

(a) マネジメント監査は採算性中心に行う。

・リソース・マネジメント

・コスト・マネジメント

・リスク・マネジメント

(b) セキュリティ監査は安全性中心に行う。

・エラー

・フィジカル・セキュリティー

・コンピュータの悪用

(c) 会計システム監査は信頼性中心に行う。

(d) オンライン・リアルタイム・システムや，コンピュータ・ネットワーク・システム等高度なシステムの監査は信頼性中心に行う。

(e) プライバシー保護のための個人データ監査は準拠性，機密性を中心に行う。

又，システムの企画から稼動に至る過程の全業務を監査の対象とする一般基準と，稼動時におけるシステムの品質を保障するための品質基準を以下に示す。

一般基準

・準拠性

ポリシー，法律，規定その他のルールに対する準拠性である。

・採算性

・適時性

・生産性

開発，稼動，保守における効率性

品質基準

・安全性

人為的，自然現象的事故に対する保障

・信頼性

ハードウエア，ソフトウエアの信頼性

・機密性

プライバシー，企業機密の保障

以上２つの特殊事例をあげたが，一般業務システムにおいても，業務目的にそった具体的なシステム評価尺度の設定が望ましい。

5.3 システムの効果

ここではシステムの効果を認識し，その内容を理解する。ついで計量方法を考え，数量化できないメリットについても考察を行う。

(1) システム効果の認識

システム効果の把握の仕方を理解させる。

(a) 目的との関連における効果

・主要効果

・副次効果

効果の積算において両者の区分をしておく必要がある。メリット的に限界的なプロジェクトの場合，副次効果の評価の仕方が大切となる。

(b) 直 接 効 果

システムの稼動によりもたらされる直接的な効果である。下記の例で説明する。

(i) 金額的に計測できるもの

例えば，

・情報処理サービスの社外への販売

（市場価格による）

・情報処理サービスの社内組織への内部取引販売

（社内仕切り価格による）

・コンピュータ費用，通信費用のコスト・ダウンによる経費の節減

・事務費用のコスト・ダウンなど。

(ii) 物量で計測されるが，金額換算ができるもの

例えば，

・システムによる省力化効果

事務員の削減

操業度の削減

・システムによるその他投入資源の削減

（例１） コンピュータ・コントロールによる燃焼制御でエネルギーを節減する。

（例２） 材料計画の合理化や工程管理の合理化による使用材料の原単位を削減する。

(c) 間 接 効 果

システム稼動により，間接的にもたらされる効果である。

（例） 適切な在庫管理システムが導入されたことにより在庫が削減され事務処理の要員負荷が軽減された場合を考えると，この場合の直接効果は在庫減少にともなう金利負担の減少，倉敷料の減少，事務経費の削減などであるが，間接効果としてはディリバリの迅速化による顧客サービスが向上し，誤った指図が減少するなど各種のメリットをカウントすることが出来る。

(d) 数量化又は金額換算が困難な効果

数量化又は金額換算の困難な効果については，数量化をして効果を評価するための工夫が必要である。数量化の困難な効果としては次のようなものがある。

① 迅速性

迅速な情報処理が可能になった。

② 正確性

正確な情報が得られる様になった。

③ 信頼性

システムが安定しており，中断することが少ない。

④ 安全性，機密性

天災，人災から安全である。機密の漏洩がない。

⑤ 余 裕

現在及び将来にわたり，多少の負荷の増大に対しては対処出来る。

(2) 数量化困難な効果の評価について

数量化が困難な効果をどう評価したら良いか。ここでは一つの考え方について説明し，数量化できないものに対する認識の仕方について理解させる。

数量困難な効果の評価の例として，

・情報を早く得ることによる利益をどう評価すべきか。

・正確な情報を得ることの価値をどう評価すべきか。

などがある。この場合，システムとしての総効果は次の様になる。

総効果＝効果－費用＝（数量化出来る効果＋数量化できない効果）－（変動費用＋固定費用）

ここで効果の計測の仕方に2つの考え方があることを説明する。

・保守的な計上

・積極的な計上

保守的な計上は，数量化できない効果は計上しないという考え方である。システム化により現実に，あるいは確実に，金銭として回収出来ないものはこの際計上しない。その結果，確実なリターンが期待できるものに対してのみ投資するということになる。又，回収期間の観点からも短期間で回収できる効果のみを評価するといった場合もある。

積極的な計上は，これと対照的である。計量化できないものも，企業戦略上必要なものは積極的に計上する。現在のみの観点だけでなく，将来のニーズを先行して取り込み，将来のポテンシャルもあわせ評価するといった立場である。

保守的な計上の場合は次のようになる。

数量化できない効果＝0

この場合コストに対してもリスク・ミニマムがとられる。すなわち，開発コストや機益導入等による初期費用の最小化がはかられる。これらの固定費はリスク回避の観点から，

固定費→0，ないしミニマム化

がはかられる。

保守的な計上の欠点は，効果が確実であっても，金額換算しにくい性格のものは実現しにくいことである。

次に積極的な計上の例を，グレゴリーとバンホーン〔11〕の提示した分析モデルで検討する。

グレゴリー，バンホーンによると，システム・デザイナーが考慮しなければならない6つの数量化できないものの評価に関する問題は，

① 正確性

② 状態（STATUS）に関するレポートの新しさ

③ 操業に関するレポートの新しさ

④ 予測可能性

⑤ 適切性

⑥ 結果の重要性

であるという。

情報の価値Vは，次の一般型で表現できる。

$$V = K\left[\frac{1}{F}\sum_{n=1}^{F} f(A, T, Y, n) - g(P, Y)\right] - h(A, F, T)$$

ここで，

V＝（一定期間の情報を用いて得られる価値）

K＝（正しい決定を下すことによって得られる利益）

A＝（情報の正確さ）

T＝（情報を得るまでの時間的遅れ）

Y＝（期間内に行なわれる決定の回数）

P＝（情報がなくても正しい決定が下せる確率）

F＝（情報が使われる期間）

である。Vの第1項は情報の効果を示し，第2項は情報のコストを示す。この関数の持つ特徴を記述すると次のようになる。

(a) 情報の正確さ（A），情報の時間的遅れ（T）と情報の効果の関係は，図5－1で示される。

- 情報の価値は，正確さ（A）の増加関数である。
- 正確さ（A）について考えると，その情報がなくとも正しい決定が下せる確率(P)

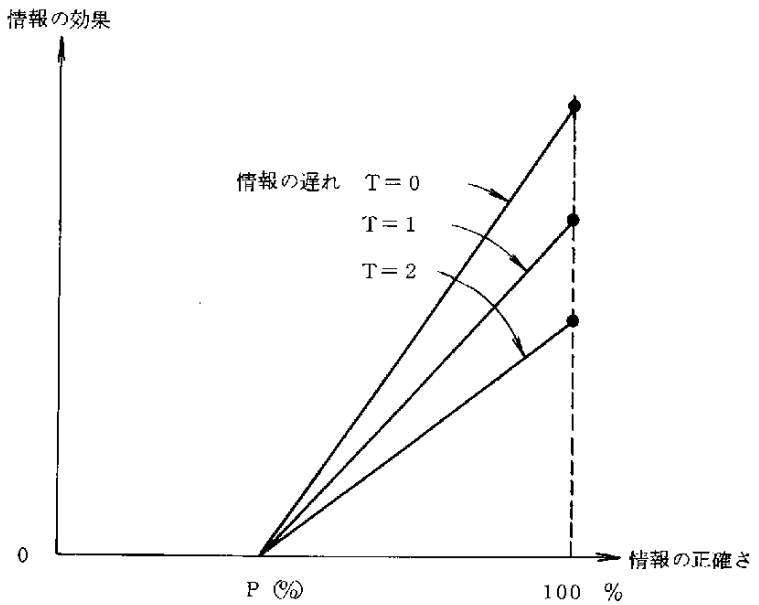


図5－1　情報の効果の線型衰退

のところまで下ったところで，その情報の効果はゼロになる。

・情報の効果は，情報の時間的遅れ（Ｔ）の減少関数である。

(b) 情報のコストに関しては，次式で示される。

$$h\ (\mathrm{A},\ \mathrm{F},\ \mathrm{T}) = \frac{C_T + C_A}{\mathrm{F}}$$

ここで，

C_T ＝（リスポンス・タイムに関するコスト）

C_A ＝（正確な情報をつくるのに要するコスト）

コスト関数の特徴としては，

・正確さ（Ａ）の増加関数である

・時間的遅れ（Ｔ）の減少関数である

ことが想定され，図示すると，図５－２（a），（b），のような曲線になる。

図５－２(a)のようなコスト曲線を想定すると，Ｔ＝０，即ち時間的遅れ０では，情報の正確さがａ点以下では情報コストが情報の効果を上まわるが，ａ点とｂ点の間で情報効果が情報コストより大となる。更にｂ点以上の正確な情報を求めると再びコストが効果を上まわる結果となることがわかる。

更に図５－２(b)のような効果，費用曲線を想定すると，情報の遅れゼロ（Ｔ＝０）の場合には，情報コストが情報効果を常に上まわり，情報の価値がないことになる。逆に，情報の遅れが１（Ｔ＝１）の場合には，情報の効果が費用を常に上まわる結果，情報の価値が明らかな場合となる。

具体例として，Ｔ＝０の場合をオンライン・システムの導入による迅速な応答を可能にするケース，Ｔ＝１をバッチ処理システムのケースとすれば理解しやすいであろう。

グレゴリーとバンホーンのモデルは，情報の効果及び費用曲線を極めて単純化しているが，情報の価値の性格を理解させるのに役立つことと思われる。なお次のような考察をするのも良い。

・情報は，時間経過とともに効果を減少させるが，この曲線は直線なのか，EXPONENTIAL な減衰曲線なのか，上に凸な衰退曲線なのかといった問題。

・正確な決定によって得られる利益の具体例は何かといった問題など。

なお，リスポンスを早めるための情報コスト，正確さを増すための情報コストについて考えると，以下の様な実際の例が考えられる。

・リスポンス時間向上のための所要コスト

・電話回線コスト

・計算機の能力増コスト

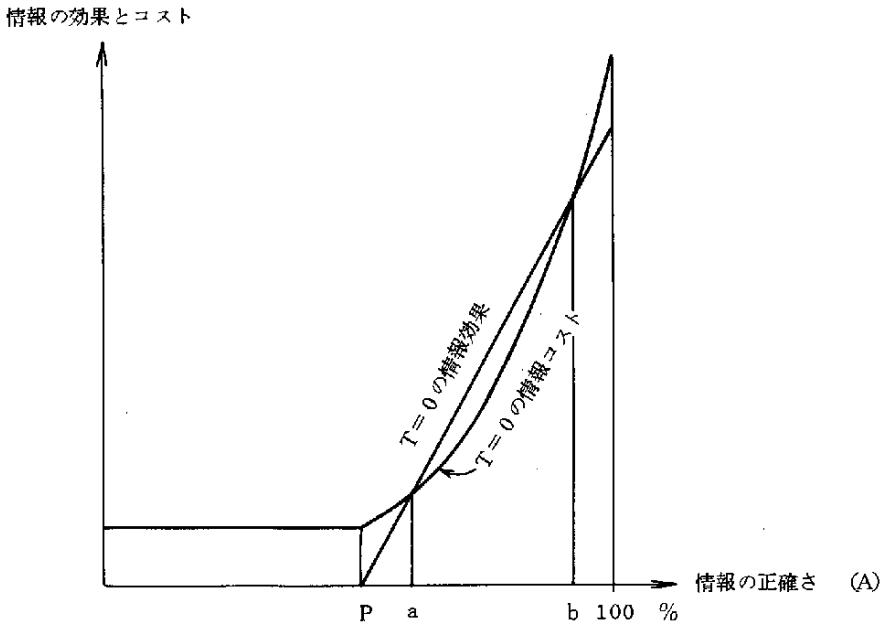


図５－２（ａ）情報効果と情報コスト

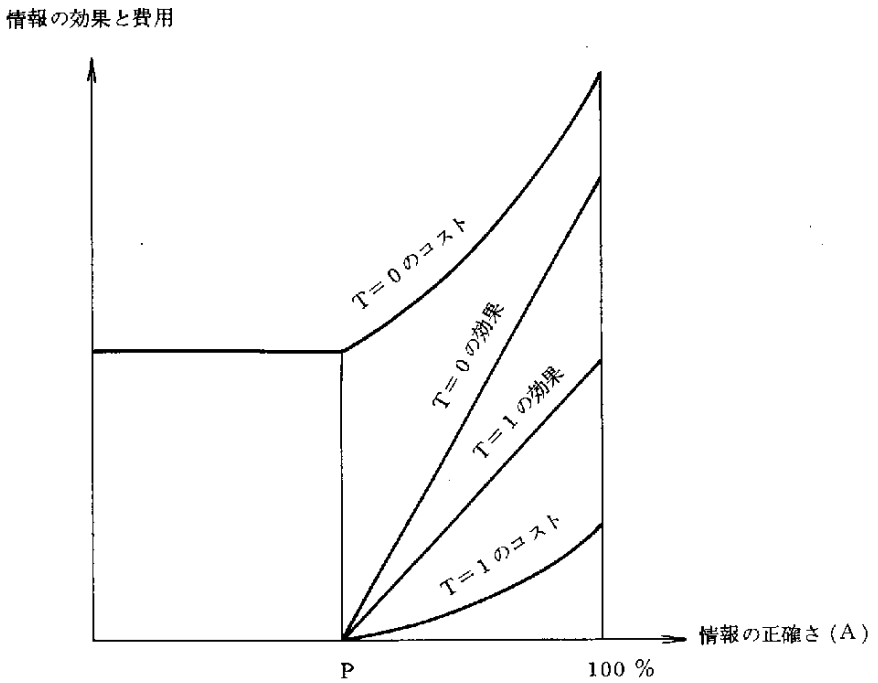


図５－２（ｂ）情報の効果と費用

・オペレーション効率化のコスト

・オペレータ増員等人件費の増加

・正確性向上のための所要コスト

・ベリファイ機器導入のコスト

・ジュプレックスするための費用

・監査するための費用

・データ・チェック等のための人件費の増加など。

5.4 情報システムの費用

システム部門の費用管理も，他の経営活動一般の費用管理と異なるところが少ない。特殊化せずにオーソドックスな分析手法を理解させることを第一に考えたい。基本を正しく学ばせるのが第一となる。

(1) 情報システムの原価計算の目的

原価計算の目的を以下の項目別に説明する。

① コストの集計

② コストの管理

③ 情報システムに関する種々の方針をきめるに必要な基礎データの提供

コスト管理については，誰が責任を持ち，誰が管理の権限をもつか，効果的にコンピュータ資源の活用をはかることの意義と必要性について説明する。又，方法として，管理のための標準値設定の必要性についても述べる。システム開発，運用における対標準値との差異分析，差異発生の原因の追求とその改善についてもふれると良い。

方針決定のための基礎データの提供については，業務推進計画，設備投資計画，要員計画との関連について説明する。

(2) コストの計算方法

原価計算についての概要を説明する。

① 費目別原価計算

② 部門別原価計算

③ プロジェクト別原価計算

の手順で計算することと，内容について説明する。

費目別原価計算は，システム部門における原価の種類（費目という）別計算で，労務費，経費，材料費といった区分がこれに当る。

部門別原価計算は，原価の発生場所別把握である。原価の発生場所別管理や，間接費のプ

ロジェクト別配賦に是非必要となる。

プロジェクト別原価計算は，プロジェクト管理のために必要となる原価計算である。又，完成したプロジェクトを販売したり，ユーザ部門に費用付替えする場合にも必要となる。

以下の項で，システム部門を対象にやや詳しく説明する。

(3) 情報システムの費目別原価計算

費目別原価の体系となる費目分類について説明する。システム部門の費目分類をその一例として示すと以下の様になる。

直　接　費

・直接材料費

・直接労務費

　　システム・アナリスト労務費

　　プログラマー労務費

　　コーダー労務費

・直接経費

　　ＴＳＳ使用料

　　外部計算機使用料

・外　注　費

　その他の直接経費（調査費，資料費等プロジェクトへ直課されるその他の経費）

間　接　費

・間接材料費

・間接労務費

　　データ部門労務費

　　オペレータ労務費

・間接経費

　　コンピュータ・レンタル料（又は，リース料）

　　コンピュータ機益減価償却費

　　その他機器賃借料

　　保険料

　　修繕費

　　備品及消耗品費

　　用紙，テープ等

　　電力料

雑　費

又，上記費目については，

・変動費

・固定費

区　分

・管理可能費

・管理不可能費

区分があることを理解させる。

(4)　部門別原価計算

部門別原価計算の目的は，

・プロジェクト別原価計算のための配賦を行うこと。

・部門別の原価管理を行うこと。

であることを理解させる。

システム部門の部門区分としては，

・システム管理部門

・データ処理部門

・システム企画部門

などがあり，

・直接部門

・間接部門

区分があることを説明する。

間接部門費は直接部門へ配賦される。配賦の基準は，コンピュータ使用時間などが使われることを説明する。

(5)　プロジェクト別原価計算

プロジェクト原価計算の手順と方法について理解させる。

①　部門別計算を行わない場合

②　部門別計算を行う場合

両者を図示すると図５－３になる。

③　間接費のプロジェクト別配賦

間接費の配賦については以下の点を理解させる。

・間接費を一括配賦するか，又は要素別配賦をするかの決定は，管理の目的による。

・間接費の配賦基準については，時間基準や数量基準等が考えられる。

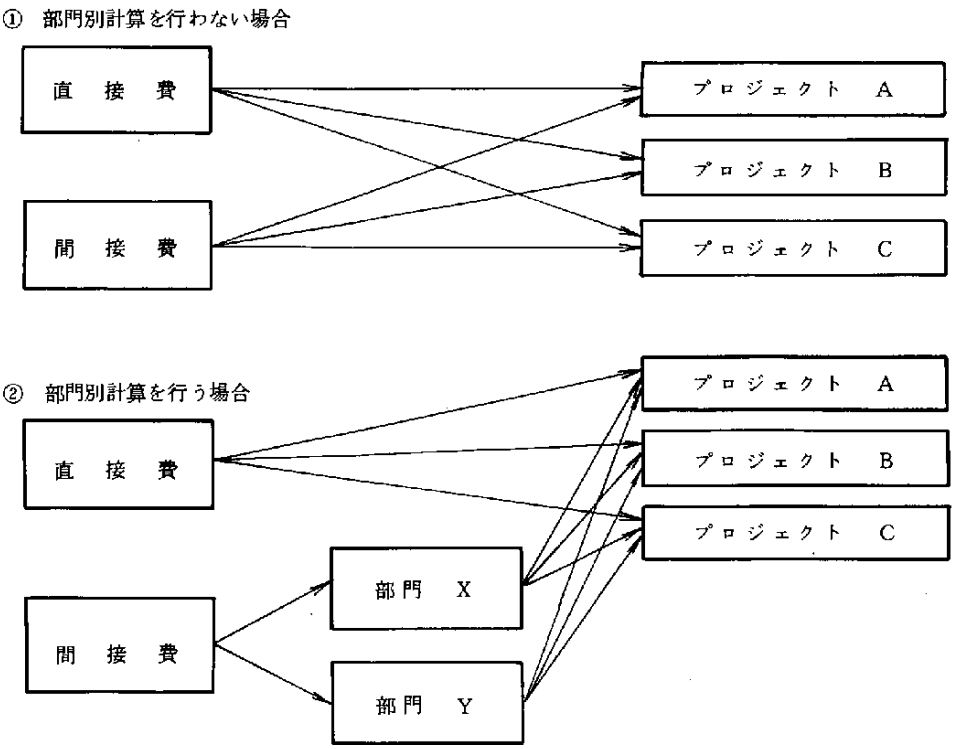


図５－３　プロジェクト別原価計算

・間接費の配賦基準については，時間基準や数量基準等が考えられる。

・間接費配賦率の求め方については，実績を毎月求めこれを適用する実際配賦率と，一定期間の予定間接費と予定基準数量によって求められる予定配賦率とがあること。

5.5　システムの費用効果分析

システムは大別するとハードウエア，ソフトウエア，アプリケーションウエア（ヒューマンウエアを加えても良い）から成り立つ。システムの費用効果分析を行う場合，まず個々のコンポーネントの効率追求がなされ，次いで全体としての効果が検討されることになる。

ここでは，マクロ的な見地からコンピュータの費用効果について考え，最後に費用効果分析の一般的な手法について説明する。

(1)　コンピュータ・システムの経済性評価

ここではハードウエア自体の経済性評価について検討する。

①　コンピュータの規模の利益

大規模システム程，単位処理量あたりコストが安くなることを規模の利益（Scale of Economy）という。コンピュータについてこの原則を定式化したのがグロシュの法則である。

グロシュの法則

$$C = k\sqrt{E}$$

ここで，C＝（コンピュータのコスト）

E＝（システムのパーフォーマンス）

システムのパーフォーマンスとしては，

スループット

処理スピード

等を考えれば良い。

グロシュの法則を図示したのが図５－４（a）（b）である。

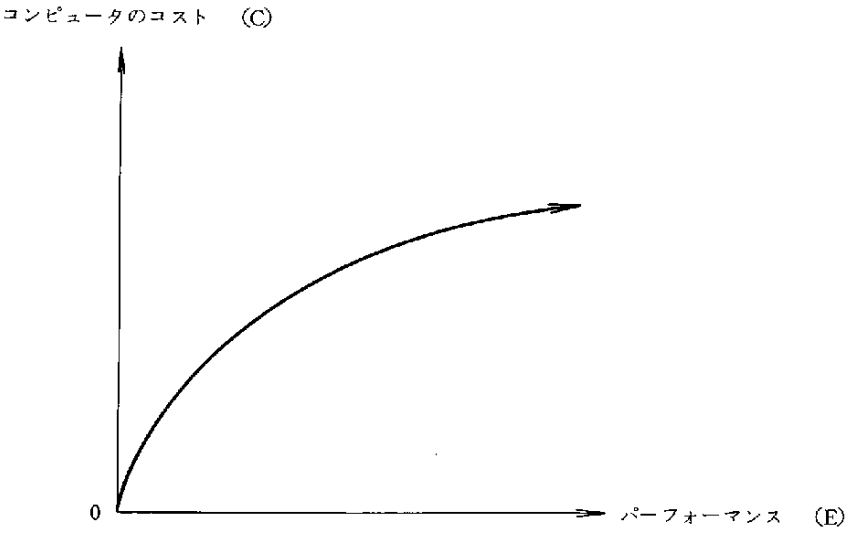
まず（a）図の方は，パーフォーマンスEが増加しても，コストは１／２乗でしか増加しないことを示す。単位パーフォーマンス当りコストで見たい場合は両辺をEで割れば良く，

$$\frac{C}{E} = \frac{k}{\sqrt{E}}$$

となる。

これを図示したのが図５－４（b）である。すなわち，単位パーフォーマンス当りコスト（C／E）は，規模の大きさを示すパーフォーマンスが大きくなる程小さくなることがわかる。

(a)　コンピュータ規模とコストの関係図



(b)　単位処理量当りのコストの逓減図

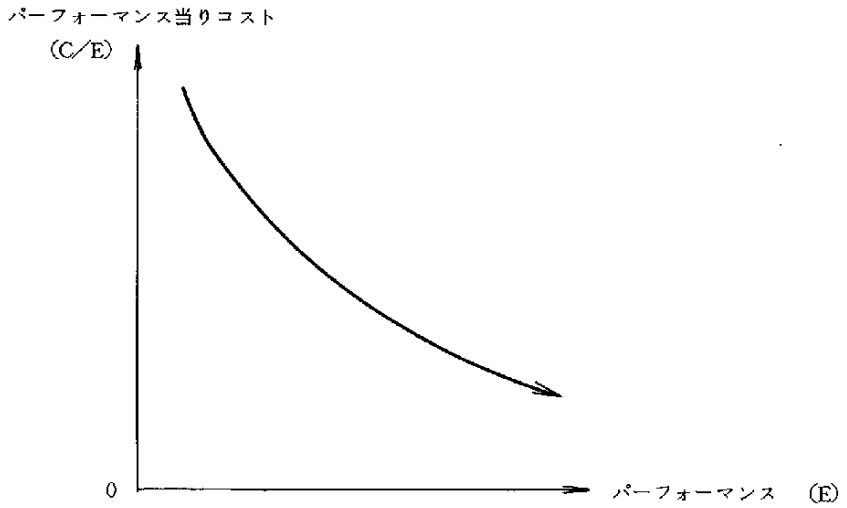


図５－４　グロシュの法則

規模の利益が成立することは経験的にも言えるのであるが，これを実証したものとしてソロモン〔15〕がある。

②　メモリーの経済性評価

記憶媒体ごとの経済性について説明する。

以下メモリーの，単位ビット当りコストの概略をおさえる。

・磁気テープ

・磁気ディスク

・磁気ドラム

・コア記憶装置

・バッファー記憶装置

③　技術進歩と経済性

コンピュータの技術進歩は著しい。新旧システムのコスト採算性の趨勢を正しく理解させる。

次の点を指摘する。

・技術進歩はカレンダ・イヤーの関数である。毎年一定率で向上する。

この場合の技術進歩は，

（コスト）／（パーフォーマンス）

で計測すれば良い。図５－５参照。

・技術進歩は，コンピュータを構成するいくつかのコンポーネントから合成されたある

関数である。

・技術進歩は，ある種の技術革新が導入された時に起る。

・技術進歩はＩ／Ｏ機器より，メモリー，プロセッサにおいてより大きい。

④ コンピュータのパーフォーマンスについて

コンピュータのパーフォマンス当りコストとか，パーフォマンスの良し悪しを論ずる場合のパーフォマンスとは何かについて簡単に説明する。

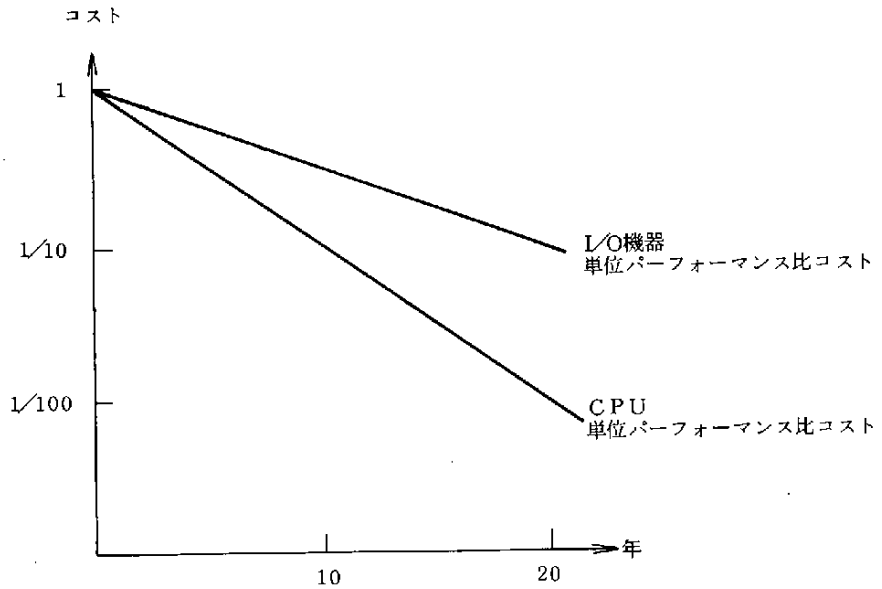


図５－５　技術進歩とパーフォーマンス比コスト

(a) 簡単な指標を用いる場合

経験的な方法である。例えばCPUについては，

E＝M・N

を用いる。ここで，

E＝（効率）

M＝（内部記憶装置の容量）

N＝（サイクル数）

＝（サイクル，タイムの逆数）

(b) インストラクション・ミックスを用いる場合

用途に応じ，

・サイエンティフィック・ミックス

・コマーシャル・ミックス

を用いる。

代表的な命令としては,

・固定長加算と比較命令

・浮動演算加算

・乗算と除算

・移動命令

などが使われる。

(c) カーネル法(Kernel Timing Estimate)

核となる代表的な業務における処理時間の比較でパーフォーマンスを求める。

核となるモデル業務としては,

・逆行列計算

・平方根の数値解法

・フィールドの移動

・エディティング

・フィールドの比較

・社会保険税計算

などが用いられる。

(d) ベンチ・マークによる方法

カーネル法の欠点は, I/Oを含めたシステムとしてのパーフォーマンス測定に不向きなことである。ベンチ・マーク法はシステムとしてのトータルな効率測定を行う。

比較の対照となる業務は,

・シークエンシャル・ファイルの更新

・ランダム・ファイルの更新

・分　類

・逆行列計算

・LP計算

等が使われる。なお業務の選択にあたっては,

・自社内の運用実態を調査し, 偏りのない分野から選択すること。

・ファイルを用いる場合には, 処理データの状態等, 後の分析に役立つと思われる内容を記録しておくことが必要である。

(2) 全体システムの経済性評価

対象となるシステムの効果及び費用の見積りが出来た後の総合評価をどのようにするかに

ついて説明する。

① 収益額による法

代替案A，Bがあったとき，それぞれの収益を計算し，比較選択する方法である。表5－2はシステム化における効果および費用を簡略化して示したものである。表中のシステム化収益（P＝S－C）で判定する。

代替案A，Bが狙う効果は同じで，ただ手段が異なる故の費用のみが異なる場合には，単純な原価比較法に帰着する。この場合にはCの大小によって判定する。

表5－2 システム化損益表

システム化による効果	S
直接費	
材料費	M
労務費	L
経費	E
間接費	
部門間接費	O
他部門配賦額	OT
資本費	
償却費	D
利子	I
システム化費用	C＝（M＋L＋E）＋（O＋OT＋D＋I）
システム化収益	P＝S－C

② 資本回収法

毎年の回収額をP（P＝S－C），当初の投資額Kをn年で回収するとする。投資の利回りをrとすると，回収額の現在価値K′は，

$$K' = P(1+r)^{-1} + P(1+r)^{-2} + \cdots\cdots + P(1+r)^{-n}$$
$$= P\frac{1-(1+r)^{-1}}{r}$$

回収額の現在価値と，初期投資額の大小で，投資の有効性を判定するのであるが，資本回収法は種々の経済計算の基本となる。したがって応用が可能でなくてはならない。例題を用意するのが良い。

又，この手法の長所，欠点も指摘しておく。

例えば，

長所：時間価値を考慮していること。

欠点：投資の効率的使用という規準が入っていないこと。又，投資の利回りを適正に決定することの困難性など。

③ ROI法（Return on Investment）

資本回収法では欠けている投資の効率的使用の観点と，投資の利回りについて補充する方法であることを理解させる。

ＲＯＩ法は，

$$K=P(1+r)^{-1}+P(1+r)^{-2}+\cdots\cdots P(1+r)^{-n}$$

から，投資の利回りｒを求める。代替案ごとのＲＯＩを計算し，最も利回りが高く効率的な投資案件を選択する。

④ 回 収 期 間 法

初期投資額Ｋを回収する期間Ｔの短いものを優先実施する方法である。毎年の回収額をＰ（Ｐ＝Ｓ－Ｃ）としたとき，回収期間は，

$$T=K/P$$

となる。この方法の長所，欠点を理解させる。

長　所

- 概念がわかりやすい。
- プロジェクト選択にあたり，リスクを小さくしたい場合に適している。
- 投資枠に制約があり，限られたプロジェクトに資金投資し，早期に回収したい場合に適している。

短　所

- 時間価値を考慮していない。
- 回収期間を終ったあとの価値を無視している。

などをあげれば良い。

⑤ その他の方法

その他の方法として，

- 投資収益率法
- 投資額一定のもとで，毎年の回収額最大をとる方法
- 毎年の回収額一定のもとで，初期投資額最小をとる方法

などいくつかあるが，上記①－④の応用問題として理解させることができる。

指導上の留意点

(1) システムの経済性評価を行うときに求められる能力は，システム・アナリシスにおける費用効果分析を正しく行える能力といえる。第５章全体の構成はかかる観点でなされている。

(2) 5.1情報の価値は，評価の対象となる情報を経済財という観点から再構築することを狙いとしたい。

(3) 5.2システムの評価基準においては，多次元の目的関数があることの事実に加え，システムの多様化に対応した新たな視点からの評価が可能であることを示したい。システム監査にかける評価基準やＭＩＳの有効度評価のための基準の例を加えたのはその点を意図としたものである。

(4) 5.3システムの効果においては，特に金額換算が困難な効果に対する考え方について理解させたい。金額換算の手法は未だトライアルの段階にすぎなく，本文中の事例も一つの考え方としての提示にすぎないが，基本的な取り組み方の事例としてグレゴリー，バンホーン・モデルを引用した。

(5) 5.4システムの費用においては，効果把握の場合とは対照的に，伝統的なコストアカウンティングを主体とした。効果把握とは異なり，費用把握後の部門別管理，プロジェクト管理が正しく行えることが目的となるからである。

(6) 5.5システムの費用効果分析では5.3，5.4でカバー出来なかった事項の整理と，経済性評価手法のレビューを行う。

参　　考

1．システムの有効度

完全自動化装置でない限り，すべてのシステムには何らかの形で人間系の関与がある。情報処理システムの場合は特に人間の関与に関する考慮が必要なことは言うまでもない。米国における宇宙開発計画に伴って検討された各種の評価手法の中にはマン・マシン・システムの設計に関して参考になるものが多い。その一例として**システム有効度**という概念をとりあげてみる。これは設備のコンフィギュレーションの選択の評価尺度を数値的に把握しようとする試みである。有効度の定義については未熟な部分もあるが，広義の設備選択のための判断として適切なガイダンスになる。これに類似する概念として次のようなものがある。

- システム有効度（system effectiveness）
- システム性能有効度（system performance effectiveness）
- 人員システム有効度（personnel system effectiveness）
- システム安全有効度（system safety effectiveness）
- コスト有効度（cost effectiveness）

有効度の概念とともにこれの測定評価のプロセスがさだめられている。

有効度の評価要因の例は下記のようなものである。

近づきやすさ	accessibility
責　務	accountability

正確性	accuracy
適応性	adaptability
一致性	compatibility
恒常性	consistency
制御性	controlability
拡張性	expandability
自在性	flexibility
互換性	interchangeability
保全性	maintainability
可動性	mobility
可搬性	portability
判読性	readability
信頼性	reliability
安全性	safety
安定性	stability　　等々

評価尺度として有効度を定義するためにはこれらの要因を構造化して構成する必要がある。その例をあげると，

システム有効度

$$E = f\ (A, D, C)$$

E：システム有効度

A：アベイラビリティ

D：ディペンダビリティ

C：ケイパビリティ

コスト有効度

$$E_c = \frac{E_s}{C_a + C_c} = \frac{f(P, A, U)}{C_a + C_c}$$

E_s：システム性能有効度

C_a：開発コスト or 取得コスト

C_c：運用コスト

P：性能指標

A：アベイラビリティ

U：ユーティライゼーション

$$A = \frac{MTTF}{MTTF + MTTR}$$

MTTF : Mean Time To Failure

MTTR : Mean Time To Repair

有効度の考え方は，システムそのもののシステム目的に対する適合の度合を定量的に表現しようとする試みである。

参考文献

〔1〕Gorry, G.A. and Morton, M.S.,"Framework for Management Information Systems",Sloan Mangement Review, vol. 13 no.1, 1971

〔2〕坂井利之 「情報学」 筑摩書房，1970

〔3〕北川敏男 「組織と情報」 日本放送出版協会，1966

〔4〕北川敏男 「情報学の論理」 講談社，1969

〔5〕野口悠紀雄 「情報の経済理論」 東洋経済新報社，1974

〔6〕A.M.マクドノウ著 （松田武彦訳） 「情報の経済学と経営システム」 好学社，1966

〔7〕講座 「情報社会科学」 学研，1971

〔8〕NHK情報科学講座，1968

〔9〕三浦大亮（編） 「システムの分析と設計」 オーム社，1972

以上は情報の性質について言及している参考文献である。

〔10〕Amstuz,"Managing the Multi-Nationals",Business International, 1972

〔11〕Gregary, R.H. and Vanhorn, R.L.,"Automatic Data Processing System -Principles and Procedures", Wadsworth, 1960

〔12〕Sharpe, W.F.,"The Economics of Computer", Columbia University Press, 1969

〔13〕Boutell, W.S.,"Computer-Oriented Business System", Prentice-Hall, 1973

〔14〕Fisher, G.H.,"Cost Consideration in System Analysis", American Elsevier, 1971

（OR学会 PPBS 部会訳 「システム分析における費用の扱い」 東洋経済新報社）

〔15〕Solomon,"Economics of Scale and IBM System／360",Communication

of the ACM, June, 1966

〔16〕 Seiler, K.,"Introduction to Systems Cost Effectiveness," Wiley, 1969

〔17〕 村川武雄 「設備投資の経済計算とその理論」 日科技連, 1961

〔18〕 青木茂男 「原価計算新稿」 中央経済社, 1962

〔19〕 津田直躬 「管理会計論」 国元書房, 1977

〔20〕 G.A. ウェルシュ著 (諸井勝之助編) 「企業予算」 日本生産性本部, 1957

〔21〕 S. ステイムラー著 (大野豊訳) 「コスト／パフォーマンス」 日本経営出版, 1973

〔22〕 林喜男・大川雅司・井口雅一編 「人間・機械システムの設計」 人間と技術社, 1971

〔23〕 中山諭事務所著 「電算コスト管理実務集」 日本能率協会, 1976

〔24〕 日本公認会計士協会 「ＥＤＰ監査の進め方」 大蔵財務協会, 1976

〔25〕 森谷亘暉, 加藤昭編 「ＥＤＰ会計ハンドブック」 中央経済社, 1971

第6章　システム仕様書

キーワード

システム仕様書（systems specification），ロジカル設計とフィジカル設計の差異（distinction of logical design from physical design），システム出力要求（system output requirements），出力の方法と様式（output methods and formats），要求仕様技術（requirements specification technique），TAG, ISDOS, SOP, HIPO, SADT, etc

目　　標

このカリキュラムでは，システム設計を便宜的にロジカル設計とフィジカル設計の2段階に分けている。前者がシステムの機能設計であるのに対し，後者はその機構設計である。システム開発の段階わけとして，ロジカル設計からフィジカル設計への移行に際して，アプリケーション中心のシステム・アナリストから，コンピュータ技術中心のシステム・アナリストに，仕事がバトンタッチされる。この接点にあたるものが，システム仕様書である。まずシステム仕様書の内容を検討し，この学習を通じて，ロジカル設計とフィジカル設計の質的相違点を理解させるとともに，D2の単元に対する導入部も兼ねる。

要求仕様技術としてのソフトウェア工学の議論が，最近活発になっている。これらの動向をこの章で概観しておくと，D2で学ぶべきことの将来の趨勢を知る上で参考になる。

学習内容としては，システム設計の仕様書の要件を整理して把握することに重点をおく。システム分析・設計のパッケージ化された具体例としてのSOP, HIPO, ARDI, SADT などについても，その一，二の検討をおこなう。

内　　容

6.1　システム仕様書

(1)　システム仕様書の目的

システム仕様書の目的を明確にする。システムの機能についてのロジカルな設計段階を規定するものであるので正しい認識と位置づけが必要である。

システム仕様書の目的は，各階層の関係者に異なる役割を持つことを説明する。

例えば，

①　システム部門の管理者には，

プロジェクト開発の総合管理者として，管理下の資源の有効活用，業務遂行の管理のための資料として機能すること。

② システム・ユーザには，

システムに期待した事が，効果的に正しく遂行してくれることの保証書であり，かつ，新システムへの移行時には，運用要件の記述書として機能すること。

③ システム設計者（次段階のフィジカル設計担当のシステム・アナリスト）には，

システム・ニーズの達成のために，解かねばならない課題は何か，又その課題解決のための効果的方法に関する記述書として機能すること。又，入手可能な資源制約と環境，満足規準を明示すること。

④ システム運用部門，ハードウエア管理部門の関係者には，

現行のハードウエア運用条件を考慮し，新システム移行後の問題を事前に提示すること。例えば，最大負荷時への対応，スケジュール上の特記事項等を明らかにし，必要な対策をたてる上での条件を記述すること。

などである。

(2) システム仕様書の記載事項

上記目的のために，システム仕様書を構成する文書，及びその文書に記述しなければならない内容について説明する。

主な文書とその記載事項は以下の通りである。

① システム要約書

- システムの概要
- システムの目的
- システムの解答
- コスト・効果の評価
- 開発スケジュール
- システムの評価基準

② 新システムの機能要件書

- 目　次
- システム・フローチャート
- 情報の流れ図
- インプットの記述
- アウトプットの記述
- ファイル

・データ・コミュニケーション

③ コンバージョン（切替え）のための機能要件書

・コンバージョンの概要

・コンバージョン実施計画表

・コンバージョンの内容

・ファイルのコンバージョン

・データの作成

・コンバージョンのためのシステム概要

・コンバージョンのための教育訓練

④ 経済性分析書

・所要コスト

・期待できる効果

・コスト効果分析

・検討を行った代替案とそのコスト効果分析

⑤ 開発スケジュール表

・開発スケジュール

・全体スケジュール

・要員スケジュール

・設備導入スケジュール

・開発体制

・開発体制，組織，分担，要員

6.2 ロジカル設計とフィジカル設計の機能分担について

第1章でのシステムライフ・サイクルの段階分けにおいて，システム開発を上述の2段階に区分別けした。その際の定義は，

ロジカル設計 ― 機能の設計（情報分析）

フィジカル設計 ― 機構の設計（システム設計）

であった。ここでは，更に具体的な例によって両者の機能を明確にさせる。そうすることにより，両者の設計にたずさわる者に，欠落のないシステムの設計を行わせることを目的とする。

この両者の性格の差を把握するには

システム開発過程の各段階での判断基準を整理してみるのも一方法であろう。

段階区分の一例をあげると

① 問題の発生⟶問題の提示
② 構想立案⟶予備検討及びシステム計画　} 情報分析（ロジカル設計）

③ 基本設計
④ 詳細設計　} ⟶システム設計（フィジカル設計）
⑤ プログラム設計

⑥ プログラム作成

⑦ システムテスト

⑧ 本稼動

システム開発のアプローチとして，さらに概括的に

① 問題発見

② 問題形成

③ 問題解決

と区分する場合も多い。この場合には，システム仕様書が問題形成に該当する。

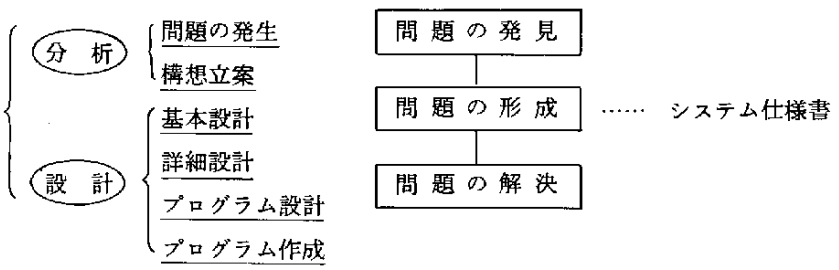


図６－１　システム仕様書の位置づけ

6.3　要求仕様技術

大規模なソフトウェアの開発には，開発スケジュールのおくれがしばしばみられた。この原因の多くはシステム設計にあったことが指摘される。さらにさかのぼれば，システム仕様があいまいであったことに根本原因があることが多い。初期段階での要求のあいまいさや記述の不完全さが，開発の終了段階に近くなって開発の費用を大幅に増大させることがある。このことの重大さに対する認識が高まって，最近要求仕様技術として研究されるようになった。仕様化の方法論を学習することによってシステム分析の性格を把握することができる。

要求仕様技術は具体的なシステム或いは規範として整備されつつある。それらの事例について仕様作成を実行してみることも必要である。しかし現状ではこれらの技術はシステム分析のレベルのものよりもむしろシステム設計レベルの詳細仕様の表現に適したものが多い。したがって個々の技法に深入りするのは得策ではない。むしろそれらの技術が持つ文書化に関する工

夫にヒントを得て，明快で正確なシステム仕様書を表現する方法を研究することに重点をおくのがよい。

SOP	IBM社
ARDI	Philips社
SADT	Softech社 (Ross)
Young & Kent の方法	
TAG	IBM社
HIPO	IBM社
ISDOS (PSL／PSA)	ミシガン大学 (Teichrocw)
SREM	TRW社
SSDM	Boeing社
ADS	NCR社

このような標準化されたパッケージが開発されて，試用されている。

指導上の留意点

(1) 単元D1の要約の章でもある。第1章から第6章までの要点をふりかえって各章の主要事項の復習をおこなう。

(2) システム分析の結果を集約した文書がシステム仕様書である。したがってこれに記述すべき要件は，上記の主要事項がすべてもり込まれていなければならないことを強調する。

(3) 単元D1と単元D2との間の連結章でもある。したがって分析と設計の相互の性格の違いを明確に把握させるとともに，分析と設計とがシステム開発の流れの中で一貫性をもたねばならないこともあわせて教育する。

(4) システム仕様書に盛り込むべき項目（内容）について説明するだけでなく，標準的な書式（形式）についても検討を加える。

(5) システム仕様書は伝達のための文書である。誤りなく仕様を伝える技術として，ソフトウェア工学の研究の必要であることに注意をむけさせる。

参考文献

[1] Clifton,D. H.,"System Analysis for Business Data Processing" Auerbach, 1970

[2] Gray M.,"Documentation Standards,"Brandon Systems Press, 1969

[3] IBM, corporation,"SOP Documentation Techniques", Technical Report No C20- 8075-0, 1963

〔4〕 Myers, D.H.,"A Time-Grid Techniques for the Design of Information Systems," IBM Systems Research Institute, 1963

〔5〕 IBM Corporation,"The Time Automated Grid System : Sales and Systems Guide," Y 20-358, 1966

〔6〕 Kelly, J.F.,"Computerized Management Information System," Mac Millan, 1970

〔7〕 Myers, D.H.,"Reliable Software through Composite Design," Manson / Charter, 1975

（久保，国友．訳 「高信頼性ソフトウェア」 近代科学社，1976）

〔8〕 Hartman, W., Matthes, H., Proeme, A.,"Management Information Systems Handbook," McGraw-Hill, 1968

〔9〕 Teichroew, D., and Sayani, H., "Automation of System Building," Datamation, Aug. 15, 1971

〔10〕 Teichroew, D., "A Survey of Languages for Stating Requirements for Computer Based Information Systems Proceedings," FJCC, 1972

〔11〕 Teichroew, D., and Hershey, E.,"Computer-Aided Structured Documentation and Analysis of Information Processing System Requirements," ISDOS Project, Michigan Univ, Prepared for SHARE XIVⅡ, Aug. 1976

〔12〕 National Cash Registed Company,"Accurately Defined Systems," 1967

〔13〕 Lynch, H.J.,"ADS : A Techniquevin Systen Documentation," Data Base, vol.1, No.1, 1969

〔14〕 Young, J.W. and Kent, H.,"Abstuct Formulation of Data Processing Problems," Journal of Industrial Engineering, Nov.-Dec., 1958

〔1〕,〔2〕は文書に関するもの，文書化に関する参考文献は，このほか，2章の参考文献を参照のこと。

〔3〕：SOPに関するもの，日本IBMでこれを翻訳したものがある。3章の参考文献に引用してある。

〔4〕,〔5〕,〔6〕：TAGに関するもの

〔7〕：HIPOに関するもの

〔8〕：ARDIに関するもの

〔9〕,〔10〕,〔11〕：ISDOSに関するもの

〔12〕,〔13〕：ADSに関するもの

〔14〕：Young & Kentのシステムに関するもの

〔15〕久保未沙，国友義久　「効果的プログラム開発技法」　bit　Vol. 8及び Vol. 9に連載，共立出版社，1979

〔16〕宮本　勲，東谷秀夫　「ソフトウェア・エンジニアリングへの道」　bit　Vol. 9　及び Vol. 10に連載，共立出版社，1977

〔17〕国井利泰（監修）　「ソフトウェア工学：要求仕様技術」　共立出版社，1978

〔15〕はHIPOの解説

〔17〕は要求仕様技術の全体を見通す参考文献

単元Ｄ２

システム設計

単元Ｄ２
システム設計

目　　次

第１章　基本的な設計手法 …… 101
1.1　システム要求 …… 101
1.2　システムの設計の重点 …… 103
1.3　トレードオフ …… 103
1.4　モデル化 …… 105
1.5　情報処理系の型 …… 107
1.6　システム設計の文書化 …… 109
1.7　プロジェクト計画 …… 111
1.8　テクノロジー・アセスメント …… 111

第２章　システム資源の選択 …… 123
2.1　ハードウェア資源の選択 …… 124
2.2　ソフトウェア資源の選択 …… 128
2.3　システム効率の評価 …… 131
2.4　設備計画 …… 134
2.5　設備選択 …… 136

第３章　ソフトウェアの設計 …… 143
3.1　ソフトウェア開発の現状 …… 143
3.2　モジュール化 …… 146
3.3　プログラムの設計 …… 152
3.4　サジシステムのインターフェイス …… 155
3.5　ソフトウェアの標準化 …… 158
3.6　ユーザとのインターフェイス …… 160
3.7　プログラム技法集の整備 …… 163

第4章　データベース …… 170
4.1　データ管理 …… 171
4.2　ファイルの設計 …… 174
4.3　データベースの概念 …… 177
4.4　ＯＢＭＳの導入計画 …… 179
4.5　データベース・システムの設計 …… 181

第5章　プログラム作成と検査 …… 196
5.1　よいプログラムとは？ …… 197
5.2　プログラムテスト …… 199
5.3　デバッグ …… 203
5.4　プログラミング・スタイル …… 206
5.5　プログラム作成の日程計画 …… 208
5.6　プログラム開発工程上の管理作業 …… 210

第6章　システムの評価と改善 …… 217
6.1　システムの性能の評価 …… 217
6.2　システムの保守 …… 220
6.3　評価の方法 …… 223
6.4　システムの寿命 …… 226

第1章　基本的な設計手法

キーワード

文書化 (documentation)，標準化 (standardization)，システム工学 (systems engineering)，ソフトウエア工学(software engineering)，図式化 (graphic representation)，トレード・オフ (trade off)，プロジェクト計画 (project planning，技術予測 (technological forecasting)，テクノロジー・アセスメント(technology assessment)

目　　標

本章では，Ｄ２の単元に対する導入部として，各章の組立をシステム・ライフサイクルに対応して説明する。骨格はすでにＤ１の第１章で説明してあるので，復習程度でよい。

システム設計の重点を説明する。この際Ｄ１の第３章との対比をおこなって，その特徴を明確に示すことが必要である。

設計手法の中で，文書化の占める役割の重要さを認識させる。システム開発における管理的側面としての文書化については既にＤ１の第２章で学習してあるので，ここでは主として技術的側面から文書の役割をつっこんで検討する。

分析手法の中心的な技術が「代替案」の検討であるのに対比して考えれば，設計手法の中心的な技術は「トレード・オフ」という考え方である。ここではトレード・オフについて詳細に検討する。ソフトウエア工学の中の設計手法に関する部分の考え方について概観する。

この章では個々の手法を個別に検討するよりも，システム分析・設計の底流になっている考え方を教育する。モデル化の効用，プロジェクト計画手法，テクノロジー・アセスメントなどのようなシステム設計に関連する重要な留意事項にも言及する。Ａ３，Ｂ１の単元との関連にも触れて，システム設計の骨組を明らかにすることを主目標にする。

内　　容

1.1　システム要求

第１章の学習上の位置は，これ以降で学ぶシステム設計のためのコースに対する序章である。まず，単元Ｄ１で学んだ事項を，システム・ライフサイクルの概念と結びつけて復習することから着手するのが適切である。単元Ｄ１のシステム分析で学んだことは，結論としてのシステ

ム仕様書に集約される。システム仕様書は，システム要求を記述した文書である。システム開発の成否を支配するものは，システム要求が正確に表現できているかどうかにかかっている。

ここでシステム要求とは何かをふりかえって考えてみることにする。システム開発は，おおづかみに次の三つの型に分類できる。

① ユーザが注文するプロジェクト

② ユーザが吟味するプロジェクト

③ ユーザとは独立に実施されるプロジェクト

これらの三つの型でシステム要求がどのようにして作られるかを考えてみると，①はユーザが独自で要求を作って，開発グループに下請契約をさせる形になるため，要求の解釈で誤りをおかす可能性がある。③は一般に商品としてのソフトウエアパッケージなどの開発の場合で，ユーザの要求を適切につかまないことがありうる。したがって望ましい形は，②であるということができる。結局システム開発のための要求を用意するには，何らかの形でユーザを含ませることが重要であるということになる。①の場合も，③の場合も，何らかの方法によって②の場合のようにユーザを要求作成段階で深く巻きこむ必要がある。

システム設計の出発点としてのシステム要求についてその意義を考えてみる。

要求仕様技術の学習ですでに学んだように，要求記述の重要性に着目して，その記述の自動化には，情報代数，ADS，TAGなどが開発されているし，様式の制定には，HIPO，SADTなども考案されている。さらに表現言語としてISDOS（PSL／PSA），SSDMなどの研究も進められてはいるが，これらでシステム要求のすべてが表わされるとはいえない。これらの要求仕様技術がねらっているものは主として機能的な要求であって，このほかの要求，たとえば，システムの信頼性とか性能（効率）とかコストなどについても記述がなければならない。

システム設計に着手する前に，システム仕様書に記述されていることが，システム要件のすべてを満しているかという検討をしなければならない。ユーザ側と開発側との協議のテーマは，要求の正当性に対する議論である。

ユーザは要求の完全性と正確さを調べる責任がある。

開発側は実現可能性と理解しやすさについて調べる責任がある。

システム設計の中心は，後に述べるようにトレード・オフになる。システム要求には，このトレード・オフのためのよりどころになる情報がなければならない。システム仕様書に，「これこれの機能について，性能が最大で，コストが最小になるようにせよ。」と書いてあるだけでは，トレード・オフのための判断になる情報を何も提供していないことになる。

システム仕様書の要件とは何かを，システム設計の側から再吟味するのが，この章の学習の焦点である。

1.2　システム設計の重点

システム設計で最も重要なことは，システム要求を全体として正確につかむことと，その開発のための目標を設定することである。要求については，前節にあらましを述べた。目標は次の二つに分けて考えるとよい。

① プロダクトの目標

② プロジェクトの目標

前節で分析した二つの立場，ユーザと開発側に対応させて考えれば，①はユーザの立場からみたシステムの到達点を設定することで，②は開発側からみたスケジュール・コスト，などの程度や制約を明らかにすることである。この両者の間の妥協点を決めることが，トレード・オフである。

このように要求と目標とをどのように設定するかという技術が，基本的な設計手法である。

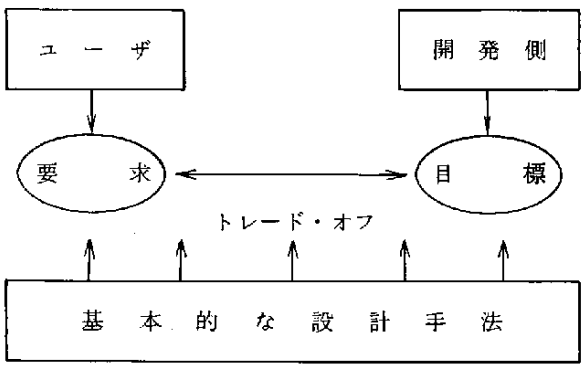


図１－１　システム設計の概念

1.3　トレード・オフ

システム設計とは，要求と目標との間にある相反する矛盾点をどの線に妥協させるかという技術であるということもできる。要求とは漠然としたイメージであって，目標とは明確な量的な規定であるということもできる。設計とはトレード・オフであるという定義をさらにあきらかにするために，Myers の分類によって説明を加えると次のようになる。即ちソフトウエアの目標には次のような分類が考えられる。

① 信頼性（Reliability ）

② 一般性（Generality）機能の数，能力，範囲

③ 人間因子（Human factor）使いやすさ

④ 適応性（Adaptability）拡張性，オプション追加

⑤ 保守性（Maintainability）修理のしやすさ

⑥ 保護（Security）ユーザやデータの分離

⑦ 文書化（Documentation）

⑧ 製品コスト（Cost）

⑨ スケジュール（Schedule）

⑩ 効率（Efficiency）性能（Performance）

これらの要素の間には相互に密接な関係があって，場合によってはその一つを良くすれば，他の一つが悪くなるということがある。又その反対に一つの効果をよくすれば，それにつれて他の一つの効果もよくなるようなプラス効果のあるものもある。

過剰な要求をすれば，製品コストは高くなり，開発期間も長くなる。この間の調整をして，どこに目標をおくのかを決める問題がトレード・オフである。これらを実行に移す基本的な設計手法は，単純に次のような手順になる。

(1) プロダクトの目標を書く
(2) プロジェクトの目標を書く
(3) 上記の目標の評価をする
(4) 外部設計をする
(5) ユーザ・インタフェースの設計をする
(6) 外部仕様を書く

これらの記述について注意すべきことは，「できるだけ良いものを」とか「なるべく早く」とかというような表現では，何もトレード・オフの検討には役立たないという点である。

ここでシステム分析と，システム設計の性格の差異について考えてみる。分析のための中心技術は代替案であり，設計のための中心技術はトレード・オフである。

簡単ないい方をすれば，前者は「やるかやらぬか？，あれかこれか？」の問題である。即ちWhat である。後者は「どのくらいやるか？，どうやるか？」の問題である。即ちHow である。

前述した基本的な手順を踏んで作成される文書がシステム計画書である。

システムの信頼性や性能に対する要望と，開発費や期間の制約との間の相反する力関係をどこに落着かせるかが基本計画である。したがってその決定までのプロセスは，システム分析と同じように，やはり反復的，収斂的な過程をたどることになる。このためには，システム仕様書には，トレード・オフのための目安になるような情報が具体的に記述されていることが望ましい。

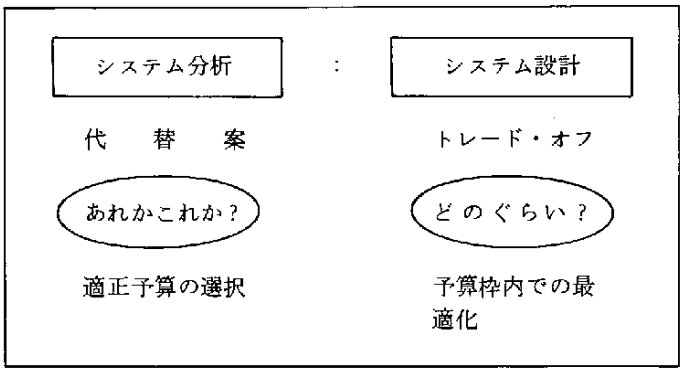


図1－2　分析と設計

このような作業を実施する上で，それを支援するための各種の手法が開発されている。例えば次のようなものである。

- 決定表（decision table）
- シミュレーション
- 見積方法，予測技法
- ウォークスルー（walk-through）
- 原因結果グラフ

1.4　モデル化

システムには目的が存在する。ある目的を実現するために，利用できる資源を活用し，与えられた制約の下で，その目的を達成するシステムを実現化する方途を考えるのがシステム設計であるが，その開発の途上で，往々にして見うしなわれてしまうものが，システムの目的である。システムの目的と，システムの特性とを結びつけて考察する哲学的資質が，システム設計者には要求される。

システムの型の分類とか，システムの一般理論とかは，多くの場合，出来あがったシステムを後からながめた観察にすぎないもので，システム設計には役立たない知識であることが多いが，システムの目的をみうしなわないためには，その基本的部分は何か，その本質的部分は何か，と問いただすという考え方の修練が大切である。

システム設計で対象とするシステムとは，通常人工的（アーティフィシャル，マン・メイド）と呼ばれる世界を考えている。この分野で重要な考え方の基礎はモデル化という概念である。

このためには，システム工学上の知識を，システム設計と結びつけて，自分の頭で整理する必要がある。具体的な学習項目としては次のようなものがあげられる。

(1) システム構成の階層性

要素と全体

システムのレベル

サブシステム

(2) システムの構造

入力・変換・出力

コントロールの方式（フィードバックループ等）

モジュール

結合方式

(3) システムの進化

代 替 案

システムの運用と保守

システムの改善・廃止

(4) 部品化・標準化の役割

独 立 性

汎 用 性

モジュール

システム設計とは，システムのモデルを作ることであるといってもよい。概念の明確なモデルには，それに対応する明快な図式化が開発されている場合が多い。前節の文書化の考察と対応して，システム構成やその構造の表現のための図式化についても学習するとよい。図式化とは次のようなものをいう。

集合	ベン図
状態	グラフ，遷移図
プロセス	フローチャート，HIPO図
コントロール	ブロック・ダイヤグラム
構成，結合	階層図，サブルーチンツリー，系統図
判断	ディシジョン・ツリー，ディシジョン・テーブル
要因	レリバンス・ツリー，特性要因図
待行列	GPSSのブロック図
工程	PERT図

これらの図式，図表の類は，システム設計におけるモデル化の表現の有力な技法であるとともに，場合によってはモデル化の思想そのものであることもある。

最近のシステム開発技法に，各種の図式化の提案がなされていることにも着目する必要がある。

ORのモデルは問題そのものの抽象化であるが，システムモデルは問題そのものよりはむしろ，その問題を解決する体系の抽象化である。自然あるいは社会の中に何らかの人為的な「しくみ」(man-made world又はartificial system)を新たにもたらす時に，システム設計が必要になり，システムモデルが必要になる。システムモデルの考え方の整理には，ORのモデルでの研究は参考にすべきところが多い。即ちモデル化のための共通的な性格について考察すべきである。例えば，

・必要な条件を落ちなく拾いあげること

・目的を明示的，定量的に示すこと

・機能を簡潔に表現すること

・これらを一個の全体として関連づけたものとして記述すること

などである。

表1－2　ORモデルとシステムモデル

ORモデルの例	決定論的モデル	LP
	確率的モデル	ゲーム
	日程計画モデル	PERT
	シミュレーションモデル	GPSS　など
システムモデルの例	資料検索型，問合せ応答型	
	報告書作成型，定例報処理型	
	オートメーション型	
	自動設計型	
	マン・マシン対話型	
	伝達表示型	など
問題把握モデルの例	デミング　サークル	
	オプトナーのフィードバック・コントロール	
	サイモンの意思決定のメカニズム	
	フォレスターのシステム・ダイナミックス　など	

1.5　情報処理系の型

前節でとりあげたシステムモデルという設計対象は，システム設計の「内容」による考え方の整理である。これに対して一方で「形式」による考え方の整理もありうる。本節で主としてとりあつかうのは，情報処理の方式や形態の分類に基づくシステム設計の型の考察である。

情報処理系と人間行動系との結びつき方の組合せで，情報処理の方式には様々な型が存在し，それらの利害得失も経験的に或程度検証されつつある。これらの知識を足場にして新たなシステムにむかうのが有効なことは言うまでもない。しかしながら様々な「形式」の評価も，現在未だ流動的な面も多く，今後の技術の進展につれて，変化することも十分予測される。この節でシステム設計の型について研究する態度としては，まずさまざまな形で混在する情報処理の形式に関する知識を整理して，基本設計として役だったものとして把握すること，システム目的に最も適合する型を採用するための判断基準を整理することなどである。ここで言うシステム設計の型というのは次のようなものである。

・設備系に着目した場合
　大型と小型（コンピュータ）
　集中と分散
　階層と並存
　計算と通信
・処理手順に着目した場合
　ルーチンとランダム
　バッチとインタラクティブ
　オフラインとオンライン
　マルチジョブとＴＳＳ
　メカニカルオートメーションとプロセスコントロール
・資源配分に着目した場合
　処理時間の割合
　プログラム負荷の割合
　データ領域の割合
・人間行動系に着目した場合
　人間系ですべてを処理する場合
　マン・マシンシステム
　完全自動化系

これらのシステム設計の型の分類考察にはまず比較検討の足場を明確にする必要がある。実在する複合的なシステムを様々な分類基準でだだ眺めまわすだけでは，むしろ混乱の増した知識の累積に終ってしまう。このことはシステム設計の内容について考察した前節の場合と同様である。そこで，学習のための着眼点を設定する。

一つの着眼点は，これらの処理形態の多様化をもたらしたニーズは何であったか，これらの

処理形式を技術的に可能にしたニーズは何であったかという歴史的追跡による把握である。他の一つの着眼点は，システム設計の型の多様性を構成しているシステム要素の分析である。即ち前節の「内容」との対応関係を整理して考察するという方法である。特に後者の方向での学習の精密化は，システム設計の基本になる。したがって重点を後者におくことが望ましい。

情報処理系の型の考察の場合も，あれかこれかという単純選択であってはならない。システム要素の配分を少しづつ変更して，次第に適正なところに追いつめてゆく作業がトレード・オフである。

特にマン・マシン系における人間の役割をどのように設定すべきかという配慮は，これからのシステム設計における重要な課題である。

1.6 システム設計の文書化

分析の単元で文書化についてすでに学習したが，その時の重点は文書の管理的側面であった。ここでは上記の学習を補足するために，文書化の技術側面を学ぶ。即ち設計の内容の表現技術を学ぶ。内容の記述は短期的にみれば，作業指示，運用指導，保守記録などがあり，長期的にみれば，教育資料，システム改善点の指摘などがある。技術的側面の文書化の特徴は，管理的文書がその場かぎりの性格が強いのに比べて，何回も参照される継続的性格をもつ点である。

しかしながら文書の中に記述されるものは詳細であればあるほどいいというものではない。情報伝達の役割をはたすためには，多くを書くよりも出来るだけ簡潔に書く工夫が必要になる。

文書作成のための着眼点を整理すると次のようになるであろう。

(1) 文書化の目的に関して

- システムの機能，用途を知らせる。
- システムの運用法，管理法を知らせる。
- システム改良，変更のための資料
- 次のシステム設計の参考資料

(2) 文書化の方法に関して

- システムの環境，目的との関係を明らかに。
- 詳細を多く書くよりもポイントをわかりやすく書く。
- システムが変更された時の追従を容易にしておく。
- システムの機能と方法の分離を明確にしておく。
- システムの構造と標準共用部分を図解しておく。

(3) 時間，費用の制約に関して

- 文書の作成には時間と費用がかかる。従って一般には一個の文書を多目的に使用する場合が多い。このためその文書を使用目的の側から見た場合，帯に短かし，襷に長しになることが多い。
- 文書化は，システム開発に着手する前に行うのが望ましいとされていながら，現実には事後処理として作られるのが実状であり，時間と費用がかかるとの理由でいい加減になされる結果に終ることが多い。
- これらを解消するための方策として，文書化についても，モジュール化，標準化の工夫が試みられている。

(4) 文書化のチェックポイント

- システムのライフサイクルの各段階
  ーステップとの対応　〔区切をつけるため〕
- 使用目的と伝達対象に合致した精度
  ーレベルとの対応　〔わかりやすくする〕
- 伝達，伝承のための時間に対する考慮
  ー時間，費用の制約との対応　〔間にあわす〕

(5) 文書化の工夫

- 形式に関する工夫
  文章形式（書式，文体，用語，用字，記号）
  絵による伝達（グラフ，図表，マンガ，写真）
- 表現に関する工夫
  概要の抽出，条件の箇条書，全体から部分へ，時系列にそった記述，論理形式に合致した記述，文章の簡明さ，概念図や立体図による全体把握，分解図による機構や結合の説明表やグラフによる計数的整理
- 標準書式の採用
- 文書化用器材の活用
- 標準的表現に対するなれ
  レター，出願，議事録などのまとめ方

システム内容を説明するための文書は，伝達のためにも，自分自身の考え方の整理のためにも，技術伝承のためにも必要である。

1.7 プロジェクト計画

(1) 予　　算

システム設計を開始するための意思決定をする時のよりどころは，具体的な形でいえば，予算である。システム設計のためのプロジェクトの足場は，時間的制約と予算である。したがってフィジカル設計の前提としての予算決定のしくみのあらましをあらかじめ知っておくことは，システム設計者への要請となる。

目標に照らして代替案を比較評価する方法を定形化したものにPPBSがある。PPBSに直接準拠しないまでも，プロジェクト計画の予算化の考え方に，合理的手順をもちこむことは，システム設計者の立脚点を明確にすることになる。

(2) 日　　程

プロジェクトの計画手法として用いられるものにCPMがある。これはPERTなどと同様に作業をネットワークに組んで，クリティカルパスをもとめて，計画と管理のための指標とする方法であるが，この時各作業に対する費用を見積り，プロジェクトのコストカーブを作り，日程計画を資金効率の面から評価し選定できるようにする手法である。

基本設計のためのトレード・オフ技法として重要なものの一つである。

(3) 数値計画法及びシミュレーション

ＬＰで代表される最適化技法は，数式表現されたトレード・オフの特殊な場合と考えることも出来る。目標設定のためにこのような抽象化が可能なものについては，これらの数式モデルを積極的に利用すべきである。

ここにあげたプロジェクト計画に関連する手法は，すでにシステム分析のところでも学習したものであるが，システム設計にとっても基本的な技術である。

1.8 テクノロジー・アセスメント

システム設計は，本質的には人工的なものを，あらたに人間の生活の中になかば無理やりに挿入するという行為である。したがってシステム目的に対する適合性を直接的に評価することのほかに，設計された「システム」が環境に対してどのような影響を副次的にもたらすかということもあわせて考慮しておかなければならない。システム稼動後の反作用の予測をすることである。

通常のシステム設計の「目的に対する適合性の評価」とはまったく別次元の評価が要求される。所謂公害問題の予測等の必要性である。これを一般にテクノロジー・アセスメントと呼ぶ。マイナス効果に対する考慮と対策が必要になりつつある現在，テクノロジー・アセスメントの研究は，これからのシステム設計者の必須課題になる。

テクノロジー・アセスメントは，技術予測を下敷にして検討しなければならない。技術予測の方法は最近各種のものが開発されている。例えば

- デルファイ法（RAND）
- システム・ダイナミックス（Forester）
- 形態学的検討
- 関連樹木法
- 環境アセスメント　マトリックス

などである。システム設計は目的指向的なアプローチであるが，それと同時に自らその内容を客観点に批判反省する態度を同時に養成するためにテクノロジー・アセスメントの節を附加した。したがって技術予測の技法そのものに多くの時間をさく必要はない。分析のための手段の一部としての予測手法については，すでにＤ１の単元でも学習済であるので，ここでは，システムの基本設計の完了した時点で，一歩しりぞいて今までと異った視点から，システムの副次的影響を客観的に反省してみることの必要性を強調しておく。

指導上の留意点

(1) ロジカル設計（分析）とフィジカル設計（設計）のシステムズ・アプローチとしての性格の差異を認識させる。

(2) トレード・オフの基本思想を説明する。

(3) システム内容の表現に焦点をあてて，文書のあり方を教育する。モデル化の役割をそれとの関連で説明する。特に図式化の研究に重点をおく。

(4) 設計の良否は，結局は設計者の洞察に負うところが大きい。これにはバランスの感覚が必要になる。近視眼点判断におちいらないためには，両極的な考え方も知っていなければならない。情報処理系にその例をあげたのはこのためである。

(5) 具体的な手法や手順についても学習する。

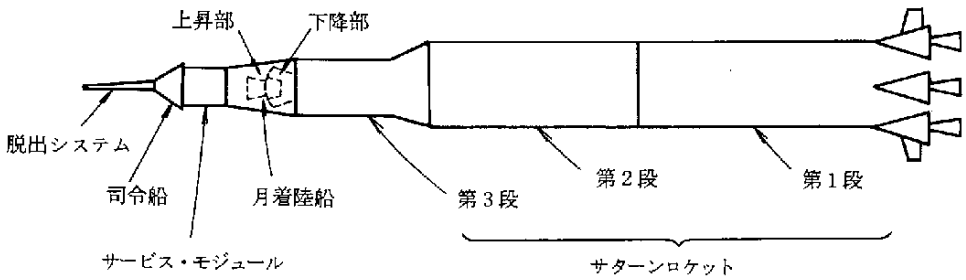
(6) システムは人為的な技術であることの認識もまた必要である。テクノロジー・アセスメントの項目を設けたのはこのためである。

(7) Ｄ１では比較的単純な単一システムの設計を念頭において教育する。複数システムの同一コンピュータ内での運用などのような現実的な応用分野の問題については，Ｄ４での学習にまつ。

参　　考

(1)　FAMEについて

FAMEとはForecasts and Appaisals for Management Evaluationの略で，アポロ計画のサターンロケットの生産過程で，その重量管理のために開発された技法である。アポロ宇宙船は下図のようになっている。



もし月着陸船の上昇部の重量が予定より100kg増加したとすると，月を脱出するのに100kg余分に燃料が必要になり，上昇部重量は200kg多くなる。月着陸のためのブレーキ分がさらに200kg余分になり，月着陸船は総計で400kg多くなり，そのためにロケットの一段目の燃料が400kg増加し，合計800kg重量超過になる。重量という点でみた時，どのモジュールの分がどのモジュールにどれだけ影響をあたえるかという係数がtrade-off factorとしてあらかじめ設定されていて（上の例ではtrade-off factor ＝1として計算した）これにもとづいて，全体的重量管理がおこなわれる。

NASAのアポロ開発では，上記の各モジュールは，別々に企業で，平行しておこなわれていたので，これの全体系としての重量管理は重要な問題であった。各モジュールでの重量をE／C／A（E＝Estimated，C＝Calculated，A＝Actual）の三段階にわけて，製作工場から月毎に報告される重量をつみあげて行き，日程の座標の上にプロットし，そのデータを使って先行予測をおこない，且つE→Aの割合の増加につれて絶えず予測の修正をおこない，重量を限界内に収検させるようにコントロールする。コントロール限界を切る可能性が予測されれば計画変更をおこないながら全体の開発を進行させる。

FAME技法の骨格は予測技術の応用であるが，モジュール間の重量に関する関連性をtrade-off factorとして定量的に表現し，ダイナミックな管理に使用している点は参考にすべき点である。

(2)　技術資料の整備について

次にあげるような技術資料の整備は，システム計画を順調に進めるための基盤である。

(a)　図　　書

・ハンドブック，年鑑，事典，辞典

・書籍，論文集

・定期刊行物，学術誌，雑誌

・ジャーナル，新聞，ニュース

・報告集（学会，研究所，調査団体）

・情報センター提供の特殊文献

・講習会，講演会の資料

・コンピュータ等の説明書

・パンフレット

(b) 資　　料

・雑誌などから抜粋したコピー

・調査報告，研究会，委員会資料

・カタログ

・各種センターの案内書

・パッケージプログラム説明書

(c) 文　　書

・開発プロジェクト関係文書

・オペレーションマニュアル

・教育訓練用マニュアル

・サマリー　リーフレット

・マニュアルリポート

(d) 書　　類

・名刺，名簿（人脈は大切な資料である）

・委員会

・レター

・通知，・依頼

・計画・契約

・予算・実績

・整員・人事

・外　注

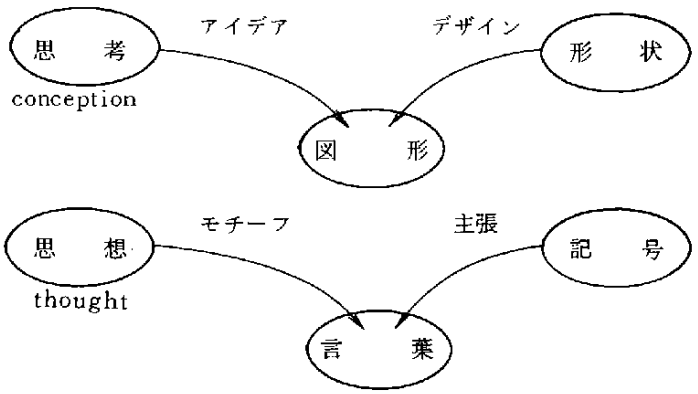
・機械稼動状況

・手順書，備品，用紙，様式

これらの各項目について，検索の容易さ，個人用，公共用の妥協点，開発時用，管理時用の妥協点，捨てる時のタイミング，整理のための備品資材など考慮すべきことは多い。

(3) 図形の機能について

表現手段としての図形の機能を，言葉の働きと対比して考えてみる。



・図形の特長

幾何学的，アナログ的，描写的，視覚的，全体認識，スタティックなパターン

・言葉の特長

代数的，ディジタル的，記述的，聴覚的，時系列認識，ダイナミックなロジック

図形の機能を分析すると次のようなものがある。

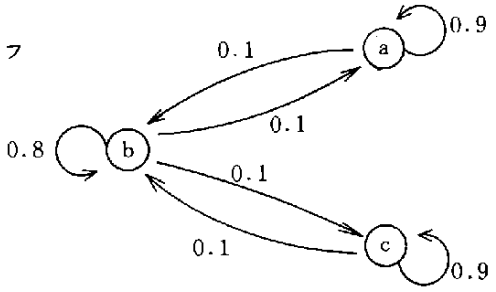
- 構造（structural）
  - 位相（メビウスの帯，折紙，概念図）
  - 結合（系統図，組織図，リンク図）
  - 論理（流れ図，結線図，ロジック図）
- 計数（computational）
  - 順序（方向図，時刻表）
  - 幾何（比例，写像，投影，透視）
  - 座標（カルテシアン表示，計算図表，統計図）
- 配置（allocational）
  - 形（shape）
    - form，fashion，contour，out line, type, mold, cast
  - 位置（map）
    - plan, chart, diagram projection, elevation, atlas
- 絵　(pictorial)
  - マンガ（carton）
    - caricature, parody, copy, draft, picture, sketch, portrait, drawing

図形の機能に色彩の働きを加えると，さらに複雑な機能構成となる。視覚的表現の問題は今

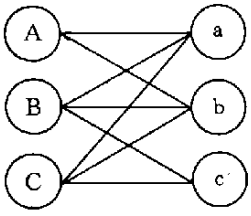
後の技術課題として研究すべき項目が多い。特に発想のための触媒としての図形の役割は重視するに値する。

・図形表現の二三の例

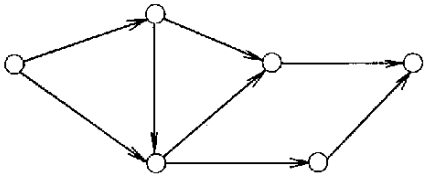
推移グラフ



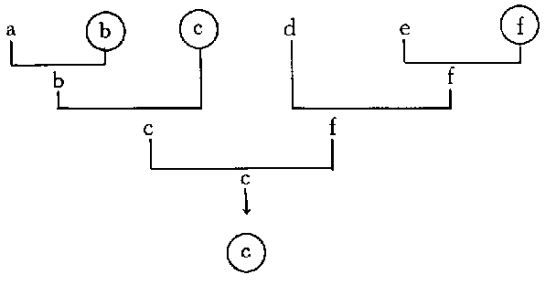
マッチングのグラフ



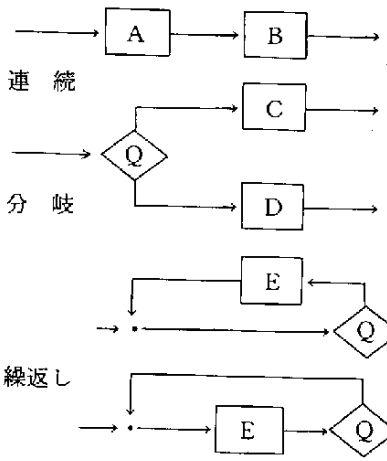
ネットワーク・グラフ



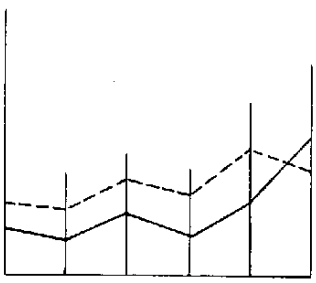
トーナメント図



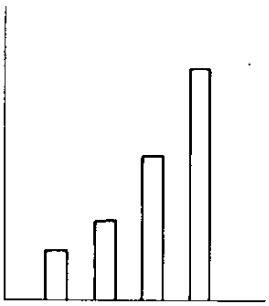
流 れ 図



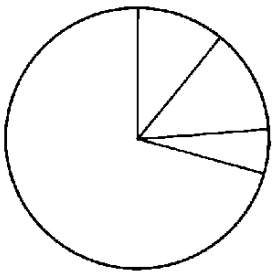
統 計 図



線グラフ

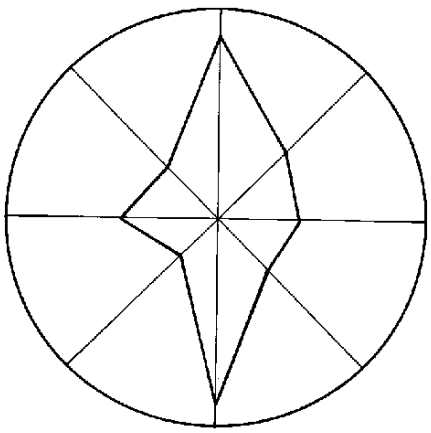


棒グラフ

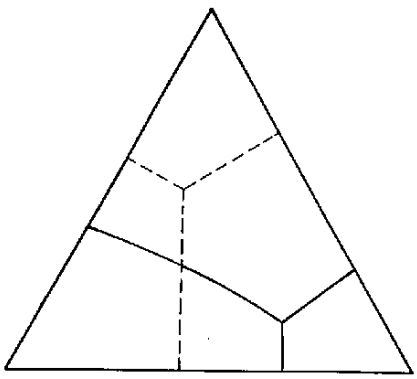


円グラフ

分 析 図

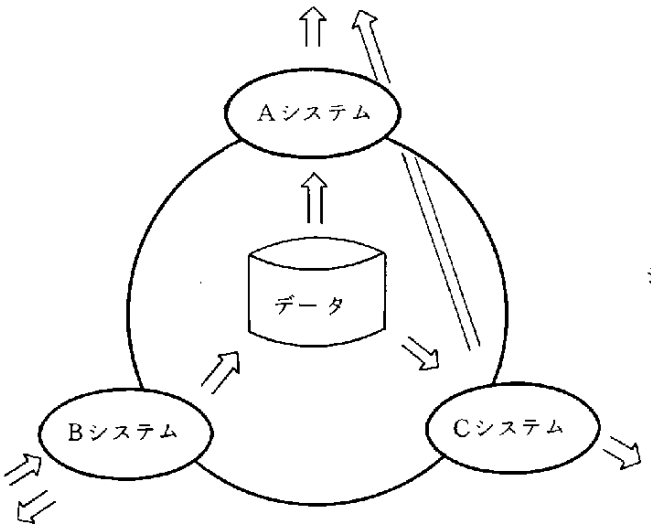


キビエット・グラフ
レーダ・チャート

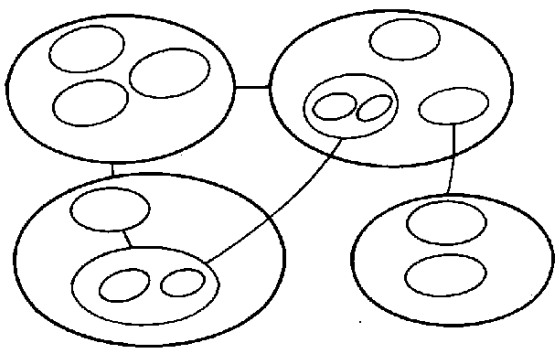


三角図表
メビウス図表

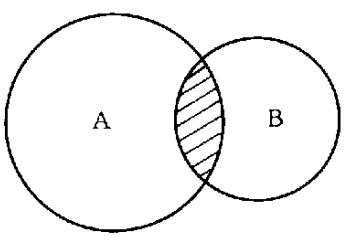
概　念　図



システム全体図



KJ法A型図解



A∩B

集合の表現
ベン図

参考文献

[1] Ackoff, R. L. and Sasieni, M., "Fundamentals of Operations Research," Wiley, 1968

[2] Albrecht, L. K.,"Organizations and Management of Information Processing Systems", Macmillan, 1973

[3] Archibald, R. D., "Managing High-Technology Programs and Projects," Wiley, 1976

[4] Benjamin, R. I.,"Control of the Information System Development Cycle", Wiley, 1971

[5] Brooks, F. P., "The Mythical Man-Month", Addison-Wesley, 1975

[6] Couger, J. D., and Knapp, R. W. (Ed.),"System Analysis Techniques", Wiley, 1974

[7] Davis, G. B. and Everest, G. C. (Ed.), "Readings in Management Information Systems", McGraw-Hill, 1976

[8] Dijkstra, E. W.,"A Discipline of Programming," Prentice-Hall, 1976

[9] Grindley, K.,"Systematics : A New Approach to Systems Analysis", McGraw-Hill, 1975

[10] IBM Corporation,"Business Systems Planning Guide, Application Manual", GE 20-0527-1, IBM, 1975

[11] Jackson, M. A., "Principles of Program Design,"Academic, 1975

[12] Katzan, H.,"Systems Design and Documentation : HIPO," Van Nostrand Reinhold, 1976

[13] Kernighan, B. W. and Pleuger, P. J.,"Software Tools", Addison-Wesley, 1976

[14] Martin, J.,"Design of Man-Computer Dialogues", Prentice-Hall, 1973

[15] McLean, E. R. and Soden, J. V.,"Strategic Planning for MIS," Wiley, 1977

[16] Metzger, P. W.,"Managing a Programming Project", Prentice-Hall, 1973

[17] Myers, G. J.,"Reliable Software through Composite Design," Mason／Charter, 1975

[18] Myers, G. J.,"Software Reliability : Principles and Practics," Wiley,

1976

[19] Smith, R. S., "Written Communication for Data Processing", Van Nostrand, 1976

[20] Thierauf, R. J., "Systems Analysis and Design of Real-time Management Information Systems", Prentice-Hall, 1975

[21] Yeh, R. T. (Ed.), "Current Trends in Programming Methodology", Prentice-Hall, 1977

[22] Yourdon, E. and Constantine, L. L., "Structured Design", Yourdon, 1975

[23] Gunther, R. C., "Management Methodology for Software Product Engineering", Wiley, 1978

[24] Hartman, W., Matthes, H. and Proeml, "Information Systems Handbook," Philips data system, 1968

[25] Glans, T. B., Grad, B., Holstein, D., Meyers, W. E. and Schmidt, R. N., "Management Systems", Holt, Rinehart and Winston Inc.,

[26] Weinwurm, G. F., "On the Management of Computer Programming", Van Nostrand Reinhold, 1970

[27] Brandon, D. H. and Gray, M., "Project Control Standards", Auerbach, 1970

[28] Brandon, D. H., "Management Standards for Data Processing", Van Noatrand Reinhold, 1963

[29] Gray, M. and London K., "Documentation Standard", Auerbach, 1969

[30] Kuehue, R. S., Lindberg, H. W. and Baron, W. F., "Manual of Computer Documentation Standards With Forms", Prentice-Hall, 1972

[31] Walsh, D. A., "A Guide for Software Documentation", Advanced Computer Techniques, 1969

[32] Katzan, H., "Systems Design and Documentation", Van Nostrand Reinhold, 1976

[33] Pollack, S. L., Hicks, H. T. and Harrison, W. J., "Decision Tables", Wiley, 1971

[34] Mckeever J. M., "Management Reporting Systems", Wiley, 1971

[35] Press, L. I., "Conversion of Decision Tables to Computer Programs", CACM, Vol. 8, No. 6, 1965

[36] Pollack, S. L., "Conversion of Limited Entry Decision Tables to Compu-

ter Programs ", CACM, Vol. 8, 1965

〔37〕 Reinwald, L. T. and Soland, R. M.,"Conversion of Limited Entry Decision Tables to Optimal Computer Programs", CACM, Vol.13 , 1966

〔38〕 Fisher, D. L.," Data Documentation and Decision Tables", CACM, Vol.9, No. 1, 1966

〔39〕 King, P. J . H.," Decision Tables", the computer journal , Vol.10, No. 2, 1967

〔40〕 Kemp, J . E.," Planning & Producing Audiovisual Materials", Chandler Publishing company, 1963

〔41〕 Minor, E. and Frye, H. R.," Techniques for producing visual instructional media", Mc Graw-Hill , 1970

〔42〕 高浜節夫編著「システム設計」産報

〔43〕 H. A. サイモン著（高宮晋訳）「システムの科学」ダイヤモンド社, 1969

〔44〕 西田, 疋田, 秋葉, 八柳「リアルタイム・システムズ・マネジメント」企画センター, 1972

〔45〕 川原裕「オンライン・システム導入のためのドキュメンテーション」企画センター, 1974

〔46〕 IBM社「SOP の考え方」「SOP の手法と文書化の手引き」1972（翻訳）

〔47〕 久保末沙, 国友義久「効果的プログラム開発技法」bit Vol 8, Vol 9 共立出版

〔48〕 宮本勲, 東谷秀夫「ソフトウエア・エンジニアリングへの道」bit Vol.9, Vol.10, 共立出版

〔49〕 国井利泰監修「ソフトウエア工学：要求仕様技術」共立出版, 1978

〔50〕 三浦宏文「システムと評価」共立出版, 1974

〔51〕 デービット・ノービック編（福島康人訳）「PPBS の理論と手法」日本経済新聞社 1969

〔52〕 Myers 著(久保末沙, 他訳)「高信頼性ソフトウエア」近代科学社, 1976

〔53〕 G. J . Myers 著（有沢誠訳）「ソフトウエアの信頼性」近代科学社, 1977

〔54〕 H. Mc Daniel 著 （岸田孝一訳）「ディシジョン・テーブル入門」日本経営出版, 1970

〔55〕 奥村誠次郎「グラフの知識と技術」実業之日本社, 1972

〔56〕 金田正也「教授メディアの整備」研究社, 1973

〔57〕 グレゴリー著（金子隆芳訳）「インテリジェント・アイ」みすず書房, 1972

〔58〕 南川利雄「表・グラフ作成の要領」同文館出版, 1966

〔59〕 中村正一「グラフとチャート」日刊工業新聞社，1967
〔60〕 サイエンス「コミュニケーション特集」日本経済新聞社，1977
〔61〕 日本の科学と技術「特集コミュニケーション」日本科学技術振興財団，1975
〔62〕 サイエンス「イメージの世界：特集　視覚の心理学」日本経済新聞社，1975
〔63〕 堀淳一「地図のたのしみ」河出書房新社，1973
〔64〕 河原淳「新版イラストレーション」ダヴィット社，1971
〔65〕 真鍋一男，大辻清司「マーク／フォト／イラストレーション」美術出版社，1966
〔66〕 植草甚一「ぼくがすきな外国の変った漫画家たち」青土社，1971
〔67〕 Engineering Concepts Curriculum Project 著（平松啓二訳）「技術評価の工学入門」オーム社，1973
〔68〕 白根禮吉「テクノロジー・アセスメント」日本経済新聞社，1973
〔69〕 高橋希一編「テクノロジー・アセスメント入門」竹内書店，1972
〔70〕 石本幹雄，猿橋皓「テクノロジー・アセスメントの進め方」日刊工業新聞社，1974
〔71〕 土木工学大系「環境アセスメント」彰国出版，1976
〔72〕 吉川博也編「環境アセスメントの基礎手法」鹿島出版，1975
〔73〕 エーリッヒ・ヤンツ著（マネージメントセンター訳）「技術予測」マネージメント・センター，1968

第２章　システム資源の選択

キーワード

反復性 (iterative nature), 保全性 (integrity), システム設計の型 (types of system design), 資源 (resource), 設備計画 (equipment planning), ハードウエアの選択 (hardware selection), ソフトウエアの選択 (software selection), 設備能力配分 (equipment allocation), 自動評価法 (automated evaluation techniques), ハードウエアの発注(hardware purchase), 評価手続(evaluation procedure), 競争入札 (competitive bidding), 提案書 (proposal), 性能表(property list), 価格表 (price list), カタログ(catalogue), 費用見積 (cost estimating)。

目　　標

学習体系の構成として，システム設計の基本事項を次の３章で分担する。即ち，

２章　　設備計画（ハード／ソフト）

３章　　プログラム計画

４章　　データの設計

本章では，システム設計の前提条件としての比重の大きい基本枠組に関する問題を主に採りあげる。それはシステム資源の選択の判断をどう考えるべきかという着眼点の整理である。細目としては次にあげるような学習項目である。

・ハードウエア資源の評価方法

・ソフトウエア資源の評価方法

・自動化された評価手法

・システム資源の詳細見積

これらの学習にはＣ３，Ｃ４で学んだ知識を前提として必要とする。

Ｄ１での学習について触れると，Ｄ１の４章で，システム開発予算について学んでいるが，予算について見るとＤ１の４章でとりあつかうのは，「予算の決め方」であって，この章で学ぶのは「予算の有効な使い方」である。またＤ１の５章で，費用の考え方を学んでいるが，その際の着眼点は，システムの「有効性」に焦点があてられていた。この章で学ぶのは，システムの「効率」という点からみた費用効果の分析である。内容的に重複する面もあるが，着眼点

の差異に留意する必要がある。

本章の主目標は，システムを成立させるために必要な条件としての広義の設備計画（ハードウエア／ソフトウエア）の評価方法の基礎を明確にすることである。評価手法の実例と，費用見積りの実施例についても触れる。但し，見積り作業に必要な具体的資料等の例示についてはＤ４の単元でとりあつかう。

内　　容

2.1　ハードウエア資源の選択

情報処理システムの中心になるハードウェアは言うまでもなくコンピュータ・システムである。情報処理の形式を大別すると，バッチ処理を主とするオフラインシステムと，リアルタイム処理を主とするオンラインシステムの２つの形態がある。コンピュータは歴史的には前者を中心にして発達して来たため，その機器構成はバッチ処理型のものが主流になっているが，情報要求の本質（即時性など）からみると，今後の設備検討には，オンライン・リアルタイム処理のための機器構成を重視しなければならない。設備規模の検討は，中央処理装置の能力の想定から始めるのが従来のパターンであるが，むしろ端末と通信網の計画から出発する方法も検討してみるに値する。

学習の便宜上，ハードウェアの選択，ソフトウェアの選択，端末及び通信系の検討などの項目にわけて学習するが，これらは個々独立に検討すべきものでなく，これらのシステムエレメント間の相補的な関係の研究を重視すべきである。

(1)　能力の定量的評価

ハードウエア能力を静的に評価するものであり，カタログ値にもとづく選択基準の作成ともいえる。

学習方法として、一定の条件（例えば，投下費用，中央処理装置の速度など）のもとに，各社のカタログを取り寄せて比較表をつくることなどが有効であろう。

評価項目としては次のようなものがある。

①　中央処理装置（central processor）

- 処理装置の数
- 命令のオーバーラップ機能の有無
- サイクルタイム
- 演算方式
- 四則演算の速度
- 割り込みのレベル

・レジスタ数

・命　令　数

・アドレス方式

・実時間回路

などが主要な尺度である。

② 主記憶装置 (main storage)

・語長

・サイクルタイム

・インタリーブ

・オーバーラップ

・素子・集積度

・記憶容量

・増設単位

・ページ語数

・記憶保護

・停止時保護

③ バッファ記憶装置 (buffer storage)

別名キャッシュメモリ (cash memory) と呼ばれており，主記憶装置と中央処理装置との間のデータ転送に使われている。

この装置は，処理の高速化に対処するため考え出されたもので，採用されている機種もある。

・記憶容量

・サイクルタイム

・素子と集積度

など，主記憶装置を同様の尺度で評価する。

④ 入出力制御装置 (input output control)

・チャネル数

・チャネルのタイプと数

・転送速度 (transter rate)

⑤ 補助記憶装置

・記憶容量

・記憶密度

・検索時間（access time）

・回転待ち時間

・ディスク・パックの場合の取りはずしの可否

・操　作　性

⑥　入出力装置

・速　　度

・操　作　性

・プリンタの場合

活字体，複写，可能枚数，活字交換

⑦　通信制御装置

・直並列変換

・回線数

・通信速度

・信頼性

・バッファ機能

・時分割機能

などがある。

これらの性能尺度はその価格との対比で整理する必要がある。或いは傾向の推移をみるための尺度として，記憶装置の１ビット当りのコストとか演算装置の１命令当りのコストなどを時系列にプロットして，現在及び将来の概略値をつかむことも，しばしばおこなわれている。

(2)　附帯的条件の評価

ハードウエア選択についての附帯的条件について，次のような項目を評価する。

①　保有機種との互換性

プログラムの互換性が最も要求されるものの一つであるが，

・磁気テープ

・通信回線

によるデータ受渡しの可能性についても検討する。

②　導入に要する期間

メーカまたは機種により異なるので検討を要する。

③　設備条件の検討

・設置面積

・消費電力

・電源条件

・空調条件

・床荷重

・調整期間

・搬入出条件

④ 保守

・保守費

・保守体制

・保守の容易性

⑤ 拡張性

各装置の拡張性を検討する。

⑥ バックアップ

・他のコンピュータの利用の可否

・導入前のテストの可能性

⑦ 信頼性

故障のひん度，修理に要する時間などハードウエアの信頼性を評価する。ただし，新機種の場合は，導入事例が少ないため，メーカまたは販売会社からの情報入手に努力する。

(3) 処理系としての機器の結合方式

（機能の特性，信頼性などの定量的表現）

(a) 単一システム

(b) 複数システムの結合

・機能分割型

・負荷分割型

・待機・予備型

・二重化型

・階層型

・並列型

(c) コンピュータと通信系との結合

端末機器の種類と数

通信網の結合形式

回線の種類，媒体

集配信，交換，変復調の方法

伝送方法の実際例

(4) システム仕様要求の適合性評価

システム仕様を設備要求の観点から検討し，コンピュータ能力の定量化と対応させて要素に分解する。

① システム仕様に対するコンピュータ能力の適合性の判定

・処理速度の要求

・プログラムの大きさと容量の対応

・データ量と容量の対応

・処理のタイミングの適合

・入出力形式の人間工学的考慮

② 拡張性に対する予測

（システムの拡張性設計）

・将来のシステム拡張計画を想定する。

・システム・コンバージョンの手順を考察する。

③ 信頼性の予測

（システムの信頼性設計）

・故障発見の方法をあらかじめ検討しておく

・修復の手順を設計しておき，場合によっては組込む等の予備手順をとっておく

・特に端末も含めて系全体のMTBF，MTTR等を算定しておく。

2.2 ソフトウエア資源の選択

ソフトウエアについては，個別機能の抽出が難しいため，ハードウエア選択のような定量的な尺度の整理はあまりおこなわれていない。

(1) ソフトウェア単体の評価

評価するためにとりあげるソフトウェアは，次のように分けて説明する。

・オペレーティング・システム

・言語プロセッサ

・オンライン・サポート・ソフトウェア

・データベース・マネジメント・ソフトウェア

① オペレーティング・システム

オペレーティング・システムについては，その発展の経過で，機能の多様性が望まれる

ため，性能評価の定量的尺度は二義的に考えられていた。しかしながら今後はその選択という立場を考えると，例えばオペレーティング・システム自身のレジデンスの大きさとか，オペレーティング・システムの働きによるシステム性能の効果の良否については，きびしく検討すべきである。オペレーティング・システムの機能はそのような項目に分類して考えることができる。

- 入出力制御
- タスク管理
- データ管理
- ジョブ管理
- ＴＳＳ機能

さらにつけ加えると，次のような点があげられる。

- 機能の特性
- 入出力管理
- データ管理
- 主記憶域に対するオーバヘッド
- 常駐域に対するオーバヘッド
- 各機能の処理速度
- 言語機能
- ライブラリの種類
- 通信制御機能
- デバック支援機能
- 使いやすさ
- メモリ・サイズ
- 障害管理

② 言語プロセッサ

- 複雑なプログラムの能力
- オブジェクトプログラムの能率
- プロセッサのオーバヘッド
- プログラム作成の費用
- コンパイルの時間
- 言語の説明文書
- デバッグのための支援機能

・互換性

・データ通信との連結

・データベースとの連結

・その他ソフトウェアとしての信頼性など。

③ オンライン・サポート・ソフトウエア

・リエントラント機能

・多重度の管理

・バッチとの相互乗り入れ

・メモリ管理

・障害対策

・使用メモリ数

・オペレーティング・システムとの関連

④ データベース・マネジメント・ソフトウエア

・データ・ベース構造

・サポート言語

・利用形態

・記憶構造

・障害対策

(2) ハードウエアとの組合せ

ハードウエアとの関連も強いので，分離してソフトウエアだけで比較検討できないこともある。

実際的な評価尺度として，ハードウェアとソフトウェアの組合せ全体に対して，次のようなチェックポイントを設定する方法が習慣的におこなわれている。

① 仕様に対する応諾（compliance）

② 価格性能比（price-performance）

③ スループット（throughput）

④ 拡張性

⑤ 信頼性

⑥ ユーティリティ・プログラム等（デバッグ支援も）

⑦ 言語プロセッサ（機能と効率）

⑧ オペレーティング・システム（機能とオーバヘッド）

⑨ 保守費

⑩ 記憶装置の拡張性，入出力装置の拡張性

⑪ 入出力装置の種類，速度，費用

⑫ 納入時期

⑬ 命令の種類と能力

⑭ 通系制御，マルチプログラミングの機能

⑮ 入出力チャネルの特性

⑯ 装置の切換（device switching），マルチプロセッサ，共有ファイル（shared files system）などの能力

⑰ 電力，床面積，空調などの要求事項（physical requirement）

⑱ 内部処理速度

⑲ 他のコンピュータによるテストの可能性

⑳ バックアップ態勢（他のコンピュータの利用）

㉑ 保証事項

㉒ レンタル制の場合のキャンセル等の契約上の制約条項

⑲～㉒はメーカーに対するチェックポイントである。

2.3 システム効率の評価

前節でとりあつかったハードウェア／ソフトウェアの選択手順はシステム設計としてのオーソドックスな手続を踏んだものである。このような方法とは別に，コンピュータの能力の評価方法としてパターン化された各種の方法が存在している。ここではそれらのうち二三を学習してみる。

パターン化された評価方法とは次のようなものをいう。

- ギブソンミックス法等
- ベンチマークテスト
- シミュレーション
- 自動評価のパッケージプログラム
- コンピュータ稼動診断

これらの評価方法を説明する場合，長所，短所についても整理しながら解説する方法をとる。

(1) ギブソンミックス法

ギブソンミックス法はコンピュータの能力評価の方法として米国宇宙局で開発されたものである。基本的な演算機能の処理時間に重み係数を乗じてその積和を計算してコンピュータの能力比較の指標にしようとする考え方である。ギブソンミックスの係数では，科学技術計

算の比重が高いので，事務計算用のコンピュータの評価にはむかない面がある。これを修正する意味で分類集計の計算機能に対する重みづけを高くしたコマーシャルミックス法なども提案された。これらの評価法はコンピュータの計算能力のみに偏向した考え方であったので，さらにこれらを修正して，ファイル更新能力を重視するアウェルバッハ法などが提案された。これはベーシック・ソフトウェアの能力の評価も加味したものである。評価尺度の指標を設定するこの種の方法は，比較評価を概括的に行うのに便利である。既製の係数をそのままうのみにして使うだけではなく，対象問題のシステム設計に対応させて最も適合する係数を新たに設定するという試みも学習の一助となる。

(2) ベンチマーク・テスト法

ベンチマーク・テストとは，実際の業務をそのままに反映させるような標準問題を設定して，そのプログラムをそのままコンピュータで処理させてみて，その総合的な処理能力を判定する方法である。コンピュータの能力を要素毎に評価するのではなく，全体としての処理時間，使い易さ等が把握できるという利点があるが，難点として標準問題の設定が難しいという点と，テストに使用出来るコンピュータが実在していない場合には不可能であるという点がある。候補機種を実際に使用して優劣の比較が出来るという点は最も確実な評価方法である。システム設計が十分に進行した段階で，そのプログラムを使って確認することが出来れば最も確実な評価方法になる。対象システムに適合する標準問題作成のための着眼点などを整理してみるのもよい。

計測には，オペレーティング・システム組み込みのソフトウエア（ＳＩＰ）によるカウントとハードウエア計測機（ＨＩＰ）による計測があり，次のような点を比較する。

平均命令実行時間

オペレーティング・システムのＣＰＵオーバヘッド

プログラムのＣＰＵ利用率

Ｉ／Ｏの利用率

ターンアランド（またはＥＬＡＰＳＥＤ）時間

バッファ・ストレージがある場合のＨＩＴ率

コンパイル，アセンブル時間

(3) シミュレーション法

シミュレーションによる予測は，システム条件の中の不確実な部分を確めたい時，あらゆる分野で利用出来る。ハードウエアの機能，ソフトウエアの機能の両面に利用出来る。複雑な処理系を考えた場合の総合的な応答時間の推定などには待行列のシミュレーションが使われる。シミュレーション・モデルの設計の良否に左右されるという難点はベンチマークテス

トの場合と同様である。シミュレーションに使用するプログラムとして新たに専用のものを作ってもよいが，パッケージ・プログラムとして用意されている汎用シミュレーション・プログラムを使う方が能率的である。

設備検討のためのシミュレーショングラムの使用方法を研究してみる。

(4) 自動評価のパッケージプログラム

自動評価のプログラムとは，最近コンピュータ・システムそのものが複雑化巨大化したため，単純な評価方法を採用することが次第に困難になったことの打開策として開発された評価専用のプログラムを云う。この種の評価プログラムではオペレーティング・システム等のシステムプログラムの性能評価も行えるし，コンピュータ・システムの要素の組合せ方やオペレーティングシステムの各機能の有効な利用方法の探索にも使える。この種の評価プログラムがどのような比較項目をもっているかを研究してみるのもよい。

SCERT，CASEといったパッケージ化された専用のプログラムが発表されている。

(5) コンピュータ稼動診断

シミュレーションやベンチマーク・テストはテストのために一定の環境を人為的に設定するものであるが，稼動診断は実行中の通常の環境で測定をおこなうものである。自動評価のパッケージプログラムの場合はソフトウエア・モニタであるため短所としてモニタ自身による外乱の影響をまぬがれないし，OSの変更に追従するのが困難になることもある。この短所をおぎなうために，ハードウエア・モニタが使用される。コンピュータのメーカ自身が，CE用として診断用のモニタを保有していることが多いが，たとえば一定時間内に処理する業務量をもう少し増せないか，業務のピーク時の稼動率を平均化できないか，結果ができるまでの時間を短縮できないか，などという問題を検討するための診断には，CE用のハードウエア・モニタだけでは不足する。これらの問題解決のためにハードウエア・モニタと分析評価のシステムとを組合せた稼動診断がおこなわれる。稼動診断は次の手順でおこなわれる。

① 調査と目標設定
② ハードウエア・モニタによる測定
③ 分析
④ ボトルネック抽出→対策
⑤ データの表示と記録
⑥ 評価

上記の手順のうち②と⑤に稼動診断のためのハードウエアが用意されている。

ますます複雑になるリアルタイムシステムの効率向上のためには，今後はこのような稼動診断も必要になる。

表２－１　コンピュータの自動評価法

(1) 評価尺度の指標化
ギブソンミックス法
コマーシャルミックス法
アウェルバッハ法
(2) 業務プログラムによる試計算
ベンチマークテスト
実プログラムによるテスト
(3) シミュレーション
応答時間の予測
業務拡大状況の予測
回線，チャネル，ディスクＩ／Ｏの頻度
(4) パッケージ型評価専用プログラム
ＳＣＥＲＴ等
(5) ハードウエア・モニタとの組合せに稼動診断
実機に対し，実行環境中での診断

2.4 設 備 計 画

システム仕様を設備要求の観点から検討し，コンピュータ能力の定量化と対応させて，設備の選択をおこない，その結論を記述したものが，設備計画書である。

設備計画の標準的な手続は次のように要約される。

(1) ハードウェア／ソフトウェアの能力

- コンピュータの能力をたしかめる
- コンピュータの結合方式を決める

(2) システム仕様との対応

- システム仕様をコンピュータの要素能力の表現形式に分解する。
- システムの拡張性を考慮する
- システムの信頼性を算定する

(3) 設備要素の能力配分表を作る

- 設備選択だけでなく，負荷配分をあらかじめ決めておく。
- これにはプログラム構造，データの流れの概略を想定しておく必要がある。

設備計画書には，どういう考え方でその設備を選んだかを記述しておく必要がある。その一例として，システム仕様からコンピュータの負荷をどう推定したかという点があげられる。こ

のためには，コンピュータ等の各システム要素に対する負荷の配分表の形にまとめることが望ましい。負荷の配分予定表の形に表現しておけば，単一処理系による複数システムの運営の場合にも計画資料として使用出来るし，システム実施後の予定と実績の対比検証の資料とすることも出来る。システム規模の拡張の参考資料にもなる。

設備計画を行うためには，上記のような設備そのものに関する知識のほかに，システム設計の内容についてもあらかじめ概要をつかんでおくことが必要になる。

システム設計の詳細については今後の章で学習する予定ではあるが，設備計画のための判断に資する範囲で最小限次のような項目についてあらかじめ把握しておくことも望ましい。

① プログラムの構造の把握

- プログラムの簡素化（標準ルーチンの利用）
- メモリスペースの効率的利用（セグメンテーション，オーバレイ等）
- プログラムのモジュール構造（リロケータブル・リンケージ等）
- 入出力装置の割等と処理手順（オペレーティング・システムのタスクコントロール）
- 言語処理プログラムの機能

② データの流れの把握

- インプットデータの仕様
- アウトプットデータの仕様
- 中間結果
- 非同期処理の状況

これらに関連する知識として

③ オペレーティングシステム

④ データベース管理プログラム

などの性能についてもあわせて検討しておくことが要求される。

設備の選択は，フィジカル設計の出発点である。設備を決める場合，何が重要な要因になるかどうかという判定は，システム設計の考え方でもある。

このように，コンピュータを含む設備計画には、技術的にみて，通常の設備計画よりも，考慮しなければならない注意事項が多いが，費用面からみれば，通常の設備投資計画のための手法が適用できる。例えば，MAPI法（Machinery and Allied Products Institute）あるいは，New MAPI法などによって設備更新の時期を検討してみる試みなどは，投資がもつ意味をつかむ上で有効である。但しコンピュータの場合は一般機械の稼動劣性に該当するものは陳腐化であるが，これらの推定は難しい。設備の更新についての経済的判断をどう考えるかという点を学ぶことに重点をおく。システム設計のための設備計画でもう一つの難しさは，ハ

ードウェアの選択が，それだけで独立に経済評価が出来にくいという点である。ハードウェアの選択が，ベーシックソフトウェアの選択も含んでいて，しかもその性能が，以後のシステム開発コストに影響をあたえるという因果関係がある。

機能の価値或いは投資の採算性という分析的な評価方法の他に，いくつかの評価要素（安定要因，成長性要因，市場性要因，経済性要因など）をあらかじめ設定しておいて，これらについて評点法，プロファイル法（評点をグラフ表現したもの），チェックリスト法などによって総合的にとらえて判定する方法などもある。

設備選択の問題は，以上のようにいろいろな方法があるが最終的には意思決定の問題に帰着して，純粋に合理的に決められない面もある。重要な要因は，その設備選択に対して，システム開発グループの合意が得られているかどうかという点である。

設備計画を順調に進めるためには常日頃機器に関するメーカーのカタログを整理して収集しておく必要がある。ハードウェア，ソフトウェアの性能や機能を比較検討するための一次資料はこれらのカタログ類である。しかしながらこの記載にはメーカーの宣伝的脚色も含まれているので客観的比較資料にはなり難いという欠点も多い。これらに対する補足修正資料として，その機種のユーザーからの情報，他メーカーからの批判等が二次資料となる。概括的な比較資料として，アンケート集計表，ハンドブック等の一覧比較表は三次資料となる。特定の機能や性能のテストの実施結果が学会などに公表されることもある。

設備計画の基礎は実務上の視点からみた場合，結局のところ，技術資料の収集とその整理蓄積である。したがって，ここでの学習項目は最終的には，技術資料をどのような方法で整理しどうやって活用するかを検討することになる。

- 技術資料の分類区分
- 収集蓄積方法
- 活用のための整理方法
- 設備計画に必要とする項目の抽出

2.5 設備選択の実務

前節までに学習した設備選択のための基本は，システム設計として理想化された手順にそったものであった。しかしながら理論に対する実際という立場も忘れてはならない。この節では，実務的プラクティスを概観することにする。

(1) システム評価の体系

コンピュータの実用化の発展に伴って，様々な経過をたどって，現在実施されているシステムは，大型化，複雑化し，システム管理の方法の見直しの必要が感ぜられるようになって

いる。このような認識は，主として，システム開発費とその維持管理費の増大をくいとめたいという意図に発している。そしてそのための対策として何らかのシステム評価制度が，組織的に実施されているのが実状である。その骨格はおおむね次に示すような手順になっている場合が多い。

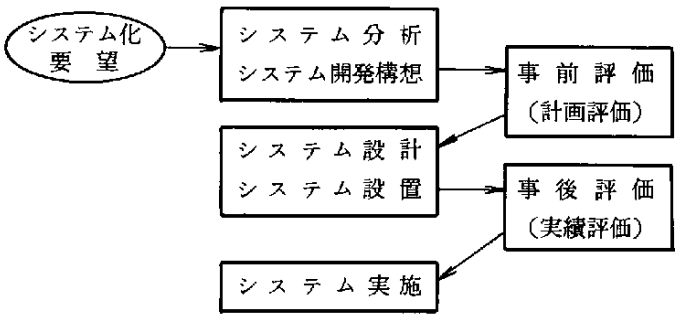


図２－１　システム評価の体系

このような評価制度のねらいは，組織的承認を明らかにすることと，開発段階の区切りを確認することである。システム評価制度は一つにはこれから開発にとりかかろうとするシステムに適用される場合と，他の一つにはすでに稼動中のシステムに対して適用される場合の二通りがある。設備選択の問題は，これからとりかかるシステムの場合の事前評価の一部分に組込まれている項目であることが多い。したがって実務的学習としては，現存している評価制度の中の具体例について，そのチェックポイント等を調査することなどが考えられる。

(2)　設備計画のコスト分析

設備には耐用年数がある。設備計画の前提に想定しておくべき事項は，想定した耐用年数に対する経済性の試算である。即ち

- 設備投資の資本回収計画
- 設備の想定寿命，耐用年数，陳腐化率
- 減価償却の計算

などである。購入した場合のソフトウェア・パッケージなども最近は固定資産とみなされるので，設備とは単純にハードウェアだけと考えることは出来ない。

コンピュータの価格については，さらに詳細な検討が必要になる。その理由の一つは，コンピュータ費用に対するレンタル制の普及である。コンピュータの費用については，買取りの場合とレンタル制の場合の比較，さらには，買取りの形の変形としてのリース契約の場合との比較なども考慮しなければならない。一方コンピュータそのもののもつ技術革新の早さのため陳腐化の速度も普通の機械にくらべて早いため，単純な経済計算のみで判定出来ない点があることが，設備計画問題を複雑にしている。これらの事項に関する実務的な考察事項

を学習しておく。

(3) 機種選定の実務的手順（提案書）

システム設計に基づいて設備を選定するという手順にしたがった理想的な設備計画の方法では，システム・コンポーネントをメーカにこだわらずに任意に選択し，その組合せとしてシステムを構築するという過程になるはずである。しかしながら実務的には，このような理想化された手順をとることはまれである。実際上普通におこなわれている設備計画は，機種選定のために，メーカーから提案書を提出させて，その内容を比較検討するという方法が採られている。このことは，情報処理システムのほとんどの機器が，それぞれのメーカーで総括的に製造されているという現状に起因するものである。このような事務的習慣による「提案書方式」の実態について一通りの知識も心得ておく必要がある。

この場合，発注者側は，システム設計の概要をシステム仕様として複数のメーカーに明示し，夫々のメーカーは提案書という形式で，メーカー側からみたシステム設計を提出する。その上で競争入札によって最終的にメーカーを決定する。

- 発注者のシステム仕様
  - システム・フローチャート
  - システム・ファイルの詳細
  - データ量の増加の予想
- メーカーの提案書
  - ハードウェア
  - ソフトウェア
  - 要員
  - パフォーマンス
  - 拡張性
  - メーカーからの開発援助体制
  - レンタルと買取の価格その他
- （他の納入先ユーザの状況）

極端な場合には発注者側はみずからシステム設計をおこなう必要もなく，競争入札のため設備費用も自然に押しさげられ，きわめて楽になる方法であるが，この方法によって最適なシステム設計を得られることは少ない。しかしながら多くの類似システムを経験したメーカーの場合には無難な設備計画を提案して来ることもあるという利点がある。

表２－２　設備選択のプラクティス

①　システム評価の体系の一部として
（組織的システム評価制度）
・事前評価……チェック・ポイント・リスト
・稼動評価
②　設備計画のコスト分析
・耐用年数と減価償却
・買取とレンタル制の比較
③　提案書制度
・発注者側のシステム仕様
・メーカー側の提案書
・競争入札

指導上の留意点

我が国でのシステム計画は，まずいくつかの主要なＥＤＰＳ化対象業務を想定し，それらの全部を消化するのに必要な能力に幾分かの余裕をもたせたハードウェア計画が先行する。導入計画にたずさわる人々の中でも対象業務についての考え方に認識の精粗の差があるのが普通であるので，コンピュータの能力をどの程度にすべきかという点ではその決定のためのよりどころははっきりしないことが多い。極端ないい方をすれば，ハードウェアの選択に関しては，その時期での同業他社の動向とか，設備予算の余裕とか，宣伝効果とか，システム設計とは別次元の要因で決定されることが多い。ハードウェアの選択の決定がまず先におこなわれて後に，想定されていた複数の対象業務があらためて検討されるので，それらの実施は，プログラム作成の完了した順に，さみだれ式に実行に移される。したがってプログラム作成のおくれたものほど，ハードウェアの容量の不足や，ハードウェア／ソフトウェアの組合せの不整合が多くなる。これらの不足不整合がめだちはじめる段階に来て再び設備の増強が検討される。これらの経過を一言でいえば，ハードウェアとソフトウェア（アプリケーションプログラム）の両者を同時に考えたシステム設計がなされてないと言ってよい。

システムとは共通の目的を達成するために結合された多くの部分の複合体である。システムの部分集合には，この章でとりあつかうシステム資源としてのハードウェアとソフトウェアのほかに，アプリケーションプログラム（次章以降で学習する内容）や，システムの運用に関与する人間，システム運用の操作や手順（広義のソフトウェア）などがある。

本章で学習すべき項目は，その中の基本的ハードウェアとソフトウェアの組合せをシステム目的に最も適合した形に選択する方法を研究することである。この基本線をあきらかにしてお

けば，我が国で実際におこなわれているシステム計画の実状（一つのコンピュータに異質のシステムを乗合自動車の乗客のように押し込むやり方）に対しても実際的な判断基準をさらに明確な形で提供できるであろう。

この節での学習に最も関連の深い予備知識はＣ３の単元である。

(1) カリキュラムの中には，やや詳細にわたる，検討項目の例示がおこなわれているが，これらの一つ一つを検討する必要はない。設備計画とは，ここに例示してあるような項目を多角的に検討することが，大切な点であることを指摘し，システム設計の枠組の大要を教示する。

(2) システムの効率を重視するという視点から，パターン化された自動評価手順を採りあげてある。これらの評価方法が，システムの効率をどのような着眼でみているかを研究させる。

(3) 設備投資計画を費用面からみて研究する。即ち減価償却とか運営費用とかについて考察する。この章で学習する費用の問題は運用費に重点がある。（単元Ｄ1の第５章と対比すると，Ｄ１の場合は開発費に重点がある。）

参　　考

MAPI法

M.A.P.I法というのはTerborgが1949年に発表した“Dynamic Equipment Policy”に述べられている考え方を基にして，MAPI（Machinery and Allied Product Institute）が刊行した“MAPI Replacement Manual”（1950）による設備投資のための方法論である。

MAPI法は設備問題の考え方を数学的に単純化して，従来漠然といわれていた陳腐化という概念に計量的基礎をあたえている。即ち，稼動劣性（operating inferiority）という変数を考え，設備の稼動劣性は線型（linear）に増加すると仮定する。その年間成長度を劣性度（inferiority gradient）g とすると，処分価格を０とした時の第１次近似式は

$$n=\sqrt{\frac{2C}{g}}\ ,\quad AM=\sqrt{2gC}\ +\frac{iC-g}{2}$$

n：新設備の最も経済的な使用年数

c：新設備のコスト

i：設備投資に用いる資金コスト

ここでＡＭ（adverse minimum）は稼動劣性（年々増大する変数）と設備の年平均資本費（年々減少する変数）の両者の平均値の和（ＡＡ：adverse average）の最小値である。新旧両設備のＡＭを比較して，費用最小にするには，

$AM_0 > AM_1$ ｛０：旧，１：新　をあらわす｝

の不等式が成立する時を投資時点とするのがMAPI法の骨子である。

new MAPI 法は，上記のold MAPI 法を発展させて，Terborg が1958に発表した。"Business Investment Policy" の考え方である。ここでは独立投資と取替投資を同一の基盤で評価するための比率として，一種の収益率(relative rate of return)を提起している。この比率によって投資の順序づけをおこなうので，Terborg はこれをurgency rating と呼ぶ

$$r = \frac{P_1 + (R - R_1) - (C - V_1)}{C - R}$$

C ：新設備の現在価値

P_1：取替を1年のばした時の損失

R ：旧設備の今日の残価

R_1：旧設備の1年後の残価

V_1：新設備の1年後の残価

これで独立投資の場合は $R = 0$，$R_1 = 0$ で

$$r = \frac{P_1 - (C - V_1)}{C}$$

となる。

old MAPI は設備投資のタイミングを考えるための手法であり，new MAPI は投資の順序づけをおこなう手法である。

new MAPI 法は，

- new MAPI system（資本回収法）
- new MAPI formula（設備減価の減耗率）の二つで構成されている。

参考文献

〔1〕 Fisher, G. H.,"Cost Considerations in Systems Analysis", American Elsevier, 1971

〔2〕 Seiler, K.,"Introduction to Systems Cost Effectiveness", Wiley, 1969

〔3〕 English, J. M.,"Cost Effectiveness", Wiley, 1968

〔4〕 Head, R. V.,"A Guide to Packaged Systems", Wiley, 1971

〔5〕 村川武雄「設備投資の経済計算とその理論」日科技連，1961

〔6〕 千住鎮雄，伏見多美雄「設備投資計画法」日科技連，1974

〔7〕 石尾登「企業の採算計算」日刊工業新聞社，1964

〔8〕 G. H. フィッシャー著（OR学会訳）「システム分析における費用の扱い」東洋経済新報社，1974

〔９〕　S. ステイムラー著（大野豊訳）「コスト／パフォーマンス」日本経営出版会，1972

〔10〕　横山　保編「設備問題への経営科学的接近」培風館，1964

〔11〕　デービッド・B・ハーツ著（松田武彦監訳）「マネジメント・サイエンス入門」ダイヤモンド社，1974

〔12〕　岩城三郎編「コンピュータ・システム拡張変換の手引」企画センター，1973

〔13〕　佐藤　良「コスト工学」産業能率短大出版部，1967

第3章　ソフトウェアの設計

キーワード

モジュール化 (design modularity), サブシステム (subsystem), サブシステムのインターフェイス (subsystem interfaces), 構造化プログラミング(structured programming), モジュールの強度 (module strength), モジュールの結合 (module coupling), 機能 (function), 性能 (performance), 構造 (structure), 階層図 (hierarchy chart), 機能図 (factoring diagram), 流れ図 (flow chart), 標準手続(standard procedure), 技法集 (programming standads), コード設計 (code design)

目　　標

本章では，ソフトウェア設計の中核部分であるプログラム設計の技術を学習する。

モジュール化，サブシステムのインターフェイス，プログラムの標準化などの項目を学ぶ。特に最近その体系化が進展しつつあるプログラム設計手法の大略を全体的に概観する。内容的にはＣ４の６章でとりあつかった事項の各論であるが，トピックスの紹介という程度でなく，その底流にある設計思想を考察させるという方向が望ましい。

設計手法としては，Ｄ２の１章で，網羅的知識として一応学んであるが，それは或意味では道具或いは材料としての知識であって，この章で学ぶことは，それらの材料をもとにして，プログラム設計を構築する技術である。ソフトウェア設計の成果物は，プログラム設計書である。この章の主目標は，プログラム設計書の要件を整理し把握することに集約される。

内　　容

3.1　ソフトウェア開発の現状

プログラム開発は，コンピュータで実際に実行されるプログラムを作成することであるので，その内容は具体的現実的な方法の集積である。ところが，その内容が具体的であるのに反して，開発技術を体系的に整理することは最もおくれている分野である。

ここではまずはじめに，この点から考えてみることにする。コンピュータ技術は，誕生後日が浅く，他の工業技術にくらべて熟成していない面が多い。技術というよりも職人技能であると言われることもある。この点はプログラム開発のようなソフトウェアの分野で特に顕著である。

職人技能であると指摘される点はどんな面だろうか？　その理由は何なのか？　技能から技術へ昇格するための改良はできるのか？　技能にとどまるような本質的な難点があるのだろうか？　等々について観察し，考慮してみることは，開発技術を深く学ぶ上で価値がある。

着眼点として次のような事柄から考えてみるとよい。

「プログラム」を，一つの生産物としてみて，一般の工業製品と比較してみると，次のような諸点が指摘される。

(1) コンピュータ技術は言語を媒介にして，ソフトウェアの生産をおこなっている。この点で，ハードウェアを媒介にしている一般工業製品と性格の違う面がある。ソフトウェア（言語）を媒介にすることによって，その生産活動はハードウェアから独立し，ソフトウェアがソフトウェアを生むという増殖系になっている。（工作機械が機械を作り出し，その循環が次第に機械工業を精密化した歴史と似た点もある。）

……プログラム開発の現状を認識する上で，言語の研究と言語の選択の問題は一つの重要なテーマである。コンピュータ技術での言語の発展経過を概観すると次の図式のようになる。

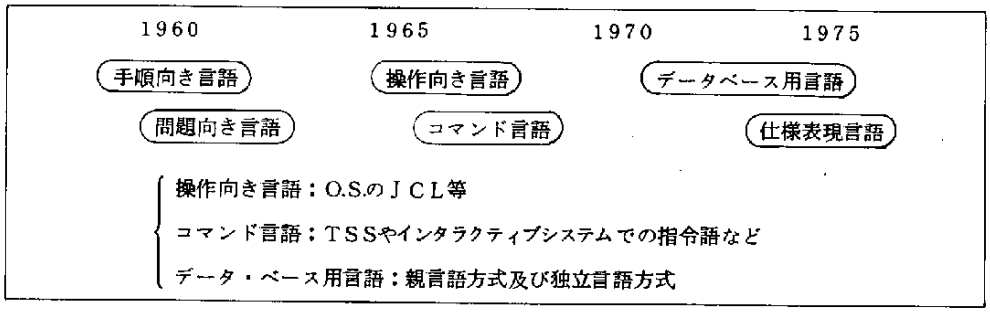


図３－１　人工言語の開発の焦点の移動

プログラム言語，特に手順向言語それ自身の本質の研究は，プログラム開発のための前提として重要な学習項目であるとともに，ソフトウェア技術の発展の方向を認識するためにも有効な学習上の足場である。言語をプログラムを書くための単なる道具であると軽視してはならない。言葉は思想でもある。腕のいい職人は道具を大切にする。プログラム開発技法の最近の話題も，プログラミング言語の選択と密接な関係をもっている。

(2) プログラムは試作品であり且つ実用品である。工業製品の場合は，試作品がそのまま商品として市場に流れる例は少ない。造船やプラント建設の例のように，注文があってその都度設計して製作するものの場合でも，設計段階で数個の代替案に対する試行錯誤がおこなわれ，比較評価の結果製品が作られる。プログラム開発の現状では，プログラム設計に対して，その設計方針のよしあしを論議する習慣が少ない。多くはプログラマの好みによって技術的決定がおこなわれる。システム設計の段階では，或程度の代替案の比較検討が

おこなわれているにもかかわらず，最も具体的で定量的比較も容易なはずのプログラム設計の中味は，それにもかかわらず比較評価の外光の下にさらされることが少ない。技能と呼ばれる所以の一つがここにある。現在のプログラム開発は，次のような経過でおこなわれることが多い。

（i）システム設計仕様書が渡されると，プログラマは，流れ図を作成する。

（ii）モジュール分割が必要であると聞かされているので，プログラマは前記の流れ図を大体の流れに沿って適当に分割する。

(iii) 分割ができるとただちに夫々のモジュールのコーディングとデバッグにとりかかる。

（iv）作業の進行中に設計変更が次々におこる。

（v）各モジュールは設計変更のためのつぎはぎで，予期以上にふくれあがる。

（vi）一応全体が完成し，実行に移るとまもなく，さまざまな理由でプログラム修正の要求がおこる。

（vii）プログラマは保守と修正に明け暮れる。

（viii）プログラムの修正をくり返すうちに，全体の関係がこみいって来て個々のモジュールの修正では対応できないようになり，修正作業をあきらめる。

（ix）利用者はこのプログラムを使わなくなり，そのプログラムは捨てられる。

（x）莫大な量のプログラムに関する文書はプログラム自身も含めて誰からもかえりみられない。

このような経過をたどる間，プログラマ以外の者にとっては，プログラムの中味がどうなっているのかは結局わからないままで，ただ現象としては，プログラマのほとんどの時間がエラーを直すことだけに費やされているという事実を眺めたということになる。

……このようなプログラム開発の個人依存を脱却して，技術としての安定を確保するために考えるべきことは何か？

既に学習した「基本設計」と「ソフトウェアの設計」では基礎事項の整理に重点があった。そのためここで指摘したような具体的問題点については答えてはいない。観念的に「標準化」とか「文書化」という念仏で解決できる問題ではない。プログラムの設計過程を明確に形式化してみて，一つ一つ具体的に検討すべき点を列記してみることから問題点を整理してみる必要がある。

- よいプログラムとは何か？
- プログラムの文書化はどこに着目すべきか？
- プログラム技術を蓄積するにはどうしたらよいか？
- プログラムの標準化はどこに着目すべきか？

(3) プログラムの作成は個人々々の孤立した思考作業にたよっているので，その生産性は低い。適用業務の範囲が拡大しているため，プログラム開発の生産性をあげる必要が急に高まっている。一人一人の生産性が低いため，グループによる共同作業の必要がおこっている。この事情は100年前の機械工業の発展過程と類似している。家内工業的生産から工場での集団作業への移行期と同じである。コンピュータのハードウェアの発達はソフトウェアのプログラミングの方法の変更を次々に要求し，結局，ハードウェアの発展と需要の伸展の両者の成長の谷間にはさまってソフトウェア技術は産業革命期の幼年労働者のようにみじめな要求過重にあえいでいる。期限も品質も確保できないまま不完全なプログラムを次々に生産しているのが現状である。

ソフトウェアの生産は，未だ工業生産ではないのにもかかわらず，工業生産らしい形式を採用しようとしているところに無理がある。工業生産の生産性をあげるための基本方式は，家内工業から現在の装置産業まで含めて，工程の分割による分業形態の採用であるということができる。プログラム開発とプログラム構成の基本事項として，プログラムのモジュール化が強調されていることの由来は，このような工業生産の考え方によるところが大きい。

プログラム開発技術の現状を把握する上で，このようにプログラムを工業生産物と考えることの出来る面と，ソフトウェアという特殊性から考えて，工業生産物とみなすことのできない面との両者を正しく認識してみることが大切である。

表３－１　ソフトウェア開発の性格

(1) プログラムの生産は言語を使っておこなわれる。
・人工言語の研究が基礎になる。
(2) プログラムは試作品で且つ実用品である。
・一個だけしか作らないという点では工業製品ではなくて芸術品である。
(3) プログラムの生産は個人の思考作業の生産性にのみたよっている。
・共同作業の場合でも産業革命前の工場制生産以上の生産性は望めない。
技術の伝承が難しいという点で，徒弟制生産と同じである。

3.2　モジュール化

モジュール化とは，一つのプログラムを数個の細分化されたプログラム群によって構成することをいう。一つのシステムを数個のサブシステムで構成する場合にも使われる概念である。モジュール構造の概念の由来，有効性を研究することは，ソフトウエアの設計と作成の基礎になる。

モジュール化の意味を研究するための足がかりとして，言語プロセッサでのモジュール化をふりかえってみる。

(1) FORTRANのサブルーチン

モジュール化の基本形はFORTRAN 文法の**サブルーチン**にあらわれている。サブルーチンという概念が必要になった理由は，次の2点であった。

① メモリ領域の節約

② 同一処理手順を何度も書く労力の節約

これらを実現するための手段として，FORTRAN のコンパイラでは，

ⓐ クローズド・サブルーチン

ⓑ オープン・サブルーチン

の形式で，夫々のモジュールの結合をおこなっている。前者では，メモリの節約に重点があり，リンケージのための時間は余分になる。後者では実行用のプログラムにその都度挿入されるので，メモリは多く必要とするが，標準的なルーチンが組込まれるので，プログラマの負担にはならず，実行時間も速くなる。

即ちソフトウェアのモジュール化とは，基本的に，メモリ領域の節約と計算時間（プログラム作成時間及びプログラム実行時間）の短縮という二つの制約項目に対する解決の工夫から生れた概念であって，さらに広義にとらえれば，工業化社会の基本技術でもある。

産業革命以降の機械化の歴史をふりかえってみると，20 世紀に入ってからその発展が急速に加速された一つの原因には，機械の**部品化**とそれをささえる**標準化**の普及があげられる。動力の伝達機構の中に組込まれている歯車，機械組立の要素のネジ，ボルト類の**共通性，互換性**の徹底が，工業化の基礎を作ったといっていいであろう。ソフトウェアのモジュール化の思想にも，同じような工業社会での部品化，標準化の考え方が含まれている。

サブプログラム単位にプログラム作成をおこない，これを**コンパイラ**によって，**オブジェクトモジュール**に翻訳し，次にそれをリンケージエディタによってリロケータブル形式に変換し**実行モジュール**として結合して使用するという現在の方法は，モジュール化を実現した一つの実行方法である。

アセンブラでは，FORTRAN のサブルーチンに該当するものは，マクロライブラリという呼び方で採用されている。これらは処理手続のモジュール化の思想である。

(2) COBOL の作業分担

COBOL の文法の中には，モジュール化のもう一つの視点がある。それは作業分担の考え方である。FORTRAN の場合と同様に工場生産の場での比喩でみてみると工場又は会社での機能別組織構成にたとえられる。COBOL でのプログラム作成には，見出し部(ide-

ntification division)，**環境部** (environment division)，**データ部** (data division)，**手続き部** (procedure division) にわけて書くという規約がある。

(3) ＯＳの作業分担

ソフトウェアのモジュール化をさらに進展させたものは，**オペレーティング・システム**の構成である。オペレーティング・システムの萌芽は同じくFORTRANのプロセッサの中に含まれていたとみることが出来る。即ちそのIOCS（入出力制御機構）の分離である。これはその後オペレーティング・システムの中の**タスク管理**に発展し，現在の形式の，タスク管理，ジョブ管理，データ管理の多重処理機能になったとみることが出来る。これはオペレーティング・システム自体のモジュール化でもある。

(4) ALGOLのブロック

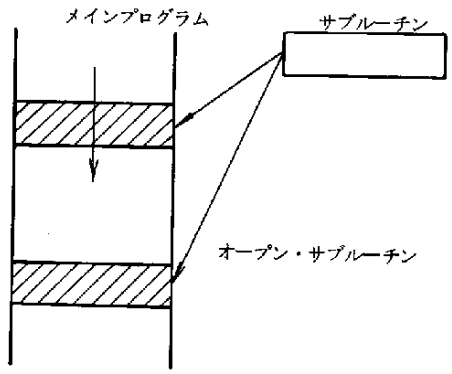
モジュールを一つの機能としてとらえる立場をさらに立体化した考え方が，ALGOL 文法の中に採り入れられている。

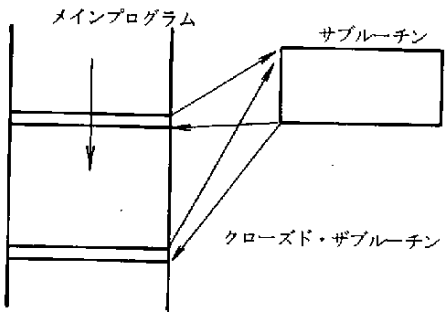
ALGOLのプログラムは**ブロック** (block) または**複合文** (compound statement) の形をしている。両者とも前後をbeginとendでくくったものになっている。少なくとも一つの宣言文を含むものがブロックで，宣言文を一つも含まないものが複合文である。これらは見かけ上はFORTRANのサブルーチンと類似しているが，本質的に異なる点は，FORTRANのサブルーチンが一個づつ独立であるのに対して，ALGOLの場合は，beginとendでくくられた一塊が一つのシステム・レベルを表現しているという点である。即ち構造的にはbeginとendの間には幾重にもまたbeginとendの組（ブロック，複合文）が挿入され，それらは次々と一つ下のレベルになってゆく。begin－endであらわされる同一のレベルのブロック同志は交叉することはない〔(begin begin end end) となることはない。この場合は，(begin begin end end)とみなされる。〕ALGOLの中に含まれるモジュールに対する考え方は，プログラムの構成に対する基本を示すものとして，その後の議論の出発点となった。Dijkstraの「GO TO文」不要論も，ALGOL流のネストされたブロックをプログラムの構造として前提にしている。構造化プログラミングの基本型はALGOLのブロック構造である。その後に展開されたプログラム開発技法のほとんどのものにこの考え方が共通項として含まれている。IBM社のIPTでは**階層化**と**モジュール化**とが組合せられてHIPOとなっている。システムに対する考察でも，例えばG. Nadlerはこれを Boxes within Boxesと即物的に表現している。ワークデザインのホッパー表現にも同じ考え方があらわれている。工業製品になぞらえてみると，FORTRANのサブルーチンが歯車や「はめあい」のレベルの互換性や共通性をねらったものであるのに対して，ALGOLのブロックは，機械システムの中の部分機能のパッケージ化，ユニット化に対応する。

(5) 自動制御系のブラック・ボックス

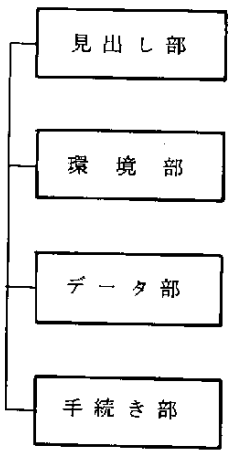
機能別にモジュールを分離独立させて，全体の構成を明確に把握しようとする思想が端的にあらわれているものは，自動制御理論の分野で使われているブロック線図である。ブロックで表現される一つの機能は，システムの外側からみた一個の作業分担であって，その内部の「からくり」は，システム全体を見る場合には，それを視野の外におこうとする考え方である。ブラック・ボックスという言葉であらわされるように，モジュール内部をシステム全体の構成よりも下位の次元のものとしてとりあつかう。システム全体を点と線の網として考えた時，その点にあたるものがモジュールである。システムを部分の集合体とみる管理活動や，部分機能を信号系で連結したものとみる制御系としてとらえる場合に有効な考え方である。経営活動のとらえ方としてフィードバック制御の比喩がとられるのもこの例である。モジュール化をこのような立場でとらえる時は，「モジュール」即ち「単一機能」ということになる。モジュールを単一機能に限定するという考え方は，システム全体を理論的に明快な構成として把握するためには，大切な着眼点である。

FORTRANの場合





COBOLの場合



データ部と
手続きの部
の分離

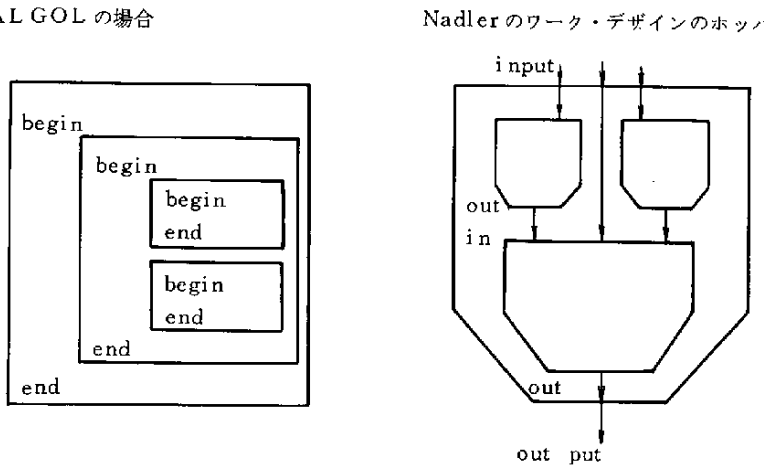


図３－１　論理的モジュール化の例

(6)　物理的モジュール化

今まで考察したソフトウェアのモジュール化の思想は，いわばすべてロジカル構造としてのモジュールの問題である。

モジュール化の考察で実際問題として大切なもう一つの側面は，フィジカル構造としてのモジュール化である。

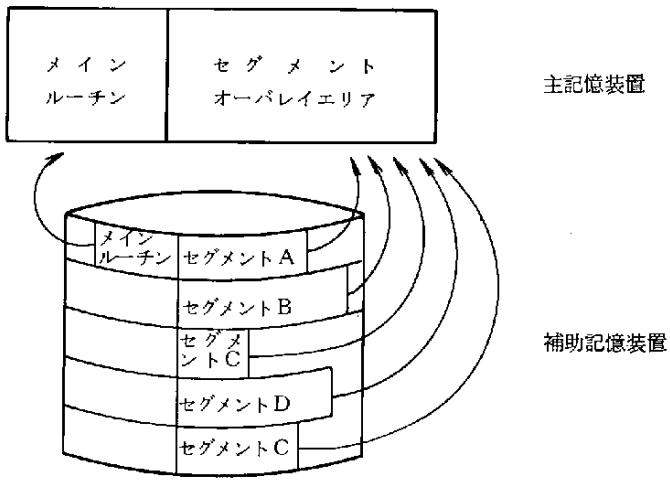
クローズド・サブルーチンを使用しても，メインメモリに収納しきれない程にプログラムが大型になってきたこと，ディスク装置等の外部補助記憶装置が発達して来たことの両者が，フィジカル構造としてのモジュール化を促進した。即ちセグメンテーション或いはページングの機能などである。セグメンテーションでは，メインメモリに各セグメントの交通整理をおこなうメイン・ルーチンのみを常駐させて，補助メモリにある各セグメントを順次ロードして計算を実行する。(オーバレイの技法)。この場合の各セグメントはフィジカルなモジュールである。前に述べたロジカルなサブルーチンの樹（親子関係の結合方式）と似かよった点はあるが，オーバレイの場合のメイン・ルーチンとロードされている各セグメントの連結の仕方は，本質的には異なるものであることをよく認識する必要がある。フィジカルなモジュール化のセグメンテーションは，オペレーティング・システム自身に多用されている。

オペレーティング・システムがV. S. に発展してきたのに伴って，さらに機械的に画一化されたモジュール化の技法が採用されるようになった。それはページング機能である。補助メモリからのローディングを能率的に実行するためには，モジュールそのもののサイズが一定になっている方が取扱いが容易になる。ページング機能とは一定単位の大きさにプログラムを分割して，プログラム実行時のオーバレイを行う方法である。プログラム作成の段階でも，フィジカル構造としてのモジュール化は，プログラムの実行効率の向上に対して最も重

要な要因となる。

データ管理でのデータ自身の階層的モジュール化は，同様に重要なテーマであるが，これについては「データ管理」の節であらためて検討することとする。

セグメンテーション



ページング

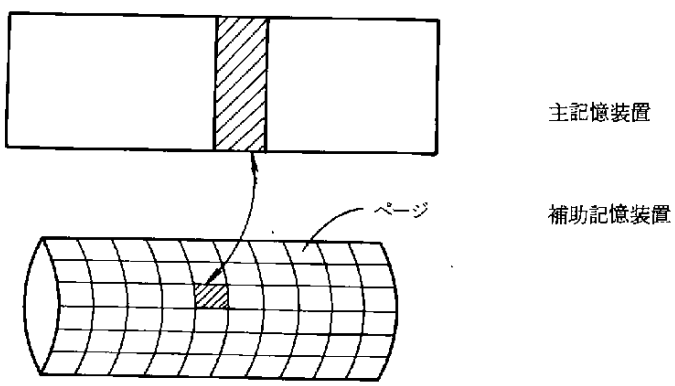


図３－２　物理的モジュール化の例

以上は主として理論的な観点の整理であるが，その他の問題として，実際的な観点からの検討も，モジュール化について考察を加えておく必要がある。例えば次のような項目である。

・プログラムの１モジュールとして適切な大きさ（ステップ数など）

・階層的構成での１レベルの中に於ける適切なサブプログラムの数

・適切なレベルの数

・モジュールとしての「まとまり」の判定基準

ソフトウェアの設計の基本的手法としてのモジュール化に関する考察をまとめると，下に示す表のようになる。モジュール化の思想を教育する方法は，この他にもシステム一般理論から演繹的に説明するやり方など，いろいろありうるが，以上のような説明が理解しやすいであろう。

表３－２　モジュール化の思想

・理論的モジュール化

(1) FORTRAN　サブルーチン

オープン・サブルーチン……時間の節約

クローズド・サブルーチン……メモリの節約

(2) 自動制御理論のブロック

ブロック線図……機能の明確化

(3) ALGOLのブロック

ブロックのネスト……階層構造の把握

(4) Boxes within Boxes

システムのレベル……レベル毎のモジュール化

・物理的モジュール化

(1) セグメンテーション……ローディング単位のプログラム

(2) ページング……記憶域の入換の単位

・実際的なモジュール化の考察

(1) １モジュールの適切なステップ数

(2) モジュール中のサブ・モジュール数……スパン・オブ・コントロール

(3) モジュールの「まとまり」の基準は何か？

3.3　プログラムの設計

プログラム設計には，経済的制約や物理的制約を特に考慮する必要がある。プログラムを内側からみる目（作る側の視点）と外側から見る目（使う側の視点）のバランスをとることが所謂エンジニアリングのセンスである。

（i）内側の視点

・単純さ（simplicity）

・作りやすさ（easiness）

・信頼性（reliability）

・保守しやすさ（maintenability）

（ii）外側の視点

・効率（efficiency）

・使いやすさ（usability）

・汎用性（generality）

・拡張性（expandability）

Meyears はその複合設計（Composite Design）で，プログラムの概念を分析して，プログラムの属性に三つの次元があるとしている。即ち，

(a) 機能（プログラムの実行すべきこと）……用途

(b) 性能（プログラムの作動状態）……大きさ，速度

(c) 構造（プログラムの構成法）……モジュール，インターフェイス

さらにプログラム設計を，モジュール設計としてみた時の着眼点を，モジュール強度（module strength）と結合度（module coupling）の二つの尺度として，分析している。即ち，

(i) モジュール強度
a) 機能的（functional）
b) 情報的（informational）
c) 連絡的（communicational）
d) 手順的（prodedual）
e) 時間的（classical）
f) 論理的（logical）
g) 暗合的（coincidental）

(ii) モジュールの結合
a) データ結合（data coupling）
b) スタンプ結合（stamp coupling）
c) 制御結合（control coupling）
d) 外部結合（external coupling）
e) 共通結合（common coupling）
f) 内容結合（content coupling）

モジュール強度は強いものが望ましく，結合度は弱いものが望ましい。上にあげた例はいずれも望ましいものが上の方にかかげてある。モジュール化設計には，このような設計上の原則を頭に入れておくことも重要である。

さらに具体的設計作業に対する目安として，各種の標準的な尺度で検討することがある。例えば次のようなものである。

a) モジュールの大きさ及びその数

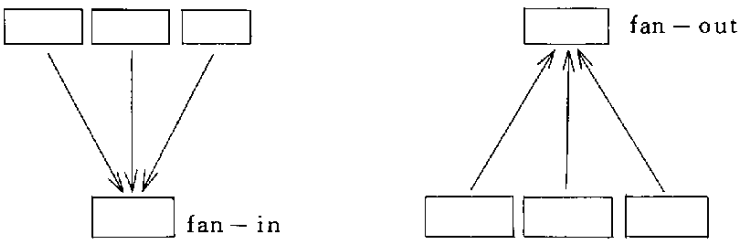
b）モジュール間の結合子の数（結合の強度）

c）モジュール内の結合及びその完全性（integrity）

さらに一つのモジュールに着目してその結合の様式と度合をあらわす尺度として次の項目があげられる。

d）流入数（fan－in）……他のモジュールから利用される数

e）流出数（fan－out）……他のモジュールを利用する数



モジュールの大きさについてさらに詳しく検討すると，プログラムの分割をすすめてモジュールの数を多くした場合，夫々のモジュールの大きさは小さくなり，個別モジュールは単純になるためその開発費は少なくなるが，反面結合の度合は複雑になり結合のための開発費が多くなる。全体の開発費は個別モジュールの開発費の和と結合のための費用の総和になる。この関係を図に示すと次のようになる。

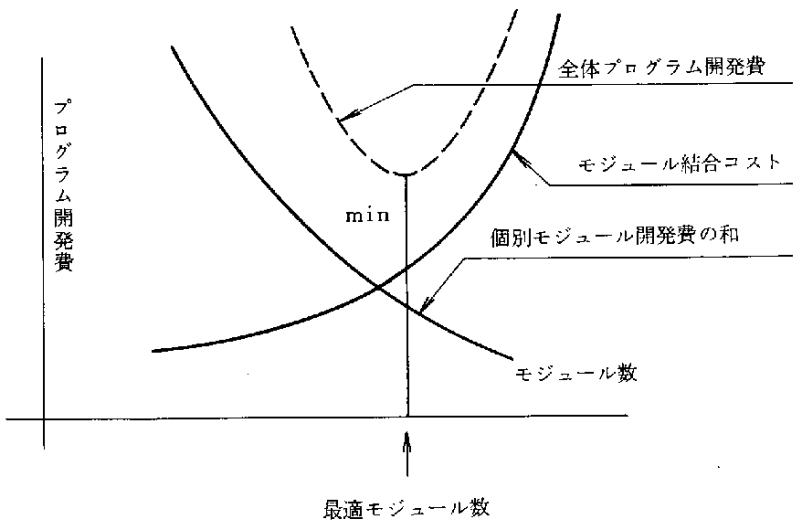


図３－３　モジュール設計での最適モジュール数

モジュール間の階層図（hierarchy chart）を考えると，その複雑さをしめす尺度に次の指標がある。

f）深さ（depth）……トップからボトムまでのレベル差

g) 幅（width）……あるレベルでの横の数

h) スパン（span） ……モジュール同志の道のり

（下図で A と C のスパンは4）

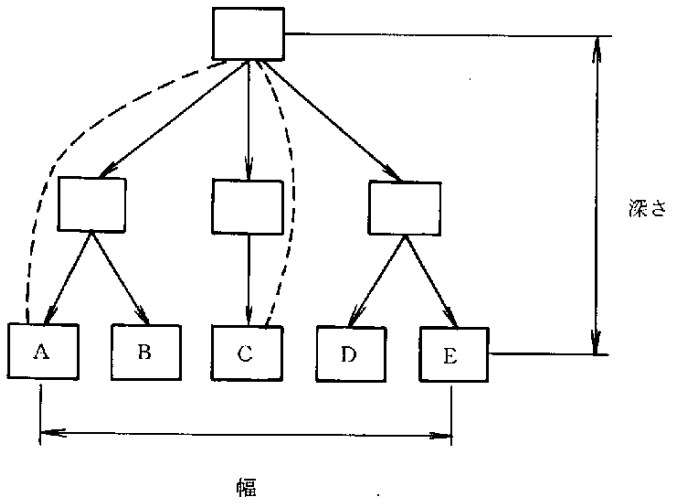


図3－4　階層図の尺度

i) かたより（skew）

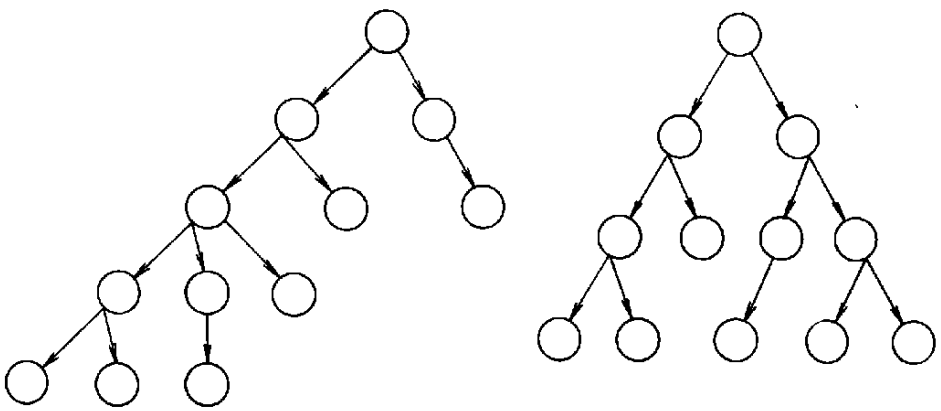


図3－5　かたよりの例

3.4　サブシステムのインターフェイス

機能の分化をおこなってこれをサブシステムに割りあてる仕事は，いわば分析，解析の作業である。このほかにシステム全体を作りあげる総合，統合の仕事がある。これが本来のシステム設計である。システム設計は分析と総合の調和の仕事である。ここでははじめに検討したモ

ジュール化の考え方をシステム設計の統合化の作業といかに調和させて行くかを考察する。それには次のような着眼点が必要になる。

(1) システムとサブシステムの関係

サブシステムはシステムに対する機能的な分担である。階層的な構成をもつシステムの場合は，中間のサブシステムのレベル毎の目的を明確にする必要がある。目的に適合するように組みあげてゆくことが統合化の中心であるが，共用的なサブシステムを使う場合などは，上位レベルのシステムの目的が見失われることがある。システムの統合化の考察の際には，不要機能が混入していないかというチェックポイントで調べてみることが大切である。

(2) サブシステムの連結

モジュールは出来るだけ独立であることが望ましい。このことは具体的に言えば，モジュールの連結子即ちインプットとアウトプットが夫々単純であることを意味する。モジュールが複合的な機能を持つ時はそれにつれて連結子も複数になる。サブシステム間を結びつける糸の数が増えれば増えるほどお互同志はからみあって，結局はモジュールの独立性が失われる。モジュールの内容を一部修正した場合にも，そのモジュールの連結している他のモジュールに，思いがけない波及効果をあたえることになる（その修正が最初に意図した改良点以外に予期しない影響をおよぼすことになる)。このために全体システムの改良が出来なくなる。システム全体をモジュールの組合せで構成することの利点は，夫々独立性の高いモジュールで組み立てられている場合に，モジュール毎の改良が実行出来て，結果的には全体システムの進化が可能であるという点である。

このためにもサブシステムのインターフェイスの設計には特別な注意をはらう必要がある。ただ漫然と部分に分割するというだけでは，このようなねらいは達成できない。

(3) 構造化プログラミング（structured programming）の考え方

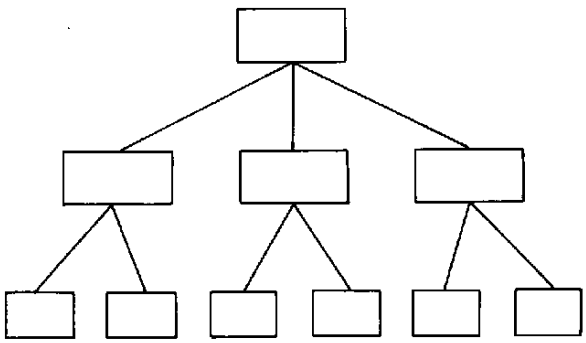
構造化プログラムでのアプローチが，トップダウン的な方法を採用しているのは，今まで述べたような統合化のねらいを明確にすることを第一義と考えたためである。部分から設計をはじめるのではなく，はじめにシステム全体の目的をあきらかにしておいて，順次部分機能に分解して行くという過程の方が，個々のモジュールも単一機能になり独立性も保てる。したがってシステムのメンテナンスも確実なものとなる。文書もシステム全体が出来あがってから書きはじめるのではなく，全体から部分に，システムの開発に平行して（むしろ開発に先行してといった方が適切）作成される。

構造化プログラムにおけるサブシステムのインターフェイスを具体的に学習することが望ましい。

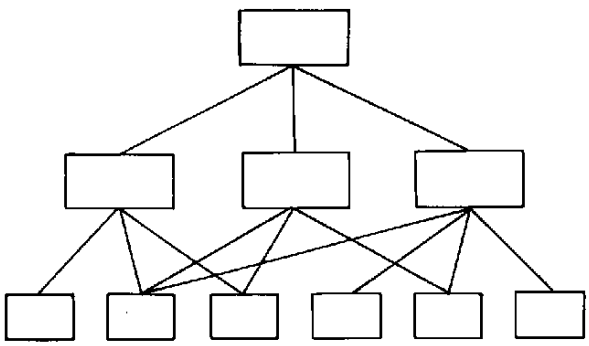
(4) サブシステム間の関係

一般にシステムの構成を図表化すると次のようなタイプがある。

(A) 関係が単純なもの

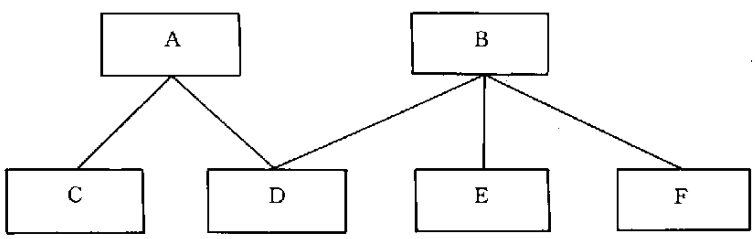


(B) 関係が複雑なもの

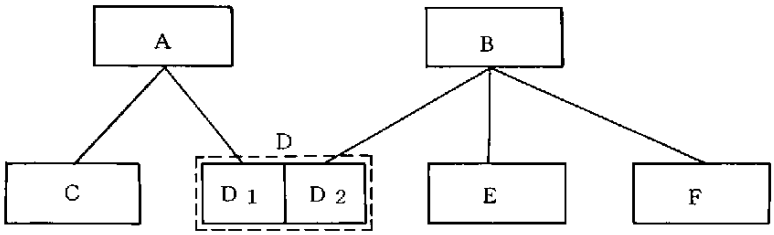


この場合注意すべきことは，汎用サブシステムと複合サブシステムは異なるということである。

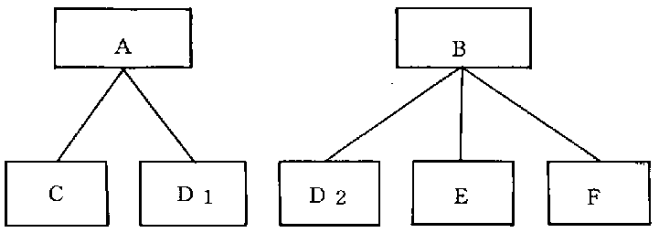
下図のような場合，Dサブシステムは汎用であるために，AとBとで共用されていることになるが，



内容的には，時として



のように便宜的にＤ１とＤ２の複合であることが多い。このような場合には，



と単純なインターフェイスにすることが望ましい。共用サブシステムの設計で，安易な寄せ集め機能で汎用性をもたせたものが多いが，これはいたずらにシステム全体の構成を複雑にする原因となる点に注意すべきである。

3.5　ソフトウェアの標準化

一般に工業製品の発達は次のような発展過程をたどる。

(i)　アイデアの段階……製品としては未熟で実用的には不成功に終るものが多い。

(ii)　試作の段階……試作的な商品が市販され，ユーザの批判や他のメーカとの競争で改良が次々に加えられる。

(iii)　大量生産期……技術的に安定し，小改良がつみかさねられる。この段階で標準化がおこなわれ，互換性のある製品の体系を形成する。

ソフトウエアの技術の場合にも，上記の段階を経過するとみなすことが出来る。但しやや異なるところは，工業製品のように同じようなものが次々に生産されることが少ないという点である。長時間の使用経験を経て基礎的，共通的なものが標準として残るということは同じである。

データ処理の手続について標準化という視点から学習するには，現在標準化されているものが，歴史的に蓄積をつみかさねてきた経過をふりかえってみて，その中にある本質的な技術を把握するという態度と，標準化されたルーチンを有効に活用して，今から作成するプログラムの開発を能率的におこなうという態度の，両面からのアプローチが必要である。技術とは先人の蓄積を踏み台にして高みに登る方法である。特にソフトウェア技術は，未だ若い技術であるので，ブラックボックス化した標準的な手続に対しても，不断の検討が必要である。

(1) 事務処理分野の標準的手続き

事務処理分野でのデータ処理は，各種事象のコード化，データの分類などが作業の主流となる。次のような学習項目がある。

a) コード体系の標準

b) 分類とまぜ合せの基本的な手法

c) 統計計算の標準形

(2) 技術計算分野の標準的手続き

技術分野は問題が広範囲で多岐にわたるので，全体を学習することは難しい。夫々の専門分野で固有な形の標準化がありうる。一例としてあげれば次のような項目がある。

a) 数値計算，関数近似などの基礎ルーチン

b) 有限要素法などのような特定応用分野の問題処理のための道具としてのソフトウェア

c) 作画ルーチン，インタラクティブ・グラフィックスなどの手続きとして頻度の高いプログラム

(3) プログラム言語の標準化

手続向言語の標準化の動向について，代表的なもので，FORTRAN，COBOL，ALGOL，PL／1などの標準化について断えず関心をはらう必要がある。さらに最近では，BASIC，APLなどのインタラクティブな言語の重要性も増加しつつある。応用分野についてはAPT，EXAPT等のNC言語，GPSS，DYNAMO，SIMSCRIPT等のシミュレーション。データ・ベースのDML，DDL等も重要である。ISO／TC97，JISでの標準化活動，ANSI，CODASYL，CCITTの活動などはプログラム用言語の標準化に密接に関係するものが多い。これらは，データ処理の標準化の基礎部分でもある。言語については1960年代の後半に問題向言語の流行があり，その後その反動としての沈静化を経て特殊分野（プロセス制御のPROSPRO，数式処理のFORMAC，リスト処理のLISP，論理処理のPASCAL）で標準的なものが安定的な評価を得ている。これらについても必要があれば研究することが望ましい。

(4) 共用サブルーチンの整備

ソフトウェアの生産性を高める最も有効な方法は，既に開発したプログラムを使うことである。このためには何回も使えるような共用サブルーチンを整備しておくことがまず必要になる。ソフトウェアの標準化の中心主題は，このようなサブルーチン，あるいはサブシステムの検討でなければならない。しかしながらこの問題は技術的に困難な問題点を含んでいる。

システム開発は目的志向で，トップダウン型の方法が望ましいとされている。これに対して，その中に使われるサブシステムを，共用の標準的なものにしようとする方向は，本来矛

盾を含んでいる考え方である。その相反する二つの方向を，どの点で妥協させるかが，共用サブルーチンの整備の中心課題になる。

このような考慮を念頭においたシステム設計の手順は次のようになる。

(i) システム全体の機能を明確にする。

(ii) それをサブシステムに分割して，そのサブシステムの機能を設定する。

(iii) このサブシステムには既存の共用サブシステムをできるだけ多く使う。（共用サブシステムには今までに検討した標準的なものを使ってもよいし，目的にあわせて新たに開発したものでもよい。）

(iv) 上記のサブシステム同志は，お互いに独立なものであることが望ましい。

(v) 共用サブシステムは，共用という点を重視しすぎて，汎用型の手足の多いものにするとかえって使いにくい中途半端なものになるおそれがあるので特に注意が必要である。

ここに述べたようなサブシステムの標準化を蓄積することが，ソフトウエア技術を前進させる唯一の方法である。

3.6 ユーザーとのインターフェイス

システムは業務処理を目的とする。前節までの学習はそのための手段としての道具の検討であった。この節で検討する問題は，その処理手順である。その中心をなすものはプロセス・フローチャートである。特にソフトウエアとして見た場合には，業務におけるデータの流れをパターン化して把握することが必要になる。

業務処理のシステム化を推進するための手がかりは，まず対象事務そのものの大網を全体として把握することである。そのための有効な方法として，「システムの図表化」がある。すなわちプロセス・フローチャートである。業務処理のフロチャートの作成には，プログラムのフローチャートとは異なる視点が必要である。

次のような諸点を学習する。これらの技法については，基本的な分析手法，基本的な設計手法などで学習ずみであるが，ここではこれらの手法を現実問題に適用する時の問題点を復習する。

(1) 業務処理のプロセス・フローの把握

使用される手法には，既に学んだ事務分析フローチャート，ブロック・ダイヤグラムなどがある。

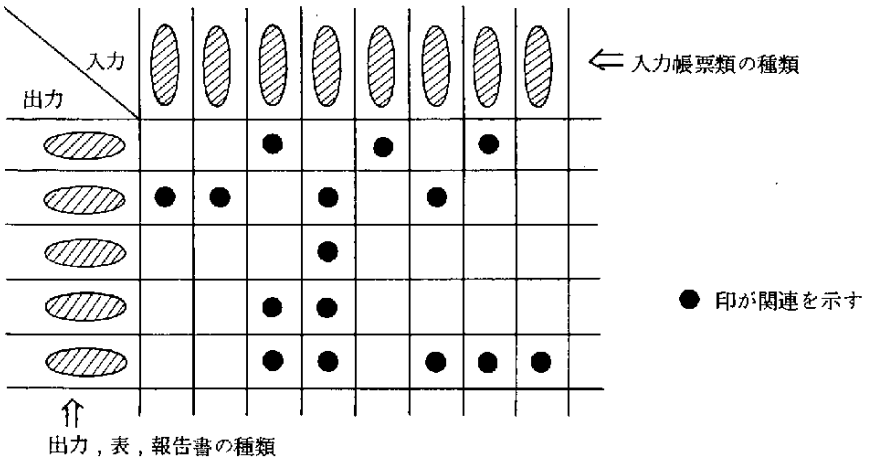
(2) 入出力作業の把握

データの形式，内容，量，時期などを整理するための標準的な書式が工夫されている。

(3) システム要素間の関連の把握

例えば次に示す例のようなものがある。

(i) インプット－アウトプット関連図



この形式の表現で大規模なものに産業連関分析がある。

(ii) ディシジョン・テーブル

処理手順を流れの形で表示するフローチャートよりも，条件と結果の組合せを表の形式であらわした方が，条件の欠落などが一目でわかる上に，全体に対する一覧性が明確である。その部分を一種のブラックボックスとしてとりあつかうまとまりもある。

ディシジョン・テーブルの基本形は次のようにあらわされる。

Condition Stub	Condition Entry
Action Stub	Action Entry

(4) 処理手順の時系列的把握

関連事項の把握には，以上述べたように，マトリックス表現が便利であるが，これは現象を静止断面でとらえようとする方法で，処理手順の動的把握には不足する面がある。フローチャートやブロック・ダイヤグラムで動的把握が可能な範囲は，処理手続が一次元に逐次的手順に従っている時であって，ランダムに生起する事象を対象に考える制御系などのシステムの場合には，時系列上での割込状況の動的把握は困難なことが多い。この場合のシステムの図表化には，グラフ理論などで使用されるグラフ図，状態遷移図などが便利である。生起事象のすべてを行に並べ，時系列上での割込状況を模式化した割込線表などが使われることもある。これをタイミング・チャートと呼ぶ。

(5) プログラムの流れ図

事務計算を主流とする業務処理プログラムの場合の実際的な標準手順は次のようになって

いる。

① コード設計
② アウトプット設計
③ インプット設計
④ ファイル設計
⑤ チェック方式設計
⑥ エラー修正方式設計
⑦ ドキュメンテーション
⑧ フローチャート設計
⑨ プログラム設計書

流れ図を中心とする伝統的なプログラム設計の手順である。この方法は制御的構造（operational hierarchy）を中心とする設計手順である。これに対して最近になって議論が集中しつつあるものに，機能的構造（functional hierarchy）を中心にする設計手順もある。

(6) 機能の階層図

流れ図を中心とする従来のプログラム設計の手順と異なる方式が，新しい開発技法で多く提案されている。その手順の要点は次のようになる。

① モジュールの機能を確認し，その入力と出力とをはっきりさせる。
② 上記機能を分割してサブ・タスクにわける
③ 機能の分解図（factoring diagram）を画き，全体と部分のタスク間の関係を把握する。（ＫＪ法での「島づくり」と同じ方法）
④ 機能階層図（hierarchy chart）を画き，判断と処理のレベルをあきらかにする。
⑤ 機能的構造に基づいて作られたモジュールについて，モジュールの大きさや結合の数などの形式上の尺度で再検討してみる。データ構造のレベルとの対応も検討して，経済的，物理的制約との調整をはかる。（functional model とphysical model の間でのtrade －off……調整）

ここにあげた手順は機能的構造（functional hierarchy）を中心にしたプログラム設計の手順である。新しい開発技法のほとんどはこの方式になっている。即ちまず機能中心にモジュールに分解する。この際，機能をはっきりさせるために「機能を文章で書いてみる」という点に注目する必要がある。（ナドラーのワークデザイン，川喜田のＫＪ法など，機能把握に文章表現を足がかりにする例が多い。……発見，探索のプロセスに共通する手段として興味深い。）構造化プログラミングのトップダウン方式で疑似モジュールを使う方法も，文章を使うという点では共通点がある。入力→変換→出力という処理は機能の一部分ではあるが，

機能そのものではない。ここまではロジカルに分解されたモジュールであるが，次の段階でこのモジュールをフィジカルな面からみて（大きさや数について）調整する。

(7) 独立タスクの集合

制御的構造を中心とするプログラム設計の手順のもう一つの方法として，プログラムの処理の流れを状態遷移としてとらえて，これを有向グラフで表現してみる方法がある。チューリング・マシンの原点に帰る考え方でもある。この方法は特に最近，再たび関心を増しつつある視点である。O.S.のシステムプログラムのように，お互いに独立なタスクの集合であるプログラムや，事象の生起が確率的に発生するオンライン・リアルタイム処理のプログラムや，コンピュータ・グラフィックスやプロセスコントロールのプログラムのように割込処理によって，全体の処理の流れが支配される場合などは，流れ図による把握は無力である。このような場合には状態遷移図のようなグラフ理論的な把握や表形式の言語の表現が有力になる。

(8) 流れ図と機能モジュール

流れ図は，プログラムの各部分の関係をあらわす一番自然な方法である。コーディングに直接反映できる形でもあるので，機能的構造を中心にするプログラム設計の場合にでも，その部分として有効に使われる。他人の書いたプログラムや自分の作ったプログラムでさえも時間をおいて見直す時には，流れ図はプログラムの論理を図示する最良の方法である。プログラム設計の学習に流れ図を中心におくことは基本として間違いのないところである。よいプログラムは流れ図の形も美しくなっているものである。

制御的構造は実行順序の問題であるので，機能的構造とかならずしも調和するとはかぎらない。制御的視点からはすべてのモジュールをリエントラントな形にすることが望ましいが，これには全体のプログラムのオーバヘッド（over head）の増加が必要になり，且つプログラムの統合（integration）もむずかしくなる。将来のプログラム設計の方向は，データ・ベースの分離独立とともに，適当な機能のプログラムモジュールを任意に結合して実行する形態（プログラムベース）も考えられる。

これにはモジュール機能の標準化の整理が必要になる。

3.7 プログラム技法集の整備

ソフトウェアの設計の基本的な技術資料として，既に開発されたプログラミング技法の整理はきわめて大切な知識である。それらは，技法として整理することが望ましい場合とサブルーチンとしての標準形を用意してこれをブラックボックスとして使用する場合とがあるが，いずれの場合もプログラム設計を効率的に実行するためには，これらの基礎的な技術に精通してい

ることが前提になる。プログラム設計の基本はこれらの知識を多年の経験と学習によって蓄積することである。

このような過去の技術的知識の集積の上に乗っておこなう設計方法を上昇型（bottom-up）のアプローチという。最近の開発技法の中心になっている下降型（top-down）のアプローチは問題そのものに直結するという点では望ましいが，工業技術の大局的基盤は「失敗」という経験を踏まえたボトムアップ型であることを認識して，数多くのプログラミング技法を一つずつ確実に消化することを心掛けなければならない。

これらのアルゴリズム集も，標準的手続きと同じように，言語別の集積が必要になるのが現実の姿である。なお事務計算の分野では，プログラム技法の整備のほかに，コード設計がきわめて重要な位置をしめているので，特に学習する必要がある。

事務処理業務を能率化するために，各種のコードが使用されている。コード化による標準化の考え方は PCS 処理の頃からの歴史に立脚したもので，コードの種類の各種のものが存在している。これらについて一通りの知識を整理する必要がある。但し注意すべき点はコード設計にとらわれすぎるとシステムが硬直化する危険があるという点である。

コード設計については次の諸点を学習する。

a) コード化の目的とコードの機能

コード化の目的は，データの体系化と処理システムの効率化であるが，コードのもっている問題点も多い。これらについてあらかじめ検討する。

コードの機能には分類の機能，標準化の機能，区分の機能などいろいろの面がある。これらを十分把握しておく。

b) コードの種類

順番コード（Seguence Code）

区分コード（Block Code）

10進コード（Decimal）

桁別コード（Group Classification Code）

数字表意コード（Significant Digit Code）

文字表意コード（Mnemonic Code）

c) コードの管理

コードブック，コーディング，ディコーディング，コードのチェック

d) コード設定の手順

調査，整理分類

e) コードの標準化

指導上の留意点

ソフトウエアの設計での学習項目は，実際上の問題であるので詳細に入り込めば際限なく深入りする事が考えられる。学習知識としては必要最小限度にとどめ，むしろ既に学んだ個別の知識をどのように活用して具体的なソフトウエアをまとめあげるかという考え方の方に重点をおく方がよい。それには，各種の標準的な手続がなぜ使われるようになったのか，ソフトウエアの技法上の特徴やねらいは何なのか等の考察に時間をかけた方がよい。ソフトウエアの設計の実習はＤ３の単元でおこなうので，実務的な注意事項は除いてもよい。データ管理に関連して，データ構造の研究が必要になるが，これについては次の章でまとめて学習するものとしてここでは一応除外した。

次にあげるような事項を重点的に教育する。

(1) モジュール設計の基礎

分離（モジュール強度）と結合（モジュール結合度）

モジュール設計の目安（深さ，幅，スパン等）

(2) プログラム設計の手順

機能的構造（階層図）

制御的構造（流れ図）

(3) 標準的技法の集積－標準化

言語別標準技法集（COBOL，FORTRAN）

アルゴリズムの例（分類，数値計算）

コード設計

参　　考

(1) コンピュータ用言語の歴史

1954　FORTRAN

1956　APT (Ross)

1958　ALGOL

1959　DYNAMO (MIT) フォレスターのI. D.による

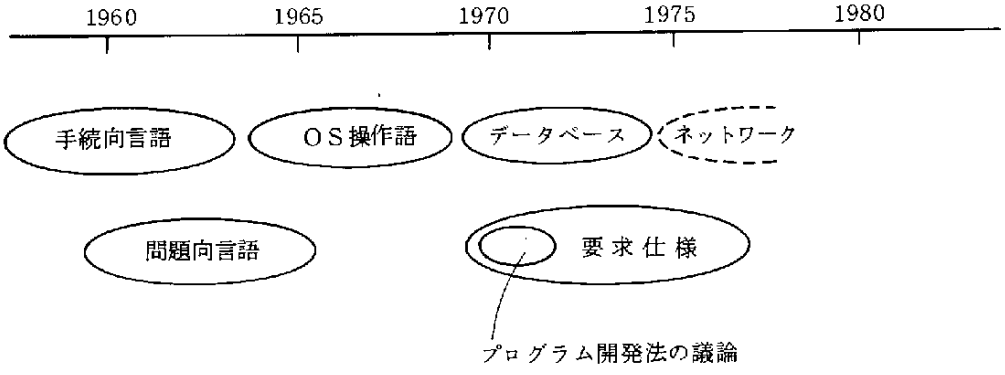
1961　COBOL (CODASYL)

1961　GPSS (Gordon) IBM

1962　APL (Iverson)

1962	LISP 1.5 (Mc Carthy)
1963	SIMSCRIPT (Markowitz) RAND
1964	BASIC (Dartomouth 大学)
1965	COBOL 仕様追加 (disk 関係)
1966	IBM 360 用 O.S.
1966	GO TO文有害 (Dijkstra), 2−formation rules (Böhm)
1968 ~ 69	AS1−ST, MARKⅣ, GIS, IMS等
1969	CODASYLのSCによるDBMS調査
1969	DBTGの提案
1969	structured programming (Dijkstra)
1970	IBMのGUIDE／SHAREの要望
1971	DDLC (データ記述言語委員会)
1971	chief programmer teams (Mills, Baker) IBM
1972	HIPO, Walktru (S. S. D.) IBM
1973	IPT (IBM)
1974	PASCAL (Wirth)
1975	composite design (Myers) IBMのS. R. I.
1976	SADT (Bracket) Softech 社
1976	PSL／PSA (Teichroew) Michigan 大学

全体的傾向



参考文献

[1] Allen, F. E. and Cocke, J.,"A Program Data Flow Analysis Procedure", CACM, 1976

[2] Wirth, N.,"On the Compsition of Well-structured Programs",Compuling Surveys, 1974

[3] Wirth, N.,"Program Development by Step-wise Refinement", CACM, 1971

[4] Böhm, C., and Jacopini, G.,"Flow Diagrams, Turing Machines and Languages with only Two Formation Rules", CACM Vol 9, 1966

[5] Dijkstra, E. W.;"GO TO Statement Considered Harmful",CACM, Vol.11, No. 13, 1968

[6] Dijkstra, E. W.,"Notes on Structured Programming,"T. H. Report 70-WSK-03, 1970

[7] Mills, H.; "Chief Programmer Teams Principle and Procedures", IBM, FSD June 1971

[8] Baker, F. T.; "Chief Programmer Team Management of Production Programming",IBM Sys, Journal, Vol.11,No.1, 1972

[9] Wirth, N;"On the Design of Programming Languages", Proc.IFIP 1974

[10] Wirth, N;"An Assessment of the Programming Language PASCAL", ICRS, 1975

[11] Teichroew, et al; "PSL/PSA-A computer-aided technique for structured documentation and analysis information processing systems",Proc. 2nd Int. Conf.on Software Eng., 1976

[12] Ross, D. T. and Bracket, J. W.;"An approach to structured analysis, Computer Decision", Vol.8, No. 9, 1976

[13] Irvine, C. A. and Bracket, J.W.; "Automatied Software Engineering Through Structured Data Management", Proc. 2nd Int.Conf. on Software Eng., 1976

[14] Dijkstra, E. W.,"A Discipline of Programming",Prentice-Hall, 1976

[15] Gilb, T. and Weinberg, G. M.,"Humanized Input ; Techniques for Reliable Keyed Input", Winthrop, 1977

〔16〕 Jackson, M. A.,"Principles of Program Design", Academic, 1975

〔17〕 Kernighan, B. W. and Plauger, P. J.,"Software Tools", Addison-Wesley, 1976

〔18〕 Martin, J.,"Design of Man-Computer Dialogues", Prentice-Hall, 1973

〔19〕 Myers, G. J.,"Reliable Software through Composite Design", Mason／Charter, 1975

〔20〕 Stevens, W.P., Myers, G.J., and Constantine L.L.,"Structured design", IBM systems journal, Vol. 13, No. 2, 1974

〔21〕 Myers, G. J.,"Software Reliability; Principles and Practices", Wiley, 1976

〔22〕 Thierauf, R. J.,"Systems Analysis and Design of Real-time Management Information Systems", Prentice-Hall, 1975

〔23〕 Yeh, R. T. (Ed.),"Current Trends in Programming Methodology", Prentice-Hall, 1977

〔24〕 Yourdon, E. and Constantine, L. L.,"Structured Design", Yourdon, 1975

〔25〕 国井利泰監修「ソフトウェア工学」共立出版, 1978

〔26〕 G. F. ウェンワルム著（吉原賢治訳）「ソフトウェア生産技術」学習研究社, 1976

〔27〕 半沢孝雄「プログラミング技法集」竹内書店, 1974

〔28〕 渡辺茂編「プログラム例題集」共立出版, 1977

〔29〕 J. T ゴールデン著（戸田英雄訳）「FORTRANプログラム技術」培風館, 1974

〔30〕 ハーバート・D・リーズ, ジェラルド・M・ワインバーグ著（藤井純, 山門四十夫訳）「電子計算機プログラミングの基礎」産業図書, 1972

〔31〕 ブライアン・W・カーニハン, P.J. プローガー著（木村泉訳）「プログラム書法」培風館, 1976

〔32〕 bit 臨時増刊「プログラミング言語」共立出版, 1974

〔33〕 F. W. ダイクストラ, 他著（野下浩平, 他訳）「構造化プログラミング」サイエンス社, 1975

〔34〕 栗生晴夫「プログラムのモジュール化のテクニック」ソフトウェア技術, Vol. 2, No. 11, 1970

〔35〕 N. Wirth著（野下浩平, 他訳）「系統的プログラミング入門」近代科学社, 1975

〔36〕 Myers著（久保未沙, 他訳）「高信頼性ソフトウェア」近代科学社, 1976

〔37〕 G. J. Myers著（有沢誠訳）「ソフトウェアの信頼性」近代科学社, 1977

〔38〕 J. D. ワーニエ著（鈴木君子訳）「ワーニエ・プログラミング法則集」近代科学社,

1975

〔39〕 久保末沙，国友義久「ストラクチャード・プログラミングの周辺」bit Vol.8, No.6～Vol.9, No.1 共立出版，1979

〔40〕 宮本勲，東谷秀夫「ソフトウェア・エンジニアリングとは」bit Vol.9,No.4～Vol.10,No.4，共立出版，1977

〔41〕 松本憲治「フローチャート入門」オーム社，1971

〔42〕 宮内儀三郎「フローチャート概論」共立出版，1970

〔43〕 横井満，西村恕彦「流れ図の書き方」企画センター，1970

〔44〕 筧捷彦「ストラクチャード・プログラミング用言語」情報処理，Vol.16，No.10，1975

〔45〕 bit 臨時増刊「アプリケーション・プログラム」共立出版，1977年7月

〔46〕 ジェームズ・マーチン著（北原安定訳）「リアルタイム・プログラミング」日本経営出版，1967

〔47〕 竹下享「ドキュメンテーションの標準化」日本経営出版，1974

〔48〕 海宝顕「データ処理の標準化」産業図書，1971

〔49〕 情報処理学会「電子計算機の国際標準」情報処理学会，1971

〔50〕 サンダーズ編（松浦四郎訳）「標準化の目的と原理」日本規格協会，1974

〔51〕 日本電子工業振興会「ソフトウェア信頼性に関する調査」 1977

〔52〕 佐貫亦男「発想のモザイク」中央公論社，1972

〔53〕 G・ポリア著（柿内賢信訳）「いかにして問題をとくか」丸善，1954

第4章　データベース

キーワード

データベース (data base)，データベース管理システム：DBMS (data base management system)，データ管理者 (data administrator)，データ独立 (data independence)，データの機密 (privacy of data)，データの保護 (protection of data)，DB／DC (data base／data communication)，情報構造 (information structure)，論理データ構造(logical data structure)，物理データ構造(physical data structure)，記憶構造 (storage structure)，データ記述言語：DDL (data description language)，データ操作言語：DML (data manipulation language)，固有データ構造記述 (schema)，個別データ構造記述 (subschema)，データサブランゲージ (data sub-language)，CODASYL (the conference on data systems language)，リレーショナル・モデル (relationl model)，エンティティ (entity)，属性 (attribute)，マッピング (mapping)

目　　標

本章でとりあつかうテーマは，データに焦点をあてたソフトウェア設計である。第3章がプログラム設計であるのと対をなす形で，データ設計を中心に学習する。

特にファイル設計の基本事項を体系的に把握する。業務側からみた立場として，入出力仕様を具体的に決定すること，コンピュータ側からみた立場として，データ管理の方式を設定することの二面から整理する。

データ・ベース概念の成立の由来を概観し，ソフトウェア設計の中心テーマが，データ・ベース・システムの設計であることを把握する。データ・ベース・システムの構築には，必然的に，とりあつかう業務範囲の全情報を十分に分析して構造をもった体系に整理することが必要になる。したがってデータ構造の学習に重点をおく。

データ・ベース・システムの学習には，抽象的議論は有効ではない。現実に稼動しているDBMSについて，それを利用したデータ・ベース・システムの設計をおこなってみて，DBMSの機能を体得してみることが望ましい。

しかし，一方ではデータ・ベース概念は，現在未完であるし，問題点も含んでいる。これらの将来方向に関する議論の一端にふれることによって，情報処理システムの本質を学習する。それには手続表現言語（FORTRAN，COBOL等）→操作表現言語（OSのJCL等）→ デ

ータ管理用言語（DBMSのDDL，DML等）と発展して来たソフトウェア技術の重心の移動を鳥瞰するのも一方法である。

ＤＢＭＳそのものを学習するのではなく，道具としてこれを考える立場で研究する。しかし道具を熟知しているかどうかは，ソフトウェア設計の巧拙のきめ手になる。

内　　容

4.1 データ管理

コンピュータをデータ処理の道具として使用することは，伝票や台帳類をコンピュータ・システムがもっている各種の媒体の上で，読み書きすることである。すなわちカード，磁気テープ，磁気ディスクなどの媒体の上での記録がデータ管理の実体である。これらの媒体をコンピュータ・システムのデータ・ファイルとよぶ。したがってデータ管理の設計をファイル設計とも呼ぶ。

特定の有意の情報を集めて蓄積したものがファイルである。実際上は，貯えられた情報のなかから特定の情報をとりだしたり，その中の一部に変化がある場合，その変化のあった部分に変更を加えて報告書にまとめてアウトプットすることが情報処理システムの処理の内容である。

ファイル設計の主眼は，情報をある目的にそって引出す場合に，合理的で，能率的に引出しやすくなるように設計することにある。

データ管理の設計が独立項目として検討されるようになった由来は，オペレーティング・システムの発達による。したがって，まずオペレーティング・システムの概略の機能を復習しておく必要がある。ファイルにはプログラムが指示する形（論理ファイル）と磁気テープ装置や磁気ディスク装置のような記憶媒体（物理ファイル）とがある。オペレーティング・システムでは，この両者の概念の区別をつけるために，前者をデータ・セット（data set）と呼び，後者をボリューム（Volume）と呼ぶ。

ファイル設計の学習には，次のような項目について，単元Ｃ３で学んだことを復習して予備知識を整理しておく必要がある。

- ファイルの種類
- ファイルの内容
- ファイル装置の見積
- ファイル処理のパターン
- データ構造

(1) ファイルの種類

a) ファイルの内容に基づく分類

・マスタ・ファイル (master file)

・トランザクション・ファイル (transaction file)

・記録ファイル (history file)

・合計ファイル (summary file)

・付加ファイル (trailer file)

・インデックス・ファイル (index file)

・プログラム・ファイル (program file)

b) ファイルのアクセス方法に基づいた分類

・シーケンシャルファイル

・パーティション・シーケンシャルファイル

・インデックス・シーケンシャルファイル

・チェインド・シーケンシャルファイル

・ランダム・ファイル

(2) ファイルの内容

・ファイル名

・作成，更新サイクル

・種別

・媒体とファイリング形式

・件数

・1レコードの長さ

・固定，可変の区分

・1レコードの件数，1件あたりの桁数

・保存期間

・項目，桁数，固定，可変等の仕様

・分類項目

・ラベルの特記事項

・関連ファイル

(3) ファイル装置の見積

・磁気テープ装置

・磁気ドラム装置

・磁気ディスク装置

・マス・ストーレジ装置

(4) データ構造

a) データの論理構造

・データ・フィールド，データ・レコード，セグメント，セット

・順次構造 (sequential structure)

・木構造 (tree structure)

・網構造 (network structure)，(graph structure)

b) データの記憶構造

・リスト構造 (list structure)

・リング構造 (ring structure)

c) データの編成

・階層順次編成 (hierarchical sequential organization)

・階層索引順次編成 (hierarchical indexed sequential organization)

・階層索引直接編成 (hierarchical indexed direct organization)

・階層直接編成 (hierarchical direct organization)

ファイル設計の中心はファイル編成の選定である。

目的の業務に対して，どのようなファイル装置で，どのくらいの磁気テープやディスクパックが必要になるかを算出する方法を学習する。業務側から検討するポイントとして，平常時とピーク時のデータ量の変化，アクセスの頻度，同時にアクセスされるファイルの種類，ファイルの移動性，拡張性，活動性に対する考慮，なども考えてみる。この検討にはコンピュータ側の見方と業務側の見方の協力が必要になる。

データ構造をどのように設計するかという問題は，データに対するモジュール化の問題である。ソフトウェアの設計で考察した事項をふり返ってみる必要がある。即ち，データとプログラムとの間には相対的な関係がある。

プログラムの設計でモジュール分割の目安がつかめず，いたずらに複雑になりすぎる時には，問題を裏がえしてみて，データの側からプログラムを見なおしてみることもヒントになる。且つまたプログラムの階層の設計に対して，データ構造の階層のレベルとの整合性を検討しながらプログラム開発を進めることも必要である。この点の見落しがあると，後になって設計変更が困難になることがある。

(5) データ管理のパターン

データ処理の作業には，いくつかの基本的パターンが存在する。磁気テープが主な媒体であった時期から磁気ディスクがファイルの中心に変ったのにつれて，そのパターンにも若干の変化があったことに注意する必要がある。

a) 入力変換（input）

入力装置と外部記憶装置の間に処理速度の違いがあるために，インプットデータは一度外部記憶装置上に記録して，後の処理にそなえるのが一般的方法である。これを入力変換と呼ぶ。

b) 生成（generation）

1個又は複数のファイルから新しいファイルを生成することをいう。

c) 分類（sort）

事務処理の大部分の仕事は，分類・照合の作業である。あるキーコードに基づく分類をおこなって作業用のファイルを作ることはファイル管理上重要な仕事である。

d) 更新（update）

ファイルの内容の変更，追加，削除の仕事をいう。このための技法として各種のものが考案されている。データ管理技術の中心である。

e) 出力変換（outpot）

記録されている情報を，人間が利用できる形式で印刷したり，表示したりする手続で，このための専用のパッケージも各種のものが開発されている。

f) チェック（check point）

情報処理システムの正確性と信頼性は，適切なチェック機構がその中に組ま込まれて得られるものである。データ管理の設計は，このチェックポイントの設計であるということも出来る。

4.2 ファイルの設計

前節で復習した項目は，いわばコンピュータの側からみたファイル設計のための基礎事項である。業務内容の側からみたファイル設計が必要になる。次のような検討が必要になる。

(1) インプットの把握

システムとユーザとの最初の接点はインプットである。どのような内容をどのような仕様の周辺機器を介してシステムに入力するかを検討しなければならない。設備については2章で学習したので，ここではインプット・データの分析を主とする。細目として次の項目がある。

- インプット・データの種類
- インプットの周期
- 原始データの内容
- 原始データの作成部門

・データ量

・項目，桁数

これらについて，選択された周辺機器に対する適合性も検討する必要がある。

入力部門での原始データの作成の手順，チェック項目の検討などは，ユーザとのインターフェイスとして重要な項目である。

インプット方式の選択には入力装置の特長もつかんでおく必要がある。原始データをそのまま入力するOMR，OCR，MICRや，図形データを入力するタブレット，グラフイックスなどの端末装置なども検討する必要がある時がある。オンライン端末では入出力を兼ねている場合が多いのでその考慮も要る。装置については設備計画のところで既に検討したが，ここでは，インプット方式の選択という見方でもう一度それを見直してみるとよい。読取り装置類のおおよその読取る速度を把握しておくことはインピットの設計をおこなう上での基礎でもある。

(2) アウトプットの把握

アウトプットはそのシステムが本来何を目的とするものとして設計されたものかを示すものである。アウトプットを必要とする人の立場で，その形式と内容を検討する必要がある。さらにその出力のタイミングは，アウトプット・データが，本質的に情報としての役割をはたすか否かの鍵になるものである。検討項目には次のような諸点があげられる。

・アウトプット・データの種類

・アウトプット周期，時期

・アウトプット・データの内容と形式

・作成部数，そのデータの量

・配布先，配布ルート

・アウトプット・データの使用目的→行動

・アウトプット・データのチェック項目

・内容の細目，項目，桁数，表現，集計区分

アウトプット・データの送付の時期，方法の検討は，ユーザとのインターフェイスとして重要である。

アウトプット方式には，通常のラインプリンターによる表形式のもののほかに，図形表示の形式のX－Yプリンターによる図面，グラフや，オンライン端末による文字や図形のディスプレイ表示もある。これらは対象業務に効果のある形式を選ぶことが望ましい。

(3) 中間結果の把握

インプット・データがアウトプットに変換される過程は単一の処理で終ることは少ない。

その途中経過をどのように構成するかというところが，プロセス・フローでもある。中間に何段階もの処理のフェーズが加わる時には，その中間結果は，全体のプロセスをそのまま続行してもいいかどうかを確認するために，出力されることもあるし，中間結果自身がアウトプット・データとして必要な時もある。会計業務処理などの場合はこのような形態をとるものが多い。この場合日報，月報の類はチェックのためと，経過観察のための二重の目的でアウトプットされる。

中間結果の検討は，アウトプットの検討と同様の項目でおこなう。但し無用の中間結果を多く設けることは，システム全体の硬直性を増す結果になるので，特にこの点に注意する必要がある。

さらには，オペレーティング・システムの発達の結果，多重処理がおこなわれるようになったため，非同期処理がおこりうるので，特別な注意を中間結果に対して払う必要が生じつつあることに留意しなければならない。即ち中間結果に対して書込みと読み出しの要求が同時におこる場合に論理的に矛盾がおこらないようにしなければならない。非同期処理が可能なものと，不可能なものとのお互いの関係を明らかにしておくことが，中間結果の整理には特に必要である。

(4) 関連の把握

システムの対象が広範囲にわたるときは，インプット，アウトプットを個別に把握しても，全体をつかむことが出来ない。単純な判断作業の一次元の連続であるような処理手順であらわされないシステムもある。このような場合には，それらの部分の関連表をマトリックスの形式で表現する方法が便利である。

オンライン・システムの場合には，インプット，アウトプットの個別の項目の検討よりも以前に，その事象の生起の確率の推測などの方が，設計上考慮すべき点としての比重が高くなる。これのための準備設計として，待行列型のシミュレーションを利用することも必要になるし，処理手続そのものの把握にも，シミュレーションが用いられる。インプット・アウトプットの把握の方法も，一般業務処理の場合と幾分異なるところがある。

前節で学んだコンピュータ側からみたデータ管理の機能と，上に述べたような業務側からの要求とを結合させるものがファイル設計である。コンピュータの適用業務が小規模な頃には，業務内容を時系列に沿ってそのままプログラムに移せばよく，その場合はファイルの役割は，プログラムに対して従属的であったが，システムの規模が大きくなり，プログラムとデータとが互に独立なシステム・エレメントとして考えられるようになると，ファイル設計の内容が大きく変った。

ここでは単純な形式のファイル設計の基礎事項を先ず学習して，次に新しいファイル設計の

概念となったデータ・ベースについて学習する。

4.3 データ・ベース概念

(1) データ・ベース概念の背景

トータル・システム（Total System）や，ＭＩＳ（Management Information System）がコンピュータの応用分野の体系化の考え方であるのにくらべると，データ・ベースは，これらよりもコンピュータ技術側に近い一種の体系化を指向した概念である。この考え方は1960年代の後半に生れて，その後急速に具体化が進展しつつあるコンピュータの応用技術である。

コンピュータによる情報処理の普及につれて，データ量の増加に伴う次のような各種の情報処理上の難点が指摘されるようになった。即ち

- データの範囲の拡大によるプログラム構成の複雑さが，プログラム作成費用を増大させる。
- データの範囲が増加すると，次にはデータ項目間に新たな関係づけを追加したいという要求がおこる。
- 個別業務への適用を，次々に増加していると，そこであつかわれるデータ同志の間で，お互に重複しているものが発生し，その重複しているデータに異なる値のものが見出されるなどで，全体としての整合性が失われるようになる。
- 既存のデータを利用するad-hoc的情報処理の要求が増加する。
- データとしては全般にわたるものを多角的に使用し，情報処理の目的としては部分的であるような処理要求が増加する。
- そしてこれらの情報処理の要求はプログラムの変更として，全体に影響するような手直しを度々おこなわなければならない結果になる。

これらの問題点を解決する手段として考えられた一つの方向が，ここで学習するデータ・ベース概念である。

データ・ベースという考え方は，処理体系の中でデータとプログラムをお互いに独立にとりあつかえるように分離して，全体の処理系を拡張性のあるものにしようとするものである。この考え方を実現するための手段として具体化されたシステム・プログラムが，データ・ベース管理システム，DBMS（Data Base Management System）である。即ち，要約すると，DBMSとは，「定常的な業務処理と，ad-hoc的な要求との両者に対して，共有できるようなデータの管理方法として，ランダム・アクセス・ファイルを有効に利用するシステム」であるということができる。

ソフトウェア設計の趨勢が、データ・ベース・マネジメント・システムに依拠するようになった経路には，次のような三つの方法があった。

(a) O.S. のアクセス方式の延長

(b) データ・ベース・パッケージの大型化

(c) 情報検索システムの拡大

データ・ベース概念の把握には，これらの発展経過をふりかえってみるのも一方法である。

(2) データ・ベース概念の特徴

現在のO.S.には各種のアクセス方式が採用されている。即ち

BSAM(Basic Sequential Access Method)

QSAM(Queued Sequential Access Method)

BPAM(Basic Partitioned Access Method)

BISAM(Basic Indexed Sequential Access Method)

QISAM(Queued Indexed Sequential Access Method)

BDAM(Basic Direct Access Method)

BTAM(Basic Telecommunication Access Method)

QTAM(Qued Telecommunication Access Method)

TCAM(Tele-Communication Access Method)

これらの各種のアクセス方式が出現した基盤は，結局メッセージ操作 (message-handling) とメッセージ処理 (message－processing) の両者の間での非同期処理の拡大とその精密化である。アクセス方式の整備はデータ・ベースという考え方が成立するための，技術的な背景である。しかし，このような各種のアクセス方式が順次整備された経過では，大容量のランダム・アクセス記憶装置の有効活用という技術上の要求のみが先行し，その後でデータ・ベース概念が追従したという歴史的事情から，O.S.のアクセス方式はかならずしも現状のＤＢＭＳに融合しているとはいえない。DBMSは端的に云えば，O.S.のデータ・マネジメントの部分と，アプリケーション・プログラムのデータ処理部分を抜き出して，融合させようとする試みである。D.B.M.S.がねらっているものは，複雑化しすぎた処理体系の中からデータ管理の部分を抜き出して，処理系の考え方をさらに単純化しようとする意図である。しかしながら，多年にわたって蓄積されたアクセス方式の技術と無関係に，これを実現するのは現実的には不可能なため，実際のD.B.M.S.は既存のO.S.の上に附加されたパッケージ・プログラムの形成をとっている。このためにその意図するところと反して，処理系としてはむしろ複雑な間接的構成になっているものが多い。学習に際しては，混乱気味のＤＢＭＳの構成にまどわされることなく，その本質的なねらいである単純化の

意図を把握することを心掛けなければならない。

(3) ＤＢＭＳのねらい

以上概観したようなデータ・ベース概念の成立経過と関連して，それでは現実のＤＢＭＳは，システムとしてどのような目的と機能をもっているかを整理して把握する。

次のような項目をあげて考察してみる。

a) データ重複の排除

b) データの記述とプログラムの分離

c) データの論理構造と記憶構造の分離

d) 入力と出力の分離

e) データ体系の設計とデータ・ベース・システムの運営の分離

D.B.M.S.が上記のようなことを目的とするシステムであることの必要性は何によるものであるかをまず検討し，次にこの目的をはたすために，どのような方策がとられているかを，各種の提案や具体的なD.B.M.S.のパッケージについて検討する。これらの学習のための予備知識として，データ・ベース・システムに関する述語の定義について正確に把握することも必要になる。これにはCODASYLのシステムズ委員会の調査報告，DBTG(data base task group)の提案，GUIDE／SHARE要望などを参考にするとよい。

さらに具体的なＤＢＭＳのパッケージについて，その機能を検討してみることが可能ならば，データ・ベース・システムの認識をうるのに有効である。

ＤＢＭＳの機能を把握するための着眼点として，次のような項目がある。

表４－１　D.B.M.S.の機能

(1) 制御－同時アクセス，排他制御，スケジューリング，監視
(2) 言語－ＤＤＬ，ＤＭＬ（親言語，独立言語）
(3) 記憶－マッピング，多様なアクセス，データの保護
(4) 定義－スキーマ，サブ・スキーマ
(5) 保全－機密保持，データの保護，回復・修復
(6) 通信－オンライン，端末の多様化
(7) データ－関連，非冗長，完全，アクセス，機密，効率
(8) その他－拡張性，互換性，汎用性，問合せ応答

4.4　ＤＢＭＳの導入計画

データ・ベースの概念は，これからのソフトウェアの開発に欠くことの出来ない前提となるものである。今までのファイル中心のアプローチとは異なる設計上の留意点も必要となる。ＤＢＭＳはシステム設計に対してはあくまでも道具であるので，今までに学習したシステム設

計の大綱はデータ・ベース・システムの場合にも変化はないが，ＤＢＭＳによる場合には，まずこの道具について，その得失をよく心得ておくことが必要になる。それには実際に使用するＤＢＭＳの機能を実地に調査してみることも有効な学習方法である。そのためにはまずどのような点をねらってデータ・ベース・システムを設計するのかという意図をはっきりさせることと，データ・ベース・システムを採用した場合にはどのような問題点が新たに発生するかということを考えてみる必要がある。

(1) データ・ベース・システムを採用するねらい

今まで述べたような一般論としてのデータ・ベース・システムの効能書とは別に，現在システム設計を意図している問題に対応させて，データ・ベース・システムのメリットを具体的に想定して，そのためには，どのようなＤＢＭＳが最適であるかを考察してみる。即ち

(a) 情報の総合管理をねらうのか？

(b) システムの拡張性をねらうのか？

(c) 非定形の情報要求に応ずるためなのか？

(d) 情報提供の即時性に応ずるためなのか？

などというシステム設計上の目的を明確にして，このテーマに対応することが大切である。

ものには表があれば裏もある。データ・ベース・システムの採用に対して，そのメリットと共に，それに伴う問題点もあらかじめ，予測して対応策も考えておかなければならない。

データ・ベース・システムを採用する場合の問題点として考えられる諸点は次のようなものであろう。

(a) 効率の低下，汎用的なものが，専用的なものに劣るのは当然であって，その代償としてデータ管理の一元化等のメリットが得られるのであるから，その収支のバランスが，システムのねらいにかなっているかどうかを検討することが大切である。

(b) ファイル中心の方法からのコンバーション

すでに運用されているシステムをデータ・ベース形式に変更するには，そのためのコンバーション作業が必要となる。これらの余分の仕事を追加してでも，システムの拡張性を望むかどうかが一つの判断事項となるであろう。

(c) データ集中による障害危険の増加

データ・ベース・システムに移行することは，企業にとって重要な情報を集中化することであるから，そのための危険が増加することになる。ＤＢＭＳの重要な機能として機密保護やデータ修復があげられるのもこのためである。データの保全対策について十分な考慮が必要になる。

以上に述べたような諸点を着眼点として，ＤＢＭＳの機能調査をおこなって，目的とする

システム設計に対応させた評価を実施してみるとよい。

(2) データ・ベース・システムへの移行手順

データ・ベース・システムへ移行するには，段階的な手順をとることになるであろう。その時に考慮すべき諸点を学習してみる。

(a) データ量に対する考慮

実際に稼動しているシステムでは，データ量に対する考慮をおろそかにすると，運用面で重大な支障をもたらすことがある。

(b) データの種類に対する考慮

特定の種類のデータに着目して，順次移行するという方法も考えられる。

(c) サービスに対する考慮

データの収集，提供の頻度を考えて，段階的に移行するという方法も考えられる。

(d) 管理体制に対する考慮

データ・ベース・システムでは業務中心から，データ管理者中心に移行する。従ってこのための要員の配置も十分に考えて運用しなければならない。

以上の諸点を，目的としているシステム設計と関連させて実地に検討してみるとよい。さらにそのためのスケジュールを作成してみることが望ましい。

4.5 データ・ベース・システムの設計

データ・ベースの設計には，論理構造の設計と，記憶構造の設計の二つのフェーズがある。前者は業務側からのアプローチになり，後者はＤＢＭＳの利用者（専門技術者）側からのアプローチになる。さらにデータ・ベース・システムが実施運用されるようになると，データ管理者（GUIDE／SHARE要望の用語ではData Base Manager，CODASYLのDBTG提案の用語ではData Administrator）がこの中間に介入する。

データ・ベース・システムの場合には，一般にこのような二重構造になる面が多いために，システム設計にあっては，問題把握のための概念を分析的に明析にしておかなければならない。

データ構造の設計には，使用するＤＢＭＳで固有な呼び方をしている述語の意味をよく把握しておかないとならないことが多い。現状では処理効率を重視するために，記憶構造に対する配慮に重点がかたよりがちになるが，システム設計という立場から考えると，論理構造に重心をおいたシステム設計が本来の姿であることを忘れてはならない。

データの性格をよくみきわめて，分類し，データ間の関係づけをし，その使用頻度や処理の流れを考え，そのデータ構造を決定し，実際のＤＢＭＳによってそれを運用してみて，従来のファイル中心のプログラムとの差異を確めてみることが望ましい。

データ・ベースの設計の場合でも，従来のファイル設計と，その本質において異なるところはない。ソフトウェアの設計の章で学んだことと同様に，分析的なアプローチは前提として必要になる。特長的な点は，記憶構造に関する技術的な検討に触発されて，問題そのものがもつ論理構造が意識的に採りあげられるようになった事である。

この節では特にデータ構造に重点をおいて研究することが望ましい。

ＤＢＭＳの採用の理由は，適用業務の拡大に伴うＣＰＵの処理能力の限界に達したためその打開策である場合が多い。システムの運用に未経験であって，或日突然ＤＢＭＳを使用する大規模なシステムの設計に入るということは考えられない。したがって通常の場合はバッチ処理システムとオンライン処理システムを合体して統一的なデータ・ベース・システムに移行することが多い。以上のような状況を考慮に入れると，データ・ベース・システムの設計とは，既存の運用中のシステムからの移行計画であると見てもよい。即ち次のような手順が考えられる。

- 現行のアプリケーション・プログラムの見直しをする。
- データの統合をすることを前提にして考えて，無駄の多いバッチ処理部分を整理する。
- 今までの個別のプログラムで，同一対象に対して異なるコードや用語を使用しているものに対して統一的なコードや同一用語を使う。
- 個別プログラム毎の入出力部分を分離して，データの直接入力，利用部門の直接検索の手順を新たに設計する。
- ＤＢＭＳでは通常入力は入力待行列に入り，出力は出力待行列から端末に出力されるので，それらの操作のためのメッセージを設定する。（ＤＢＭＳが用意してあるものを学習することになる。）
- データの階層構造を設計する。以前のバッチ処理の時の形をそのまま引継ぐと，応答が遅くなることもあり得る。応答の速さを確保するためには３階層位までに編成することが望ましい。（スキーマ）
- 二重更新のおこりうる場合の調整も必要である。（やはり応答時間に影響する。）
- データの保全，機密に対する対策を考える。
- 拡張性，変更の容易性を考えて，新しいデータ・ベースのデータを基調にして，より有効な次の新たな適用プログラムを計画する。（サブスキーマ）

以上の手順を要約すると次のように整理できる。

① データの分析

ⅰ）使用データの分類（データの機能）

ⅱ）データ使用方法の分析（工程，用途）

ⅲ）データ量の予測（業務とデータの対応）

② データ構造の設計

i）階層及びセグメント・タイプ

ii）記憶構造の設計

③ プログラム群の設計（入力プログラム，出力プログラム）

④ 安全・機密対策

⑤ その他，データ・ベース管理者の決定

利用部門の組織，権限の制定

ＤＢＭＳ使用の効率向上方法の検討

データ・ベースの保守の予定

適用範囲の拡大の計画

このような段階は，通常のシステム開発の手順ときわめて類似している。

現実的には，はじめから大規模なデータ・ベース・システムの設計に突入するのではなくて，まずパイロットとしてのモデル・データ・ベースの設計と作成をおこなって，これによって，データ・ベースの創成方法を検討し，効率チェック・運用チェックをするためのテストを繰り返し，その後に範囲をひろげてデータ・ベースの設計とそのためのデータ収集（或いは現在運用中のシステムからのデータ変換）をおこなって後に本稼動に入るという二段階による移行計画がおこなわれる。

ＤＢＭＳの効率や応答速度などは，小規模なパイロットモデル・データ・ベースの場合と，実際の大規模データ・ベースの場合とでは，その傾向に大きな差異があることがあるのでその余裕も考慮に入れる必要がある。（応答時間の速さを維持するために大きい主記憶域を占有することもあり得る。）

学習としてはパイロットとしてのデータベース・モデルの作成をおこなうのがよい。

指導上の留意点

(1) 前章の学習がプログラム設計であるのに対応して，この章の学習がデータ設計である性格をあきらかにする。

(2) プログラム重点志向が，データ重点志向に転じた現在の情報処理技術の一般動向について言及する。データ中心の考え方が，本来の情報処理の本質であることについても注目する。

(3) ファイル設計の標準的な手順を基礎事項として学習させる。

(4) データ・ベース概念が発展した技術的，社会的な原因を検討させて，それによってソフトウエア技術がかかえている現在の問題点を認識させる。

(5) リレーショナル・モデル等の新しい技術問題などにも言及し，現在のＤＢＭＳの技術及び

考え方などに，未完成な側面もあることを認識させる。

(6) データ・ベース概念の発展がもたらした一つの成果は情報構造に対する分析的な視点である。この意味で「データ構造」の研究的学習に或程度の時間をわりあてることが望ましい。その場合は，記憶構造の問題よりも，論理構造の問題に重点をおく。

(7) ＤＢＭＳとして重要なテーマに，通信技術との結合があるが，これについては単元Ｃ３で学んでいるものとして省略する。但し情報処理システムの将来指向が，コミュニケーション・ネットワーク主導の傾向にあることについては，データ・ベース・システムの発展の経緯のところで言及する必要がある。

参　　考

(1) データ・ベース概念の成立に関係の深い二三のトピックス

a) CODASYL (The Conference on Data Systems Languages)の活動

(i) システムズ委員会（SC：Systems Committee）の汎用D.B.M.S.に関する調査報告

この調査報告は，データ・ベース・システムの分野の述語の標準化に貢献している面が大きい。1969年と1971年の２回の調査があるが，特に後者の調査報告が，概念や述語の整理に役立つ。

CODASYLのシステム委員会が，調査対象として採りあげたDBMSは次の10種類であった。

表４－２　CODASYLのSCの調査対象

GIS（IBM）	1969
Mark Ⅳ- (Informatics Inc)	1968
NIPS／FFS (DCA　IBM, NMCS)	1968
TDMS (SDC)	1969
UL／1 (RCA)	1969
COBOL (CODASYL)	1970
DBTG (CODASYL, PLC)	1971
IDS　(H. I. S. 当時GE)	1963
IMS　(IBM, North American Rockwell)	1969
SC－1 (Western Electric, Auerbach)	1970

(ii) データ記述言語委員会（DDLC：Data Description Language Committee)のスキーマ，ＤＤＬなどの整理活動。(この委員会のデータ・ベース用データ記述言語の報

告書（1973）は，ＤＢＭＳの具体案に関する報告である。）

(iii) プログラミング言語委員会（ＰＬＣ：Programming Language Committee)のD.B.T.G.提案

DML，サブスキーマなどの整理，COBOL仕様検討。

b) **ＤＢＴＧ（data base task group）について**

W.G.Simmons の提案によって，CODASYL の中のCOBOL 言語小委員会の一環としてリスト処理作業班が形成され，その後そのグループは名称をデータ・ベース作業班（DBTG）と変更した（1967)。このDBTGがCODASYL言語委員会に提出した内部報告書（1969年）はデータ・ベースの開発の指針となった。この報告書はその後変更や拡張がおこなわれ1971年に再び報告書が作成された。これらの活動がきっかけになって，新しい委員会ＤＤＬＣが出来ることになり，ＤＢＴＧは発展してDBCTGとDMLTGに細分化された。ＤＢＣＴＧはデータ・ベース概念作業班でありＤＭＬＴＧはその後ＤＢＬＴＧ（data base language task group）と名称変更をおこなっている。

CODASYL の活動分野を委員会別に分類してみると，システム委員会は全体的な概念の把握，データ記述言語委員会は独立言語方式に関連する項目，プログラミング言語委員会は親言語方式に関連する項目となる。

c) ＧＵＩＤＥ／ＳＨＡＲＥの要望書

ＩＢＭのユーザ・グループから提起されたＤＢＭＳに対する要望書であって，ＣＯＤＡＳＹＬの調査ほど組織だった整理はなされていないが，実現の可能性は度外視して，ＤＢＭＳに対する本質的要望項目に，データとプログラムとの分離独立を強く主張している。

d) リレーショナル・モデル（relational model）

E.F.Coddの提案するリレーショナル・モデルの展開は，今後のデータ・ベースとして注目に値する考え方である。

従来のＤＢＭＳを順序ずけへの依存（ordering dependence)、インデクシングへの依存（indexing dependence)，アクセスパスへの依存（access path dependence）などにとらわれすぎているとして，ユーザからみて，今までよりもデータの独立性の高い体系が可能であるというのが，Coddのリレーショナル・モデルの提案である。データのモデルに対する考え方について，情報構造（information structure)，データ構造（data structure)，記憶構造（storage structure)の3つのレベルを考える。ＤＢＭＳの実現のための方法として，従来のようなポインタを使用するのをやめて，そのかわりに，データの関係をあらわす表（table）を使用する。ユーザからみて，より自然に情報の体系を表現しうるようにした。テーブルで表わされるものは，ドメイン（domain)(列）とタプ

ル（tuple)(行）で構成されるリレーションであって，このリレーションの集合がデータ構造になっている。これらのリレーションの間には階層関係はない。各種のリレーションの中のドメインにあらわれる同じデータ項目そのものがおのずからデータ構造を表現している。このような単純で均質的なデータ表現はかえって複雑な現実問題のモデルを作るのに適している。この考え方は今後のＤＢＭＳの方法として期待がよせられている。

e） ACM(association for computing machinery）の研究グループの活動，例えばＳＤＤＴＴＧ(stored data definition and translation task group）では記憶構造の定義の研究をおこなっている。

(2) ＤＢＭＳの特徴

a） データ重複の排除

従来の方法によると，個々の適用業務ごとにデータ・ファイルが作成され，その中味はコード化と分類によってファイルの量が増加する。

・分類には倍倍のスペースが必要になる。

・分類のステップが増えると必要スペースは幾何級数的に増加する。

・コードの標準化のためにブランクのファイルスペースが増加する。

少数のデータのエラーが幾多の計算ステップのリランを要求する。

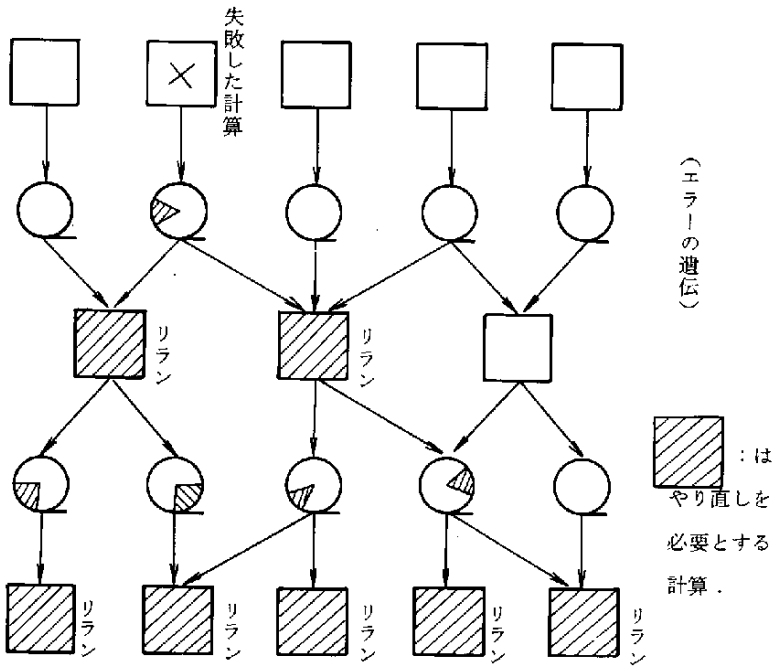


図４－１　エラーの伝播

システムを更新する場合に，同一データがいくつものファイルにあると更新に時間がかかるばかりでなく更新のし残しが生ずる。

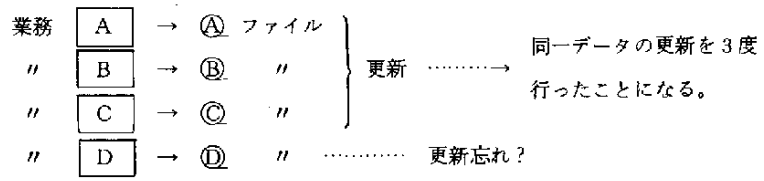


図4－2．更 新 の 忘 れ

データ重複の排除というねらいを，概念的に図式化すると次のようになる。

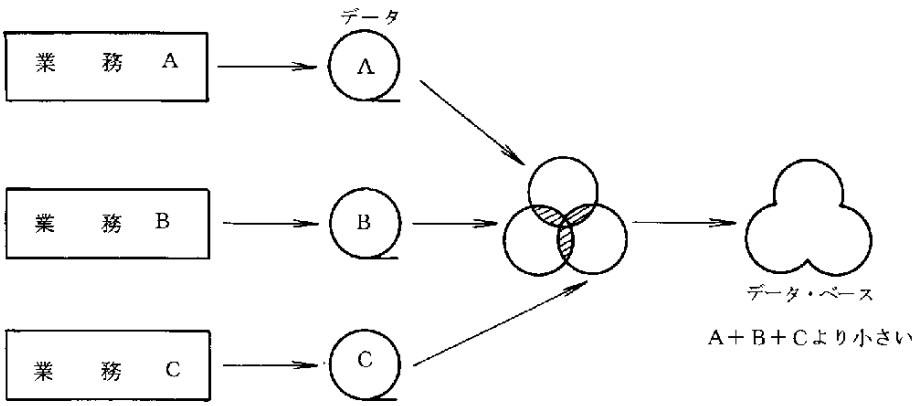


図4－3 データ重複の排除

b） データ記述とプログラムの分離

プログラムの中にデータ記述が組込まれていると，適用業務の一部分のみが変更されるだけで，データ記述部分への影響によって，プログラムの変更作業が大規模になり，保守要員も多くを必要とする。業務内容の変更は，データ部分の変更の場合と，処理手順の変更の場合の2通りがありうる。したがってデータ部分と処理部分とを分離しておけば，業務内容の変更による保守作業を局所的に狭めておくことが出来る。

c） データの論理構造と記憶構造の分離

記憶装置がもっているその物理的機能（記憶密度，書込みや読出しの手順や速度など）によって，データの記憶方式はおのずから制約をうける。ところが一方では，データの体系の論理的構造は，データ自身がもっている属性に基づく独自の構成をもつはずである。データの独立性を保つためには，データの記憶の形式にひきずられることなく，データ自身が内在的にもっている関連（論理構造）を体系化するデータの記述形式が望ましい。

従来のプログラムでは，この点は，むしろ，データの記憶形式にひきずられたものが多く，そのためにソフトウェアの設計が無理に複雑になっているものもあった。

現状のＤＢＭＳでも，データ記述に関してみると，データの物理構造（記憶形式）

(storage structure) と，論理構造（内在的関連づけ）(data structure) とを，完全に分離しているとはいえない。

この点は現実的な処理能率との妥協の結果ではあるが，データの論理構造と記憶構造の分離を介してデータとプログラムの間の独立性を獲得しようとする努力は継続されている。

このような技術動向は，情報検索プログラムの発展した結果とみることもできる。

データの独立という観点でのソフトウェア技術上の工夫は

- ＤＢＭＳの親言語方式（host language）と独立言語方式（self－contained language）
- データ記述言語（ＤＤＬ：Data Descriptive Language）とデータ操作言語（ＤＭＬ：Data Manipulation Language）

などである。

d）入力と出力の分離

情報は時間とともにその価値を減衰する。情報の提供には，即時性，迅速性が要求される。このことは情報処理のオンライン化の要求になる。ひとつのトランザクションの処理が終るまで，次のトランザクション処理がおこなえないという方法から脱して，マルチプログラミングへと発展し，メッセージ操作とメッセージ処理の分離を非同期に処理するようになれば，次に来る段階は，入力時期，及びその方法，データの蓄積及びその方法，出力時期及びその方法の三者の分離独立である。情報検索のプログラムは，上記のような特徴をもっていたため，ＤＢＭＳの源流の一つとなった。

データ・ベース・システムの前面（front－end）は通常のオンラインシステムと同じものであり，後面（back－end）はＤＢＭＳである。データ・ベース・システムの範囲が拡大するとともに，上記のように，入力と出力との独立性は高くなり，データの集中化によって，データ損壊時の危険が高まった。且つオンラインであることから，システムの修復の困難さ，機密保護の必要性など，新しい問題点が加わるようになった。データ通信技術が，ファイル処理技術と共に，データ・ベース・システムの技術上の両輪とみなされるようになりつつある。そのため，データ・ベース・システムのことをＤＢ／ＤＣシステムと呼ぶこともある。

オンライン化の面から，次の諸点に焦点をあてて，データ・ベース概念の発生の経過を観察するとよい。

- データ通信技術との結合
- 狭義のデータ・ベース・システム（情報検索）
- データの安全性，機密保護の工夫

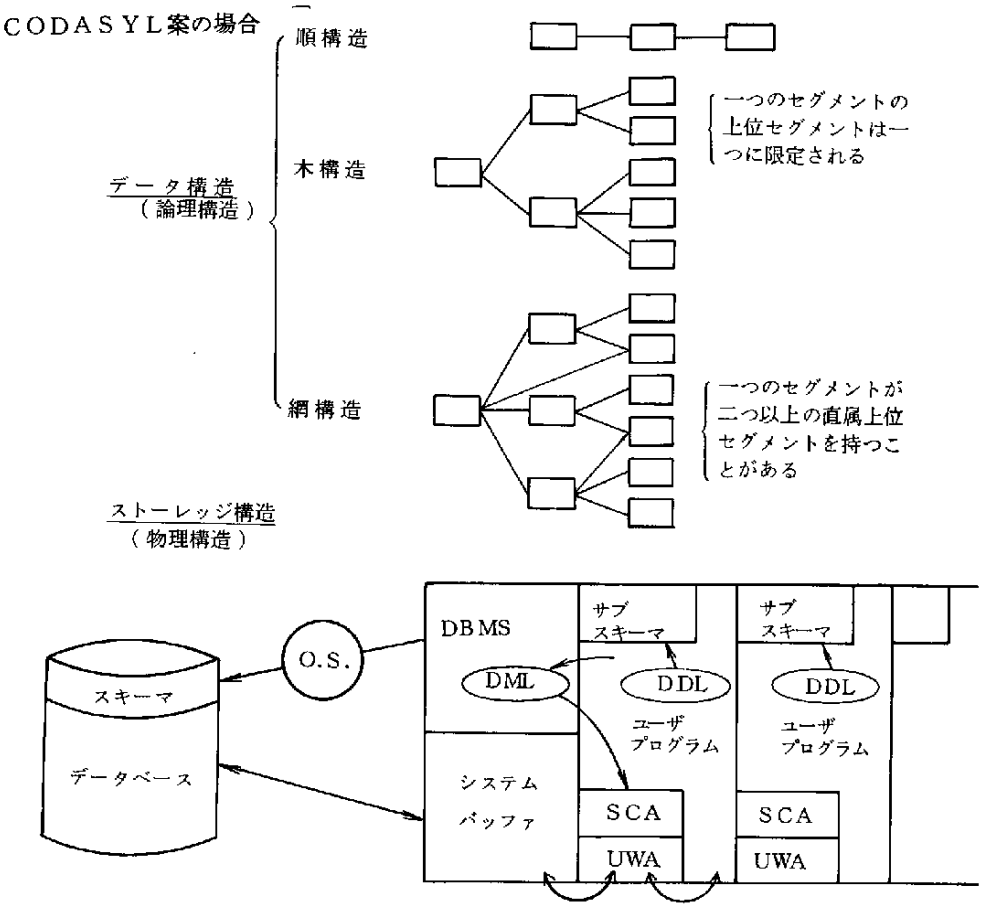
(3) ＤＢＭＳの機能

ＤＢＭＳは情報の蓄積機能，所要の情報を検索する機能，これらを独立に実行するための制御機能，たくわえられたデータを安全に保持し，データ処理を監視し，場合によっては損壊されたデータを修復する機能などをもっている。

(a) データ構造

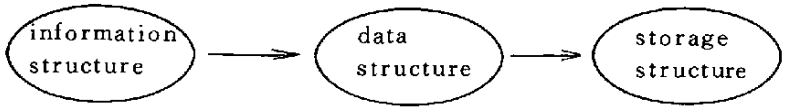
データ構造に関する研究が活発になったことは，ＤＢＭＳの普及に伴う現象である。従来は，データ構造の問題は主としてＯＳの内部に限定された技術的テーマであるとみなされていたが，データ・ベース・システムの普及によって，最近ではむしろソフトウェアの設計問題の中心課題がデータ構造の想定であると考えられるように変化しつつある。

データ構造の学習というのは，一面では，ＤＢＭＳの内部機構としての基本部分を把握するためであり，他の面では，システム設計としてデータの性格を分析して，ＤＢＭＳを有効に活用するためである。

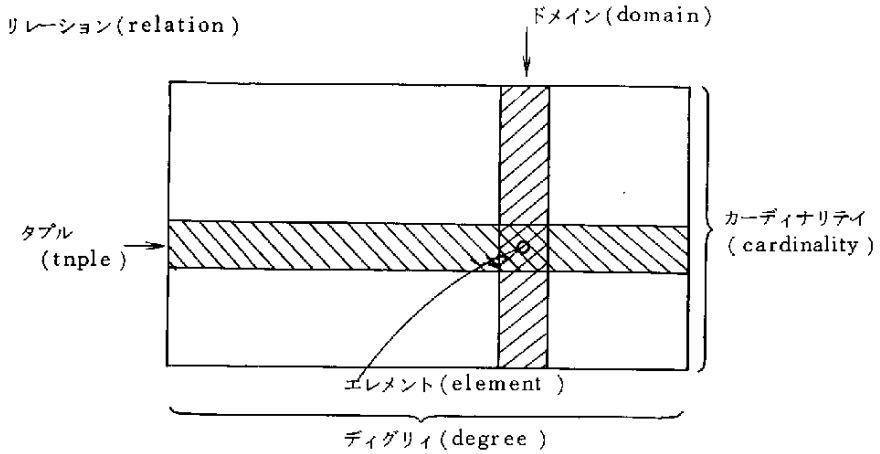


SCA：System Communication Area，
UWA：User Working Area

Coddのリレーショナル・モデルの場合



リレーショナル・モデル概念図



リレーショナル・モデルの表形式

CODASYL 方式のデータ・ベースでは，データ構造は木構造が中心になっていて，場合によってこの木構造の網構造への拡張が可能になるようになっているものが多い。そしてデータの構造定義をあらかじめ処理の時点より前におこなっていて，データ・ベースが使用できるのが通常の形式である。

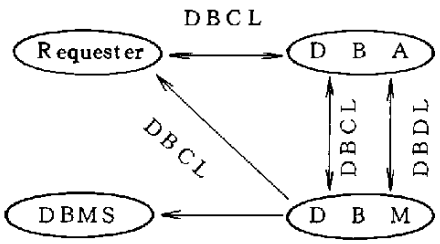
リレーショナル・モデルの場合は，既に実用に使用されている従来のＤＢＭＳの機能をそのまま利用して（例えば，保全とか通信のシステムはそのままとして),データ構造のとりあつかいの部分のみをパッケージ化することによって，その実現化を計画しているものもある。この場合のデータ管理機構としては，O.S.のアクセス方式，木構造形式のＤＢＭＳ，リレーショナル・モデルをとりあつかうデータ構造操作パッケージという具合に三重構成となり，原理的には単純でありながら，現実的には複雑なものになることもありうる。

b) データ管理業務の分解

データ管理の方式は，従来のプログラムの作り方では，個別のアプリケーションの設計思想に支配されて，夫々異なる手順をとることになり，データはプログラムに対して従属的な位置にあった。

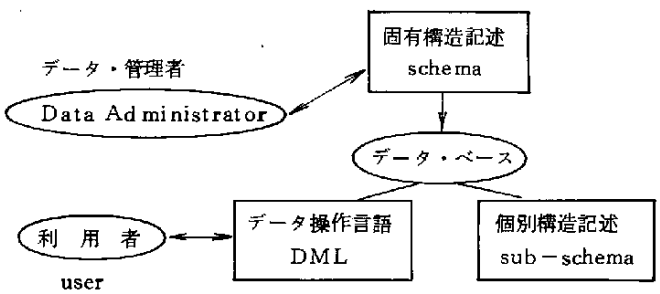
ＤＢＭＳの採用によって，データ管理の方式は，その立場を逆転した。データ管理業務に専従する技術者が必要となり，この専門技術者が，データ管理に関する基本的な部分の運用をつかさどり，プログラマはむしろそのデータ・ベースを利用する立場となった。

データ・ベース・システムの構成を，ＩＢＭのユーザ・グループのGUIDE／SHARE要望によって図式化すると次のようにあらわされる。



DBA　：Data Base Administrator　人
DBM　：Data Base Manager　ソフトウェア
DBMS：Data Base Management System　実行ソフトウェア
DBCL：Data Base Command Language
DBDL：Data Base Descriptive Language

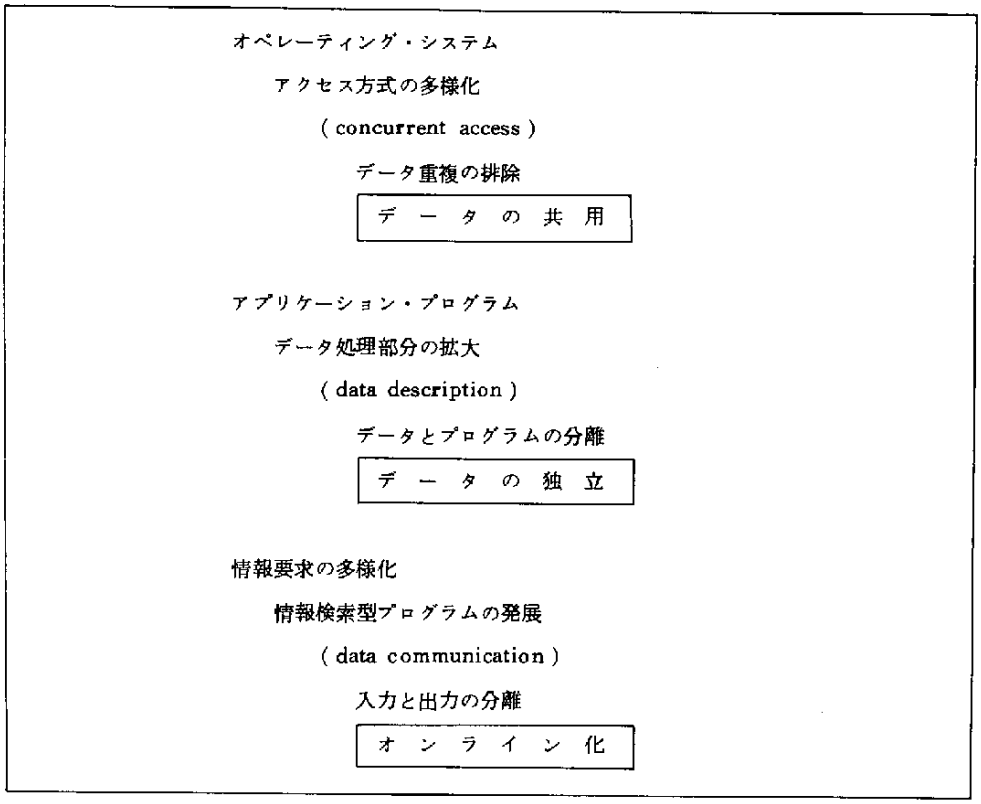
或いはＣＯＤＡＳＹＬのＤＢＴＧ提案による表現であらわすと，次のようになる。



(4)　データ・ベース概念の発展にともなって派生した各種の新しい概念語

項目（item），値（value），レコード（record），エントリ（entry），集団（group），ファイル（file），属性（attribute），事象（entity），具体値（instance），　実現値（occurence），論理関係（logical relation），論理構造（logical structure），記憶構造（storage structure），写像（mapping），固有構造記述：スキーマ（schema），個別構造記述：サブスキーマ（sub－schema），順構造（sequential structure），木構造（tree structure），網構造（network structure），リスト構造（list structure），リング構造（ring structure），連鎖（chain），ポインタ（pointer），　ハッシュ化（hash），転置ファイル（inverted file），セグメント（segment），セット（set），タプル（tuple），ドメイン（domain），データ・サブランゲージ（data sublanguage）

(5) データ・ベース概念の発展



(6) ＤＢＭＳの導入のための機能調査項目の例

(1) 設　備	(i) ハード・ウェア	(a)ハードウェアからの独立（上位，下位機種への流用） （端末選択の自由） (b)記憶域の専有度（ＤＢＭＳの必要とする記憶域）
	(ii) ソフト・ウェア	(a) O.S.からの独立（O.S.改訂の影響？） (b)プログラマ言語（親言語は？　独立言語は？） (c)アクセス方式（使用するアクセス方式は？） (d)データの重複率（処理能率との関連で） (e)プログラムの独立（プログラムの更新の自由） (f)データの保全（データ保全の単位は？）
(2) 使いやすさ	(i) 導入時	(a)D.B.の定義（システム設計者からみて） (b)データのローディング（方法，処理時間）
	(ii) 変更時	(a)データ構造の変更（relation設定の手順） (b)D.B.の拡張（新しいＤＢ追加の手順）
	(iii) 稼動時	(a)D.B.管理者からみた使いやすさ (b)プログラマ，利用者からみた使いやすさ
	(iv) サービス	(a)D.B.M.S.提供者の行う教育（カリキュラム，テキスト） (b)技術援助（トラブル発生時の応答性）
(3) 一般機能	(i) 共用性	(a)D.B.運用上の各種ユーティリテイ (b)マニュアルの完備
	(ii) 検　索	(a)検索言語 (b)出力編集機能
	(iii) 統　計	(a)監視用 (b)使用実績ロギング
	(iv) 保護回復	(a)チェック・ポイント・リスタート (b)バック・アップ

参考文献

〔1〕 Bertiss, A.T. :"Data structures-Theory and Practice", Academic , 1971

〔2〕 Cagan, C. :"Data Management Systems", Melville, 1973

〔3〕 CODASYL Systems Committee :"Selection and Acquisition of Data Bare Management Systems", ACM, 1976

〔4〕 Date , C. J. : "An Introduction to Data Base Systems", Addison-Wesley, 1975

〔5〕 Dougué, B. and Nijssen, G. M. :"Data Base Description", North Holland,

1975

〔6〕 Lyon, J.K. :"The Data Base Administrator," Wiley, 1976

〔7〕 Nijssen, G. M. :"Modelling in Data Base Management Systems", North Holland, 1976

〔8〕 Sprowls, R.C. :"Management Data Bases," Wiley, 1976

〔9〕 CODASYL Systems Committee :"Feature Analysis of Generalized Data Base Management Systems," ACM, 1971

〔10〕 Flores, I. :"Data Structure and Management", Prentice-Hall, 1970

〔11〕 GUIDE／SHARE : "Guide ／ Share Data Base Management Systems Requirements," Technical Report, 1970

〔12〕 Lyon, J.K. :"An Introduction to Data Base Design", Wiley, 1971

〔13〕 Horwitz, E. and Salni, S. :"Fundamentals of Data Structures", Computer Science Press, 1976

〔14〕 CODASYL :"COBOL Data Base Facility Proposal", 1973

〔15〕 The COBOL "Data Facility in CODASYL" COBOL JOD, 1975

〔16〕 Codd, E.F. :"A Relational Model of Data for Large Shared Data Banks", CACM Vol.13, No.16, 1970

〔17〕 IBM :"IMS／VS Utilites Reference Manual," 1969

〔18〕 MRI :"System 2000 Reference Manual", 1974

〔19〕 Cincom Systems, Inc :"TOTAL ／7 Reference Manual", 1977

〔20〕 Software AG :"ADABAS Reference Manual", Utilities Manual, 1977

〔21〕 Cohen L.J. ;"Data Base Management Systems", Performance Development Corp, 1973

〔22〕 Date, C.J. :"Relational Data Base Comcepts," Datamation, April, 1976

〔23〕 小林功武「データ・ベース入門」事務管理11巻, 1972

〔24〕 小林功武「データ・ベースの管理」企画センター, 1970

〔25〕 ケイガン著（西村恕彦訳）「データ・ベース入門」近代科学社, 1975

〔26〕 D. レフコビッツ著（横山保, 他訳）「オンライン・ファイルシステム」日本生産性本部, 1976

〔27〕 ジャド著（戸川隼人訳）「ファイル処理」サイエンス社, 1974

〔28〕 I・フローリズ著（久保寛彦訳）「データ構造」竹内書店, 1972

〔29〕 I・フローリズ著（久保寛彦訳）「データ管理」竹内書店, 1972

〔30〕 浦　昭二「データ構造」共立出版，1974

〔31〕 K・R・ロンドン著（穂坂有郎訳）「ダイレクト・アクセス技法」学研，1976

〔32〕 西村恕彦，中川友康監修「データ・ベース・システム」共立出版，1973

〔33〕 情報処理学会，データ・ベース言語研究会委員会訳,「CODASYLデータ・ベース用データ記述言語」　1973

〔34〕 西野博二「データ・ベース・システムの研究開発動向」情報処理，Vol.17,No.10,1976

〔35〕 朴木　実「データ・ベース導入と設計」企画センター，1975

〔36〕 西村恕彦監修「データ・ベースの選び方」共立出版，1977

〔37〕 穂鷹良介「データ・ベース要論」共立出版，1978

〔38〕 情報処理振興事業協会「エンジニアリング・データ・ベース・システム調査報告書」1976

〔39〕 西村恕彦監修「FORTRANデータ・ベース」bit 1978，10月，共立出版

第5章　プログラムの作成と検査

キーワード

アルゴリズム (algorithm)，インプリメンテーション (implementation)，プログラムテスト (program test)，機能テスト (functional test)，効率テスト (performancetest)，環境テスト (environmental test)，信頼性 (reliability)，拡張性 (expandability),デバッグ (debug , debugging)，エラー恢復 (error recovery)，予防保全：PM (preventive maintenance)，デバッグ文組込み(built-in debug)，ダンプ(memory dump)，スナップショット (snap shot)，モジュール化 (modulality)，標準化 (standardization)，机上チェック (desk checking, dry run)，コンパイラ診断 (compiler diagnostics)，　作業見積 (work estimation)，日程計画 (scheduling)，品質管理 (quality control)，　費用管理 (cost control)，システム切換 (cut over, system conversion)

目　　標

システム設計について，1章，3章，そしてこの5章でとりあつかっている。システムライフサイクルに対応づけすると，1章はシステム設計の基本，あるいはシステム計画と呼ぶべき項目にあたり，3章はプログラム設計，5章はプログラム作成に該当する。

システム設計とプログラム設計の違いを考えてみると，前者は何を (what)システムとして作るかに重点があるのに対して，後者はいかに (how) システムにまとめあげるかに重点がある。さらにさかのぼればシステム分析は何故 (why) そのシステムを作る必要があるのかを考えることに重点がある。

システム設計が機能的モデル (functional model) を問題にするのに対して，プログラム設計は手順的モデル (procedual model) と物理的モデル (physical model) を作ることが仕事の中心になる。ここで学習することは，第3章のソフトウェアの設計で研究した原則的な項目の具体化である。

システムのライフサイクルの一局面であるインプリメンテーション (implementation) を学習することがここでの中心課題であるが，このimplementationという述語には適切な日本語訳が見出されない。「作成」とか「実施」とかの訳語もあるが，コンピュータ関係の用語としては，これらの訳語であらわされる意味よりももっと限定された内容をあらわしている。即ちプログラム設計，プログラミング，プログラムテスト，デバッグ，関係先の訓練などの一

連の作業をスケジュールにしたがって実施することがインプリメンテーションである。プログラムという現物を道具（implement）として作りあげることである。

システム設計全体の連鎖を見渡した時，現在，最も欠落している部分は，設計仕様が与えられた時，それをどうやって実際のプログラムに実現するかという方法である。設計はこうあるべきである。プログラムはこうあらねばならない。等の個々の議論は多いが，この両者を連結するところは，実地にプログラムを作ってみて体得するよりほかに学習の道はない。

プログラムを作成するには，数多くの予備知識が必要になることに気がつく。例えば，ハードウェアの機能，O.S.の役割，特にＪＣＬの書き方，データ・ベースの利用方法，デバッグ用ユーティリティを知っていなければならない。フィジカルモジュールを決める段階では，リンケージ・エディター，オーバレイの方法等の言語処理プログラムの性能なども把握していないと，実際の仕事はできない。プログラムテストの効率，実行時間の効率なども考えなければならない。コード変換，割込処理などのように処理が特に要求される部分にはアセンブラ・マクロのモジュールを使う必要もある。多くの経験的知識を動員しなければならない。さらに現実問題として既存の共通モジュール，標準モジュールを利用するための調整など，考慮すべき点は数多くある。このためには標準的な手続や標準サブルーチンについてのあらかじめ仕入れられている豊富な知識が前提となる。さらに中心的な課題として，プログラム内部のアルゴリズムについて十分に学習しなければならない。ここでは実際問題に対するその応用を研究することに主眼をおく。

さらに高い目標としては，プログラム技術を技能から技術へ前進させるために，ひとりよがりの技巧に浮することなく，前人の開発したものを利用し，その足場の上に立って仕事を進める方式を研究することを念頭におく。即ちプログラムの標準化，共同開発，共通プログラム等の重要性を認識させる。

内　　容

5.1　よいプログラムとは？

ここでは「よいプログラムとは何か？」を考えてみる。それには開発技術の現状で観察した諸々の問題点に対する解決の方向づけをするようなプログラムを考えてみればよい。

問題点をもう一度整理してみると次のような項目がひろいあげられる。

① 試作品がそのまま実用に供せられていて十分なテストがおこなわれていない。

② プログラムの中味がプログラマ以外にはわからないものが多い。

③ プログラムの修正要求にすぐに応ぜられないことが多い。

④ プログラムの共同作業による開発方式が確立されていない。

⑤ プログラムの品質を表現する尺度が明確でない。

⑥ 技術的蓄積を積みかさねて質的向上をはかるという工業技術の一般的発展過程の道が未発達である。

よいプログラムの定義を，常識的な判定基準によって，次のように設定して考察してみる。

- テスト可能なプログラム
- デバッグ可能なプログラム
- 発展性のあるプログラム

(1) テスト可能なプログラム

まず何をテストするのかが問題になる。論理的に内部状況を証明（prove）すること，実際的にテスト状況に対する合致を実証（verify）すること，外部状況に対する有効性をしめす（validate）こと，利用者の経験に対してその実効性を保証する（certify）ことなど，単にテストといっても色々な内容が含まれている。プログラマが独断的に内部の論理の正しさを主張するだけがテストではない。性能（performance）をテストすることは，実行時間や資源などの制約の間のバランスを決めることであるので，その測定には非論理的なものも介入する。自然科学分野ではテストの原理として「テストの反復可能性」を重視している。これからの類推では，非同期処理の場合の結果の変動を最小にする工夫，繰返しのきく入力，実行の記録などに対する注意などもテスト条件を考える場合に重要になる。

ソフトウェアの設計の章で考察したような諸点，すなわち

- プログラムのモジュール化（logical）
- プログラムのセグメント化（physical）
- プログラムの機能把握（functional）
- プログラムの構造化（structured）

などが，その答の一面ではあるが，これらは形式的な側面である。さらに基本的な問題点としてプログラマが自分の作るプログラムに対して何が望まれているかがはっきりしていないことが多いということを注意しなければならない。その結果としてテストは，プログラムがうまく働く場合だけに絞って作られてしまう例が多い。とりあつかう入力の本質について無知であることも多い。

(2) デバッグ可能なプログラム

まずプログラム作成にはデバッグが伴うことの認識から出発しなければならない。テストとデバッグは質的に異なる行為である。テストは誤りが存在することを示すだけのもので，デバッグは誤りの原因をさがし出すものである。誤りは開発のすべての局面に存在する。設

計の段階，プログラミングの段階，コーディングの段階，コンパイルの段階，部分テストの段階，全体テストの段階，実行の段階，運用の段階，修正の段階，プログラム保守の段階，等々多くの局面で誤りは発生する。

デバッグ可能なプログラムを作成するための工夫は数多く考えられているが，それらは一つ一つが思いつきのレベルであって，全体として体系的な知識にまではまとめられていない。

(3) 発展性のあるプログラム

テスト可能なプログラムやデバッグ可能なプログラムは，問題解決に直接結びついた条件を調べるための手段であるが，これらの要求とは性格的にやや矛盾する要求が，プログラムの評価の尺度として存在する。それは変更容易性，汎用性，使いやすさなどの尺度である。プログラムの信頼性や保守性の要求は，技術的には，融通性や拡張性の要求と調和させるのが難しいことが多い。

1960年代の前半頃までは，プログラマは適用業務そのものにわりあてられ，プログラムの設計，プログラミング，オペレーション等開発から運用迄の全体に対して責任をもった。このため先見性のあるプログラムを作ることは，プログラマの資質として訓練されたが，その後の発展に伴って，プログラマの作業範囲の限定が次第にせばめられて，プログラマの能力にたよることは次第に困難になった。

発展性のあるプログラムをどうやって開発するかは，今後の重要な課題である。これに関連する問題として，文書化は，「そこにあるものを記述する」ことよりも，「そこで意図したものが何か」をも表現する方を重視する必要が生じつつある。プログラマは技術的決定を個人の好みでひとりぎめしてしまうばかりでなく，設計方針を仲間と議論しあう習慣もないし，その意図を表現する能力もうしないつつある。他人の仕事の上に乗ることをせず，多くの場合，はじめからやってしまう。したがって何時でも一，二，三の繰り返しで，十，百，千の進歩にはほど遠い。トップダウンの開発手法はプログラムの信頼性を高める反面，他人の仕事を足場にしてその先に進むという工業技術の一般型から逆もどりして職人技能に足踏みする傾向に加担する危険も含んでいる。

発展性のあるプログラムとは，短期的には，そのプログラムが組込まれているシステムが，時間的経過によって，変更改良が加えられる時に，それに容易に順応できるようなプログラムであり，長期的にはそのプログラム自身が汎用性，互換性をもつものであることである。即ちソフトウェアの標準化の問題である。

5.2 プログラムテスト

テストとは誤りを発見するためにプログラムを実行する過程のことである。まず誤りとは何

かを解析する必要がある。

プログラムのエラーのケースを経験的に整理すれば次のような例があげられる。

- 物理的誤り
  - コントロールカードの誤り
  - カードの忘れ，位置の間違い
  - 余分のカードの捨て忘れ
  - データ形式の間違い
  - 古いプログラムの使用
- 論理的誤り
  - 判断の行先の間違い
  - ループの終了の間違い
  - ループの回数の間違い
  - ループの開始手順の間違い
  - 起りうるすべての場合を考慮しなかった
- 計算上の誤差
  - 有効桁数の制約からの誤差
  - 計算手順にもとづく誤差
  - 丸め誤差，桁落ち，累積の誤差　等々
- コーディングの誤り
  - 部分の脱落
  - 未定義記号，重複記号
  - 記号や命令の綴りの間違い
  - 入出力指定の誤り
  - サブルーチン指定の誤り

さらにプログラムの内容に入って，考えれば各種の誤りが考えられるが，ここではその一例として，数値計算プログラムの場合について検討すると，次のような各種のエラーが考えられる。

- 誤差解析
  - 伝播誤差（transmitted error）
  - 切捨誤差（analytic truncation error）
  - 丸め誤差（round off error）
- 領域解析

常微分方程式の数値解などの場合，特にクセの悪い問題むきに作ったものは，一般問題をとくためにはかえって非能率になる。したがって将来はとりあつかう領域によってプログラムの分担が分化することになるであろう。

・時間解析

計算時間の制約も考慮しなければならないものもある。

・その他

収束判定基準，負の引数，オーバフロー，アンダフロー，デフォルトの設定などに対する考慮。

問題領域によっては，それぞれその分野個有の誤差が存在する。

このように分析してみると，テストの問題はきわめて範囲の広い検討事項を含んでいることが想像できる。

まず，テストの種類を一通り調べ，次にテストの方法を検討する。

テストの種類としては，形態的にみて，単純に

① 個別テスト

② 全体テスト

が考えられるが，これをさらに範囲別，時系列別に分解して，次の4段階に分けて考えることが実際的であるとみられている。

① 論理テスト（logic test），モジュール・テスト（module test），ユニット・テスト

② 機能テスト（function test），モジュールの結合テスト（coupling test），インターフェース・テスト

③ 環境テスト（environment test），統合テスト（integration test），システム・テスト

④ 実施テスト（live test），受入れテスト（installation test）

このような分類が，テストの学習にとってどれほどの意味があるかは，疑問ではあるが，実際上の問題として，テスト計画の策定の上では，どの段階でどのテストを行うかは，重要な問題である。大々のテストが何を主目的とし，どのような項目をテストするかを把握する必要がある。特に第三段階のシステム・テストには，次のような範囲の広いテスト項目が含まれる。

・プログラム負荷テスト，能力テスト

・構成テスト，オプション選択テスト

・適合性テスト，保護テスト

・記憶域判定テスト

・性能テスト，効率テスト

- 設置可能性テスト
- 信頼性テスト・回復性テスト
- サービス性テスト・人間因子テスト
- 文書テスト・手続きテスト

テストとは誤りを発見する過程であるという定義にたちもどって考えてみると，これらの多種類のテストに於いて，その誤りとは何であるか？　これらを考えることはプログラムとは何であるのかを考えてみることになる。

次に，その誤りを見出すための方法について学習する。次のような手順がある。

① ボトムアップ・テスト
② トップダウン・テスト
③ 一斉テスト
④ 混合形（①＋②）のテスト

テストのための方法論としては，プログラム開発技法に関する議論と同じく，各種の意見がある。前にあげたテストの種類との対応で，これらの方法の特長を研究することが望ましい。

テストの学習で，本質的な問題は，テストそのものがもっている性格の分析である。

Myers の次のような主張が，この問題に対する一つのヒントを提供している。

『エラーがないことを示そうとすればエラーはあまりみつからない。エラーがあることを示そうとすればエラーは沢山みつけられる。システムに故障を起させるのがテストの成功を意味し，プログラムに故障を起させなかったらそのテストは不成功である。』

また，Weinberg は egoless programming の主張で，

『自分のプログラムを自分自身の自我の延長だと思うプログラマは，自分のプログラムのエラーをみつけようとしない。逆に自分のプログラムが正しいことを証明しようとする。ほかの人の目では大きなエラーと見えるものでも見過してしまう。』

と述べている。

このようなテストに関する心理学的な検討は，テストの分類やその細目の学習よりも基本的な事柄であるし，次に学ぶデバッグのための立場を明らかにすることにもなる。

さらにテスト及びその評価判定について，次の二つの立場あるいは傾向があることにも注意をうながす必要がある。

① 仕様についてのテスト→プログラムの内容はみない（入力の組合せを重視）
② プログラムの中味についてのテスト→仕様との対応はみない（プログラム処理の経路の組合せを重視する）

このことについて注釈を加えると，プログラム・テストについての技術的問題点として，プ

ログラム・ロジックの全部をテストするのは有限時間では不可能であるという根本問題があることに注意する必要がある。ロジック中にある判断分岐による経路数の組合せの数を試算してみれば，きわめて単純なプログラムの場合でも，それは天文学的数になる。（プログラム・テストの方法の最大の弱点は，プログラムにエラーがないことの保証ができないという点である。これに関連して「プログラムの正当性の証明」「プログラムの検証理論」という研究分野があって，プログラムの基礎に関する研究がおこなわれていることにも言及するとよい。）

テストとは，経済的制約の下で，どうやって適当なテストケースを選ぶかという経験的，直観的な視点に帰着する問題であるということを認識させることが重要である。

実例について，完全なテストをおこなうには，どの位の時間が必要になるかを試算させてみて，その不可能さがわかった段階で，それではどうやれば，最も信頼できるテストになるかを検討させる。……これには当然何をテストしたいのかという目的に対する反省が必要になる。

5.3 デバッグ

プログラム設計と文書化の作業を除くプログラム作成の費用の50％以上は，プログラムの手直しに消費されている。プログラマがコーディングのために鉛筆を持つ時間はきわめて短かい。大部分の時間はデバッグのための試行錯誤に費されているのが現状である。

デバッグをどのようにおこなったら，実効があがるかという議論も，前節のプログラム・テストの諸説と同じように数多くの原則や提案がなされている。ここでは，実際的な手法の例を採りあげて検討する。

a) **机上デバッグ**（desk checking）の徹底

デバッグの時間を大きくしている最大の原因は，プログラムを書きあげると再検討せずにコンピュータでテストを実行してしまうこと，エラーの原因の一つをみつけると，そのほかにもエラーの原因になるものがないかを調べずに，その一箇所だけを修正してすぐにコンピュータでのテストにかけてしまうことである。

b) **コンパイラ診断**（compiler diagnostics）のエラー指摘，デバッグのための参考情報のリストなどの利用方法に精通すること。

c) **ＴＳＳによるデバッグ**

通常のバッチ処理形の言語処理よりもインタラクティブな方法によるデバッグの方がエラーの箇所が局所的に追求できて，デバッグ時間が短縮できる。

d) **第三者の手によるプログラムテスト**

プログラム作成者には思い込みによる思考の短絡のためエラーがみつけだせないこともある。第三者による客観的探索がデバッグに有効なことが多い。

e) **プログラムの修正は**，コーディングの訂正に直ちにかからず，一歩しりぞいて，機能図，流れ図等の**文書の修正を先にして**，その後にコーディングの訂正にとりかかる癖をつける。

さらに次のような経験的，注意事項も重要である。

- エラーの発生は作業のつぎ目（between processes）におこることが多い。工程間検査（inprocess inspection）が重要である。

  流　れ　図 → コーディング

  コーディング → カードパンチ

  カードデック → オブジェクトプログラム

  開　　発 → 実　行

- 流れ図とコーディングは注意の焦点が異なる。全体を見る時は部分が見えなくなり，部分を見る時は全体が見えなくなる。

  流　れ　図 → 全体の構成

  コーディング → 局部の論理

- デバッグには規則的な消し込みが有効。

  すべてのデータの場所の確保

  すべての変数の宣言の有無

  すべての流れ図のラベル

- 目で追うだけでは犯人を見うしなう。手を使うことの重要さを忘れてはならない。

  流れ図の書き直し

  チェックリストによる全項目の消し込み

- プログラム言語の性格によってエラーのおこりやすい箇所が異る。

  エラー統計をもとにして，言語別にチェックリストを作ることが有効である。

- プログラムテストがすべて完了しても，実行時のエラーが残る可能性は未だ残っている。
- デバッグとは結局は見出されたエラーにたいしてその原因を推察し，対策をたてる作業で本質的には後始末である。エラーの修正よりも，エラーの早期発見が望ましく，さらにはエラーの防止の方がさらに望ましい。プログラムの作成方法にも，機械故障の防止と同様に，予防保全（preventive maintenance）の思想が必要である。
- 工業製品で組立用治工具足場などをあらかじめ製品の中に作りつけにする例や，検査用のぞき穴，補修用部品の予備の取付と同じように，テスト用のルーチンの組込みや，プログラムの状態観察用，計算進行のトレースのためのデータやチェック用変数域の設定などをはじめからプログラムの中に仕込んでおく。（デバッグ文の組込み，ドッグタグ）

・プログラム修正を容易にする対策

長年の使用を経た確実な標準モジュールを使用する。モジュールの互換性を考慮する。

モジュールの独立性を高める。モジュールの結合を弱める。

さらにデバッグの手法の細部に立入れば，エラー発見の方法として次のような種類があり，

・機械的方法

全ダンプ（post mortem dump）

スナップショット（snap shot）

トレース（trace）

チェックポイント（check point）など

・論理的方法

制御論理追跡（ジャンプ先，ループ回数）

データ領域の検討（アレイの範囲）

・破壊検査

異常データの入力によるテスト計算

サブルーチンの着脱，交換の試み

言語別デバッグ用チェック・リストの例の一部をひろいあげれば次のような箇条がある。

FORTRANの場合

・配列の宣言を忘れていないか？

・添字なし変数と重複する配列名を使っていないか？

・ゆるされない添字の形式を書いていないか？

・添字の値が実行時に配列宣言の範囲からはみだすおそれはないか？

・GO，TO文の行方に間違いはないか？

・DOループの中で指標パラメータを変えていないか？

・FORMAT文の書きあやまりはないか？ 等々。 其他コンパイル時に検出されない

ALGOLの場合には，beginとendとの対応がくずれると，プログラムの構造がガタガタになり，エラーメッセージの洪水となり，真のエラーの原因を探しだすのに困難であるという基本的問題点がある。構造化プログラミングにも共通する問題点である。

COBOLの場合

事務処理は，一般に入力データの範囲が広いために，正当なデータのみが受けつけられるとは限らない。このことに起因する各種のエラーに対する防止策と修正方法を幾重にも用意しなければならない。データチェックの問題はむしろプログラム設計の課題ではあるが，エラーの問題としてここで検討しておく。

データ設計のチェック・リスト

合理性チェック（reasonableness check）

妥当性チェック（validity check）

リミットチェック（limit check）

照合チェック（matching check）

トータルチェック（total check）

バランスチェック（balance check）

シーケンスチェック（sequence check）

論理チェック（logical relationship check）

桁づれチェック（format check）

自動番号（self－checking number）

抹消記号（delete code）

重複のチェック（duplication error）

n を法とするチェック（module n check）

デバッグについては，このように必要な知識として数えあげられるものがきわめて多岐になるが，これらについても，テストの節で学んだと同様に，心理学的考察が必要になる。

結局，デバッグの最善の方法は，プログラムをよく読んで，そのロジックに意欲と集中力とをそそぎ込むことだけである。悪いデバッグの方法は，「ここが原因かもしれない。こう変えてみて何が起るか見てみよう」という実験的な修正である。しかしながら，人間の習性によって，この最悪の方法が多くの場合おこなわれている。デバッグの学習はこのことを避ける方法の心理学的研究である。

5.4 プログラミング・スタイル

よいプログラムを作るための方策について，色々な議論がなされている。すでに学んだ構造化プログラミングや，モジュールの性質に関する考察も，その一部であるが，ここではさらにブレークダウンした実務的注意事項に関する事項について考えてみる。例えば，Myers は，次のような点をあげている。

① プログラムをわかりやすいものにする。

- 意味のある変数名を使う
- 似ている名前を使わない
- 中間変数を使わない　　等々の原則

② 言語の機能になれる

・トリック的なプログラムにしない

・警告メッセージを無視しない　　等々

③ 効率について

・プログラムが正しく出来あがるまでは効率のことは考えない

・効率向上のために読みやすさを犠牲にしない

・最適化はコンパイラにまかせる

④ データの定義を確実におこなう

・FORTRANのEQUIVALENCE 文のようなものは有害である

・ひとつの変数を複数個の意味に用いない

・特殊な意味をあらわすために変数に特別な値をあたえるような技巧的なことをしない

・表で作動するようなモジュールを使用する　　等々

⑤ プログラム構造を整然とさせる

・最後のＥＮＤで複数のＥＮＤを兼ねさせない

・ＴＨＥＮとＥＬＳＥをかならず対にして使う

・自分自身を書きかえるプログラムにしない

・ローカル変数とグローバル変数のケジメをつける

この例で気づくことは，効率についての考え方にかなりはっきりした個性的な主張があるという点である。ここにあげたものは一例にすぎない。効率的なプログラムを書くことの必要性を強調している例も多い。プログラム作法に関する主張の二三の例をあげて，その中にある思想と，そのよって来たる由来などを研究することは，個々の細部に関する＂How to＂をおぼえることよりも，技術的知識の習得に有効である。

プログラム作成のための中心的な学習項目は，プログラムの中味としてのアルゴリズムの発見方法である。今まであげて来たデバッグの方法とかプログラミング・スタイルの問題は，プログラムの形成であって，プログラムの内容ではない。しかしながらアルゴリズムの学習は，問題分野について，すべて異なると考えられる。実際の教育の場においては，学習者個人にとっての関係の深い問題分野について，問題別のプログラム例題集などを教材にしてアルゴリズムの学習を行うべきである。

（但し，アルゴリズムの学習の項は，一面では，各論第三部の枠からはみだす大きなテーマでもある。）

一般論として，アルゴリズム発見のためのヒントとしてPolyaの「いかにして問題を解くか」などが参考になる。実務的視点では，言語別のプログラム技法集などが教材として有用である。特に，プログラミング・スタイルを論じている際に採りあげてあるプログラムの例題を

議論に沿って熟読することはアルゴリズムの理解に最もよい方法である。

5.5 プログラム作成の日程計画

前章で述べたように，インプリメンテーションとはスケジュールに従ったプログラム開発作業のことである。プログラミング，テスト，要員訓練等の一連の作業を順調に実施するには，実行可能な日程計画が準備されなければならない。ここではその日程計画のたて方を主として学習する。

(1) プログラム開発作業の見積り

システム設計の結果からプログラミングの作業工数を予測するのは通常の場合は困難な仕事である。システム設計に基づいてプログラム設計がかなりの程度に進行して，モジュール分割の具体的目度が立ち，個々のモジュールの規模もわかった段階ではじめて，作業工数の予測がたてられる。しかしながら，この頃には実際の開発作業はほぼ全期間の半ばに達していることが多い。これでは日程をたてても何んのための日程なのか意味をなさなくなる。このために一般には日程の決め方は，完成期日から逆算して日程に各作業を割りふるという方法がとられている。

a) プログラム規模から推定して各々のプログラム作成工数を算出し，その積和にもとづいて日程計画を作る場合

①プログラムの複雑度のランク

②担当者の技能のランク

③担当者の業務知識の程度

④開発のネットの時間を計算する

① × ② × ③

⑤損失因子を見積る

管理業務，文書作成，担当者交代，やり直しの可能性，使用するハードウェアやソフトウェアの不備（デバッグ時間の割当），会議時間

⑥其他の損失因子を加味する

教育，休暇

b) 完成要求期日から逆算して日程を決める場合

この場合に考慮すべきことは，前の方法と同じく，損失時間に対する余裕を十分にとることである。

①プログラム設計

②コーディング及びデバッギング

③個別テスト

④全体テスト

程度の節を設けて日程とするが，さらに厳密にいえば，出力仕様の確定時期，ファイル項目の決定時期，テストデータの準備完了時期等とプログラムの内容に密接した日程の方が望ましい。

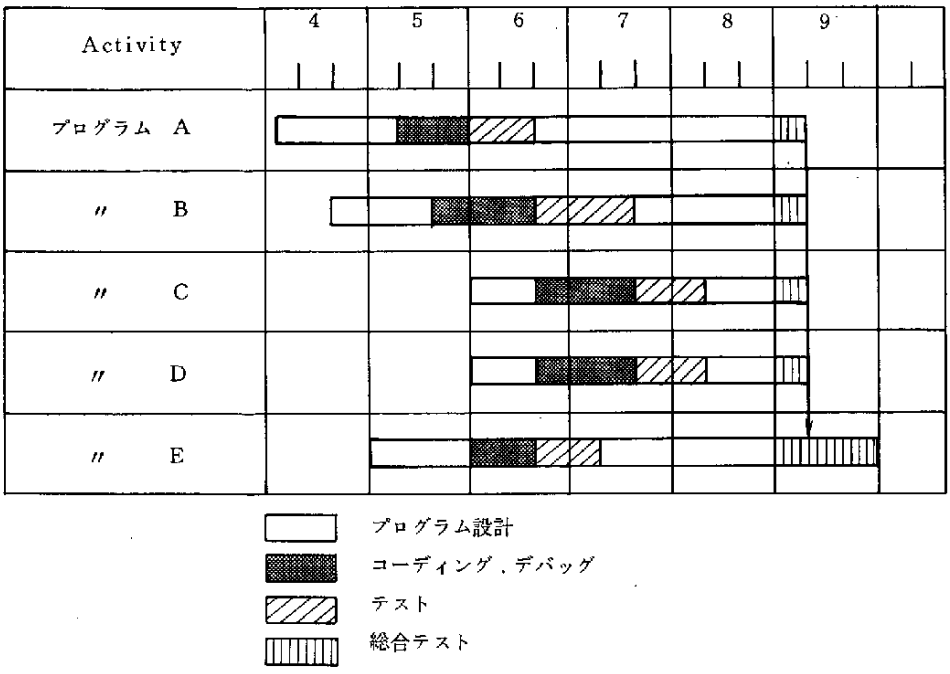


図５－１　ガントチャートによる日程計画

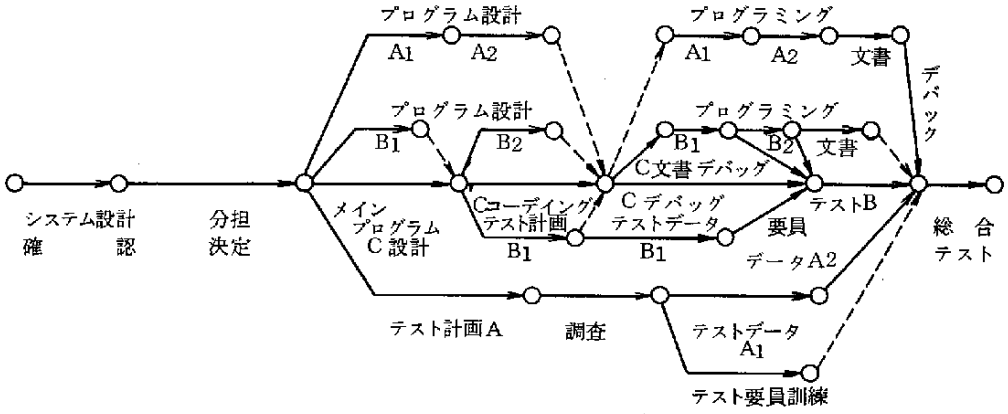


図５－２　ネットワークによる日程計画

5.6 プログラム開発工程上の管理作業

プログラムの開発工程は細分化された作業の集大成である。その中にはデバッグという日程計画を乱す不確定要素を含んでいるので，整然とした管理体系がととのっていないと目的通りのプログラムを完成させることが困難になる。それらの諸管理業務のうちで主なものをとりあげて，その概要を学習する。実務的な詳細については，単元Ｄ４で見学あるいはケース・スタディとして学習する予定であるので，ここではプログラム作成のための支援システムの概要という見方でその全体像をとらえる。

(1) 工 程 管 理

工程管理の基礎は，作業の進行状況を定量的に記録することである。日程計画に対応して工程管理があることを考えると，この両者の表現形式は管理尺度に応じて同一であることが望ましい。日程計画の書式として，

・ガントチヤート

・ネットワーク

の二形式があることはすでに学んだが，工程管理の記録も，これに対応したものがよい。

モジュール別総括スケジュールに対して開発状況一覧表

プログラム別週間作業予定に対して作業進捗状況表

などが使用される。

(2) 作 業 管 理

作業管理は工程管理の細目の一つである。文書や記録にもとづく管理が望ましい。作業管理を実行するための書類にも「開」と「閉」の対応が整っていることが望まれる。即ち

作業指示書－報告書

作業予定表－作業日報

作業管理の記録は，その後見積り資料としてフィードバックされて，次のシステム開発計画の参考となる。

(3) 品 質 管 理

プログラムの品質管理は，制度的にはプログラムのテストの問題である。デバッグのためのプログラムテストやプログラムの性能評価のテストについては幾度か検討した。ここではすでに学んだことをふりかえって，次の２つの着眼点について整理してみることにする。

a) 与えられた仕様を満足していることを検査するにはどんな方法があるか？

b) プログラムの標準化による品質の向上方法としてどんなものがあるか？

(4) 費 用 管 理

プログラム開発の工程管理で難しい問題は開発費用を適正な予算額の中におさめる管理で

ある。例えばコンピュータの使用時間がデバッグの見通しの悪さによって，予定を超過しても，その時点でコンピュータの使用を中止させるわけにはいかない。費用の予定枠の中に仕事を収めるための正確な情報に乏しいことが多い。

開発コストの要素のうちコンピュータ費用については，実績値の集計計算が自動化されるようになったため次第に正確に予測できるようになったが，人件費のうちで，プログラム開発期間の予測は最も予測しにくい部分である。

費用管理について大切なことは，開発コストの推定だけでなく，運用コストの推定も大きな問題である。実際的な管理業務としては開発費，運用費の推定だけでなく，利用部門への費用の振替などをするために，標準単価の設定や操業度，実績などの計測，集計などをおこなう必要がある。これらの作業には，費用管理という観点からと，作業管理という観点が二重に絡みあっている。

(5) プログラムの検査

プログラム開発の各段階の完了は，プログラムの検査によってけじめがつけられる。前に述べたデバッグのためのテストは，むしろプログラム作成の途中に於ける手段であったが，ここでとりあげるテストは確認とか証明とかのための方法であって手段というよりも目的に近い性格のものである。

検査はその性格上第三者の手による確認を原則としなければならない。プログラム検査制度を定めることは，プログラム開発技術の向上をもたらすための礎石になる。制度の意味には法律と類似の性質がある。制度の正常な運用には検査規準が定められなくてはならない。次のような検査規準が考えられる。

① モジュール検査規準

② コンポーネント検査規準

③ 総合テストの検査規準

(6) システム切換

プログラムの総合検査を終了した段階で，システムの運用管理の責任はシステム開発グループから，運用実施のグループあるいは利用者のグループにひきわたされる。この引継ぎ作業はドラスティックに期日を決めて品物を手渡すようにはおこなわれない。システム分析の単元でも学習したように，いくつかの段階に分割して順次移行されるのが普通である。プロジェクト管理の理想からいえば，ターンキー(turn－key)プロジェクトとして品物を引渡すようにひきついで，プロジェクトチームもその期日をもって解散できるようなシステムを作ることが出来ればそれが理想ではあるが，システムは通常人間とソフトウェアとハードウェアの集合体でなりたっている。特に人間とマシンの間のトレードオフの問題は逐次的な順

応が最も大切である。システム切換のための平行作業の計画がシステム開発の仕上げである。

(7) 要員の教育訓練

システムの運用開始を順調におこなう礎石として関係部門の参画が特に大切である。そのシステムが本質的な需要に根ざすものであって，且つシステムそのものが要求に合致していれば，始めの不馴れによる障害が除かれるのは時間が解決してくれるものではあるが，はじめから見知らぬものを急におしつけられる場合と，実施に入る前から熟知していて開始の命令の時期を待っていた場合とでは，稼動直後の評価も違う。運用費そのものにも実質的な差が生じる。

関係部門又は実施運用部門の参画協力を得るための方法は，直接要員に対する教育訓練である。特に運用テキスト（user's manual）の作成には運用者自身にその作業に協力してもらうようにすることが望ましい。プログラム設計上のエラーや不備を発見する緒ともなって，最も望ましい結果を生む。

(8) 文 書 管 理

システム開発の基礎として，文書管理が重要な役割をはたしていることは既に学んだが，実施運用の面では，文書はさらに重要な役割をもっている。運用面の能率化という視点でプログラム関係の文書がどうあるべきかを検討してみる。標準的に準備されるべき文書として次のようなものが考えられる。

① 業務の概要（システム説明書）
② プログラム運用マニュアル（プログラムの使い方）
③ 運用マニュアル補足（その他の共通ユティリティ）
④ ファイル保守マニュアル
⑤ プログラム保守マニュアル

指導上の留意点

(1) プログラム設計書にもとづくプログラム作成の学習である。実務的，経験的知識の整理であるので，採りあげ方いかんでは際限なく学習項目が多くなり，木を見て森を見失う。プログラム・テストとデバッグに焦点をあてて教示し，その他は全体のスケッチをする程度にとどめるのが，教育時間の配分上から適当である。

(2) 単元Ｃ４との関連が深い。Ｃ４ですでに理論的検討のすんでいる項目は省略し，主としてプログラミング技術としての経験的原則などを中心にして指導する。

(3) プログラム作成上，アルゴリズムの発見と，アルゴリズムの改善は重要なテーマであるが，この問題をとりあつかうことは，単元Ｄの枠組の中に入れるには大きすぎるテーマでもある

ので，ここでは省略する。受講者のもつ問題分野に応じて，教材として特定のアルゴリズムを採りあげて，これを詳細に検討することは望ましい。

(4) プログラム作成上，必要な予備知識として，コンピュータの使用法に関する各種の専門的知識があるが，これについても省略する。単元Ｄ３での演習で，準備作業として或程度の学習をおこなうことになるであろう。

(5) プログラミング・スタイルに関する討議は，或程度のプログラミングの実務経験を持ったものに対して有効な学習テーマであるので，プログラミングの実務経験の少ない学習者に対しては，削除してもいい。逆にプログラミング経験者の場合には，重視すべきテーマである。

(6) 管理作業の話題は，教育時間に制限がある時は全部削除してもよい。単元Ｄ４でとりあつかう。

参　　考

(1) structured walktruについて

ＩＢＭ社が提唱しているプログラム開発技法（ＩＰＴ)は複数（多数）の人員によるプログラム開発を念頭においた技法である。（structured programmingの論争に刺激されて，その影響も強く受けているが，Dijkstra，Wirthなどのねらいと異質な面が多い）Mills，Baker Constantine，Myers等の多くの人々が発案し，改善を加え現在にいたっている。その内容は，

a）チーフ・プログラマ・チーム

b）ストラクチャード・コーディング

c）トップ・ダウン・プログラミング

d）開発支援（ライブラリアン）

等の各種の技法の集合体である。その中に上記のstructured walktruも含まれている。

これは，プログラム検査のための一方法であって，次のような原則が用意されている。

① レビューをうける立場の人が主催し，問題点を指摘出来ると思われる人を４～６人選定する。

② 管理者は，その会議には出席しない。プログラムの品質向上をねらうものであって，評価はしない。

③ レビューアは会議の前にあらかじめ資料を与えられその内容を理解しておく。

④ 会議は本番前のリハーサルの性格をもつ。

⑤ エラーの発見を中心にする。エラーを文書化する。エラーの訂正は walktru ではおこなわず，その後の別の局面で訂正案を検討する。

⑥ walktruの時間は短時間でおこなうように心掛ける。

walktruのねらいは，複数者のチェックによる品質の安定化であり，エラーの早期発見によるシステム開発費の削減である。またこの会議はプロジェクトメンバーに対する動機づけともなり，すぐれた設計とはどんなものであるかという点を学ぶための適切なチャンスを与える。

(2) テスト支援ツールについて

自動化されたプログラム分析用の各種プログラム・パッケージが実用化されている。その種類を分類してみると次のようなものがある。

a) ソースプログラムの静的分析用

- コード分析
- 変数分類表（ダンプリスト）
- プログラム構造チェック
- 流れ図自動作成
- モジュール・インターフェース・チェック
- イベント・シーケンス・チェック（トレース）

b) プログラムの動的分析用

- モニタ
- テストケース自動生成
- アサーション・チェック
- ソフトウェア・ディフェンス

c) 保　守　用

- 修正による影響範囲指示
- 文書化，図式化ツール
- 修正確認
- 改善点予測

d) 評　価　用

- アカウンティング
- 性能検査
- 信頼性計算
- ディシジョンテーブルによるテスト条件整理

参考文献

〔1〕 Polya, G.: "How to Solve It," Princeton Univ Press, 1971

〔2〕 Kernighan, B.W, and Plauger, P.J.: "The Elements of Programming Style,"

Mc Graw-Hill, 1974

〔3〕 Weinberg, G. M. :"The Psychology of Computer Programming",Van Nostrand Reinhold , 1971

〔4〕 Myers, G.J. :"Reliable Software Through Composite Design", Petrocelli/Charter, 1975

〔5〕 Hetzel, W. C (Ed.) :"Program Test Methods",Prentice-Hall, 1973

〔6〕 Gruenberger, F. : "Program Testing and Volidation", DATAMATION Vol.14, No. 7. 1968

〔7〕 Drummond, M. E. :"Evaluation and Measurement Techniques for Digital Computer Systems, Prentice-Hall, 1973

〔8〕 Shooman, M. L. and Bolsky, M. J. :"Types, Distribution, and Test and Correction Times for Programming Errors," Proc.of 1975 ICRS

〔9〕 Weinwurm, G. F. :"On The Management of Computer Programming", Auerbach,

〔10〕 Wirth, N. :"On the composition of well-structured Programs",Computing Surveys 1974

〔11〕 G.F. ウェンワルム著（吉原賢二訳）「ソフトウェア生産技術」学研, 1976

〔12〕 J.T. ゴールデン著（戸田英雄訳）「FORTRANプログラム技術」1974

〔13〕 ハーバード・D・リーズ, ジェラルド・M・ワインバーグ著（藤村純, 山門四十夫訳）「電子計算機プログラミングの基礎」改訂版, 産業図書,

〔14〕 ブライアン・W・カーニハン, P.J. ブローガー著（木村泉訳）「プログラミング書法」共立出版, 1976

〔15〕 Niklaus Wirth著（野下浩平, 他訳）「系統的プログラミング入門」近代科学社, 1975

〔16〕 半沢孝雄「プログラミング技法集」竹内書店,

〔17〕 MYERS著（久保未沙, 他訳）「高信頼性ソフトウエア」近代科学社, 1976

〔18〕 G.J.Myers著（有沢 誠訳）「ソフトウエアの信頼性」近代科学社, 1977

〔19〕 J.D. ワーニェ著（鈴木君子訳）「ワーニエ・プログラミング法則集」近代科学社, 1975

〔20〕 HETZEL編（鳥居宏次訳）「プログラム・テスト法」近代科学社, 1974

〔21〕 林, 五十嵐「プログラムの検証理論」情報処理, Vol.17, No.5

〔22〕 Zohar Manna著（五十嵐滋訳）「プログラムの理論」日本コンピュータ協会, 1975

〔23〕 ソフトウェア信頼性専門委員会報告書「ソフトウエア信頼性に関する調査」日本電子工業振興協会，1977

〔24〕 西谷真二，疋田英雄，秋葉二郎，八柳祐久「リアルタイム・システムズ・マネジメント」企画センター，1972

〔25〕 南条優，保坂岩男「システムの開発と運用」オーム社，1972

〔26〕 大日方真「プログラム開発の工程管理」情報処理，Vol.12～Vol.13.

第6章　システムの評価と改善

キーワード

開発完了後の分析（post implementation analysis），達成率(achievement)，総合判断(synthetic decision)，開発費見積り(costing of system development effort)，運用費見積り（costing of system performance），システムの寿命（system usage life），システム評価（follow-up and evaluation），稼動実積（progress records and reports），成長曲線（growth curve），習熟曲線（learning curve），長期展望（long-range forcasting)，システム監査（auditing system performance），システム改善（tuning）

目　　標

システム・ライフサイクルの最終局面に該当する項目を学習する。即ち，システム開発の完了した時点以降に発生する主要な課題である最終的なシステム評価，開発作業に対する反省，システム実施上の注意，システム保守技術，改善点の発見などである。

これらの活動は，狭義のシステム・ライフサイクルとは別次元の問題であるとする考え方もあるが，システムを生命ある実体としてみる時，忘れてはならない問題である。

システムの実施と保守まで含めた広義のライフサイクルを考えると，システム開発中の問題の研究よりも，システムが運用に入ってからの長期間の観察や，保守技術の考察の方がより大切な研究テーマである。

開発完了後のシステム評価や，開発中の経験の反省などは，再び次のサイクルを廻すための原動力であり，次のシステム分析のための資料である。

システム改善点の発見や，保守技術は，次のリサイクルに対する連結鎖である。即ち，保守改善の作業は，場合によっては，Ｄ１とＤ２を通して学んで来た事項を再び始めから繰り返すことになる。

このように広義に展望した時のシステム・サイクルの構造を検討するとともに，システムの寿命についても考察する。

6.1　最終段階でのシステム性能の評価

システム設計に基づいて，そのプログラム作成とテストを行い，デバッグも完了した時点で，システム開発は一つの区切りに達したことになる。この時点ではシステム設計で意図したもの

が，その実行によって達成できたかどうかを確めることができる。即ちシステムの性能の評価である。コンピュータの応用は，自然発生的に人間社会にひろまって来た社会制度や会社組織などと異なって，人為的人工的なしくみがその本質に含まれている。そのために一方で「思ったほど効果があがっていない。」という評価があるかと思うと，他方で「短期的効果はともかく，長期的な尺度で広く合理化に役立っている面は大きい。」という評価もあるという状況になる。システム評価については考えるべき点が多い。システムの採算性などの判断には減価償却の時間の問題も判断要素に入れなければならないのでむずかしい問題である。

システム稼動後の追跡評価という名目で，省力結果の集計表や，費用削減成果などが，システム開発者の手で自己保全・自己宣伝用に作られることがあるが，これらのものはシステム評価と呼ぶべきものではない。

システム性能の評価は，Ｄ１の５章及びＤ２の２章ですでに学習したような評価項目に対して，作りあげたものが，どの程度に合致しているかという反省である。これらの内容の復習をおこなうとともに，ここでは主として評価のための手続きの全体を把握する。

システム性能の評価の役割は，システム・ライフサイクルの終了時点で改善資料を収集し，それをシステム・ライフサイクルの開始時点に引きついで，再びサイクルをまわすことによって，システムに再生の活力を与えることである。即ち，１サイクルのつなぎめにある評価活動であるが，この種の活動は，１サイクルの中でも，各局面のつなぎめでその都度おこなわれるものである。その概略をふりかえってみると次の図のようになる。

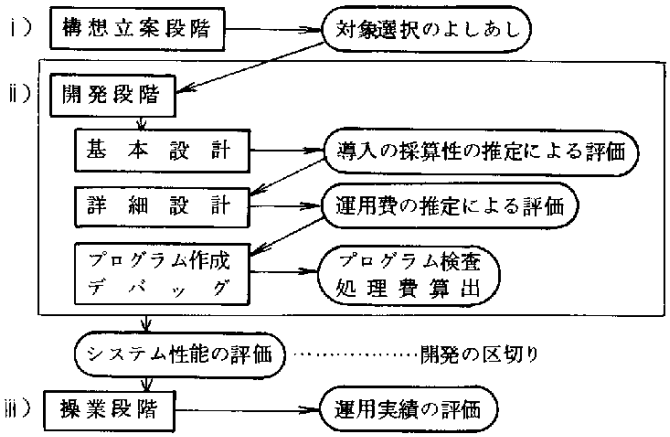


図６－１　システム開発の各局面での評価

システム開発の初期では予測による開発の適否に対する判断の性格がつよく，後期になると実測による性能の確認の性格がつよくなる。

いずれの局面でも，評価の手続きとして大切な要件は，ユーザと開発部門の両者の合議によ

る評価である。開発開始時点ではユーザ要求を正しく把握することが大切であるし，運用時点では，ユーザにとって不便・不都合になっていることを見過してはいないかという注意が大切である。

システムのもっている妥協的な性格を考えてみれば，開発側とユーザ側とは当然利害が対立することが多い。開発側が一方的にシステムの評価をすれば，システムの不備に目をつぶり，不具合の責任をユーザの無知に転嫁してしまうことになり，システム改善に対して自己防衛的になる。こうなればシステム評価の真の目的は達せられない。

システム評価の手続きの検討については，細かい評価項目のリストをつくるよりも，ユーザと開発側の合議制をどうやって運用するかを議論する方が重要である。

システム評価の結果，新システムの実施が決定された場合，新システムのスタートの準備がおこなわれる必要がある。システムの規模によってこの準備はさまざまである。

これらは実施開始直後のシステムの稼動実績の評価と並んでおこなわれることが多い。主な項目をひろいあげてみると，

① エラーのフィードバック

種々のエラーがある
- プログラムの起因によるもの
- オペレータの不馴れによるもの
- 利用者の感ちがいによるもの

② 訓練の開始
- (a)一般説明会
- (b)利用者訓練
- (c)プログラマ訓練
- (d)オペレータ訓練

③ サブシステム開発（不足部分，改正部分）

④ ファイルの変更，プログラムの変更

⑤ 運用計画・暫定計画から定常的計画へ

⑥ 設備・能力等の再検討

これらの稼動実績のデータが集録され，初期の訓練や教育が終了し，マニュアル類の整備がととのった時点で，システムの運用は利用者あるいは運用担当者に引渡される。これによってシステム開発の一つのサイクルが完了したことになる。システム開発がプロジェクトチームでおこなわれていた場合は，プロジェクトは解散になる。

しかしながら日本での実情は一般には，このようなターンキイプロジェクトとして仕上げて引渡すということは少なく，実施への移行も，仕上がったところから順次移行の形をとるので，何時そのシステムが出来上ったのかも明らかでなく，したがって開発費もつかみにくい。ここ

で学習することとしては，このような日本における実情の下で，実施に移行する場合の長所短所をできるだけ多く洗いだしてみてそれについて議論をしてみることが，単元D２全体のしめくくりにもなる。

システム開発が終了すると，開発のために編成されたプロジェクトチームは，解散することになるが，そのメンバーが開発の経過をふりかえって作る総括的報告(post-mortem report)はぜひとも必要な文書の一つである。これは将来のプロジェクト計画に対する貴重な資料となる。次のような項目について書かれる。

- 開発中に遭遇した困難
- 計画と実績の相異
  - 日程
  - 費用
  - 工数
  - コンピュータ時間
- 開発中の重要な事件

6.2　システムの保守

システム開発が完了すると，通常はいくつかの実施移行前の検査を経て，運用がユーザ側にひきわたされる。その後数年間の運用を経て，次の新しいシステムに切り変えられる。この運用の間にも，そのシステムに改善や変更が加えられないということはありえない。システムをとりまく環境はたえず変化している。たとえシステム開発にとりかかる前のシステム分析が完全であっても，システムに変更を加えなくて運用ができるということはありえない。環境の変化を無視することは，そのシステムの開発にかけた費用の償却もすまないうちに，システムを放棄することを意味する。改善，改修を加えながら，運用期間を延ばしつつ，最後には改修不可能という見切りをつけるのがシステムの寿命である。

変化に応じてシステムに改善改修を加えることが，システムの保守である。

前に学んだプログラムのデバッグと異なる点は，デバッグの場合は，完成する以前に誤りを発見し，それを正しい姿に直す作業であるが，保守の場合は，すでに完成した後に，何らかの必要によって，システムに変更を加えるという事である。

しかしながら現実問題としては間違いのないプログラムはありえないことであるので，実用期間に入ってからでも，誤りは断えず発見される。これに対する補修工事のことも保守（corrective maintenance）と呼ぶ。

環境の変化に追従するための保守（adaptive maintenance）には，その内容によって，次のような二つの場合がある。

(1) 運用条件の変更による保守

価格体系変更－集金業務

製造工程変更－工程計画，生産管理

賃金制度改訂－賃金計算

これらの例では，変更内容が規定されるので比較的単純な保守となる。

(2) 効率改善，操作性改善のための保守

ターミナルの待時間が長くなった。

例外処理が多くなり，手作業が増加しつつある。

不必要なレポートを廃止する必要が生じている。

このような第2の場合は，その改善を実行しなくてもシステムは使用に耐えうるため，その変化の発見におくれを生ずることが多い。システムの評価で大切なことは，この第2の場合をどうやって発見し，保守をおこなうかである。

いずれの場合にも，システム運用中の全期間を通じて，保守をうけもつ専任の担当者をもうけて，ユーザの声を正しく聞くためのアンケート，稼動状況を把握するためのシステム・モニタリングなどをおこなうことが大切である。基本的には問題を的確に認識し，原因を究明する熱意と根気をもつことが重要である。

表6－1　保守作業の分類（Swansonの分類）

① 誤りの訂正（corrective maintenance）	
誤りの種類	processing failure performance failure implementation failure
② 環境変化に追従（adaptive maintenance）	
環境変化	change in data environment change in processing environment
③ 効率の向上（perfective maintenance）	
効率向上	processing efficiency performance enhancement maintenability

システム開発の期間中の問題よりも，システムが運用に入ってからの保守作業の問題の方が，より多く関心がはらわれなければならないという主張が強くなったのは，最近の傾向である。

保守の問題はさけることのできない問題であるとすると，保守技術は重要な研究テーマである。さらにさかのぼって考えれば，システム設計の考え方として，保守性のよいプログラムの設計を検討する必要がある。保守性のよいシステムの条件として次にあげるようなものがあげられる。

a） 読みやすい仕様書あるいはプログラムを作る。

b） 拡張性のあるプログラム構造とする。

c） 改修による影響を少なくする。

d） 誤りの範囲を局所化する。

e） 改修の履歴を明記する。

等のシステム設計上の考慮が必要である。さらにこのことを具体的に述べれば，

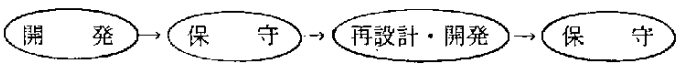
(i) 文書について，記述がformal であり，プログラムと説明書が一致していて，範囲や履歴や代替案などが明瞭にわかるようにしてあることが必要である。

② 設計について，誤動作が外部にあらわれやすく（早期発見），機能範囲が局所化されていて（モジュール化），自動回復やフェイルソフトの機能がある（冗長度設計）ことが必要になる。

③ プログラミングについて，ソースリストが読みやすいスタイル(構造化プログラミング)であることが要求される。

即ち保守の問題は，保守の局面に留まるものではなく，結局はシステム設計の出発点に回帰することを認識させる必要がある。

システム・ライフサイクルを単純化して図式化すると，



というサイクルになる。

保守技術という点でさらに細目に入って検討すると，保守作業用の各種のツールが開発されている。即ち

① パッチ適用のためのツール

・ソース言語レベルでのパッチデータが自動生成されるもの

・パッチ適用の履歴が残るもの

② プログラム再編成のツール

・システム・パラメータの動的変更

・システム生成をまたずにモジュールの追加削除ができるもの

③ 文書変更のためのツール

・テキスト・エディタ

④ 変更確認のためのツール

・変更が及ぼす影響範囲の検出

これらの保守技術用ツールは，場合によってはＯＳの機能の中に組込まれているものや，データベース管理システムの機能となっているものもある。あるいはユーティリティ・パッケージとして開発されている。このような形での保守技術の整備は今後ますます重要性が認識されるとともに，システム開発技術の中に組込まれる傾向にある。

さらにシステム保守の重視の傾向は，プログラム変更に関する情報の保存とその参照の手続きの確立をもとめる方向に進むこととなる。即ち今後の課題として，保守データベースの整備が重要になる。

6.3 システム評価のまとめ

システム性能の評価とは，その機能と費用との割合を評価基準に照らして判定することである。機能を分子とし，費用を分母とする評価項目の集合がシステム性能である。したがって評価の方法を研究する前提として，評価項目，評価基準，費用などの要素の考察をまずとりあげてみる。

(1) 評価項目と評価基準

第２章で学習したシステムの有効度の概念（定量的評価基準）なども，評価基準の一つのマクドノウ（Adrian M. McDonough)は一般的評価基準用語として次のようなものをあげている。

・精　　度
・接 近 性
・依 存 性
・有 効 性
・効　　率
・伸 縮 性
・正 確 さ
・信 頼 性
・敏 感 性
・融 通 性

これらの用語に対して，修飾語を附加すると意味のある評価項目になる。例えば，

・需要予測の精度

・設計作業者への接近性

・既組織への依存性　　等である。

これらの評価項目を決めた後に，これらの評価項目をキーワードにしてシステムのねらったものが総合的にどのくらい達成されているかを判定するのが，システム性能の評価である。

一般的な評価項目には信頼性，保守性などの技術的な視点からのものと，開発費（イニシアル・コスト），運用費（ランニング・コスト）などの経済的視点からみたものがあり，さらに有効性，効率などのように費用効果比として係数化した概念など，いろいろな要素が混ざりあっているので，評価項目を数多くあげると，その中には一つの項が他の項と従属的な関係にあるものが含まれて，総合判断の資料として不適切になることもある。

費用を多く投入すれば，機能の高いものが得られるのが自然の理である。機能と性能とは区別して考えなければならない。システムの性能評価は，機能と費用のバランス点を測定する方法である。コンピュータ自身の性能評価法について，ギブソンミックス法→アウエルバッハ法→SCERTプログラムと，その評価方法が精密化して来た原因も，単に機能が高度であればよいというのでなく，総合的にみて，仕事に対する適応の度合をみきわめる必要があるという認識が高まったためである。

(2) 費　　用

この章でとりあつかうシステム評価は，第２章で検討し計画した資源選択に対する実証と，その予測値からの変動の割合の計測である。見積りに対する実績の対比である。ここでの評価尺度は終局的には費用の問題である。

費用の内訳は大別して

(a) 開　発　費

(b) 運　用　費

開発費は開発期間と開発要員数で大綱が決まる。運用費は関与部署のひろがりとデータの量で大綱が決まる。

システム性能評価は投資効果の判定であるので，或る意味で企業診断のような複雑さをもつことになる。

システム性能はそれが使用されると予想される将来の期間に渡って評価されるべきである。そのために評価項目には保守の容易さとか拡張の可能さも考える必要があるし，採算性の計算に対しては，開発費をどのようにして減価償却するかも考えなければならない。使用するコンピュータの更新，プログラムの手直しにつぐ手直しなどが繰り返されると，採算計算の分母が不明確になるきらいがある。買取りのコンピュータに，機能を限定したプログラムを固定し，その全体を設備投資と考えて，これに対してシステム性能の評価を明確にするとい

アメリカ流の行き方と，ハードウェアもソフトウェアも次々に取り替えて行く日本流の自転車操業とがあって，一長一短がある。

以上述べたことを数式化すると次のようにあらわされる。

- 評価項目を決める（Ai）
- 評価基準に照らして項目毎の達成度を判定する（αi）
- 評価項目毎に費用を算定する（Bi）
- システム目的に対して評価項目の比重をきめる（Wi）

$$\text{システム性能} = \sum^{i} \frac{\alpha i \cdot Ai}{Bi} \times Wi$$

ここでいう評価項目は，一般評価基準用語であらわされるような抽象的なものでもよいが，むしろ現実的には，そのシステムがはたそうとしている業務の中の個別業務や，サービス機能などの方がより具体的に把握できるし，その項目に必要な費用も分離して算定出来る。この評価方法で大切なことはBiを分離することであって，$\frac{A_1+A_2+A_3+\cdots\cdots+An}{B}$ のような費用のドンブリ勘定のシステムは評価をすることの意味が少ない。

(3) 評 価 対 象

評価の方法には，はじめにあげたような，チェックポイント・リストを確認する方法，評価項目別に機能と費用の比をみくらべる方法などいくつかの視点があるが，評価の目的が単なる良否の判定ではなく，改善のための足がかりを発見するためのものであるとすれば，その目的に適した評価方法がある。即ち評価対象を，ハードウェア／ソフトウェア／人と区分にて評価する方法である。この方法の利点は次のような点である。

(a) もしハードウェアに欠点が見出されれば，ハードウェアを変更することによってシステム性能の向上がはかれる。

(b) ソフトウェアの場合も同じである。プログラムのモジュール化やデータからの独立が出来ていれば，システムの部分改善は容易である。

(c) 人の問題は色々な場合がある。開発メンバーの能力が不足であれば，増強や交代をすればよい。システム性能を最終的に評価するには「人間の問題」をよく見きわめなければならない。予期した効果があがらない場合，ハードウェアにも，ソフトウェアにも欠陥がなく，運用にたずさわる人々の教育に落ちがあったり，利用者の教育に手ぬかりがあることも多い。開発したプログラムが，開発した自分の会社では使用されずに，そのプログラムを買って使用しているところで十分活用されている例もある。

ハードウェア，ソフトウェア，人間と個々の対象に対しての評価による方法はすでに学んだように，各種の評価方法も開発されている上に，システム全体の性能評価に，このようなシ

ステム要素を個別に評価する方法は，従来から採られているアプローチなのでここでは特に学習テーマとしては考えない。

(4) 評 価 体 制

評価とは終局的には総合判断である。評価項目を厳密に並べあげ，そのデータに関する各種の作業を事務的につみあげれば，帰納的に結論が得られるというものではない。先に数えあげたチェックポイントをしらみつぶしに消し込めばいいというものでもない。システムの目的を決める場合にも，利用部門の要求をそのままうのみにしていいとも限らない。逆に要求の妥当性に大きな欠陥もありうることに注意する必要がある。R.L. Ackoff の5つの誤った前提（Management Science Vol.14 No.4 1967)は逆説的にこの点を指摘している。

(i) マネージメントには，より多くの情報が必要である。

(ii) マネージャーは自分の必要とする情報が何であるかをよく知っている。

(iii) マネージャーに彼の欲する情報を提供してやれば意思決定はよりよくなる。

(iv) コミュニケーションがよくなれば業績はあがる。

(v) マネージャーは情報経路をよく知らなくてもよい。その使い方さえ知っていればよい。

システム性能の評価の実施には，評価のための組織や，体制を整え，その体制を維持するためにも様式をととのえる必要がある。

しかしその場合に注意しなければならないことは，Ackoff の指摘するような錯覚をおかさないようにするということである。システム性能の評価という名のもとに，いたずらに形式化した無用の管理機構を附加しない方がシステム開発が順調に進むことが多いし，とりたてて中途半端な評価をするよりも，使用による自然淘汰が一番きびしい評価であることを忘れてはならない。下手な評価をするよりも，下手なシステムを利用者に強制しないことの方が大切である。

6.4 システムの寿命

システムをその運用の全期間にわたって監視するには，システムの成長の一般傾向について考察する必要がある。

成長過程を示す現象の定式化には成長曲線（growth curve）というとらえ方がある。マルサスの人口論を背景にしたVerhulst のロジステック曲線（logistic curve）や，人間の老化作用と抵抗力に着目したGompertz のゴンペルツ曲線（Gompertz curve）は成長過程を波形的にとらえた定式化の一種であるが，これらは経済発展の一般的な傾向をあらわしている。システムの運用状況の時系列モデルとしても，これらの表現は通用する。成長曲線の性格を，図6－2のように試験期，成長期，飽和期，安定期とわけて，それぞれの時期に対する対策を

論ずることが，おこなわれている。

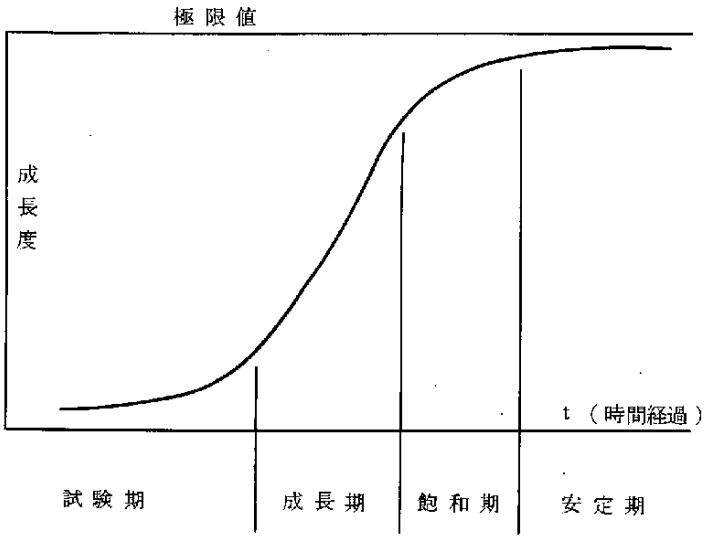


図6－2 成 長 曲 線

ロジステック曲線　$Y=\frac{K}{1+me^{-at}}$

ゴンペルツ曲線　$Y=Ka^{bt}$

システムを開発時期だけでとらえずにさらに長期的な視野で時系列での運用全体をみると，このような波形的なシステム・ライフの把握も考えられる。システムの実施状況は，何時のまにか根をはって拡大しているもの，何時のまにか忘れられ次第に使われなくなっているものなどがあり，むしろこのような成長曲線で把握される形が適当なものも多い。さらにつけ加えれば，システム運用上での習熟曲線（learning curve）や，システムの寿命の問題などの考察も望ましい。

同様な考察として，マーケット・リサーチの分野で，製品のライフサイクルに4段階或いは5段階の経過を踏むという説がある。例えば，E. J. KelleyやE. J. Marthyの4段階は次の図のようにあらわせる。

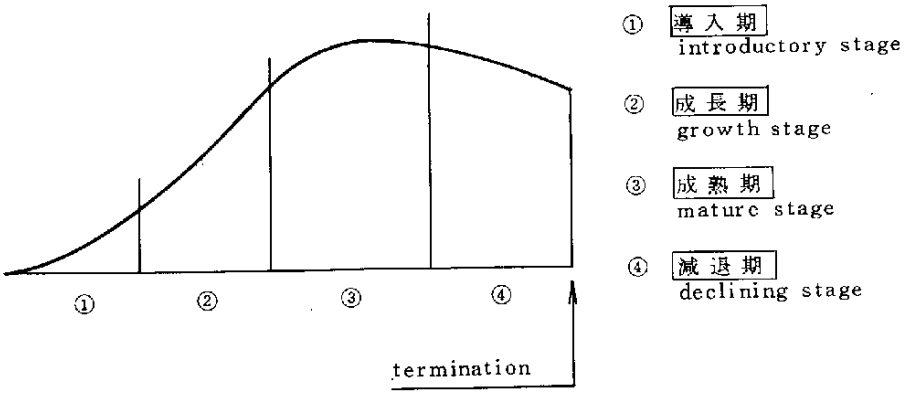


図6－3　製品のライフサイクル

同じような考え方でE. A. Pessemier, D. J. Luck, C. R. Wassonらは次のような5段階を考えている。

- 導　入　期　introduction period
- 成　長　期　growth period
- 競　争　期　competitive period
- 陳腐化期　obsolescence period
- 終　末　期　termination period

システムそのものも製品である。今まで学んだことがらは主として製品になるまでの過程におけるライフサイクルであったが，今後考えるべきことはむしろ，製品としての寿命の問題である。そのためにはここに例示したようなマーケット・リサーチの分野で研究されているような考察もまた必要である。特に不要になったシステムや陳腐化したシステムをどうやって始末するかは今後に残された大きな問題である。

指導上の留意点

(1) 単元D２のしめくくりとして，システムを全体的視野でとらえて反省してみる。

(2) システム開発完了後のシステム評価は，再び第２ラウンドのシステム開発点の出発点であるという形で，閉じた円としてのシステム・ライフサイクルの性格を説明する。

(3) 単元D１はシステム分析，単元D２はシステム設計をとりあつかっている。これらはシステム開発の主要活動であるが，このほかに関連事項として，各種の管理活動，検査活動，実施移行のための要員訓練，システムの実施，システム保守などがある。このような全体の構成を鳥瞰するとともに，その中の主要関連項目を学習する。

(4) システム評価の復習として，最近重視されつつあるシステムの品質評価，特に信頼性の問題に焦点をあてて学習することが，望ましい。

(5) 結びの章であるので，システム開発に関する技術展望など解説するのもよい。

参考文献

〔1〕 Boehm, B. W. et al, "Quantitative Evaluation of Software Quality", Proc. 2nd ICSE, 1976

〔2〕 Boehm, B. W., "Software Engineering", IEEE Trans. on Computers, 1976

〔3〕 Slaughter, J. B., "Understanding the Software Problem", NCC, 1974

〔4〕 Swanson, E. B., "The Dimensions of Maintenance", Proc. 2nd ICSE, 1976

〔5〕 Bucher, D. E. W., "Maintenance of the Computer Sciences Teleprocessing System", ICRS, 1975

〔6〕 Alberts, D. S., "The economics of software quality assurance", NCC, 1976

〔7〕 ソフトウェア信頼性専門委員会「ソフトウェア信頼性に関する調査」日本電子工業振興協会，1977

〔8〕 HETZEL編（鳥居宏次訳）「プログラム・テスト法」近代科学社，1974

〔9〕 宮本勲，東谷秀夫「ソフトウェア・エンジニアリングへの道」bit Vol.9. & Vol.10. 連載 共立出版，1977

単元Ｄ３

開発プロジェクト

単元Ｄ３

開発プロジェクト

目　　次

第１章　テーマの選択 …… 231

1.1　実在組織の課題 …… 231

1.2　仮想プロジェクト …… 232

1.3　課　題　例 …… 236

1.4　課題の作成 …… 238

第２章　チームの編成 …… 242

2.1　チームの組織 …… 242

2.2　チーム活動の意義と学習目的 …… 243

第３章　実習の進め方と教師の役割 …… 245

3.1　ケース実習の特徴 …… 245

3.2　必要な教育技術 …… 246

3.3　実習の推移の特徴と教師の対処 …… 246

3.4　実習を進める上での各種の手段 …… 247

3.5　実習中に学生が経過する段階 …… 249

第４章　開発の実習 …… 251

4.1　問題の定義 …… 251

4.2　システム開発の組織 …… 253

4.3　現状分析 …… 253

4.4　将来予測 …… 254

4.5　システム分析 …… 254

4.6　企画書，提案書 …… 254

4.7　システム開発のスケジュール …… 255

4.8　システム代替案の評価 …… 255

4.9　ハードウェアの定性的評価 …… 257

4.10 ハードウェアの定量的評価 …… 258
4.11 システムの詳細機能決定 …… 258
4.12 システムのフィジカル設計 …… 259
4.13 インプット設計 …… 259
4.14 ファイル設計 …… 259
4.15 プロセス設計 …… 259
4.16 アウトプット設計 …… 259
4.17 セキュリティとイングリティ …… 259
4.18 システム開発の工程管理 …… 259

第5章 報告書作成 …… 306
5.1 文書報告 …… 306
5.2 口頭発表 …… 306
5.3 視聴覚機材の利用 …… 307
5.4 発表及び討論の評価 …… 308

第1章　テーマの選択

キーワード

仮想プロジェクト（hypothetical project），ＯＪＴ（on-the-job training），ケース・メソッド（case method），フィールド・リサーチ（field research），インシデント・プロセス（incident process），ビジネス・ゲーム（business game），ロール・プレイング（role playing），テーマ（theme），ケースの収集（collection of case），ケースの作成（organization of cases）

目　　標

実習のテーマを適切に選定する。

どのような能力を主として養成するかによって，採用すべき教育技法がおのずから異なることをまず理解する。

次に実在の組織内でのテーマと仮想的なプロジェクトと，どちらを採用するか判断する。

仮想的なプロジェクトによる実習の代表的な例として，ケース・メソッドがあるが，この種の教育方法は日本では注目され出してからまだ日が浅いので，米国等の経験に学ぶべき点が多い。

外部の人々に面接して，ケースのデータを収集し，実習の教材用にまとめあげるという作業も，きわめて有効な実習であるということを理解する。このための調査方法等も経験によって得られたいくつかのノウハウがあるので，それを活用すべきである。

内　　容

1.1　実在組織の課題

実習テーマとして実在組織の内で発生した課題を採用する場合は，作業にあてることのできるマンパワー，期間，研修生の水準等から考えて，次のようなもののなかから選択することになろう。

(1)　中小企業のためのシステム

教師の指導をうけつゝ，実際の企業の一員としてシステム開発に従事する。

(2)　身近な要求を実現するためのシステム

研修生等が身近に関係する業務について自動化，省力化をテーマにとりあげる。

実在の組織における開発プロジェクトは，ユーザ部門からコンピュータ部門に要求されることもあれば，コンピュータ部門からユーザ部門に提案されることもある。プロジェクトの初期においては，この相違が推進の手順や文書などに影響をおよぼすので，ケースを与える場合は，どちらに属するのかを，あらかじめ想定して準備しておかなければならない。

またシステムの開発や改善要求は，一定の文書と手続きを制定して発行されることにもふれ，できればその例を示すことが望ましい。

実在の組織の中でシステム開発に従事した人に，具体的な事例や体験談を紹介してもらうとよい。実習のオリエンテーションとして，あるいは実習後の体験の再確認，一般化の助けとして実施することが考えられる。研修生はすでに相当程度の実務経験をもつと想定されているので，できるだけ有名なシステムの開発プロジェクトが望ましい。

1.2 仮想プロジェクト

システム開発技法の習得にはOJT（on-the-job training）が不可欠であるという考え方もあるが，これは教育効果があらわれるまでに長期間を要するという欠点がある。また，期間をかけた割には体系的な知識が身についていないという場合も多い。

ＯＪＴで習得する事項は大別すれば次の２種類になるであろう。

(1) システム開発技法の具体的応用による知識の強化，定着。

(2) 対象とする組織体の業務に固有な知識の習得。

ここでは少なくとも第(1)項については，ＯＪＴでなくケースによる実習でも，かなりの程度まで達成されるはずであるとの考えに立って指導を行なう必要がある。

米国等で古くから行なわれているケース・メソッドは，特に専門化された分野における実習のための代表的な手法である。これは実務において経験するのと類似した状況を数多く作り，当事者の立場で問題点を分析し，解決の代替案を検討し，仮空の経験を通して参加者を効果的に教育してゆくものである。

ケース・メソッドを用いる場合に，その導入者や製作者はとかくケースの複雑さや構造にだけ関心を払いがちである。しかしケース導入のねらいを効果的に達成するためには，テーマの選択よりもむしろケースの運営の方法が重要となることに注意しなければならない。

ケースの運営上の留意点は次のとおりである。

(1) 研修生が実習作業を通して，本当に必要性を感じ，みずから分析し設計しようという心構えになるように指導すること。

(2) 非現実的な状況が発生すると研修生の参加意欲は半減するので，このようなことを極力回避する。

(3) 実習のねらいに合わせて運営を工夫すること。新しいシステム開発環境を提供し，その中で自己啓発，相互啓発できる場をいかにつくり出すかが最も重要なことであり，技巧中心に走らないように注意しなければならない。

ケースとして仮想プロジェクトを与える場合は，特殊な知識や経験を必要としないテーマを選択するように注意しなければならない。これはテーマの内容，掘り下げの深さのどちらをコントロールしてもよいが，研修生のシステム開発能力だけが正当に評価できるような配慮をすることが必要である。

ただし，この点についてはケース・メソッドにおける経験から，さほど心配する必要はないともいわれている。特定の知識がない場合はチームの編成に配慮を加えて，各チームに他の研修生よりは詳しい知識をもつ者を平等に配属するなどすればよい。研修生同志の意見交換によって，あるいは教師への質問に答えることによって十分に対処可能であるといわれている。

特定の知識が不足しているため進歩が遅いということもあるが，実習としてこのような課題をとりあげることによって，その種の知識の必要性を認識させ急速に学習させることができる。したがって教育的に悪い効果となることはあまりなく，むしろより現実的な状況を設定しているということになる。

ただしこのような特殊知識，用語の習得が本来の目的ではないので，教師は特殊な業界だけで使用される難解な用語を避け，また同義語のなかから最も広く使用されるものを選択するなどの配慮をなすべきである。

各ケースで使用されているハードウエアの説明については，システムを理解するうえで必要不可欠のものに限定すべきである。ハードウエアについて詳述しても，システム自体を理解するのに通常は役立たないからである。また同様に，ファイルの大きさや転送速度等についても，バイトやワードやビット単位で説明せずに，文字（character）や数字（digit）単位で説明するような配慮が必要である。

開発の実習に関連した技法は，ケース・メソッドをはじめとしていくつかあるが，これらの技法がどのような能力の養成に役立つかをまとめたものが図１－１である。

ここでは能力として次の７分類があげられている。

・現状把握
・情報の収集整理
・問題形成
・代替案・解決案の提示
・意思決定
・指示伝達

・動機付け

これらの能力を養成するのに役立つ教育技法は次の６種類があげられている。

フィールド・リサーチ

インシデント・プロセス

ケース・メソッド

ビジネス・ゲーム

ロール・プレイング

グループによる活動

たとえばケース・メソッドは問題形成や代替案，解決案の提示に最も効果を発揮し，インシデント・プロセスは情報の収集整理に，そしてロール・プレイングは指示伝達の訓練に最も有効である。

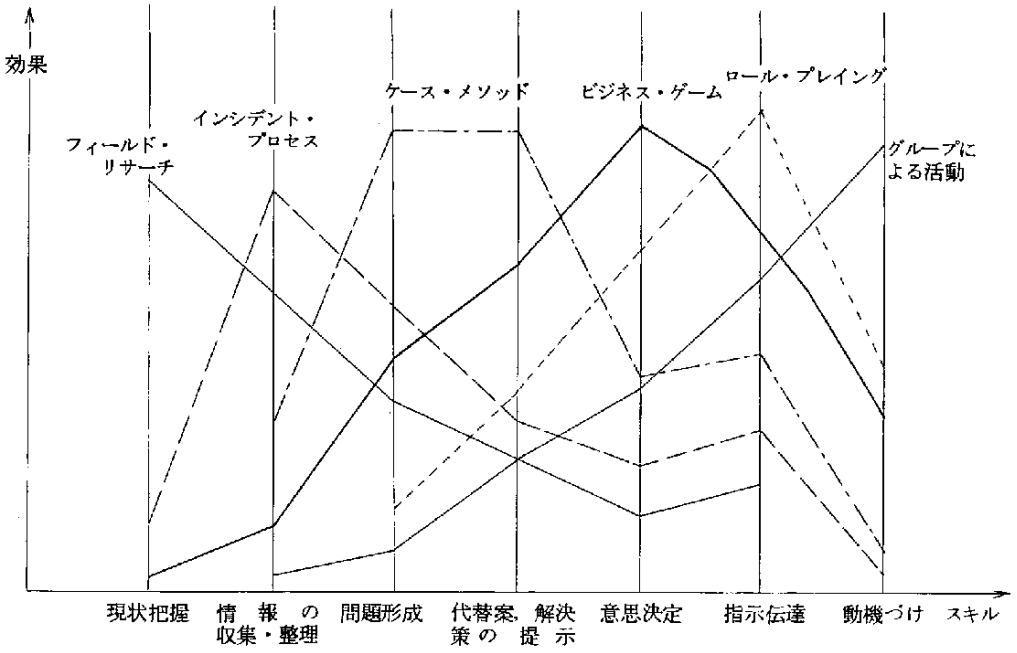


図１－１　各スキルに適する教育技法

図１－１にあげた技法のうち，主要なものの特徴を次に示す。

(1)　ケース・メソッド

企業などの組織体で実際に発生した問題や周囲の状況が，数ページから数十ページにわたって記述されたケースが与えられる。

研修生はケースの中の当事者の立場に立って，何が問題点なのか，どういう原因があるか，解決すべき代替案は何か，代替案のうちどの案をなぜ採用するのかなどを討議しなければな

らない。

討議に入る前に参加者は各自で与えられたケースの分析を行ない，独自の回答を用意しなければならない。そして討議はまず数人のチームで行ない，次に複数のチームが集まって全体で討議を進めることになる。

この討議の過程で「教師」は何も教えることはしない。研修生の議論を整理して，脱線しないように注意するだけである。

討議は数十回から数百回におよび，研修生はこの経験のなかで問題の形成や分析を迫られ，その解決のための代替案を示すように要求されることになる。

討議の過程を通して，他人に自分の意見をわかりやすく伝えること，勇気をもって自分の意見を述べること，他人が発言したことを正しく理解することといった，いわゆるヒューマン・スキルの開発も期待される。

ケース・メソッドでは，各人はすべて対等の立場であり，組織による圧力などを感じることはないから，解決案を気楽に述べることができる。

(2) インシデント・プロセス

これはケース・メソッドの変形として開発されたものであり，各人または各チームが，自分の求める情報を収集することを要求される。

ケース・メソッドでは，ケースの分析に必要な情報があらかじめ記載されているので，実際の企業等における活動とは相違していると考えられる。そこでインシデント・プロセスでは研修生に簡単に記述したケースが示され，これを検討した上で分析に必要と思われる追加の情報を要求させる方法をとる。

したがって各人または各チームは，それぞれ異なった情報にもとづいて分析を進め，種々の問題解決案を提示することになる。

このような過程を通じて，インシデント・プロセスの参加者は情報を収集することの価値や意義を体得するとともに，問題を発見し分析することを学びとるのである。

(3) ロール・プレイング

これはヒューマン・スキルの開発をねらうもので，研修生が他のメンバーの前で割り当てられた役割を演じることが要求される。

この様子をＶＴＲなどによって再生し，自己の行動を観察して，改良すべき点を見出すことができる。

ロール・プレイングは通常予告してから実施する。予告なしに取り上げてもうまくゆかないし，特定の研修生にあらかじめ割当ててしまうと他の研修生が準備してこないおそれがある。研修生を少人数のグループに分けて，そのなかから指名するということだけを予告する

と概してよく準備してくる。

1.3 課 題 例

実習の具体的な課題の例として次に電々公社中央電気通信学園で実施されたものを示す。

上級システム・エンジニア・コースとして163日間の研修を行ない，その中で183時間をついやして実施した卒業研究のテーマである。

大部分のテーマは1名の研修生が行なったが，2名の研修生でチームを編成したものもある。

実習は講義形式を主体とした科目を終了した後，コースのしめくくりとして，それまで学習した理論，手法に加えて自主的に各自の独創性を加味したテーマに対する目標を設定し，これを達成させる訓練として設けたものである。

テーマの選定，進行管理等は訓練生が主体となって行ない，指導員は助言を行なう程度にとどめた。

なお，テーマの選定にあたっては，一般に次のような点に十分な配慮をする必要がある。

① 研修生の専門分野。

② 実習にあてることのできる時間。

③ チームの人数，したがって実習で行ないうる作業量。

④ 大規模なテーマの一部分を行なうか，小規模なテーマでプログラム開発まで行なうか。

(1) 住宅の需要予測要因について

現在の住宅問題について考察し，住宅需要予測の一般的な方法を調査して，住宅意識に着目した住宅需要予測の手法を提案し，その検討を行なった。

(2) キャッシュレス社会

キャッシュレス社会の展望を得るため，米国および日本におけるクレジット・カードの現状を調査し，問題点を分析した。各種の予測資料を参考にして，キャッシュレス社会におけるコンピュータの役割，必要な端末装置，センタの機能等を考察し，実現可能なシステムへの提言を行なった。

(3) 情報検索と意味について

情報検索における意味論の位置づけ，意味の分析を行ない，情報の整理との関係について考察した。分類体系，シソーラス，統計的手法，ＫＷＩＣ等について具体的に検討している。

(4) ファイル構成についての一考察

情報検索におけるシソーラス・ファイルの設計を行なうために，情報構造の種類，ファイル構成の手法について分析を行なった。シソーラス・ファイル作成における前提条件と基本的問題を明らかにし，処理の基本方針を確立した。

(5) 情報検索についての一考察

情報検索のシステム例など一般的調査を行なった後，文献検索について分析と抽出，蓄積，探索の方法を検討した。具体的システムとしてＵＴＩＲＥＳという名前のシステムの基本設計を行ない，主題分析，文献のコード化，入力分析，構文解析の方法を決定し，ファイル構成，データ構造の設計を行なった。対象となるファイルは６種類である。

(6) 計量モデル用語とプリコンパイラの検討

計量経済モデル記述言語の要件を分析し，命令の種類と機能を決定した。この言語から，ＦＯＲＴＲＡＮプログラムを生成するためのプリコンパイラの機能を決定し，処理内容の設計，処理プログラムの開発を行なった。

(7) 医療システムについての一考察

人間と環境の制御機能について主として考察を加え，医療情報システムへの提言を行なった。

(8) 論理の正当性について

プログラムの論理の正当性を検定するプログラムについて基本設計を行なった。被検定プログラムについては，ある種の制限を課し，これをパターンにあてはめて検定を行なうこととして，必要なテーブル類処理のアルゴリズムを設計した。

(9) パターン処理についての一考察

人間システムにおけるパターン処理の例，メカニズムについて考察し，社会システムも含めたパターン処理の今後の方向を予測した。一例として指紋の照合をとりあげ，必要な機能，コンピュータによる処理の現状，各国の研究動向などを調査して，自動指紋検索システムの概念設計を行なった。

(10) ＥＤＰ会計と会計監査について

会計業務ＥＤＰ化の目的，現状，諸問題について調査分析を行ない，ＥＤＰ会計監査の具体的方法，内部統制組織の方法を決定した。情報分析の具体的資料としては，神戸料金局における事故発生状況，エラーチェック状況を調査している。

(11) 数値制御について

数値制御（ＮＣ）のハードウエアとソフトウエア（ＡＰＴ等）について調査し，プログラミングの実習をした。実習例は３件で，パートプログラムの計画，コーディングと必要な機器構成，ポストプロセッサの出力の分析などを行なった。さらに群管理，自動設計についても考察し，数値制御用プログラムの記述方法の標準化について論じている。

(12) あるシステムの一設計にあたって

コンピュータを共同利用するシステムについて予備設計を行なった。検討した事項は，シ

ステムの性能評価，データ収集と分配の即時処理方式，バッチ処理方式についてであり，最終的に機種の選定を行なった。

1.4 課題の作成

ケース・メソッドはプロブレムメソッドともよばれ，今世紀の初めに法律学校における判例研究にヒントを得て考案されたといわれる。最初のうちは民間企業から講師を派遣してもらい，具体的な問題点を紹介してもらって実習を行なっていたが，しだいに研究助手という人々が一定の方法でケースを収集し蓄積するようになった。

手順の概要を次に示すので，これを参考として研修生にケースの収集だけを実習させるのも十分に意義があるだろう。

(1) 面接の準備

通常は民間企業等の外部の人々に面接してケースデータの収集を行なうので，次のような準備が必要である。

① 関連知識を得ること。

② 課題を選択すること。

情報処理の分野で最近話題になっていることや，面接に応じてくれそうな人々が取扱っている問題等を考慮して，データを収集すべきテーマを選択する。ただし，あまりに新規なものは避け，具体的かつ典型的な状況を扱うものであるよう配慮を要する。

③ 面接の約束をすること。

訪問時間と目的の大略を知らせる。

④ 面接の対象は先方の組織内で，できるだけ高い地位にある人から始める。

⑤ 面接先についての具体的な知識を得る。

面接する相手の人について，会社について，これまでに他の人が面接した時の状況について，できるだけ知ることが必要である。

⑥ 面接の筋書きを作る。

質問表などの用意をする。ただし自己の筋書きに固執して客観的な事実を見逃してはならない。

(2) 面接中の注意

必要と思われるものを次に列挙するが，先方のようすによって適宜省略してよい。

① ケース収集の趣旨について説明する。

ケース・メソッドによる教育方法について説明し，若干の例を示す。必要な場合には会社名を秘し，数値を偽装することも可能であることを述べる。

② 必要とする資料を入手できるよう依頼する。

存在が予想される資料で必要なものは提供を求める。他の手段で入手できるものでも，面接の際に公式に依頼したほうがよい。事実や相手の意見を引き出すようにし，決して論争的になってはならない。正確な数字をメモする時以外はノートをとらないほうがよい。ただし面接が終った直後からノートを作り，その日のうちに完成させるようにすべきである。

③ 必要としない情報も適当にとり入れること。

無差別に情報を収集しようとしてはならないが，直感的に選択して若干の情報を取り入れると後で役に立つことが多い。

④ 多くのことを観察すること。

工場や事務所の様子，執務の状況など全般的な観察を行なうと有益である。

(3) 面接後のとりまとめ

① 面接結果についての報告を書く。

後日の面接の参考とするため，相手先について協力の状況などを報告するとよい。

② 秘密の保全に注意する。

資料の公表は面接先から文書による許可を得た後にすること。

③ 偽　装

会社名，数字，人名の偽装を行なう。産業の種別まで偽装することは通常は行なわない。数値は一定の係数を掛けるなどして偽装するが，本質的な関係をそこなわずにさらに変更できればそれが望ましい。

④ 公開許可の申請を要するもの。

・面接その他で得た非公開の資料を用いる場合。

・業界誌等の記事を用いる場合。これは会社がその記事を承認しているわけではなく，場合によってはクレームをつけることもありうると解釈すべきである。

・ケース作成者が公表資料をかなりの程度にわたって組み替えたり，圧縮した場合。

⑤ 公開許可の申請を要さないもの。

・会社発行の公表資料，報告書等だけを引用したもの。

・裁判所等の決定事項を利用している場合。

・特定会社に関係しない政府統計などを利用した場合。

(4) ケースの作成要領

① ケースの書きはじめは，現在の状況を分析するのに必要な背景を説明するところから始めるとよい。これに続いて問題の定義を行なえば，自然にシステム開発の各段階に導いてゆくことができるであろう。

② 情報を提供した会社を守るために，全部の記述を過去形とするとか，あるいはケースを作成した日付を明示する。

③ 図や表を活用する。ただし教材を整然とした形にまとめることだけがよいのではない。たとえば雑多な情報の中から必要なものを選択させる実習では，人々の会話をそのまま教材として使うこともよく行なわれる。

④ 数値をチェックする。とくに架空の数字は相互に矛盾がないように注意しなければならない。

⑤ 事実を記載するようにし，意見を述べてはならない。

⑥ 実際の設計結果を示すことは研修生の関心を高めさせるが，これをケースの記述に含めるかどうかは利用方法から判断すること。

⑦ 参考文献や引用は正確に書くこと。

(5) 調査をともなわないケース作成について

現実の組織体では情報が収集できなかったようなケースについては，架空のケースを机上で作成することがある。しかしこのようなケースは妥当性をチェックできるような外部事実がないので，内部矛盾をひきおこし，非現実的なものとなりがちである。したがってこのようなケースはできるだけ使わないようにすることが望ましい。

指導上の留意点

(1) 実習のテーマは参考文献に示したものから選択してもよいが，米国の事例（ケース）が多く，わが国で調査されたものは少ない。このような実状から考えれば，ケースを収集してケースブックを作ること，すなわち現場でのヒアリングと文書化の実習を行なうだけでも十分に有意義であろう。

(2) ケースデータを収集する活動に重点を置いて実習する場合は，ヒアリングに必要な面接の技術をここで復習し，強化しておく必要がある。

(3) ケースは通常，文章による記述と数値データや図の組合せによって与えられる。数値データを与えるためには，それが単に背景を説明するためだけのものか，あるいは主要な問題点の解決に不可欠のものなのかを研修生に示唆してやる必要が生じることもある。

参考文献

〔1〕J. Dearden, F. W. Mc Farlan, W. M. Zani,"Managing Computer-based Information Systems" Richard D. Irwin, Inc, 1971.（安藤嘉昭，島田照代，一楽信雄共訳「新・経営情報システム」 建帛社 1973）

〔2〕柴田典男，許斐義信著「講座・ビジネスゲーム1＝入門編」中央経済社，1977

〔3〕日本電信電話公社中央電気通信学園編「昭和45年度電算機応用科上級システムエンジニアコース卒業研究」　1971

〔4〕Malcolm P, Mc Nain,"The Case Method at the Harvard Business School," McGraw-Hill Book Co, Inc. 1954.（慶応義塾大学ビジネス・スクール訳「ケース・メソッドの理論と実際　ーハーバード・ビジネス・スクールの経営教育ー」東洋経済新報社，1977）

第2章　チームの編成

キーワード

GREMEX(Goddard research engineering management exercise), ビジネス・ゲーム(business game), ケース・メソッド(case method), チームの統制(control of a team), テクニカル・スキル(technical skill), ヒューマン・スキル(human skill)

目　　標

システム開発実習のための適切なチームを編成する。

チームの規模には適正な範囲があること，チームの規模によって各構成員の職務の分担のしかたには相違が出てくることなどを理解する。

内　　容

2.1　チームの組織

チームの組織では人数と各研修生の役割の分担が問題となる。

チームの人数はビジネス・ゲーム等の経験から5人以上は不適当であるということがいわれている。ＮＡＳＡで採用した，研究開発管理者の研修システムである GREMEX (Goddard Research Engineering Management Exercise)では1チーム3人の編成がとられている。ただし現実の開発プロジェクトではこのような小人数はあまりないので，コミュニケーション技法とも関連した，プロジェクト・チーム論についても言及するとよいであろう。

ビジネス・ゲームでは3～8組のチームを編成し，各チームは4～12人のメンバーからなる組織を採用することが多い。

チームの組織や役割の分担はあらかじめ示すことなく，研修生の自主的決定にまかせるのがよいであろう。最初から精緻で固定的な組織を編成する必要はなく，仮のリーダーを決めて分析が進むにつれて必要なことを決定していけばよい。各担当メンバーに業務を固定的に課すのではなく，実習中に適当にジョブ・ローテーションを行なう工夫も必要である。

研修生のチームが決定した組織に対し，次のような一般的評価尺度にもとづいて助言を行なう。

(1)　戦略目標を達成できるようになっているか。

(2)　環境変化に適応できるか。

(3) 主要な業務を十分に果せるか。

なお，1チームの人数が8人程度以上になると一般に組織に階層構造が生じ，立案者と作業者に分かれてくる傾向がある点にも留意すること。

また，組織を設計することは，そのコミュニケーションの方法を決定することでもある点を研修生に徹底し，これらがバラバラなものではないことを理解させるのが望ましい。

ケース・メソッドではクラスの人数を6～7人から100人以上まで変えてみたところ，40～50人から80人までが最適だという結論が得られた。

ここでは議論を活発に展開させるために最適な人数ということで判断がなされる。経験的には，活発で建設的な討論を持続させるためには，少なくとも10～12人の優秀な学生が必要であるということがいわれている。したがって，あまりに少人数のクラスでは，それだけの数の優秀な学生がそろわないおそれがあるというのである。

2.2 チーム活動の意義と学習目的

このコースにおける実習の目的は，システム開発に関する専門知識，能力いわゆるテクニカル・スキルを確実に定着させることとともに，次のようなヒューマン・スキルの習得も心がけなければならない。

ヒューマン・スキルとは「グループの一員として効果的に働き，あるいは自分のリードするチームを一致協力して努力させるために必要な能力」とされている。したがってヒューマン・スキルを持った人は他のメンバーの発言や行動がどういう意味をもつか理解できるし，自分の考えることを他のメンバーにわかりやすく説明することもできる。

ヒューマン・スキルはある種の訓練や技術的方法によって開発されることもあるし，指導者の適切な助言や忠告で開発されることもある。

チームによる開発実習を通して，第5章のコミュニケーション技法とともに，十分な注意をはかって習得するよう研修生を指導していく必要がある。

指導上の留意点

選定した実習テーマに要する作業量をあらかじめ見積っておいて，実習時間を十分に考慮した上でチームの規模を定めることが必要である。

実習時間が継続して長時間とれないような場合には，研修生がその日の実習に習熟する前に所定時間が終了してしまうので，このようなことがないように注意する。やむをえない場合は前述の作業量にもこの影響を反映させるべきであろう。

実習活動の多くはチームの構成員の自主性にまかせることができるが，活動が空転しないよ

うに定期的に報告させたり，観察をすることが必要である。

参考文献

[1] H. D. Clifton, "Data Processing System Design" Business Books Ltd., 1971
（田中弘・清水克彦訳「コンピュータ・システム設計」白桃書房，1974）

第3章　実習の進め方と教師の役割

キーワード

ケース実習(case exercise, case studies), 実習活動のリード(leading an exercise), 教育技法(educational technique), ケース資料(case material), 理論と実際(theory and practice)

目　　標

実習を中心とした学習活動の運営方法を理解し実践する。

実習活動は通常の講義よりもはるかに教師と研修生との心理的な状況によって結果が左右されやすい。

クラスにおけるあらゆる兆候を迅速に把握し，適切な技術を駆使して実習をリードしていく，プラクティカルな教育技術といったものを理解し身につけていかなければならない。

内　　容

3.1　ケース実習の特徴

ケース実習の運営方法については，さまざまなタイプが存在するが，一般に次の三点が最大公約数的な要素であるといわれている。

(1)　経験の重視

企業や政府における現実の問題を基礎とし，現実の問題に焦点をあてたテーマを採用する。情報処理の実際の場で，責任ある仕事につける人間を作り出すという主要な目的を常に念頭におかなければならない。

(2)　一般論よりも個別論の重視

一般的な理論の構築能力ではなく，設計能力，意志決定能力が要求されるのであり，それは訓練によって養成されるという考え方に立っている。したがって各ケースの個別の状況における判断が強調され，ケース間の共通点よりは相違点が指導のポイントになる。

(3)　学生の経験に対する配慮

単に知的な学習となってしまうようなテーマは盛り込まず，学生達の感情的な側面と知的側面の両方の反応を促進するように多大の配慮を行なう。

3.2 必要な教育技術

実習を行なう場合，教師の役割は講義などよりも一見楽であるかのように思われるが，実は違う種類の負担がかかってくる。

これは一言でいえば，研修生の自由な思考と討論を特定の目的意識のもとに誘導することである。そのために必要な技術は次のようなものであろう。

(1) グループ討議の進行をうながすような質問を発することができる。

(2) これまでの経過を整理すること，それをわかりやすく示すことができる。

(3) グループ討論が有効に行なわれているかどうか，進行が早すぎたり，遅すぎたりしないかどうか，研修生がひとしく参加しているかどうかを判定することができる。

(4) 研修生の活動を特定の目的意識のもとに誘導するとともに，何か目新しい発見をするように，適度に脱線させること。

(5) 実習が予想しなかった方向に進行する場合もあるが，これによって学習効果の上ることもあるので，研修生を十分に観察すること。

いうまでもなく，すべての学習活動に共通する最重要な条件は，学ぶ側に関心があるということである。ケースによる実習の場合はテーマの選択とともに，その運用方法，技術が学ぶ側の関心を大きく左右することになる。前述のような技術は一般に教育技術の一種として十分に修得しうる範囲内であるといわれている。

3.3 実習の推移の特徴と教師の対処

通常の講義による授業と実習とでは活動の推移状況が異なる。主な相違点をあげれば次のとおりである。

(1) 教師は権威的にふるまい，学生は受身の立場である。

(2) 試験や討論の時間には，教師が学生の誤りや無知を指摘し，批判することが多い。

(3) 教師は学生が修得すべきことを速やかに提示し，自分の判断基準に従っておしつける。

ケースによる実習では，このような点はあまり見られず，言いかえればこのような状態になっては好ましくなく，別の運営がなされなければならない。

しかし教師の側も学生の側もケースによる実習がはじめての場合は，最初のうち混乱やケース学習の悪い面が出てくる。具体的な例は次のようなものであり，その対処方法もあわせて示すこととする。

(1) 自意識，対抗心，不安などから教室内のコミュニケーションがゆがめられ，相互不信に陥いる。

これに対しては，教師が学生を批判したり，あるいは学生相互間で悪らつな批判をしあったり

しないように注意することが必要である。特に教師は，すべての学生の発言を理解してやるように熱意を持って耳を傾けるようにしなければならない。教師が身をもって手本を示せば，このような状況は容易に回避できる。また教師は賛成とか反対の意志表示をして議論に枠をはめてしまうこともすべきではない。

(2) 一部の研修生が発言を独占したりして，良い意味のリーダシップが見られず，他の参加者がこれに反発する。

これはたとえば，グループ全体に落着きのない状態が現われたり，だらけた雰囲気から察することができる。教師は発言の少ない者を指名して発言させ，それを批判するのではなく建設的に受け入れるように指示することが必要である。

(3) 研修生がさじをなげて，教師に対して問題の「解答」を与えてくれるように頼むことがある。

これは兆候としては，問題に無関係な発言，退屈感，研修生のおじけ，ささいなことへのこだわり，といった点から察知される。研修生個人個人が，また全体として一種の内向状態に陥っているので，教師はまず責任を成し遂げるように激励することを試みるとよい。

最終的には教師が何らかの解答を与えるつもりであっても，まず研修生に自分の考えを述べるように指示すべきである。教師が解答を示しても，それは絶対に研修生の思考作業にとって代わりうるものではないので，解答を示さざるを得ないような状態になったということは，このケースが失敗であったと考えるべきであろう。

(4) 最後に，教育上の七つの罪とよばれるものをあげておく。

「恩きせがましい態度」，「皮肉」，「個人的な詰問」，「無作法」，「ひとりよがり」，「はにかみ」，「話しすぎ」

3.4 実習を進める上での各種の手段

教師が実習を進める上で採用する手段は，次のようなものが考えられる。すなわち，(1)質問を発する，(2)発表されたことを再述し，再構成する，(3)事実関係について自分の見解を述べる，(4)「小講義」を行なう，である。

これについて次に要領を示す。

(1) 質問

質問は前もって準備するものではなく，その場で作り出すものである。解答が困難な質問を直接的に切り出すのではなく，段階を追って進めなければならない。個々の学生の未熟さ，誤りを指摘するような質問であってはならないし，また教師の技量を誇示するだけの質問も無意味である。

たとえば討議の口火を切る場合には，その方向を前もって決めてしまわずに，「判明している事実は何と何でしょうか」といった質問から始めるのもよい。学生の発言を明確にするために,「あなたのおっしゃったことは……ということですか」と質問することもできるだろう。また考えを広めさせるために,「何かもっとつけたすことはありますか……」という質問もできるだろう。

独断に陥らないように注意してやる質問としては,「あなたはそれがすべての場合にあてはまると思いますか」とか「それはこのケースに描かれている状況にどのようにあてはまるのですか」といったものがある。

さらに実習が進むにつれて，ケースの基本的問題をより適切に理解できるようにしてやる質問，すでに定義された問題に注意を喚起させるような質問，軽率な結論に走りがちな前提を客観的に指摘してやる質問が発せられる。

ケースによる実習で学生が主として経験することは，個々の状況の間の関連，連続性の認識である。これを助けるための質問としては次のようなものが考えられる。「あなたが言おうとしていることは，Aさんの発言に関係がありますか」,「あなたの見方は……のケースにもあてはまりますか」,「私達が取り組んでいる……という大きな問題とどのように関係していますか」

問題の検討が終り，意志決定をすべき時になっても，容易に決定できないことがある。この場合は，さらに何が知られていないのかを問い，困難に直面した時の無意識な逃避を抑制することが必要である。

一般的に言えば，質問はケース・メソッドの道具として使われすぎる傾向がある。討論が順調に行なわれている時は，質問は時折りするだけで済む。まだ議論されていない点を指摘するために質問を発することも必要だが，それよりは討論の流れの中で，学生の発言に対する反応として発せられた質問のほうが，はるかに有効である。

(2) 再述

ケースの討論における学生の発言は，ほとんどすべてが即興的なものであり，30秒以内の長さであることが知られている。したがってそれは十分に考えつくされた結果ではなく，また表現も常に不十分なものである。

このような個々の発言を明確にするために，教師が共通の言葉で言い換えるのが再述である。

教師が再述を行なう場合に注意すべき点は，学生の発言が意味していることについて性急に結論を下してはならないこと，本質的なものまで付け加えてはならないこと，学生の発言に対して自分が同意するとかしないとか示唆してはならないこと，などである。むしろ学生

のアイディアを強調してやるといったことを心がけると良いであろう。また学生達が教師の再述に依存する気持を起して，明確な発言をしなくなることも時にはあるが，これに対しては教師が発言を聞こうとする意欲とそれに対する集中力を示して，見ならわせるようにすれば効果があるだろう。

再述と同様な効果は，討論を一定間隔をおいて要約することによっても達成される。しかしこれも教師が規則的に行なって，学生に依頼心を起させてはならない。要約も本来は学生がしなければならないことであり，教師は学生が関心を集中させるべき主題を強調するといった程度のことにとどめるべきである。

(3) 見解

これは質問や再述よりもさらに高圧的な方法であるが，討論をコントロールしたり実習の目的を達成させるためには，時として採用せざるを得ないこともある。

見解を述べる場合の注意点は，学生の意見が十分に出された後にし，最初から示さないようにすること，教師が自己の経験について過度に強調しないこと，あまり長時間にわたらないこと，などである。

ケース・メソッドにおける教師の役割は，権威的に指導することではなく，参加者の一員としてリーダシップを発揮することであるから，自分の見解を示したりすることはさしひかえるべきだとの説もある。しかしこの考え方もやや不自然な点がある。むしろ教師の発言が権威的に受け取られず，参加者の一員として受け取られるような，クラス内の本質的な関係が早い時期にうまく形成されるように努力すべきではなかろうか。

(4) 小講義

ケース・メソッドによって学生達が自力で到達できる高さには限界があるので，時に小講義を行なう必要がある。

これは通常は授業の最初または最後に行なうが，討論が進展しない時は随時行なうべきである。教室の状況に応じて，ケース・メソッドでも部分的に講義式授業方法を取り入れてゆくことは可能である。

以上のような手段を活用して，教師は教室で複数の役割を演じることになる。教師は参加者の一員であり，他の意見を聴取する人であり，分析者である。そして質問を発し，再述し，小講義を行なう人でなければならない。そして何よりも注意しなければならないのは，他の役者をさしおいてスポットライトをあびるようなことをしてはいけないということである。

3.5 実習中に学生が経過する段階

ケース・メソッドによる実習を経験する過程で学生は通常次のような段階を経過していくとい

われている。

(1) 個人能力の限界を知る。

チームの同僚が考えつくあらゆることを自分一人で考え出すことは誰にもできないということを発見する。これは単に作業の分量が多いから消化しきれないというのではなく，限定された課題を与えても，ありとあらゆるアイディアが出てくるため，個人ではすべて思いつかないということである。

(2) 相互の協力，援助体制ができる。

問題解決にあたって他人のアイディアを活用するようになり，相互援助の提供と受け入れが円滑に行なわれるようになる。

実習になれるにしたがって，最初のとまどいにより忘れ去っていた成績への競争心が思い出されてくるが，同時に学生達はグループで知的努力のプールを作っているのだということを認識し，喜びを見出すようになる。

(3) 独立的思考により決定を行なう。

教師といえども最適の答に到達しているのではないことを認識するようになる。同僚や教師の助言を受け入れるが，他と異なることを恐れたり，権威を探し求めることはなくなり，自分の責任で意志決定を行なうことになれてくる。

指導上の留意点

この章は教育技術に関する記述が中心であり，すべてが指導上の留意点であるということもできる。

この項で特に示すものはない。

第4章　開発の実習

キーワード

教育訓練（drill and practice），問題の定義（definition of a problem），システム開発の組織（organization of systems development），現状分析（analysis of present operation），将来予測（prediction），システム分析（systems analysis），企画書・提案書（proposal），システム開発のスケジュール（schedule of systems development），システム代替案の評価（evaluation of alternatives），定性的評価（qualitative evaluation），定量的評価（quantitative evaluation），システムの機能決定（system function），フィジカル設計（physical design），インプット設計（input design），ファイル設計（file design），プロセス設計（process design），アウトプット設計（output design），セキュリティとインテグリティ（security, integrity），工程管理（schedule management）

目　　標

具体的なプロジェクトについて，開発の実習を行なう。

第1単元の「情報分析」，第2単元の「システム設計」で学習した考え方や技法を，実際に作業をとおして用いることにより，強化し定着することが目標である。

これら2つの単元の実習の進め方は多少ニュアンスの異なったものになろう。すなわち「情報分析」とくにその管理的側面や経営組織に関連する部分では，例としてあげたケースから研修生が各自で何らかを学びとるようなケース・メソッド的運営になるだろう。一方「システム設計」では，各組織体の固有の事情にかかわりなく，情報処理という共通の立場から見た技法がいくつか存在するので，これらの習得は訓練（drill and practice）によることとなるだろう。

内　　容

4.1　問題の定義

問題の定義は，そのプロジェクトの概要を簡潔に記述させることから始める。そのプロジェ

クトの必要性は何か，目的は明確かをここで確認させなければならない。記述の例は後の「参考」に示すが，この作業はシステム・ライフ・サイクルのループの第一段階である「問題の提示とニーズの把握」に対応するものであることを十分に徹底させる必要がある。

ＥＤＰの経験がない組織体では，問題の定義の段階から外部のコンサルタントに委任することもあるので，研修生にコンサルタントの役をさせて，必要な情報を聞き出しまとめさせる実習もよいであろう。

この段階の作業の成果はシステムの機能概要決定とライフ・サイクル計画の概要とする。例は後述の「参考」に示す。

プロジェクトの必要性については，後述の「参考」に次のような例をあげた。このほかにどのようなものが考えられるか，また今までに経験したプロジェクトがどの分類に属するか，といったことを検討させるとよいであろう。

(1) 法律または政府機関等の決定によって実施が強制されるもの。

(2) 同業他社が実施の傾向にあり，社会的になかば強制された形で実施せざるを得ないもの。

(3) 業界の見通し，業容の拡大から予測して，ニーズを先取りして実施するもの。

(4) コスト低減を目的としたもの。

(5) 時間短縮により，業務遂行の効率化，質の向上をねらったもの。

(6) 業務処理のロジックを改善し，シミュレーション等を行なって質を改善するもの。

(7) 既存システムの見直しを行ない，エラー防止，信頼性向上，実施部門の拒絶反応解消等をねらうもの。

システム・ライフ・サイクルの「予備検討」において，システム実施前に予想される欠陥の指摘，評価と対策を検討する必要があることを強調する。

システム失敗の典型的な原因は以下の点であることを説明する。

(1) システムの実施が一時に広範にわたること。

(2) システムの実施が性急すぎること。

(3) 以上２つの原因により，現場が混乱状態におちいること。

(4) システム実施が本来の業務におよぼす影響を正しく把握しないこと。たとえばデータ・エントリーに時間をとられすぎるなどがある。

(5) 現場の繁閑を考慮しないこと。繁忙期は新しいシステムを理解する余裕も意欲も持ち得ない。

(6) 導入時の要員の教育不足。

(7) ＥＤＰに対するアレルギー。

(8) システム開発をコンサルタントまかせにして，ユーザの考えを十分に反映しないこと。

4.2 システム開発の組織

プロジェクト管理の対象は一口に，人（要員，組織），物（設備，資材），金（予算，コスト）といわれるが，最も重要な要素である「人」の管理について考えさせる。そして実習を行なうためのプロジェクト開発組織を研修生に決定させる。

組織はプロジェクト・チームを編成する場合と，既存組織とくに電算機部門により開発を行なう場合とがあるが，それぞれの利点は次のとおりである。

(1) プロジェクト・チームの利点

責任が明確になり，効果的な計画，管理，協調，コミュニケーションができる。

(2) 電算機部門による開発の利点

多数の専門家による効果的なプログラミングが可能であり，しかも要員の有効利用ができる。プロジェクト・チームも解散すれば結局は電算機部門に配属される人が多く，組織のまとまりを保つことができる。

実際の運用ではたとえば，すべての要員をプロジェクト・チームに入れるのではなく，システムはプロジェクトの指揮下に，プログラミングは電算機部門の指揮下に組織するようなことが行なわれる。システム開発組織の実例は「参考」に示す。

なお大規模なソフトウエア・システムの開発については，コミュニケーションのオーバヘッドの面から巨大な組織の不具合が指摘されていること，ＩＢＭのチーフ・プログラマ・チームというアプローチがあることなどについてふれ，説明をしておくとよい。

4.3 現状分析

(1) ヒアリングの技術

ヒアリング技術の重要性については，Ｄ１単元第３章で述べたとおりだが，ここでは具体的な要領の例を示して復習させる。

時間を限定して研修生にヒアリングを行なわせ，収集した情報を文書にして報告させるのもよい。

ヒアリングの要領例は「参考」に示す。

(2) 事務分析

次のような調査，分析の例を「参考」に示す。

a. 事務分析フローチャート

b. ＳＯＰ

c. 業務量分析

d. コンピュータ使用の現状

情報分析の対象となる帳票類は現物を集めるのが理想であるが，かならずしも実行できないことがある。このような場合には，標準様式例として市販されているものを活用する実習指導も必要である。また各組織体における文書作成の規準を入手して，これを分析検討させるような実習も有意義であろう。

4.4 将来予測

(1) 動向の定性的予測

業界誌等の予測を参考にして，市場の動向も考慮し，当社の業務およびシステム化の動向を定性的に予測する。

予測の記述例を「参考」に示した。

(2) 業務量，コンピュータ利用率の推移予測

システムの利用期間内で，業務量の増加等を予測して，これに対処できるかどうか検討する。

予測例を「参考」に示す。

4.5 システム分析

(1) 経済効果の分析

次のような事項について分析した報告例を「参考」に示す。

① 人件費の節減額

② 業務別の費用と節減額見積り

③ キャッシュ・フロー分析

(2) システム開発の着手優先度決定

経済効果分析にもとづく着手優先度の決定は次のような考え方で行なう。

① 費用節減効果の大きいものから早く着手する。

② 現行のシステムで不満足ながら稼動しているものは後で着手する。

③ 他システムと密接に関連するアプリケーションは遅く着手する。

④ 他のプログラムの完成に影響を与えないものは後で着手できる。

⑤ 複雑で開発に長期間を要するものは，実施そのものを再検討する。

⑥ データ通信網など，外部の施設等に依存するものは注意を要する。

4.6 企画書，提案書

次のような文書の例を「参考」に示す。

(1) システム提案書

(2) システム機能概要

(3) 出力帳票一覧表

(4) 出力帳票の内容説明

4.7 システム開発のスケジュール

(1) 計画上の要件

システム開発計画は次のような事項との関連も考慮して立案する。

① 計算機室のスペース，空調，電源等の設備計画。

② ソフト開発要員と操作員の計画。

③ 導入前のテスト用計算機の利用計画。

④ メーカの提供する資料，教育訓練の利用。

⑤ プログラム移行のスケジュール。

⑥ 新システムによるサービス提供のスケジュール。

(2) スケジュールの概要

① 準備期間

人員配置，教育訓練，ツールの開発，標準の設定などを行なう。

② 設計仕様の設定

システム範囲の決定，システム移行方法の決定，ハードウエアの発注などを行なう。

③ システム設計とプログラミング

④ システム移行

旧システム（手作業または自動）との平行操作をへて，新システムへ移行。

特にオペレーショナルなシステムは移行に注意を要する。（「参考」）

(3) スケジュール例

バーチャートの例を参考に示す。

4.8 システム代替案の評価

定量的評価方法の一例として，シミュレータによる評価を次のような要領で行なうことが望ましい。

(1) シミュレータ利用の目的

① 対象業務に適したハードウエアとソフトウエアの選択。

具体的には，ＣＰＵの速度，所要記憶容量，オペレーティング・システム，業務処理

に適した周辺端末装置を決定することなどである。

② 対象システムのライフ・サイクル期間を検討する。

想定したシステムが増大する業務量にいつまで対処できるかといったことを検討する。

(2) 仮定事項とモデルの範囲

① 仮定事項

・使用する端末機の種類，台数

・処理するファイル，情報の種類

・センタの機器構成

② モデルの範囲

・ハードウエア構成

・オペレーティング・システム

・環境条件（プログラマの経験等）

・対象業務（入力文字数，アクセスするファイル，処理内容，結果の送信など）

・多重処理におけるシステムの優先順位

・処理の種類と件数

(3) シミュレーションの目標確認

① 区分した各時間帯において，業務処理がシステムにおよぼす負荷を把握する。

② ピーク時の処理が特定のハードウエア，ソフトウエアに与える影響を把握する。

③ ハードウエアの能力の検定。

④ 周辺装置の組合せの決定。

⑤ どのオペレーティング・システムが適当であるかの検討。

⑥ 応答時間の要求性能を満足するか否かの検討。

(4) シミュレーションの方針

① ＣＰＵと周辺装置の利用率から，ピーク時を決定する。

② ＣＰＵ速度の妥当性を検定する。

③ ピーク時の処理が不可能な場合は，ＣＰＵ速度の速い機種を用いてシミュレーションを行なうか，業務設計を変更する。

フローチャートによるシミュレーション方針の記述例を「参考」に示す。

(5) シミュレーションの各ケースの設定

① 区分した各時間帯における標準計算を行ない，機器の利用率を算出する。

② ライフサイクル期間の判定のため，ピーク時の負荷をたとえば50%増した計算を行なう。

③ 周辺装置を変更した場合。

④ オペレーティング・システムを変更した場合。

⑤ 業務の優先順位を変更した場合。

(6) 結果の分析と結論

① システム設計の基準となる時間帯を機器の利用率から設定する。

② ピーク時の負荷とシステム限界の判定。

③ ＣＰＵの選択。

④ 周辺，端末機器の選択。

⑤ オペレーティング・システムの決定。

⑥ 応答時間特性と要求性能との照合，確認。

4.9 ハードウエアの定性的評価

(1) 基本的要件

一般的に次のような要件の検討を行なう。

① 多重プログラミングと遠隔処理等の機能。

② ファミリー機種を構成し，上位機種との互換性があること。

③ 使用可能な言語。

④ ソフトウエア・パッケージの完備。

⑤ 社内で使用中の機種との互換性。

⑥ 他社での使用状況。

⑦ 対話型処理等のための端末機。

(2) 業務処理からの要件

① 業務形態（オンラインまたはリアルタイム等の必要性 ）

② 入力情報形態（カード，ＭＩＣＲ，紙テープ等）

③ 業務特質（高速演算を要する等）

④ ファイル媒体（磁気ディスク，磁気テープ等）

⑤ 出力装置（ディスプレイの必要性，月報等の集中発生の有無など）

⑥ バックアップ，多重処理の必要性など。

(3) 移行および操作上の要件

① 類似機種の導入前利用可能性と場所。

② ソフトウエア・サポート

・数学，統計ルーチン（ＳＳＰ等）

・計算機ユーザ（SHARE等） のライブラリ。

・特定分野のソフトウエア（原子力のＣＤＣなど）

(4) 定性的評価の項目

① 適用業務分野における対象機種の実績。

② 他システムとの互換性。

③ コンピュータ・メーカ等のサービス。

④ メーカ等に委託したシステム分析の正確性，綿密性。

⑤ システム・ソフトウエア（言語，サービスルーチン，パッケージ等）

(5) 候補機種の具体的比較

メーカ各社から価格，性能の面で互いに匹敵しうる機種を選定し比較する。

まず基本的要件について，次に業務処理上の要件等について，定性的比較の表を作成して検討する。ソフトウエアの比較表の例を「参考」に示す。

4.10 ハードウエアの定量的評価

(1) 一般的手順

① 評価基準を設定する。

② 機器構成とコストを検討する。

③ 販売条件またはレンタル条件を検討する。

④ 買取とレンタルの経済分析を行なう。

(2) 候補機種を対象とした経済分析

新システムによる人員削減，現行システムを継続する場合の増設費用，新システムの導入費用などを考慮した経済分析例を「参考」に示す。

(3) コスト／性能分析

分析例を「参考」に示す。

(4) ベンチ・マーク・テスト

テスト結果例を「参考」に示す。

(5) 月間レンタル額の比較

例を「参考」に示す。

4.11 システムの詳細機能決定

(1) 目的の確認

(2) 機能の決定

記述例を「参考」に示す。

(3) ロジカル設計

① 入　　力

② 処　　理

③ 出　　力

(4) 実施に失敗したシステムの再検討例を「参考」に示す。

4.12　システムのフィジカル設計

4.13　インプット設計

例を「参考」に示す。

4.14　ファイル設計

例を「参考」に示す。

4.15　プロセス設計

例を「参考」に示す。

4.16　アウトプット設計

例を「参考」に示す。

4.17　セキュリティとインテグリティ

(1) システムの改訂手続き

例を「参考」に示す。

(2) データおよびプログラムの保管方法

例を「参考」に示す。

(3) エラー防止手段

例を「参考」に示す。

(4) マスタへの転記テスト

例を「参考」に示す。

4.18　システム開発の工程管理

(1) 進捗度報告

次の例を「参考」に示す。

① 1個の作業項目の見積り，進捗度等の時系列的変化。

② 同時期におけるプロジェクト全体の一覧。

(2) 工程遅延の原因分析

例を「参考」に示す。

指導上の留意点

(1) ここでは情報システムの開発の実習に徹するという方針で指導を行なう。方法論はすでに他のコースまたはDコースの他の単元で学習しているので，ここでは指導中に状況を適宜判断し，助言を行なう程度にとどめる。必要な場合は，「まとめ」のような形で再提示するのもよいが，多くの時間を費してはならない。

(2) 開発手法の主要なものについては，できるだけ実際に作業をさせて習得させることが必要である。作業の時間がとれないものについては，文書などの実物を例示して指導することが望ましい。作業を行なう項目と，実例の提示にとどめる項目とのバランスを事前に検討し，所要時間や学習者の印象について見通しをつけておくことが必要である。

(3) ケースを与えて実習させる場合に，システムアナリストとユーザ部門との折衝の実習ができないので注意を要する。特に情報分析の段階における要件の確認といったシステムの成否を左右する重大な事項が，かなりの部分を面接の技術などに帰着されることを強調する。必要に応じて，ケースデータを一度に提示することなく，学習者にこれを引き出させるようなヒアリングの実習をさせることもよい。

(4) 実習における研修生の討論を指導する場合は次の点に留意すること。

① 討論の口火を切る方法としては，次のようなものがある。

- 「問題は何か」，「関連する事実は何か」という順に質問する。
- 「事実は何か」と切り出す。
- 「どのような方針で開発を進めるか」と問う。
- 「このケースをどう思うか」と問う。

② ケースの分析，討論を進めていくうちに研修生は自然と否定的，批判的，あらさがし的な態度に陥りがちなので，自分自身の肯定的，建設的な意見を述べさせるよう注意しなければならない。

③ 討論が進み，大方の意見がほぼ一致しかけたら，一人の研修生にこれまでの議論をまとめさせ，意志決定の案を作らせる。

(5) ＫＪ法はアイデアの構造化，組立ての技術であるが，やや広義に解釈してグループ化，構造化の手法とみれば，さまざまな実習課題が考えられる。たとえばシステムの処理に必要なデータ項目をどのようにファイル化すべきかといった，ファイル設計の初期の段階をＫＪ法を利用して実習してみるのも興味があろう。

(6) 情報システムにおける機密保護については，従来あまり関心が払われていなかったが，オンライン化，日常化の進展にともなって重要性を増していることを十分に認識させる。銀行におけるキャッシュ・ディスペンサの暗誦番号など身近な例をあげて検討させるとともに，時間があれば保護システムの設計をし，その欠陥を見出すような実習をさせるとよい。

また少し見方を変えて，情報処理サービス会社などの機密保護規則を入手し，分析検討するような実習も有効であろう。

(7) この単元における実習の目的は大きく分けて次の2点となる。

① システム開発技法の習得，定着を目標とする。

② プロジェクト管理の練習をする。

それぞれは開発プロジェクトによる実習のねらいとして十分に意味をもつが，一方にだけ重点をおきすぎると実習の効果が不十分なものとなってしまう。すなわち，次のような問題点が存在することを念頭において，ケースの与え方，指導方法を検討すべきである。

① 技法の習得，定着を目的とした場合は，一般に研修生の身近にテーマを求めて，ひととおりシステム開発の全段階を実習してみることが多い。このため小規模なプロジェクトの経験にとどまってしまう恐れがある。また，実際のプロジェクトで発生する種々の問題，たとえば工程管理，外注などの判断事項の演習にならないことに注意すべきである。

② プロジェクトの管理に重点を置いた場合は，一般に技法の習得，定着がおろそかになりがちである。通常は仮想のプロジェクトが対象となると思われるので，研修生が所属する組織の問題といった，現実味のあるテーマが扱えないことになるであろう。

以上2種類のプロジェクトを実習させるか，あるいは研修生の所属する組織で大きな問題を発掘し，その一部分を実習させて成果を積み上げて全体を完成させるような方法をとることが考えられる。

(8) プロジェクト管理では，物，金，人，時間，など各種のリソースの配分を考えなければならないが，特に実習中は「時間の利用」にポイントを置いて指導するとよいであろう。研修生の分析能力，設計能力，組織的なチーム・ワークのよさ，機能分担や責任と権限の明確さなどが，すべて時間に集約されて現われてくるからである。

参　考

以下に示す例は，システム・ライフ・サイクルの各段階を追って，典型的な状況設定，収集情報，データ，作業の成果等を示したものである。

各例にはそれぞれケース番号が与えられているが，同一のケース番号を順にたどっていけば，一貫した開発プロジェクトの例となっている。

したがって実習は一個のケースについて行ない，ライフサイクルの各段階において必要ならば他のケースの例を示すことによって，一貫性と幅のある実習を行なうことができるであろう。

(1) 紹介したケースのうち，1～9はロジカル設計に重点を置いたものであり，10～12はフィジカル設計に重点を置いたものである。

ロジカル設計については，いくつかのケースを分析させ，フィジカル設計については研修生に最も適当と思われるもの1件を実習させるのがよいであろう。

(2) 各ケースの完全な記述については参考文献を参照すること。ここでは典型的な成果物等を主として表の形で示したので，これにシナリオや補足データを追加して，米国や英国の事例を日本的に手直しすることも考えられる。

1 問題の定義

プロジェクトの目的別に分類した，問題の定義例を次に示す。

(1) 法律または政府機関等の決定によって実施が強制されるもの。

銀行支店におけるスタンド・アロン・システムの開発（ケース1）＊1

海外支店であるため，該当国の中央銀行の指示によってMICRコードの統一を行なう必要が生じた。

新機種の導入とあわせて，次の業務の処理も行なうシステムを開発することとした。

① 預金会計処理

② 取引処理

③ 銀行内の給与計算

④ 内部会計の大部分

(2) 同業他社が実施の傾向にあり，社会的になかば強制された形で実施せざるを得ないもの。

銀行オンライン・システムの設計（ケース3）＊2

他行との競争から，顧客に関する情報をオンライン・リアルタイムで管理するシステムの必要性が生じた。

銀行内のシステム開発グループに対象システムの設計とハードウエアの選定をさせると同時に，その妥当性を評価するために外部のコンサルタント会社へも調査を依頼した。

委託先のC社は，評価プログラムSCERTによるシステム評価の経験が豊富である。

(3) 業界の見通し，業容の拡大から予測して，ニーズを先取りして実施するもの。

研究開発会社の計算機リプレース（ケース2）＊3

大型計算機を2～3台もつ研究開発およびコンサルティング会社が，計算業務の増大に

応じて新型機の導入を企画した。

リプレースの必要性は次のとおりである。

① コンピュータの利用率が85%に達した。

② バッチ処理だけでなく，ＴＳＳや遠隔処理が必要になった。

コンピュータ利用上の要件は次の諸点から分析しなければならない。

① 業務の現状からの要件

② 業務の将来予測からの要件

③ システム移行上の要件

④ 結論として，次期機種の基本的要件

この企画はコンピュータ・サービス部が調査立案し，各事業部の代表から構成される委員会に報告するものとする。

なお，この会社のコンピュータ使用料は実績に応じて社内振替され，もし社外のコンピュータを利用した方が安価ならば，それも認められている。

(4) コスト低減を目的としたもの。

この目的で開発されるシステムが最も多いと思われるが，対象とする組織および業務処理の内容について十分に知る必要があるので，これが実習の障害とならないように配慮する必要がある。

石油会社のマーケティング会計システム（ケース５）＊4

収益改善のためのプロジェクトに着手し，次の３部門で推進することとなった。

① 収益改善プロジェクト・チーム

会社の活動能率を改善するためのツールや技法を開発または，その種のシステムを導入する。

② 計算センター

③ マーケティング会計部

本プロジェクトは，財務情報の収集，処理，作成，配布を行なうことによって，次の事項を実現することが目的である。

① 営業所におけるコストを年間4.5億円低減する。

② 経営情報の改善をはかる。

③ マーケティング会計のための適切なコンピュータ・システムの設計と導入を行なう。

段ボール製造会社のスケジューリング自動化（ケース９）＊5

多種多様な注文を，納期，製品寸法，材料仕様を基準に分類し，自動スケジューリングを行なうことを企画した。

これによって次のような経済効果が期待される。

① 裁断屑の減少による歩留りの向上。

② 段ボール製造機の利用率向上。

③ 在庫品品種の減少。

④ スケジュール作成の迅速化。

費用最小となる最適スケジュールは，ＬＰを用いて求められるとの見通しはついた。

しかしこの会社はＥＤＰの経験がなく，コンピュータ利用の経験をもつ従業員もいない。

そこでＬＰのパッケージと小型コンピュータをレンタルするつもりで，コンサルタントに相談した。

コンサルタントはこれに対して，各種の必要作業の工数や設備費等の見積りを報告した。

(5) 時間短縮により，業務遂行の効率化，質の向上をねらったもの。

警察の情報検索とデータ処理（ケース６）＊6

次のような機能をもったシステムを開発する。

① 捜査中のパトロール係に対して，容疑者や車に関する情報をリアルタイムで送る。

② トップから各分署の署長まで，各レベルの人々に対して管理資料を提供する。

(6) 業務処理のロジックを改善し，シミュレーション等を行なって質を改善するもの。

警察のパトロール要員配分システム（ケース７）＊7

次のような機能をもったシステムを開発する。

① パトロールの無線呼出し依頼の内容や，それに対する活動状況などを記憶する。

② 地域別，時間帯別の呼出しの分布を予測する。

③ 要員配分計算を行ない，必要最小要員数や最適割当を決定する。

④ 各種の報告書を作成する。

動力工具メーカーの製造管理システム（ケース８）＊8

システム開発の目的は次のとおりである。

① 仕掛品在庫管理の改善。

② 情報のタイミングと正確性を向上させ，スケジューリングと差立を改善する。

③ 工場内の仕掛部品の所在について，正確かつ最新の情報を得る。

システムの機能概要および処理手順の構想は次のようなものである。

① 需要予測により年度別の工具販売数量を予測する。

② 購買および製造所要数量に分解する。

③ 在庫量を考慮し，生産数量を求める。

④ 機械種別の日程計画を立案する。

⑤ 作業票と工程表とによって製造工程を統制する。

⑥ 作業終了のつど，工場内のデータ収集用端末で報告させる。

機械工具メーカーの生産計画（ケース10）＊9

販売の長期予測と受注実績から，工場の設備計画を行なっている。しかし各々の機械設備がどの程度の仕事量をこなせるかを手計算で算出しているため，正確な生産能力が把握できなかった。

そこで既に導入されているコンピュータを用いて，ある生産計画が与えられた時に工場の機械使用時間，労働工数，最低必要在庫量などが求められるシステムを開発することとした。

生産計画は任意の長さの期間を対象とすることができるようにし，数案の計画を処理して比較し，最適のものを選択できるようにする。

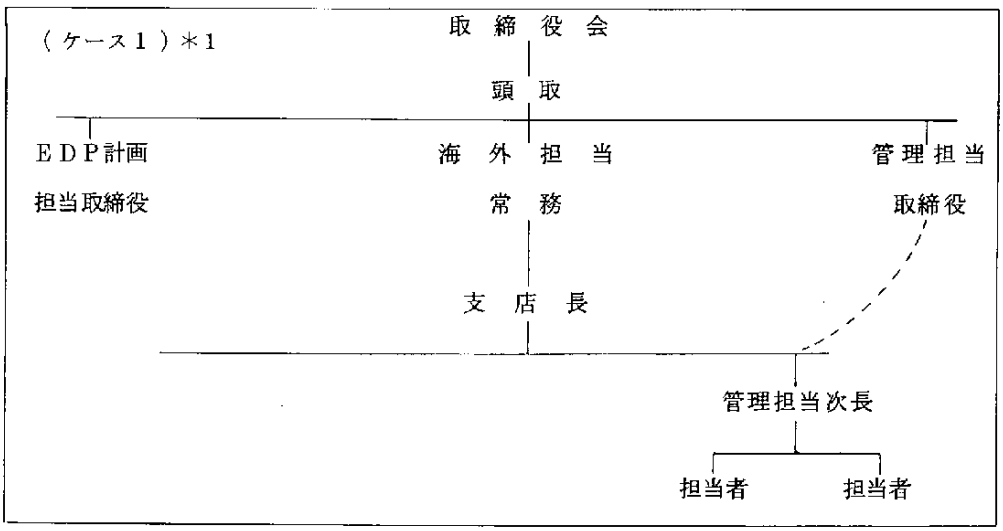
(7) 既存システムの見直しを行ない，エラー防止，信頼性向上，実施部門の拒絶反応解消をねらうもの。

電力会社の株主記録システム（ケース４）＊10

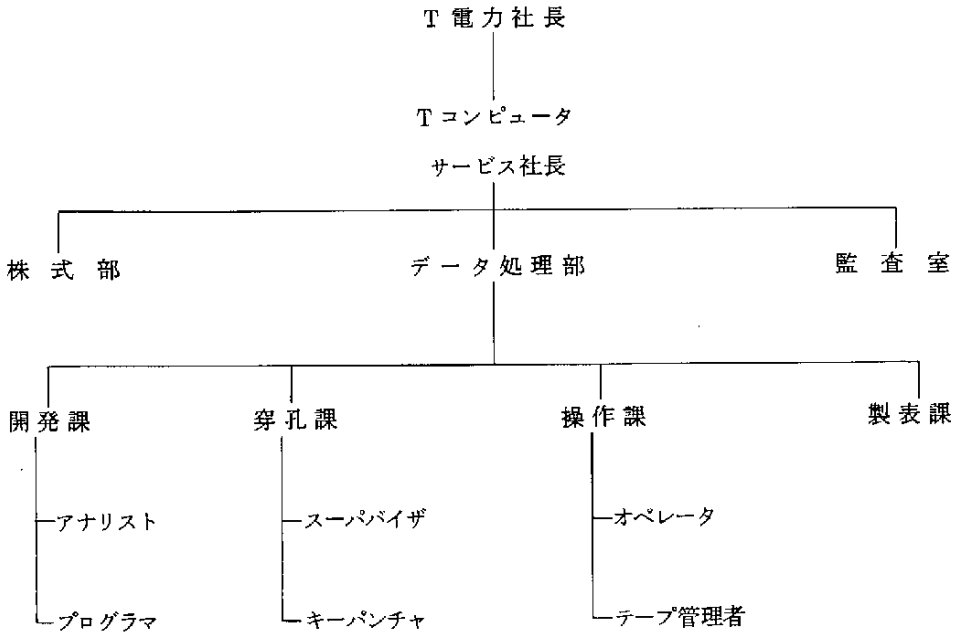
２年前に導入したシステムの再調査を行ない，システムの管理面とくにエラー発生の防止と正確性向上のための手続きの見直しを行なう。

管理手段は，人手による事務手続き上のものと，コンピュータによるチェックとがある。

2 システム開発の組織



（ケース４）＊10



3　現状分析

(1)　ヒアリングの技術

ヒアリング要領の具体例を次に２件示す。

署長に対するヒアリング（ケース６）＊6

管理報告をどのように利用しているか，どのような情報が有益か，という点を中心に次の要領で質問を行なった。

① これは大変重要な書類らしいですね。

② この数字のちがいを，どう説明なさいますか。

③ 他の管区の活動に関する情報はあまり参考にならないと思いますが，どうですか。

④ 無線指令週間分析を併用すれば，先週の状況をとりまとめるのに便利だと思いますが，いかがですか。

⑤ たとえば先週の窃盗は4.7％ですが，これは全体の水準とくらべて良いのですか。

⑥ これらの報告書の情報は，管区を管理するのに役立っているのでしょうね。

⑦ これらの情報を月間報告書にも繰り返して記載する必要があるでしょうか。

⑧ 他にも数々の警察期別報告書を受取っているのではありませんか。

⑨ 期別報告書は部下の成績評価に使えるではありませんか。

⑩ これらの報告書を定期的に受取っていますか。

⑪ コンピュータによる報告制度の発展を期待していますか。

署長に対するヒアリング（ケース６）＊6

① 朝どんな仕事から始めますか。

② そのような報告書を見るのでしたら，無線指令日報は必要ありませんか。

③ この報告書を見るには相当の時間をかけているようですが。

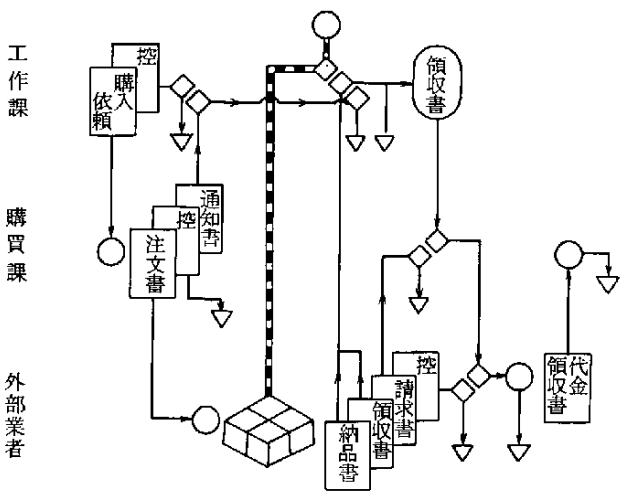
④ 無線指令の週間分析表も配布されていますか。

⑤ この報告書を使わないとしたら，どうやって部下の評価をするのですか。

⑥ あなたがＥＤＰマネジャーの相談役になって，予算も6,000万円追加されたとした，ら，どんな助言をしますか。

(2) 事 務 分 析

a．事務分析フローチャート



IBM Message Sheet

実際データ・カード
原価カード
半製品原価カード
半製品実際原価カード

K0013
M3500T

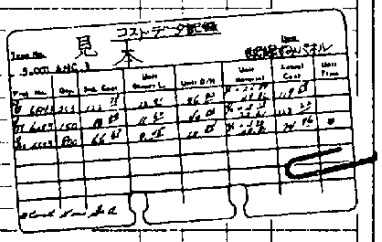
3×5 トラック・インデックス・カード　手書き

製品の価格については社外秘

CONTENTS

No.	Data Name	Frequency	Characters	Kind	Origin
01	部品番号	1	12	AN	
02	部品名	1	40	AN	
03	日付				

コスト・データ記録
見本
5,000 ANC.3
部品番号のパネル



3/12/'68　価格表　1 of 1

七星電子工業

IBM File Sheet

原価ファイル　F-4010
原価計算課 C 7 on 7　3×5 トラック・インデックス用の回転式キャビネット

部品番号順，日付順

カードは廃棄しない

001-57X，001-63X で使用
カードは廃棄しないきまりになっているので，ファイルは増える一方である。

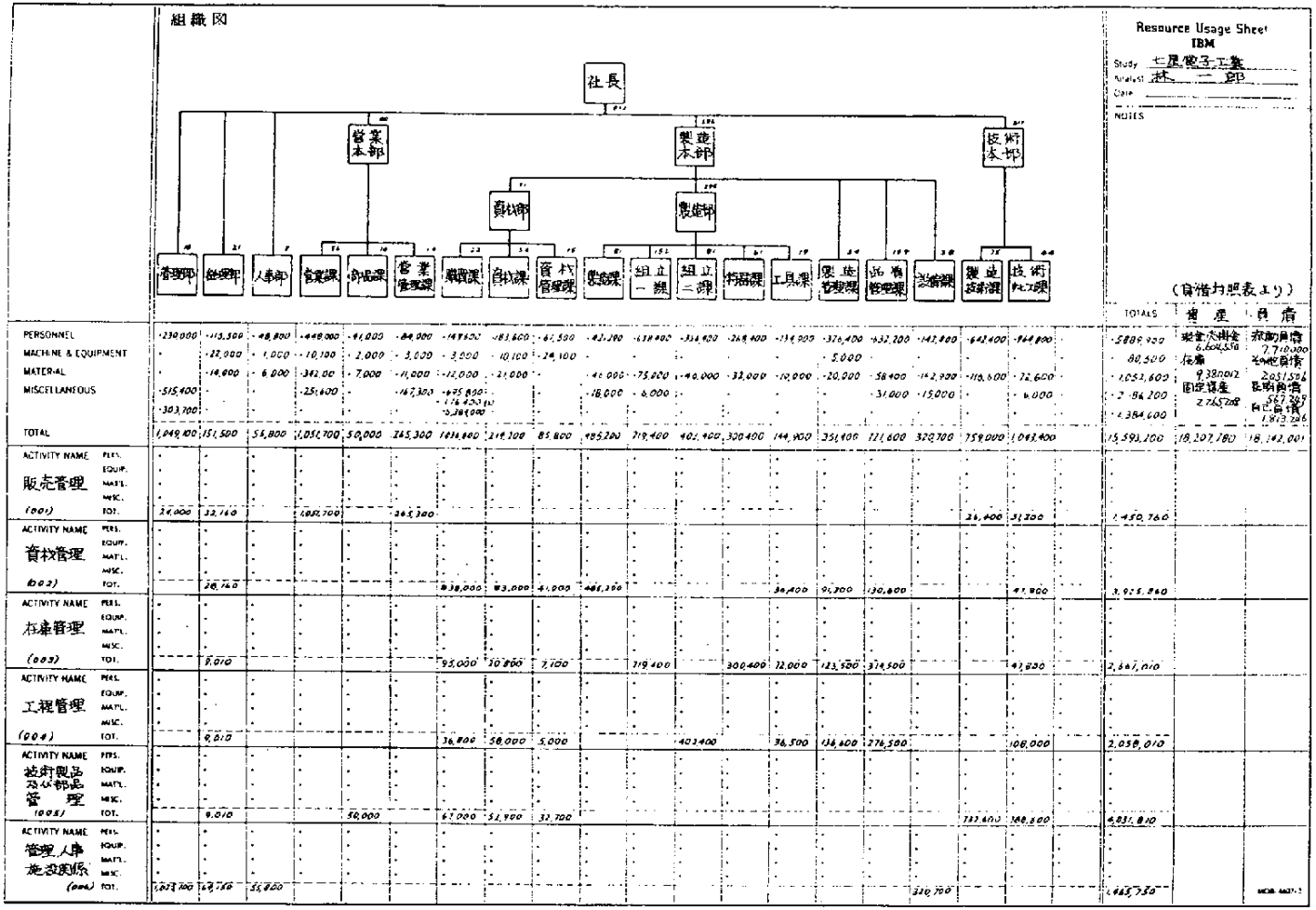
CONTENTS

実際原価データ・カード記録　1,000 K以上，ファイルは毎月4,200枚の割で増えていく。

（見本カードを参照）

鈴木太郎　価格表　1 of 1

BUTODALE


組織図
社長
営業本部
製造本部
技術本部
資材部
製造部
管理部
総務部
人事部
営業課
倉庫課
営業管理課
購買課
資材課
資材管理課
製造課
組立一課
組立二課
特品課
工具課
製造管理課
品質管理課
設備課
製造技術課
技術サービス課
Resource Usage Sheet
IBM
Study 七星電子工業
Analyst 林 一郎
Date
NOTES
（資源月照表より）
PERSONNEL
MACHINE & EQUIPMENT
MATERIAL
MISCELLANEOUS
TOTAL
TOTALS
資産
負債
販売管理 (001)
資材管理 (002)
在庫管理 (003)
工程管理 (004)
被術製品及び部品管理 (005)
管理人事施設関係 (006)

c. 業務量分析の例

（ケース１）＊1

	支　店	本　店
当座預金の口座数	35,000	283,000
当座預金の１日平均取引数	28,000	347,000
手形交換の１日平均取引数	25,600	987,000
定期預金の口座数	291,000	169,000
定期預金の１日平均取引数	2,500	3,500
有価証券取引の約手	6,900	53,000
支払延期の小切手と証書	19,800	23,100
債券数	10,700	243,000
証券の１日平均収集数	11,200	57,000
安全金庫の貸付数	21,300	9,800

（ケース10）＊9

製品の種別

80種，コードは３桁の英数字

製作に使用する機械数

110台，コードは001～110

部品点数

7,500，コードは11桁の英数字

ある部品を使用する製品数

最低１，最高80，平均15

１つの部品に必要な作業

最低４，最高30，平均10

d. コンピュータ使用の現状分析例

(1) 現有機種とその用途

（ケース２）＊3

機　種	用　途
IBM　360/75　2台	・大規模な数値計算 ・大規模なシミュレーション ・大規模なＬＰ計算
UNIVAC　1108　1台	・オンライン・システム開発 ・ＴＳＳシステム開発 ・一般の数値計算

機　　種	用　　途
IBM　370/148　1台	・対話形式の計算処理 ・一般的データ処理 ・プログラム開発
IBM　360/20　1台	・カード→テープ等の変換 ・小規模の事務計算

(2) 使用言語

（ケース２）＊3

言　　語	比　　率
FORTRAN	53％
PL／I	19％
COBOL	25％
アセンブラ	3％

(3) 部門別（業務別）コンピュータ使用割合

（ケース２）＊3

	1976年	1977年
技術部（大規模な技術計算）	28.7％	29.8％
社会システム部（計画立案，経済分析）	22.8％	25.6％
宇宙・通信部（シミュレーション）	28.4％	22.0％
生体・医用システム部（統計解析オンライン）	12.3％	14.5％
管理部（一般会計処理等）	7.8％	8.1％

(4) その他分析項目例

アセンブラ等の機種依存要因の割合

買取の場合は償却状況

操作，保守，周辺機器の費用

4 将来予測

(1) 動向の定性的予測例

（ケース２）＊3

当社の今後のコンピュータ利用の傾向を長期にわたって予測することは困難である。しかし業界誌等の予測を参考にして研究スタッフが討論を行なった結果，次のような結論が得られた。

① 数値計算以外の一般的業務処理の漸増

② 対話形式による処理の急増

③ リモート・バッチ・サービスの急増

④ BASICやAPL等の言語利用の増加

⑤ グラフィック・ディスプレイ等の周辺端末装置に対する要求

⑥ コンピュータ・ネットワーク化

⑦ データ・ベース整備に対する要求

(2) 業務量・コンピュータ利用率の推移予測

（ケース２）＊3

機　種	1967年	1968年	1969年	1970年	1971年	1972年
IBM 7094, (2)……	85	110				
IBM 360／65 ……	33	43	56	73	95	124
IBM 360／75 ……	21	27	35	45	59	77
IBM 360／85 ……	8	11	14	19	24	32
CDC 6400 ……	27	35	46	60	78	101
CDC 6500 ……	20	25	33	43	56	72
CDC 6600 ……	9	12	16	21	27	35
UNIVAC 1108 …	27	35	45	59	76	99

次の前提に基づいて計算した。

1. 2台の7094の負荷が1967年に85パーセントである。
2. コンピュータ使用量は，毎年30パーセントずつ増えるものとする。
3. 機種ごとの能力は，ベンチマーク・テストによって得られた数値をベースに計算する。

5. システム分析

(1) 経済効果の分析

a. 人件費節減分析

（ケース５）ホ4

	置換え要員数			月額（単位千ドル）	コンピュータ料金（単位千ドル）	その他の節減額（単位千ドル）	デザイン投資（単位千ドル）	計画合計（単位千ドル）	合計（単位千ドル）
	営業所	本社	合計						
1. 売掛金	47		47	29	2.5		24.2	34.7	58.9
2. ディーラー会計	71	1	72	43	3.5		47.4	33.0	80.4
3. 経費分析		9	9	5	1.5		47.0	41.9	88.9
4. 販売活動	6	3	9	5	.2		10.5	6.3	16.8
5. マーケティング統計	6	23	29	19	2.1		66.5	58.2	124.7
6. 在庫管理	13	2	15	9	.9		20.3	19.0	39.3
7. 物品税会計		10	10	6	1.6		21.7	25.5	47.2
8. マーケティング外業務					2.0		31.9	21.8	53.7
9. 給与計算	8	9	19	10	.5		74.8	60.1	134.9
10. 買掛金	12		12	8	1.0		29.1	22.2	51.3
11. 会社直営サービスステーション	2	1	3	2	1		6.6	3.4	10.0
12. 固定資産会計					1		14.6	9.4	24.0
節減額総計	165	58	223	136					
データ伝送とインプット	(34)	21	(13)	8			86.7	18.8	105.5
作表機器の取替え						10			
所要の新従業員（事務関係）		(18)	(18)	10					
正味節減額	131	61	192	113					
フィラデルフィアのRAMACの取替え				118		7			
データ伝送					18				
小計						17			
投資総計					34		481.3	354.3	835.6
正味節減月額						101			

（注）コンピュータ時間の料金請求は処理に１時間95ドル，プリンティングに１時間20ドル

マーケティング・プロジェクト監督費用は投資に配分されていない。地方および本社買掛金については，信用貸は入っていない。

給与計算については，信用貸は入っていない。

時間当りコスト……処理１時間75ドル，プラス間接費20ドル，プリンティング１時間10ドル，プラス間接費10ドル

マルチプレクサー月1,500ドル，プラス間接費10ドル

b．業務別分析

（ケース５）＊４

費用と節減額見積
（単位千ドル）

優先割当　アクティビティ	開発費	年間総節減額	年間運営コスト	年間正味節減額	総計節減額内の%
Ⅰ　売掛金	300	350	210	140	21
Ⅰ　ディーラー会計	235	440	275	165	26
Ⅰ　費用分析	90	85	10	75	5
Ⅰ　販売活動	15	65	35	30	4
Ⅰ　マーケティング統計と分析	230	230	10	220	14
Ⅰ　在庫管理	75	155	10	145	8
Ⅰ　物品税会計	100	95	10	85	6
Ⅰ　マーケティング外業務	60	10	10	—	—
小　計	1,105	1,430	570	860	84
Ⅱ　給与計算					
Ⅱ　給与計算部門と臨時要員給与	160	200	20	180	10
Ⅱ　生産および製造	—	—	10	(10)	5
Ⅲ　仕入買掛金	80	85	10	75	
Ⅳ　会社直営サービス・ステーション会計	30	20	10	10	1
小　計	270	305	50	255	100
範囲（全プログラムについて）	1,250 − 1,500	1,700 − 1,900	600 − 650	1,100 − 1,250	

C．キャッシュフロー分析（ケース５）＊4

この月で終わる四半期		EDPプロジェクト開発費						経常運営費						四半期当り按分総額	経常節減額			累計キャッシュ・フロー
年	月	出張旅費	訓練	機器および消耗品	プロジェクト要員	コンピュータ時間コスト	開発合計	コミュニケーション設備	計算センター	プログラム・メンテナンス	会計管理要員	会計管理機器消耗品	経常合計		人事	機器	総節減額	
1964	6月	$ 600	$ 450		$ 10,950		$ 12,000							$ 12,000				$ (12,000)
	9月	600	450	$ 11,500	25,450		38,000							38,000				(50,000)
	12月	600	450	1,500	34,450		37,000							37,000				(87,000)
1966	3月	600	450	1,500	56,250		58,800							58,800				(145,800)
	6月	600	450	11,000	82,200		94,250							94,250				(240,050)
	9月	600	450	11,000	106,050		118,250							118,100				(358,150)
	12月	600	450	1,500	105,750	$ 1,540	109,840							109,840				(467,990)
1966	3月	600	450	1,500	105,750	4,900	113,200							113,200				(588,190)
	6月	600	450	1,500	104,750	1,120	108,420							108,420				(689,610)
	9月	600	450	1,500	99,750	9,660	111,960							111,960				(801,570)
	12月	1,100	450	1,500	100,750	27,580	131,380							131,380				(932,950)
1967	3月	4,100	450	1,500	106,750	57,060	169,860	$ 4,323			$ 25,200	$ 3,000	$ 32,723	202,583				(1,135,533)
	6月	15,600	450	1,500	126,750	19,020	163,320	35,695	$ 19,020		37,800	3,000	95,515	258,835	$ 62,500	$ 16,000	$ 78,500	(1,315,868)
	9月	15,600	450	1,500	120,750		138,300	62,424	28,530		37,800	3,000	131,754	270,054	206,000	24,000	229,000	(1,356,922)
	12月	10,800	450	1,500	99,950		112,500	71,597	28,530		37,800	3,000	140,927	253,427	325,000	24,000	349,000	(1,261,349)
1968	3月	5,100		1,500	18,900		25,500	72,522	28,530		37,800	3,000	141,852	167,352	367,500	24,000	391,500	(1,037,201)
	6月	1,700		500	6,300		8,500	72,522	28,530	$ 12,750	37,800	3,000	154,602	163,102	384,375	43,500	427,875	(772,428)
	9月							72,522	28,530	12,750	37,800	3,000	154,602	154,602	384,375	43,500	427,875	(499,155)
	12月							72,522	28,530	12,750	38,000	3,000	154,602	154,602	384,375	43,500	427,875	(225,882)
1969	3月							72,522	28,530	12,750	37,800	3,000	154,602	154,602	384,375	43,500	427,875	47,390
	6月							72,522	28,530	12,750	37,800	3,000	154,602	154,602	384,375	43,500	427,875	320,664
	9月							72,522	28,530	12,750	37,800	3,000	154,602	154,602	384,375	43,500	427,875	593,937
	12月							72,522	28,530	12,750	37,800	3,000	154,602	154,602	384,375	43,500	427,875	867,210
1970	3月							72,522	28,530	12,750	37,800	3,000	154,602	154,602	384,375	43,500	427,875	1,140,483
	6月							72,522	28,530	12,750	37,800	3,000	154,602	154,602	384,375	43,500	427,875	1,413,756
	9月							72,522	28,530	12,750	37,800	3,000	154,602	154,602	384,375	43,500	427,875	1,687,029
	12月							72,522	28,530	12,750	37,800	3,000	154,602	154,602	384,375	43,500	427,875	1,960,302
1971	3月							72,522	28,530	12,750	37,800	3,000	154,602	154,602	384,375	43,500	427,875	2,233,575
期間累計		$ (59,800)	$ (6,750)	$ (52,000)	$ (1,311,500)	$ (120,880)	$ (1,550,930)	$ (1,117,025)	$ (446,970)	$ (153,000)	$ (630,000)	$ (51,000)	$ (2,397,995)	$ (3,948,925)	$ 5,572,500	$ 610,000	$ 6,182,500	$ 2,233,575

6. 企画書・提案書

(1) 作成報告書一覧表

（ケース９）＊5

報告書名	頻度	コピー部数	配布部門
地区別総元帳	週	3	販売
備品顧客先売上	週	2	販売
軍隊・問屋グループ(2)	月	6	販売
チェーン・ストア・要約	月	6	販売
グリーティング商品以外の代理店売上	月	6	販売
手数料付備品	月	6	販売
売掛分析（４部分）	月	6	販売
売上減／未販売	年＊	4	販売
売上減理由	年	4	販売
備品在高報告	４半期	4	販売
Ｔ／Ａ地区要約(2)	月	6	販売
Ｔ／Ａセールスマン(2)	月	6	販売
Ｔ／Ａチェーン・ストア(2)	月	4	販売
元帳・出荷	週	2	会計
顧客比較	週	2	会計
州税・利用税報告	月	4	会計
勘定集計表	月	4	会計
返品・値引／品名コード	月	6	会計
顧客別月間売上	月	4	会計
郡・市税報告	月	4	会計
キャンセル	月	4	販売
ウースワース	４半期	6	販売
取引種別	年	3	販売
会社売上記録	月	4	会計

＊ 必要に応じて作成

(2) 作成報告書の内容説明

（ケース９）＊5

報告書名	内容説明	データ	データ源	注
地区別総元帳	季節用・日常用製品の総元帳を集計，地区ごとの合計は週別および累積で計算される	1. 元帳(4)および出荷(1)レコードのみ 2. プログラム・テーブルで定義されたトランザクション・コード	週間累積テープ，前週の本年度累積額地区別集計カード	集計カードの更新は季節用品を含められる
備品勘定売上高	選択された得意先の販売取引を，得意先別の送状ごとに整理する。得意先別売上高合計　総合計を出す		週間累積テープ，選択カード・デック，顧客ファイル，送状レコード	明らかに，これは未償却備品残高のある顧客に対する販売報告書である
軍隊・問屋グループ(2)	軍隊と問屋グループ用に２組の報告書が作成される。１つは総出荷高を示し，もう１つは出荷に対する売掛を示す。各報告書は， a) 軍隊ー取引のある顧客勘定 b) 軍隊ー現在取引のない（睡眠）顧客勘定 c) 問屋グループー取引のある顧客勘定 d) 問屋グループー現在取引のない（睡眠）顧客勘定の４つに分けられて作成される。各部分の中では，店ごとに店名　住所が示され，軍隊または問題グループ・コードごとの当年および前年の当日までの累積総出荷高（または売掛金）を示す。グループ・コードごとの合計，各部分ごとの総合計を出す。各部分の最後には，グループ・コードの合計と総合計の要約がある	Ｍ－１テープからの出荷および出荷売掛レコードはマスター・ファイルを更新するために利用される。更新されたマスターは，報告書作成のために前年のマスターと顧客ファイルと組合わされる。	Ｍ－１テープ，前月の累積総額または売掛の軍隊・問屋グループ・マスター	1. 暦年基準 2. 小売店での表示価格 これは，Ｍ－１テープからの全金額が２倍されることを意味する
チェーン・ストア報告書	チェーン・コード別の店番号ごとに，季節用・日常用商品の本年累積および前年累積の純出荷額をあげる。各部分ごとに総合計が出される。各部分の最後にチェーンの合計と総合計をまとめる。	Ｍ－１テープからの出荷レコードがチェーン・ストア・ファイルの前月の累積を更新するために使われる	Ｍ－１テープ，前月の本年累積額と前年累積額のチェーン・ストア・マスター，顧客ファイル，チェーン・ストア・カード・ファイル	1. チェーン・ストア・ファイルに指定されたチェーンに対して，チェーン名の入っている要約レコードが後で利用されるために作られる。その他のときは，要約レコードはチェーン名なしで作成される 2. 暦年に基づく
チェーン・ストア要約	チェーン・ストア報告書からチェーン名のアルファベット順にチェーン合計を列挙する	チェーン・ストアがあるところのすべての要約レコード	チェーン・ストア報告書のプログラムで作成された要約レコード・ファイル	

(3) システム提案書

（ケース９）＊5

マーケティング部長　　1969年8月29日

アレン・ボードアン殿

ボブ・トルートより

販売報告システムについて

現行の販売報告システムに代わって，私から提案した種々の販売報告書についての草案を添付します。各報告書は，

頻　　度

順　　序

内　　容

に分けて説明してあります。

提案の販売報告システムは，過去２ケ月間，私の２名のスタッフによって行なわれた販売報告システムに関する設計調査の直接的な成果です。進行中の検討会は，すでに貴殿と貴部の主要メンバーとともに開催されました。調査のベンチマークの１つである現行の販売報告システムに関する完全な棚卸しが完成しました。この報告書の目録に基づいて，いまわれわれは新しい販売報告システムを開発することに関心を払わねばなりません。私の提案は，新しいシステムの設計のための出発点を与えようとするものです。

貴殿の御都合のよい時に，貴殿と貴部の主要メンバーの方々とで，この新しいパッケージ全体について御検討下されば幸いです。ただし，われわれの販売報告システムの変更についてのきわめて高い意欲を減衰させないためにも，早い機会に検討されることをおすすめ致します。

草案：販売報告書

1. 週間売上高報告書

頻度 ― 毎週（会計年度に従う）

順序 ― 国，地方，地区別セールスマン

内容 ― 当期，商品ライン別に整理された元帳と売掛の顧客取引番号の順序にまとめられたリスト；在庫，不良品，販売促進用品，およびその他に分けられた製品系列別の毎日の総元帳額，日常用および季節用注文数，日常用および季節用品ごとの製品系列別に並べられた総元帳額，売掛適正額，純元帳額，売掛不適額，および最終元帳額合計

当日までの年度累積高，季節用および日常用注文数；日常用および季節用ごとの製品系列別に並べられた総元帳額，売掛適正額，純元帳額，売掛不適額，および最終元帳額合計；割当額と割当額に対する純元帳額合計の比率（百分率）

2. 季節用品売上減報告書

頻度 ― 販売部門のスケジュールによる（一般的にはシーズン終了後間もなく）

順序 ― 一定のシーズンに対する国，地方，地区，地域別顧客

内容 ― 現シーズンと前シーズンに対する純元帳額と純出荷額合計の双方またはそのいずれかについて

注）

(1) 現シーズンの元帳額と出荷額の双方またはそのいずれかを前シーズンのそれらと比較して，ある比率限界（予め決めておく）以上減少していた場合にだけ，その顧客をリストアップする。

(2) 季節用品の売上増を示すようにも修正できる。

3. 日常用品売上減報告書

頻度 ― 4半期ごと（会計年度に従う）

順序 ― 国，地方，地区，地域別顧客

内容 ― 当期までの現会計年度累積額と前年度累積額

注）

(1) 現会計年度累積出荷額を前会計年度のそれと比較して，ある比率限界（予め決めておく）以上減少していた場合にだけ，その顧客をリストアップする。

(2) 日常用品の売上増を示すようにも修正できる。

4. 季節用品未販売報告書

頻度 ― 販売部門のスケジュールによる（一般的には，シーズン中1～2回）

順序 ― 一定のシーズンに対する国，地方，地区，地域別顧客

内容 ― 当日までの前シーズンに対する純元帳累積額合計

注）

(1) 現シーズン中に対して取引のない顧客をリストアップする。

(2) 現シーズンに購入しなかった顧客を固定客と新規の別にリストアップする。

5. 季節用品新規取引報告書

頻度 ― 定期的（たとえば，シーズン中1～2回）

順序 ― 一定のシーズンに対する国，地方，地区，地域別顧客

内容 ― 新規顧客についての現シーズン中の当日までの純元帳額合計

6. 日常用品新規取引報告書

頻度　　月または4半期

順序　　国，地方，地区，地域別顧客

内容　　新規顧客についての現会計年度の当日までの純出荷額合計

7. 系列別季節用品顧客報告書

頻度　　定期的（たとえば，シーズン中1～2回）

順序　　一定シーズンに対する国，地方，地区，地域別製品系列

内容　　現シーズンおよび前シーズンの当日までの既存顧客と新規顧客の注文数と純元帳額合計

8. 系列別日常用品顧客報告書

頻度　　月または4半期

順序　　国，地方，地区，地域別製品系列

内容　　現会計年度および前会計年度についての既存顧客と新規顧客の注文数と純出荷額合計

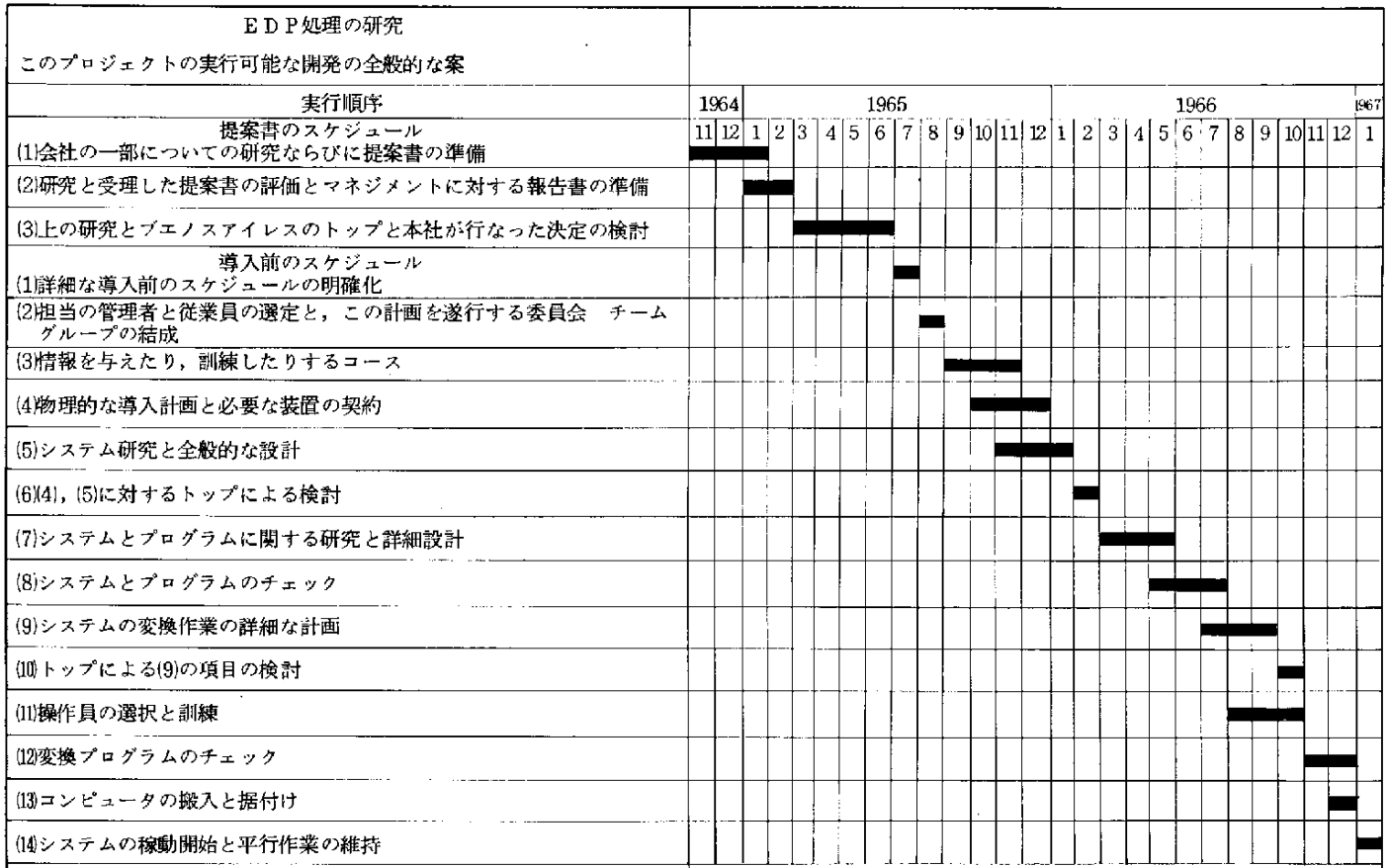
9. 確定売上分析

頻度　　シーズン終了後3カ月

7. システム開発スケジュール

(1) バーチャート

(ケース9)＊5



(2) 作業計画の時間見積り

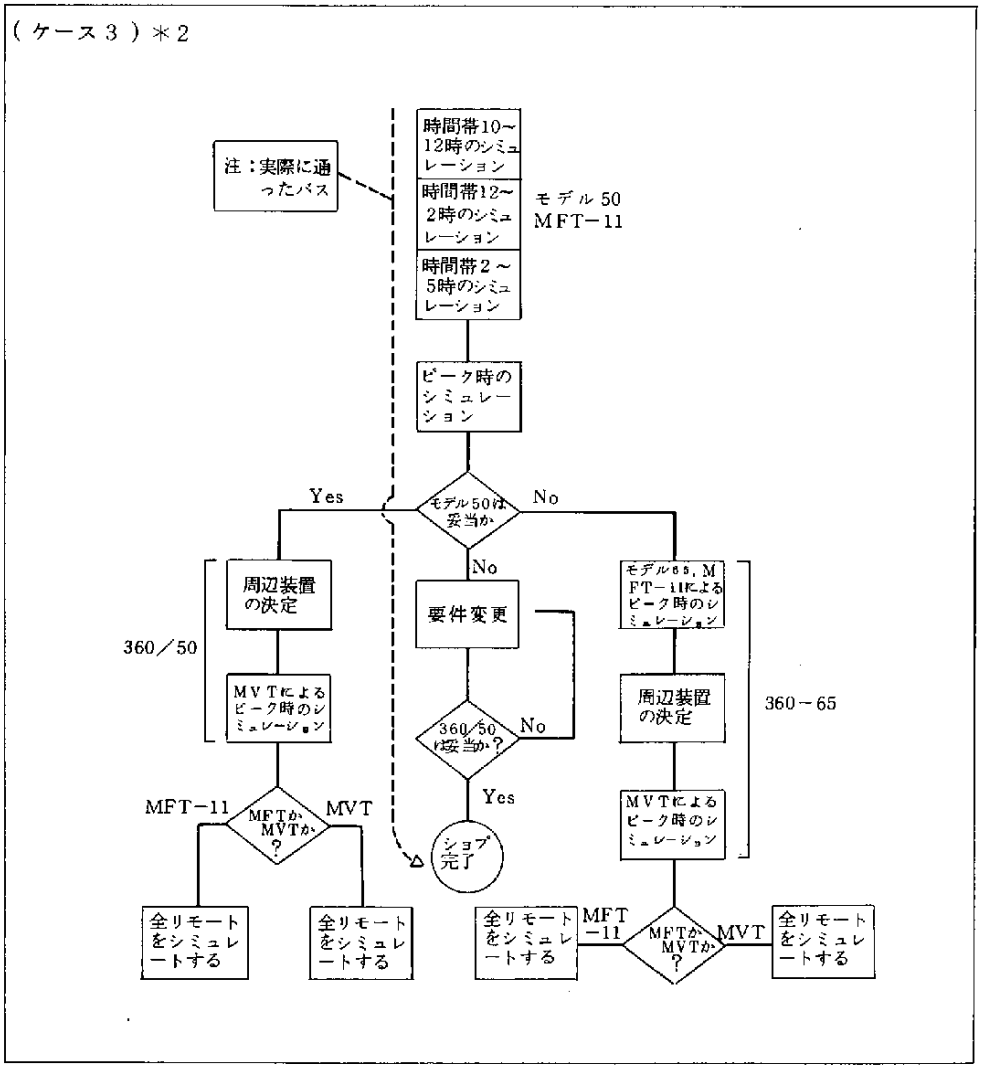
（ケース９）＊5

図５－７　トルート氏のシステム設計メモ

システム設計メモ		
アプリケーション　販売報告書	アプリケーション分野	インデックスNo.
主題　現行販売報告システムの調査作業計画		日付　1969年6月20日
参加者　R.W.Beckar　D.J.Robinson		
目的　新システムの情報に対する要件を決定することに主眼をおいて，現行販売報告システムの調査のための仕事を概略説明し，予定を示す。		

説明	人・日	完了日 計画	実績
Ⅰ　コンピュータ・システムの最新の流れ図を書く。	……	5/2/69	
Ⅱ　システムから発行される報告書を説明する。	……	6/20/69	
A　記述説明			
1. 週間合計・販売報告書			
2. 月間合計・販売報告書			
3. T／A報告書			
4. チェーン店・軍隊・問屋グループ			
5. 備品			
6. 売上減少報告			
7. その他			
B　報告書見本	……	6/2/69	
C　配布経路図：誰へ，何部，いつまでに	1 1/2	6/24/69	
D　元帳記入と出荷の間の時間遅れの説明および報告書の体裁	4	7/15/69	
Ⅲ　システムへの入力の説明			
A　累積およびM－1テープの説明をフィールドごとに書く。	3	7/9/69	
Ⅳ　例外状態に特に注意して，コードの値を設定する。			
A　顧客番号	1 1/2	6/27/69	
B　セールスマン番号	1 1/2	6/27/69	
C　トランザクション・コード	4	7/3/69	
D　売掛品名・コード	3	6/13/69	
E　売掛タイプ・コード	1/2	6/27/69	
Ⅴ　検討・完了	3	7/15/69	

（ケース３）＊2


注：実際に通ったパス
時間帯10～12時のシミュレーション
時間帯12～2時のシミュレーション
時間帯2～5時のシミュレーション
モデル50 MFT−11
ピーク時のシミュレーション
モデル50は妥当か
Yes
No
周辺装置の決定
360／50
MVTによるピーク時のシミュレーション
MFT−11
MFTかMVTか？
MVT
全リモートをシミュレートする
全リモートをシミュレートする
No
要件変更
360／50は妥当か？
No
Yes
ショブ完了
モデル65，MFT−11によるピーク時のシミュレーション
周辺装置の決定
MVTによるピーク時のシミュレーション
360−65
全リモートをシミュレートする
MFT−11
MFTかMVTか？
MVT
全リモートをシミュレートする

9. ハードウエアの定性的評価

（ケース２）＊3

ソフトウエア比較表

比較項目＼機種	A	B	C
システム名	X	Y	Z
完成後年数	2年	3年	0.5年
基本機能等			
多重プログラミング	可	可	可
リモート・バッチ処理	可	可	可
タイム・シェアリング	不可	不可	可
データ・ベース管理	可	未発表	未発表
制御用言語	多数あって，きわめて複雑	現有機種より多く必要	現有機種より多く必要
CPUコンソール	タイプライター（多重プログラミングに不適当）	ディスプレィ（操作上きわめて便利）	タイプライター（多重プログラミングに不適当）
現有機種とのFORTRAN互換性	有	有	有
COBOL	使用できるが，十分にテストしたことはない	無	使用できるが十分にテストしたことはない
LP	使用でき，現有機種より優れている	無	無
ソート	使用できるが，テストしていない	使用できるが，テストしていない	使用できるが，テストしていない
会話型FORTRAN	考慮されていない	考慮されていない	使用できるが，未完成である
統計・数学システム	FORTRANで使用可能（高速フーリエ変換あり）	未発表	マトリクスや一般数学用プログラムあり
ユーザ・ライブラリ名	PQR	UVW	IJK
汎用Ｉ／Ｏパッケージ	入出力のシステムマクロ有（非常に使いやすい）	ファイル入出力のために，制限の多いユーティリティプログラムが使える	制約された形でしか使えないが，データ取扱い用のパッケージがある

10. ハードウエアの定量的評価

(2) 経済分析

（ケース１）＊1

年間経費節減額の概算

項目	節減額（万円）
中央手形交換業務	
1つの課の課長と6人の課員	2,600
7台の会計機の減価償却と保守	420
当座勘定	
1つの課の課長と48人の課員	19,000
35台の専用機の減価償却と保守	1,600
事務所	
1人の所長と26人の所員	10,000
7台の照合機の減価償却と保守	1,700
タブ装置	
1人の長と20人の従業員	9,300
現有のタブ装置のレンタル	6,600
合計	51,220

（ケース１）＊1

システム導入コスト

項目	コスト（万円）
EDPセンターの要員	15,000
10人のコンサルタント兼管理要員	5,400
機器の減価償却と保守	21,000
保険料・運賃などの減価償却	1,500
導入費用の減価償却	960
合計	43,860
純節減コスト	7,360

(3) コスト性能分析

（ケース１）＊1

コスト／性能の係数定義例

$$\frac{a \text{ または } b}{c+d+e+f+g+h+i}$$

a：販売価格

b：レンタル月額

c：サイクル・タイムの逆数（１／μ秒）

d：１秒間に読取るカード桁数

e：１秒間に読取る紙テープ文字数

f：１秒間に読取るＭＩＣＲの文字数

g：１μ秒に転送する文字数×ディスクまたはテープの台数

h：１秒間に穿孔するカードの桁数

i：１秒間に印刷する英字行数×0.15＋１秒間に印刷する数字行数×0.85

（ケース２）＊3

コスト／性能比算出例

機種	レンタル月額（百万円）	処理時間（秒）	コスト／性能比
現有機	45	760	1.0
候補A	92	280	0.75
B	96	180	0.51
C	60	240	0.42
D	91	290	0.77
E	135	95	0.38
F	97	170	0.48
G	115	100	0.34

(4) ベンチ・マーク・テスト

（ケース２）＊3

テ ス ト 結 果 例

機種	時間（秒）	備考
現有	760	
B	180	512ＫＢを使用し，最高5本のジョブを多重プログラミングで処理
A	280	Bと同様
C	240	多重プログラミングは行なわない
D	290	最高７本をマルチプログラミングで処理
F	170	ＣＰＵ２台を使用し，最高９本のマルチ・プロセス処理

（ケース２）＊3

テ ス ト 結 果 総 括 例

ジョブ種別＼機種	X	Y	Z
中規模のFORTRANプログラム26本	○	○	○
大規模のFORTRANプログラム5本	○	×	△
小規模のCOBOLプログラム8本	○	○	×
ＬＰプログラムの実行のみ3本	○	△	○

(5) 月間レンタル額の比較

（ケース２）＊3

レ ン タ ル 総 括 比 較 表 （単位千円）

内訳＼機種	X	Y	Z
ＣＰＵ	69, 837	20, 375	65, 718
直接アクセス装置	11, 600	16, 560	12, 300
磁気テープ装置	5, 930	5, 890	4, 560
周辺装置	6, 759	6, 835	5, 805
遠隔ターミナル	1, 987	2, 456	3, 456
保守費用	−	8, 060	−
合計	96, 113	60, 176	91, 839

他機種の場合	レンタル合計
P	94, 518
Q	123, 456
R	96, 375
S	115, 684

（ケース２）＊3

月間レンタル明細

（単位千円）

機種 / 項目	X		Y		Z	
	仕様	レンタル	仕様	レンタル	仕様	レンタル
CPU						
1ワードあたりビット数	32	8,640	36	5,328	60	23,760
加算時間（μ秒）	0.7		0.75		1.1	
記憶容量（Mビット）	8	13,723	2.3	3,760	7.8	12,420
サイクルタイム（μ秒）	0.75		0.7		1.0	
コンソール						
タイプライタ		214				
表示装置						432
入出力チャネル		2,168		175	3チャネル	351
磁気ドラム						
制御装置		853		1,280		
ドラム装置	32Mビット	1,620	37.8Mビット	1,397		
磁気ディスク						
制御装置						
ディスク装置	1920Mビット	2,063	792Mビット	1,350	300Mビット	3,384
磁気テープ						
制御装置		346		515		227
テープ装置（6台）	90KC	1,750	96KC	1,620	60KC	1,361
周辺装置						
制御装置		941		796		839
プリンター（3台）		1,080		1,021		983
速度（行／分）	3,300		3,600～4,800		3,000	
カードリーダ（1台）		490		128		151
速度（枚／分）	200		60		1,200	
カードパンチ（1台）		281		175		112
速度（枚／分）	500		300		250	
端末装置						
制御装置		336		360		432
ターミナル（2台）		360		507		860
印刷速度（行／分）	240		250		300	
カード読取（枚／分）	400		200		100	
カード穿孔（枚／分）	160		75～200		—	

11. システムの詳細機能決定

(1) 目的の確認

(2) 機能の決定

（ケース６）＊６

① リアルタイムの情報検索では，パトロール員が無線で端末操作員を呼び出し，次のような情報の検索を依頼する。

- 逮捕令状
- 交通令状
- 緊急命令
- 免許取消し
- 免許一時停止
- 自動車盗難

② 管理者表は次のようなものを出力する。

- パトカー派遣の無線指令日報
- 無線指令週間分析表
- パトロール活動報告書
- 期別犯罪総括表（事件報告書も利用し月報的に出力する。）

（ケース10）＊９

次のような入力データから，出力データを得ること。

(1) 入力データ

① 部品マスタ

部品についての明細と製品のどこに使用されるかが記載されている。

② 作業マスタ

部品の工作手順と他の必要作業を示す。

③ 原価マスタ

部品製作の標準時間と材料費を示す。

④ 生産計画１

製品別の製作数量

⑤ 機械グループ・データ

110種の機械グループ別能率係数などを示す。

⑥ 生産計画２

生産計画期間などを示す。

(2) 出力データ

① 平均在庫量

② 平均仕掛品

③ 最低必要在庫

④ 最適バッチ

⑤ 機械使用時間

⑥ 機械使用時間の棒グラフ

⑦ 労働工数

⑧ 労働工数の棒グラフ

(3) ロジカル設計

(4) 実施に失敗したシステムの再検討例

ケース8

現システムの失敗を反省し，次のような方針で新システムを開発する。

① まず現場職長等に対して十分な説明を行なう。

② 1製品，1工場についてパイロット研究を行なう。

③ パイロット研究中は現場で作業カード作成なども指導する。

④ パイロット研究から本番実施に進む段階で，システムがうまく作動することを確認して進める。

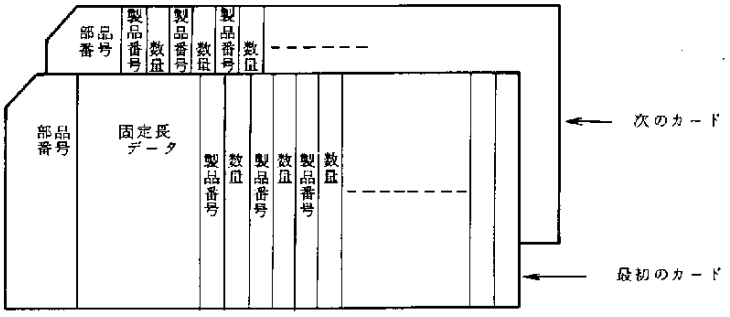
⑤ 作業の開始・終了を厳密にリアルタイムでとらえることはやめ，伝票を提出させる。これによって作業者の待ち時間はゼロとなり，現場の混乱と非能率は解消される。

⑥ 作業者の過大報告をチェックするため，仕掛数量の現物検査を行なう。

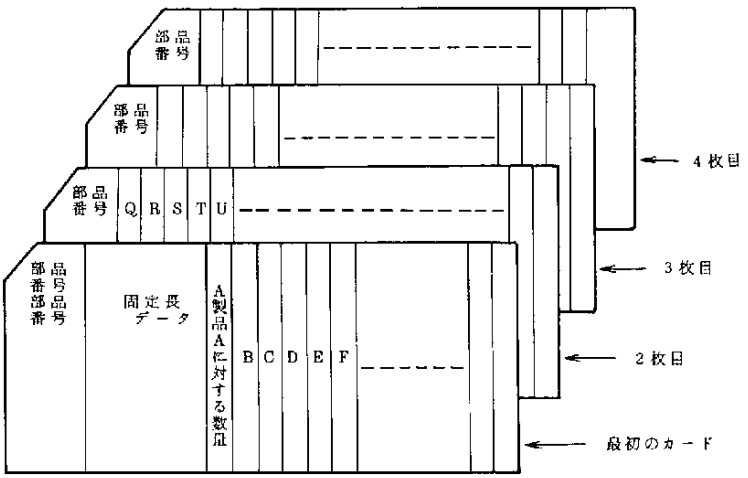
13. インプット設計

(ケース10)＊9

部品マスタ・カードの設計は次の2案が考えられた。



方法A



方法 B

A案は，ある部品を用いる製品の製品番号と，その部品の必要数量を組合せて記載するものである。

B案は各製品についてのフィールドを固定して，ここに必要数量だけをパンチする。B案では製品番号をパンチする必要はないが，その部品を用いない製品については該当フィールドを空白にしておく必要がある。

各案に対する所要カード枚数とパンチ桁数を次に示す。

		A 案	B 案
1つの部品に必要なカード枚数	最 小	1	4
	平 均	2	4
	最 大	8	4
1つの部品に必要なパンチ桁数	最 小	30	66
	平 均	132	111
	最 大	594	306
1枚のカードにパンチできる最大製品数	最 初	8	16
	2枚目	11	22
総所要カード枚数		15,000	30,000
総所要パンチ桁数		99万	88万

これからパンチ桁数についてはB案が有利と考えられたが，次の理由も考慮してA案に決定された。

(1) B案では空白桁が多いが，これは実際のパンチと同程度の労力を要する。

(2) B案では桁ずれの可能性があり，そのチェックは容易でない。

（ケース10）＊9

最終的なカードレイアウトを次に示す。

フィールド	桁数	PICTURE	注釈
カード区分	1	＝1	カード種類の区別のため
部品番号	2－12	11桁以下の英数字	
サイクル・タイム	13－14	99	固定長データ
在庫場所	15－16	99	
材料コード	17－18	99	
型コード	19	9	
完成コード	20	A	
寸法A	21－24	9999	
寸法B	25－27	999	
寸法C	28－30	999	
製品番号	31－33	A99	78桁まで7回繰返し
部品必要量	34－36	999	

2枚目以降のカード

フィールド	桁数	PICTURE	注釈
カード区分	1	＝2	
部品番号	2－12	11桁以下の英数字	
製品番号	13－15	A99	78桁まで10回繰返し
部品必要量	16－18	999	

（ケース10）＊9

部品マスタ以外のインプット設計を次に示す。

(1) 作業マスタ

フィールド	桁数	PICTURE	注釈
カード区分	1	＝3	
部品番号	2－12	11桁以下の英数字	
工作番号	13－14	99	78桁まで5回繰返し
機械グループ番号	15－17	999	
準備時間	18－19	99 （分）	
工作時間	20－23	999.9(分)	

(2) 原価マスタ

フィールド	桁数	PICTURE	注釈
カード区分	1	=4	
部品番号	2－12	11桁以下の英数字	
1セット当りの労務費	13－17	$999.99	
1セット当りの材料費	18－22	$999.99	
1セットの部品数	23－25	999	1セットが1部品の場合もある。

(3) 生産計画

フィールド	桁数	PICTURE	注釈
カード区分	1	=4	
製品番号	2－4	A99	
計画1で作られる数量	5－7	999	もし数量がゼロの場合は空白
計画2で作られる数量	8－10	999	
・	・		
・	・		
・	・		
・	・		
計画10で作られる数量	62－64	999	

(4) 機械グループデータ

フィールド	桁数	PICTURE	注釈
カード区分	1	=6	
機械グループ番号	2－4	999	1～110
特別係数	5－7	9.99	
能率係数	8－10	999	

(5) 生産計画

フィールド	桁数	PICTURE	注釈
カード区分	1	=7	
生産計画番号	2－3	99	1～20以下
期間	4－5	99	週
最終カード記号	6		最終カードのみ

14. ファイル設計

（ケース７）＊7

データ記憶方式の設計

① 空間分割は街のブロックにほぼ対応させ，ブロックと呼ぶ。

② 犯罪種別は15分類を取扱う。

③ 上記のブロック別，犯罪別の件数を記憶する。

④ 指数平滑法による平均を算出する。

⑤ 週間の変動，年間の変動を補正してデータ・ベースに記憶する。

⑥ 各犯罪種別に対する平均サービス時間を記憶しておく。

15. プロセス設計

（ケース７）＊7

① ブロック別，犯罪種別の発生件数を予測する。

② 予測モデルの件数は，ブロック，時刻，曜日，年間の第何週であるかによって異なる。

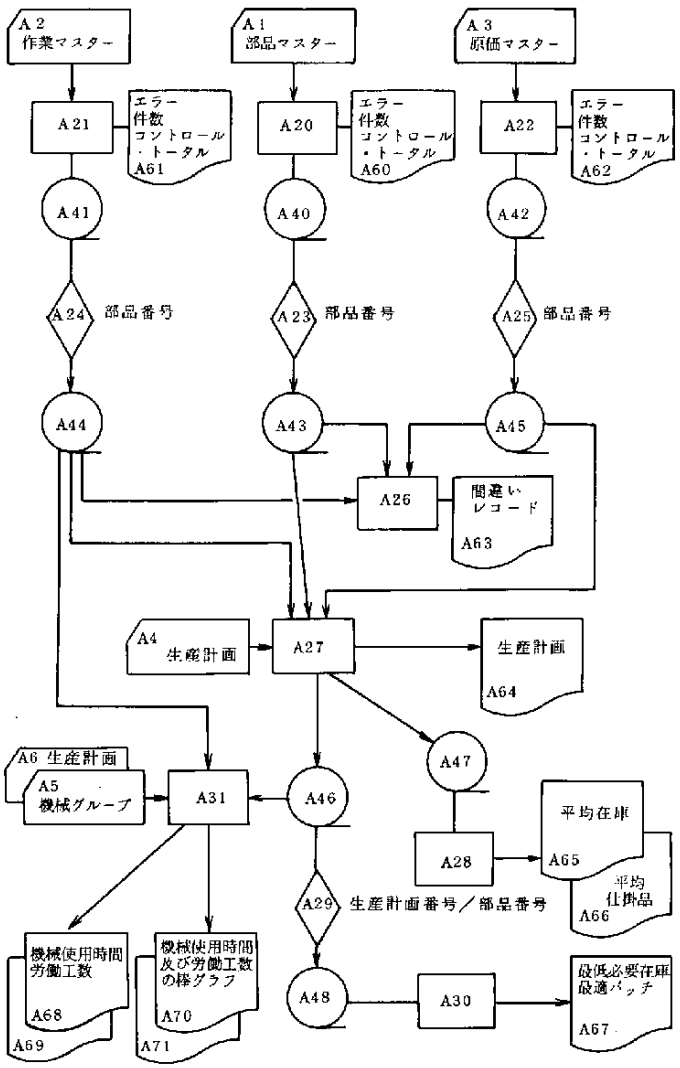
③ パトロール要員配分計画のシミュレーションは，待合せ問題的な性格を持ち，事件の到着，パトカーによるサービスといった形でモデル化する。

④ 同一地域の同一時間帯について，乱数を発生させるシミュレーションを数回行ない，待ち時間，待ちの長さなどの平均を算出する。

⑤ 犯罪種別の取扱いは，緊急性の高いものから優先順位を付し，グループ化しておく。

（ケース10）＊9

プロセス・フローチャートによる表示



（ケース10）＊9

3種のマスタ・カードのチェックは次のように行なった。

(1) 部品マスタ

① 製品番号

コード体系から最初の文字はA～H，2番目は0～3，3番目は0～9と定められているので，これらの範囲内にあるかチェックする。

② 関係チェック

寸法A≧寸法B≧寸法C

③ 組合せチェック

A×B×C≦40,000,000

④ リミット・チェック

寸法A，B，Cが最大値以下か。在庫場所コードは45以下か。

⑤ このケースでは8回のバッチに分けて処理を行なうが，各バッチでカード枚数と必要部品合計を手作業で求めておき，集計結果と照合する。

(2) 作業マスタ

① 機械グループ番号

最高を110としたリミット・チェック

② 工作時間

最高を150分としたリミット・チェック

③ 工作番号

番号順に並んでいるかどうかのチェック。ただし欠番はありうる。

④ 部品毎の工作数

3～40の間にあるかチェック

(3) 原価マスタ

① 1セットあたりの労務費

最高額を$600としたリミット・チェック

② 1セットあたりの材料費

最高額を$400としたリミット・チェック

③ 1セットの部品数

最大250個としたリミット・チェック

（ケース10）＊9

3種のマスタ・ファイルは部品番号の昇順にソートした上で，マッチング・テストを行なう。その目的は次のとおりである。

(1) 各ファイルに，ある1つの部品番号に対して1つのレコードのみが存在しているかチェックする。

(2) 各ファイルに存在する部品番号が正しく1対1に対応しているかチェックし，レコードの欠落または余剰を発見する。

この処理のロジックは次のようなデシジョン・テーブルで表現された。

ルール	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A 43の部品番号の順序	1st−	1st=	1st=	1st=	2nd	1st	2nd=	2nd=	1st	1st	2nd	3rd	2nd	3rd
A 44の部品番号の順序	1st=	1st=	1st=	2nd	1st=	2nd=	1st	2nd=	2nd	3rd	1st	1st	3rd	2nd
A 45の部品番号の順序	1st−	1st=	2nd	1st=	1st=	2nd=	2nd=	1st	3rd	2nd	3rd	2nd	1st	1st
A 43の部品番号の値	＝ ∞	≠ ∞	−	−	−	−	−	−	−	−	−	−	−	−
A 43の次のレコードを読む		×			×		×	×				×		×
A 44の次のレコードを読む		×		×		×		×		×			×	
A 45の次のレコードを読む		×	×			×	×		×		×			
A 43桁に印刷する			×	×		×			×	×	×		×	
A 44桁に印刷する			×		×		×		×		×	×		×
A 45桁に印刷する				×	×			×		×		×	×	×
終了	×													

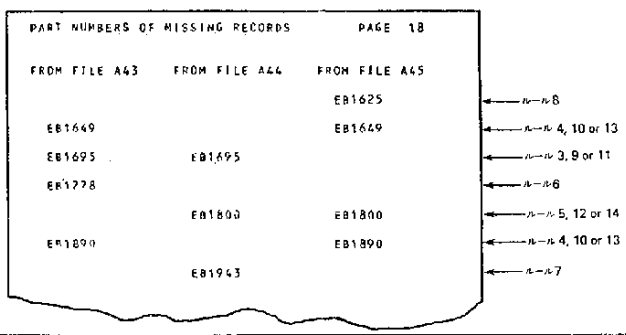
このデシジョン・テーブルによって次のような内容のファイルを処理すると，

ファイルA 43	ファイルA 44	ファイルA 45
EB1621	EB1621	EB1621
EB1625	EB1625	EB1627
EB1627	EB1627	EB1650
EB1650	EB1649	EB1688
EB1688	EB1650	EB1695
EB1700	EB1688	EB1700
EB1800	EB1700	EB1778
EB1850	EB1778	EB1850
EB1935	EB1850	EB1935
EB1943	EB1890	EB1943
EB1984	EB1935	EB1984
・	EB1984	・
・	・	・
・	・	・
・	・	・

出力される結果は次のようなものである。

PART NUMBERS OF MISSING RECORDS PAGE 18

FROM FILE A43 FROM FILE A44 FROM FILE A45



（ケース10）＊9

各部品について，生産計画との関連を次のように計算する。文章形式で処理内容を記述した例である。

ラン・A 27——処理（第2段階）

この段階では，ルーチンの前のランで作成されたA 43，A 44，A 45，のファイルを使用する。これらのファイル中のレコードは部品毎に作成されており，しかもそのレコードは部品番号順に並べられていることは既知のことである。1つの部品に関して，3つのレコードがそれぞれのファイルから読み込まれ，その部品について，それぞれの生産計画との関連が次のように計算される。

1. A 43 の製品番号が，テーブルにストアーされている生産計画数量を見つけ出すために順次用いられ，その生産計画数量に「部品必要量」が乗じられる。生産計画数量は，生産計画1～20まで20種類あるが，各々の計画に対する必要部品数量が計算される。このランで行なわれる乗算の回数は，普通225万回以下である。（7,500部品×1部品を平均15製品で使用×生産計画数20）。これらの乗算は，コンピュータで30分もあれば十分である。生産計画数は，実際には20をかなり下回る場合が多いことを考慮して，処理開始前に，生産計画数をパラメータとしてインプットすることにした。こうすると，必要生産計画数だけの乗算を行ない，ゼロの入っているエリアを計算するという無駄な時間を節約できるからである。

2. このようにして計算された必要部品数量から，次に示す最低必要在庫数量が計算される。

必要部品数量がゼロ：最低必要在庫数量はゼロ

〃　　1～249：　　〃　　10以上

必要部品数量　250～499：最低必要在庫数量50以上

〃　　499以上　：　　〃　　100以上

3. 最適バッチ・サイズは次のような式で計算される。

$$\left[\frac{1.3\times\text{必要部品数}\times(\text{A 44からの})\text{全体準備時間}}{(\text{A 45からの})\text{1単位当りの標準原価}}\right]1/2$$

4. 平均在庫額は次の式で計算される。

$$\left(\frac{\text{最適バッチ・サイズ}}{2}+\text{最低在庫数量}\right)\times\text{セント単位に4捨5入された単位当り標準原価}$$

5. 平均仕掛額は次の式で計算される。

$$\left[\frac{(\text{1セット当りの労賃}\times 1\frac{3}{4})+\text{1セット当りの材料費}}{(\text{A 45からの})\text{1セット当り数量}}\right]\times\text{必要部品数}\times(\text{A 43からの})\text{サイクル・タイム}$$

6. 1.，2.，3.によって計算された結果は，各生産計画についての各々の部品の情報として，固定長レコードでファイルA 46に書き込まれる。レコードの内容は：

生産計画番号（1～20のいずれか）

部品番号

必要部品量

最低在庫水準

最適バッチ・サイズ

7. 4.，5.によって得られた結果は，各部品について2つの固定長レコードで記録される。1つのレコードには平均在庫額を，もう1つのレコードには平均仕掛品額を記録する。これらのレコードの内容は：

〈在庫レコード〉

レコード区分（＝１）

部品番号

20の生産計画それぞれに対する平均在庫額

〈仕掛品レコード〉

レコード区分（＝２）

部品番号

20の生産計画それぞれに対する平均仕掛品額

これらのレコードは，ファイルＡ47に交互に書き込まれる。

ラン・Ａ28――処理

このランでは，ファイルＡ47が2回読まれる。まず最初は，区分１のレコードのみを読み込み，Ａ65を打ち出す。この最後に全生産計画の合計も計算して打ち出す。

２度目の読み込みでは，区分2のレコードのみを読み，Ａ66を打ち出す。Ａ66のレイアウトはＡ65と同じである。

コンピュータのライン・プリンターの幅は120字であったので，横に10の生産計画を打ち出すことができ，２行で20の打ち出しができ，１ページに11部品打ち出すことができる。また各ページ数も打ち出す。

ラン・Ａ29

このランでは，ファイルＡ46を生産計画番号の順に従って部品番号を分類し，ファイルＡ48を作成する。

ラン・Ａ30

このランでは，各生産計画についてＰ295に示すようなＡ67を作成する。ファイルＡ48を読み，その内容を，縦にミシン目が入っている用紙１行につき３組ずつ印刷を行なう。こうしておけば各生産計画について，部品順のリストが必要な場合には，ミシン目に沿って切ればそれを得ることができる。7,500の部品について20の生産計画があるので，合計150,000組印刷しなければならない。１行に３組印刷すれば時間と用紙が節約できる。また用紙の４インチ目にミシンを入れておけば，生産カードに取り付けるときに便利である。

このような方法でプリント・アウトを行なうには，メモリーの中に１ページ分の内容を記憶させておく必要があり，高速連続ライン・プリンターなどを考える必要がある。この事例の場合，１ページに27行プリントされ，１生産計画に約90ページが必要であった。１ページ分の内容を記憶させるためのコア・エリアは2,349文字（27行×87文字／行）が必要であった。この方法をより能率的に行なうにはより多くのエリアが必要である。

（ケース７）＊7

管理報告書は次のものを出力する。

(1) シミュレーション・モデルの係数

- 事件種別の発生率週間調整係数

(2) 予測結果

- 事件種別，時間別発生件数
- サービス遅れなしで処理しうる事件数予測
- ５名配置のときのサービス遅れ分布

（ケース10）＊9

各アウトプットの設計例を次に示す。

(1) 平均在庫

１つの部品につき，横方向に10の生産計画案を出力し，２行で20案をとした。１ページに11部品が収容できる。

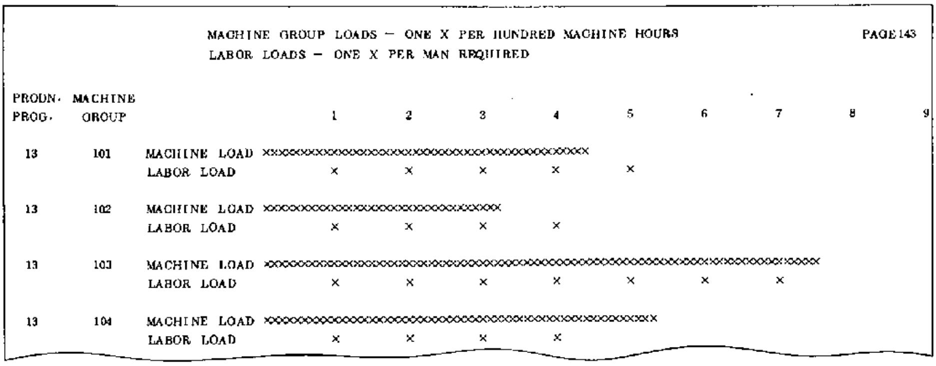
AVERAGE STOCK VALUES OF PARTS FOR PRODUCTION PROGRAMS(NOS. 1 TO 20) PAGE 236

PART NO.		1	2	3	4	5	6	7	8	9	10
		11	12	13	14	15	16	17	18	19	20
ETB 27168B	S	17630.75	18993.14	22005.39	16552.96	18315.26	12994.50	13936.08	25839.16	29570.15	17305.22
	S	9925.39	10315.03	7749.33	6168.25	5003.00	15870.06	1942.54	528.15	1122.93	780.95
ETB 27168A	S	1760.38	1908.21	1575.70	579.32	541.25	0.00	0.00	0.00	415.38	422.66
	S	1387.29	2065.43	668.27	175.95	62.11	517.39	255.76	319.28	1529.70	662.01
ETB 27167	S	1073.11	5051.61	146.92	19.85	44.92	31.52	0.00	0.00	233.11	50.28
	S	541.20	1596.58	334.66	178.73	14.86	497.19	34.25	0.00	15.36	142.36
PROGRAM	S	126973.65	122801.37	128103.30	98645.08	76196.49	100581.70	85210.41	98314.53	81537.04	79417.85
AVERAGES	S	83411.83	81057.02	93418.15	97021.16	81305.20	62152.68	77303.90	92713.22	76326.82	83389.50

(2) 各部品の生産計画

PART NO.	REQMT.	MIN. ST.	OP. RCH.	PART NO.	REQMT.	MIN. ST.	OP. RCH.	PART NO.	REQMT.	MIN. ST.	OP. RCH.
GTA 78621E 43	103537	103600	5300	GTA 79205	143291	143300	7800	GTA 80025C	7014	7100	1800
GTA 78630B 16	3749	3800	1200	GTA 79208	105712	105800	3300	GTA 80027	138	140	190
GTA 78631B 04	19180	19200	5900	GTA 79216	2016	2100	450	GTA 80028	574	600	400
GTA 78633	6411	6500	700	GTA 79225	426	450	180	GTA 80033A 10	1203	1300	800
GTA 78633A	225	230	110	GTA 79229B	18994	19000	1600	GTA 80045D	0	0	0
GTA 78634	2783	2800	500	GTA 79240	2564	2600	1200	GTA 80047	119360	119400	12600
GTA 78638C 3	12066	12100	1800	GTA 79255	13861	13900	1500	GTA 80058	0	0	0
GTA 78694B 13	69	70	20	GTA 80021A 3	6680	6700	900	GTA 91300	1385	1400	600
GTA 79200	277	300	80	GTA 80026A 4	82	90	150	GTA 91305	714	800	300
PART NO.	REQMT.	MIN. ST.	OP. RCH.	PART NO.	REQMT.	MIN. ST.	OP. RCH.	PART NO.	REQMT.	MIN. ST.	OP. RCH.
GTA 91311	6724	6800	900	GTA 91521	8366	8600	500	GTA 92355A	5183	5200	400
GTA 91326	5742	3800	500	GTA 91524A	1839	1600	300	GTA 92359B 6	485	500	150

(3) 機械グループの稼動時間数と必要労働工数



17 セキュリティとインテグリティ

(1) システムの改訂手続き

(ケース4) *10

① システムの改訂は開発課の課長が認可し，操作課に通知する。

② 新旧両システムを平行操作し，結果を照合する。照合結果の承認は，小規模な改訂ならばアナリストが行ない，大規模な改訂ならば開発課長が行なう。

(2) データおよびプログラムの保管方法

(ケース4) *10

① 株式部で施錠保管する。

② 借出しは文書によって申込む。

③ 借出し日の定時前に返却する。

④ データまたはプログラム・テープの操作中は操作課長が立会う。

⑤ コンソール・ログのコピーは株式部に送付する。

⑥ マスター・テープは3世代にわたって保管する。

⑦ トランザクション・テープはマスターに対応して保管する。

⑧ トランザクション（文書，電話）は株式部で受信する。

⑨ トランザクション・カードはランの時まで株式部で保管する。

⑩ テープの内容，保管期限などを記載したラベルを使用する。

(3) エラー防止手段

(ケース4) *10

① 入力伝票を手作業集計し，合計を照合する。

② 入力カードを機械集計し，合計を伝票と照合する。

③ 毎日の入力カードについて，件数，株式数などを集計してコントロール・カードを作る。一週間分のランを行なう時に入力カードとコントロール・カードを別々に集計し，照合する。

④ ソート・プログラムにもチェック機能があり，レコード数，株式数などを入力と出力とで照合する。

⑤ 入力データのＦＯＲＭＡＴチェックとＬＩＭＩＴチェックを行なう。

⑥ 新しい株主番号を割当てる時は，同一のものがマスターにないかどうか確認する。

⑦ チェックの結果として無効になった入力データも含めて，ランの後に合計の照合を行なう。

(4) マスターへの転記テスト

（ケース４）＊10

現在の株主が新規に買増した場合は次のようなチェックを行なってマスターを改訂する。

① 株主番号がマスターにあるか。

② 株券番号がマスターにあるか。

③ 株主番号に対応する氏名，住所が入力と一致するか。（部分的チェックですませることが多いが，それでもエラーが頻繁に発生する）

18 システム開発の工程管理

(1) 進捗度報告

1) 1個の作業項目の見積り，進捗度等の時系列的変化

（ケース５）＊4

工程名	費用配分分析						
担当者	山田一郎						
期間（第3段階）	2/5～3/25						
1. 開始日	2/8	2/8	2/8	2/8	2/8	2/8	2/8
作業遂行 2. 所要人日の当初の見積り	28	28	28	28	28	28	28
作業遂行 3. 所要人日の改訂見積り	28	28	28	28	28	28	28
作業遂行 4. 使用された人日数	4	9	13	17.5	23	26	27
作業遂行 5. 達成割合%（4÷3）	14	32	46	63	82	93	96
作業遂行 6. 完成までに必要な人日数見積り	24	19	15	11	4.5	1.5	0.5
作業遂行 7. 所要人日数の現在の見積り	28	28	28	28.5	27.5	27.5	27.5
作業遂行 8. 実際に完成した割合%（4÷7）	14	32	46	62	84	95	98
	173						174
時間現況 9. 所要人日数の調整見積り	28	28	28	28	28	28	28
時間現況 10. 経過人日数	7	12	16	21	24	27	28
時間現況 11. 経過期間の割合（10÷9）	25	43	57	75	86	96	100
時間現況 12. 工程先行または遅延（8－11）	11	−11	−11	−13	−2	−1	−2
13. 割当要員数	1	1	1	1	1	1	1
14. 改訂見積り完成日						3/22	3/28
15. 当初の見積り完成日	3/21	3/21	3/21	3/21	3/21	3/21	3/21
現在日	2/11	2/18	2/25	3/4	3/11	3/18	3/25

損失時間 － 人日数	数	所見（障害の説明も含む）
累計 － 最終報告		
2/11	3	第2段階の遅延
2/18	3	
2/25	3	
3/4	3.5	スタッフ会議
3/11	3.5	
3/18	3.5	
3/25	3.5	
現在までの累計		注：第4段階の開始日 4/1

2）同時期におけるプロジェクト全体の一覧

（ケース５）＊4

プログラム番号／名称	開始日	最近の完成日	改訂	使用人日数							
				今週	本日まで	完成までの所要数	開始から終了まで	作業指示見積	改訂	完成率 %	計算残(%)
移行	76／ 5／ 1	76／12／31	0	0.0	6.2	63.8	70.0	70	0	8.8	18.0
編集1	76／ 5／ 1	76／ 9／11	2	10.3	75.1	16.2	91.3	23	0	82.3	2.1
税務会計											
0501（日）	76／ 3／15	76／10／17	2	17.7	192.7	171.3	364.0	455	1	52.9	7.3
0502（月）	76／ 7／ 5	76／ 9／11	1	4.2	26.9	5.1	32.0	20	0	84.1	2.1
ディーラー会計											
0601（日）	75／12／29	76／10／ 1	1	20.1	412.9	111.4	524.3	393	0	78.8	5.0
0602（月）	76／ 1／13	76／10／30	3	1.4	23.1	64.9	88.0	45	0	26.3	9.1
販売統計											
0801（日）	76／ 4／25	76／ 9／ 6	3	4.4	75.2	21.8	97.0	55	0	77.5	1.4
0802（週）	76／ 4／25	76／ 9／12	2	1.9	50.8	9.2	59.9	55	0	84.7	2.3
0803（更新）	75／ 5／15	76／10／ 1	0	6.4	77.2	87.8	165.0	165	0	46.8	5.0
0804（累計）	75／ 5／15	76／ 9／19	1	9.9	113.3	44.8	158.0	145	0	71.7	3.3

(2) 工程遅延の原因分析

（ケース５）＊4

① 搬入されたコンピュータの稼動開始遅れ。

② 急成長した計算センターの要員の熟練不足。

③ 要員をプロジェクト・チームと計算センターに分割して配属させたことによる，コミュニケーションの不足。

④ プログラム，テスト・データの改訂等に関する手続きや文書化の規則が緩和された。理由は開発のオーバヘッド時間を削減することであったが，管理者は進捗状況の正確な把握ができなくなった。

⑤ 一部のスケジュールが経験からみて，非現実的であった。

＊1～10（＊9は除く）は，

J. Dearden, F.W. Mc. Farlan, W. M. Zani , "Managing Computer-based Information Systems" Richard D. Irwin, Inc, 1971

（安藤嘉昭，島田昭代，一楽信雄　共訳　「新・経営情報システム」建帛社，1973）

＊9は，

H. D. Clifton, "Data Processing System Design" Business Books Ltd., 1971

（田中弘・清水克彦訳「コンピュータ・システム設計」白桃書房，1974）

の中の，以下のページから，それぞれ引用したものである。

＊1　p.p. 105～131

＊2　p.p. 157～187

＊3　p.p. 132～156

＊4　p.p. 237～264

＊5　p.p. 344～366

＊6　p.p. 267～297

＊7　p.p. 298～319

＊8　p.p. 320～343

＊9　p.p. 1～26

＊10　p.p. 215～236

第5章　報告書作成と発表

キーワード

文書（ドキメント）（documents），口頭発表（oral presentation），視聴覚機材（audio-visual material），スライド（slide），オーバヘッド・プロジェクタ（overhead projector），発表と討論（evaluation of presentation and discussion）

目　　標

開発の実習で作業した成果を報告書にとりまとめ，それにもとづく発表を行なう。

また開発の途中でのチーム各メンバー間，教師とのコミュニケーションの実習も心がける。

文書の作成法，口頭発表の要領，視聴覚機材の特徴等を理解し，実習によって習熟する。

内　　容

5.1　文書報告

(1)　システム開発にともなうドキュメントの作成は，第1単元，第2単元等を十分に参照して行う。

標準的なドキュメント様式をあらかじめ定めておくか，あるいは研修生に制定させるのもよい。

(2)　システム開発に関連する組織内の連絡文書的なものは，通常の企業でも訓練しているが，一応は研修生のレベルをチェックしてみるのがよい。

5.2　口頭発表

(1)　口頭発表における一般的かつ常識的な注意事項は他にゆずる。

(2)　説明のための資料として提示する必要のあるものは，ビラあるいは視聴覚機材用に作成しておくこと。

(3)　黒板は特徴をよく考えて，原則として次のような目的に用いる。

①　思考過程を意識して，資料を段階的に説明する場合。

②　討論の際に意見を書き，議論の範囲を示し，要訳する。

(4)　発表の際の注意事項

①　黒板などに書く字の大きさは，6～7cm角以上とする。したがって，スライドやオー

バヘッドプロジェクタでも，投影された字の大きさが，これ以上となるように注意しなければならない。

② 口頭発表とともにスライドやＯＨＰを用いる場合は，画面にむかって右側に立ち，右手に持った指棒で示しながら説明する。スクリーンを見るだけでなく，聴取者の反応にも注意して，発表内容が理解されているかどうか確認すること。

5.3 視聴覚機材の利用

視聴覚機材を利用する場合は，各機材の特性を十分に理解して，最も適切なものが選択できるようにしなければならない。

一般に視聴覚機材とは次のものをいう。

(1) 非映写機材

写真，絵，図表，地図，掛け図，実物，標本，模型

(2) 映写機材

スライド，映画，オーバヘッド投影，ＶＴＲ

(3) 聴覚機材

レコード，録音，放送

視聴覚機材を用いると，その部分の資料提示のみが観覧者に強い印象を与え，全体とのつながりが見失われがちである。何の目的で視聴覚機材を用いるのか十分に説明して理解を徹底させた上で利用するように，発表者を指導しなければならない。

(1) スライドの特性

① 静止画面を提示する。

映写時間に制約がないので，説明や討論の時間を十分にとることができる。一般に資料の提示には好適である。

② 拡大した鮮明な画面を提示する。

同時に多数の人に観覧させうる。

③ 具象的に提示する。

記録や事象の写真画面を投映でき，具象性による迫力と説得力が期待できる。

④ 機器の操作が簡便容易である。

⑤ スライドの自作が容易である。

⑥ 発声スライドを用いれば，映画に近い効果を期待しうる。

(2) オーバヘッド投影が可能な資料の種類とその特質

① ＴＰ（トランスペアレンシー）

透明なプラスチックシートにペンや絵具で書いたり，カラーシートを貼ったもので，ＯＨＰでは最も多く使われている。

スライドとの大きな相違は，ＴＰが半完成の状態で提示できることであり，討論を行ないながら，あるいは説明を進めながら完成させてゆくことができる。また何枚かのＴＰを重ねて，図や表を合成分解しつゝ説明できるのも大きな長所である。

複写によってＴＰを作成できる点にも注意して利用法を考えたい。

② 模型

透明なアクリル樹脂やプラスチックシートを貼り合わせて模型を作り，提示することもできる。

③ 実物のシルエット

なおＯＨＰは明るく投影されるので，スライドほどの暗室は必要なく，発表を聞きながらメモを取るのに好適である。

その他の特性はスライドとほぼ同様であるが，写真の投影はやゝ劣り，発声は不可能である。

(3) 映画の特性

① 高度の具象性をもって事物，事象を提示する。経験を代行させうる。

② 時間的，空間的に拡大，縮少できる。

③ 現実を再構成し，まとまった意味や思想を表現できる。

④ 同時に多数の者に提示できる。

⑤ 情緒に強く働きかける。

⑥ 流動的に展開してゆくため，受容が安易になりやすい。

5.4 発表および討論の評価

(1) 発表および討論の進行中に評点をつけることは通常行なわない。その理由は次のとおりである。

① 発表内容への注意が散慢になる。

② 討論のペースに遅れる。

③ 採点が拙速となりがちである。

④ 研修生の側が採点を意識すると自由な発表や討論が損われる。

(2) 通常用いられる方法は，演習時間の終了後できるだけ早い時に，研修生の発表および討論の模様を思いうかべて採点する方法である。評価基準としては次のようなものが考えられよう。

① 発表の準備の周到さ。

② 発表のわかりやすさ。

③ 聴く者に興味を持続させること。

④ 質問に対する回答の明瞭さ。

⑤ グループ討論におけるリーダシップ。

⑥ すぐれた代替案を出して意志決定に貢献したかどうか。

⑦ ケースに対する分析が皮相的でないか。

⑧ 個人の直感に頼りすぎていないか。

⑨ キャッチフレーズを巧妙にもて遊ぶことで事足れりとしていないか。

(3) 単に評価するだけでなく，その結果をフィードバックして研修生の態度を修正してやることも必要である。その場合は次の点に留意すること。

① 演習の経験が浅い研修生に対しては，状況の分析をどうしたらよいのか分からないのではないか，想像力の訓練が不十分で別のアプローチがあることに気がつかないのではないか，といったことも可能性として考えてやること。

② 演習の経験をつんだ研修生は，大した準備をしなくてもなんとかグループの討論に参加できるくらいおしゃべり上手になっている。

このようなものの発表に対しては，事実はどうか，計算結果はどうか，その結論はどうして出てきたかといったことを問いただしながら迫ってゆけば，本人や他の反感を買うことなく望ましい結果が得られる。

指導上の留意点

(1) 報告書作成や発表の技法は，コミュニケーション技法の一部として，Bコースの第2単元で学習する。したがってここでは，開発プロジェクトをとおして，これらの技法を実地に応用させることをねらいとする。

(2) 本章の第1の目的はシステム開発作業の成果を正確に報告，伝達することの実習にあるが，時間が許せばシステムのPRについても指導したい。PRを行なう対象者の特性の分析，方法の選択，有効な媒体（映画など）の検討といったことを実施すればよいであろう。

(3) 研修生は特種ないし一種の情報処理技術者と同等程度の予備知識をもっているとみなされるが，本章に関連してはあまり多くを期待できないので，技法の実用面についてある程度の講義を行なう必要があるかも知れない。

参　　考

(1)　実習成果報告書の分量

前述の電電公社中央電気通信学園における上級システム・エンジニア・コース（163日間，うち実習183時間）の例

テ　ー　マ	人　員	ペ ー ジ
住宅の需要予測要因について	2	44
キャッシュレス社会	2	81
情報検索と意味について	1	40
ファイル構成についての一考察	1	42
情報検索についての一考察	1	36
計量モデル用語とプリコンパイラの検討	1	79
医療システムについての一考察	1	20
論理の正当性について	1	103
パターン処理についての一考察	2	59
ＥＤＰ会計と会計監査について	1	78
数値制御について	2	99
あるシステムの一設計にあたって	1	48

参 考 文 献

〔1〕 主原正夫，高萩竜太郎「視聴覚機材・教材の扱い方　初級編，中級編」明治図書，1975

単 元 Ｄ４

コンピュータ部門の運営管理

単元Ｄ４

コンピュータ部門の運営管理

目　　次

第１章　コンピュータ部門の組織 …… 311

1.1　企業内組織としての位置づけと特色 …… 312

1.2　組織運営形態のいろいろ …… 317

1.3　コンピュータ部門の義務と責任 …… 322

1.4　経営とのつながり …… 336

第２章　コンピュータ部門の要員管理 …… 340

2.1　要員の分類 …… 341

2.2　要員の育成 …… 347

2.3　要員管理のポイント …… 350

2.4　要員のローテーション …… 352

第３章　システム開発管理 …… 359

3.1　コンピュータ部門の組織的変化 …… 360

3.2　システム要員の職務分化 …… 364

3.3　システム開発計画 …… 366

3.4　開発とメインテナンス …… 368

3.5　大型プロジェクトの形成 …… 369

3.6　標準化とドキュメンテーション …… 373

3.7　新しいシステム開発技法 …… 375

第４章　コンピュータの導入 …… 379

4.1　コンピュータの選定と評価 …… 380

4.2　コンピュータの設置条件 …… 391

4.3　ハードウェアの保守 …… 394

第5章　コンピュータ室の運用管理 …… 397
5.1　コンピュータ室の運用方式 …… 398
5.2　設備管理 …… 400
5.3　データ管理 …… 402
5.4　コンピュータ使用管理 …… 409
5.5　コンピュータ・コスト賦課 …… 417
5.6　ＥＤＰ監査 …… 420

第6章　情報処理サービス業の利用 …… 425
6.1　情報処理サービスの必要性 …… 426
6.2　情報処理サービス業の種類と業務内容 …… 428
6.3　情報処理サービス業の現状 …… 430
6.4　情報処理サービス業の活用 …… 435
6.5　外部利用の今後の方向 …… 439

第1章　コンピュータ部門の組織

キーワード

組織内の位置 (location in the organization structure)，内部組織(internal organization)，センタ運営 (computer center management)，コンピュータ運用(managing computer operation)，コンピュータ要員管理 (managing computer personnel)，コンピュータ機密保護 (computer security)，専門化の進展 (increased specialization)，機種選択 (computer selection)，長期計画立案 (long range planning)，義務と責任 (duties and responsibilities)，ファシリティ・マネジメント (facilities management)，TSS (time sharing services)，OJT (on-the-job training)，スケジューリング (scheduling)，入力データ作成 (data entry management)，進捗報告 (progress reporting)，コスト管理 (cost control)，EDP委員会 (EDP steering committee)，プロジェクトチーム編成 (organizing project teams)

目　　標

単元D4の学習目標としては，見学や事例研究など含め，単元D1，D2を復習しながらより実務的な角度で知識を身につける態度をとるものとする。

この章では，次のことを目標とする。

(1) コンピュータ部門の管理職，または，それに準ずる立場から，企業内におけるコンピュータ部門の組織的な位置づけ，特殊性，内部管理上の大要を把握する。

(2) コンピュータ部門のもつ，複合的・総合的機能と通常部門の単一的・ルーチン的機能との差異，および，コンピュータ部門は，独立分離（スピンアウト）しやすい性格をもつことを理解しておく。

(3) コンピュータ部門は，経営合理化の先兵であり，高価な設備とその安全管理，情報集中化による機密保持など，経営的視点を置いてみる。

(4) コンピュータ部門の特殊性は，独立経営体のミニチュア版的位置づけとしてとらえる。

また，この章は，コンピュータ部門の役割り全般が理解できるよう，単元D4の総論的な位置づけで学習することを目標とする。

内　　容

1.1　企業内組織としての位置づけと特色

次のような点を説明する。

(1)　経営トップとの強い結合

コンピュータ部門と経営トップとの結びつきの理由を実例に照らして説明する。

- 投資額が大
- 省力化の先兵的役割
- 長期企業戦略の分担
- データ処理に関する責任の集中化
- 他部門との緊密な協力関係の保持が容易

などから，　～　室と呼ばれる経営直結型の組織形態が見受けられる。

このことは，経営全体の支持と協力が得られやすく，また，経営への貢献度が正しく評価できる地位に配置する必要性からであろう。

(2)　複合的性格の保持

企画立案から製作，遂行までを含む一貫生産的な複合機能をもつ，スタッフとラインを併せもつ性格がある。

特徴となる点のいくつかをあげ理解させる。

(a)　多種の要員を管理

- システム・エンジニア
- プログラマ
- キーパンチャ
- オペレータ
- ライブラリアン
- 機械室保守員　　など

これらの要員の種類の中は，シニアからジュニアに至る階層を形成しており，複雑化している。

(b)　多面的活動形態

ユーザニーズの収集・評価およびそれに答えるための開発活動，対外的には，コンピュータ・メーカー，または，販売会社とのコミニュケーションも併せもつ。

①　ユーザとの対応

- ユーザ現状調査とニーズの収集

・ユーザの運用指導

・ユーザの満足のいくシステム開発およびフォローアップ

② メーカーとの対応

・新しい技術のたえざる導入

・メーカー提供ソフトウエアの評価ならびに改善要求

・設備保守状況の把握

・設備更新に関する検討

・要員教育に関するサービス評価

(c) 会社的性格の保持

スタッフとラインをもち，企画立案から製作，運用まで一つの独立組織となりうる性格がある。

しかも，システム開発などの場合は，定常的な経営活動というより，プロジェクト的性格の仕事を内部に包含しながら進めていくことが多い。

多面的な性格としての機能を列記する。

・調査

・企画立案

・技術相談

・計算センタ

・データ管理センタ

・通信管理センタ

・システム開発（プロジェクト）

・ユーザ運用指導

(3) 分離独立の容易性

会社的性格をもっていることから，分離独立が他の企業内組織と比較して容易である。

情報処理サービス会社は，企業内組織から分離独立して形成された企業が多い例をみても，独立の容易性を示している。

(a) 要員のスピンアウト

要員は，専門化，分業化しており，かつ，かなり汎用的な技術能力を身につけていることから，他の企業へ移動しても個人の能力発揮が可能であり，要員のスピン・アウトの大きな理由である。

また，情報処理分野が成長過程にあることもその原因の一つになっている。

(b) 内部の独立性・採算性

コンピュータ部門の多くは，その位置づけから，他の組織との独立性を強く求められる。

それは，一方で，採算性が重視され，収支の明確化の必要から，組織ごとスピン・アウトする要因ともなっている。

(c) 計算センタの分離

コンピュータの運用とその関連業務を受け持つ計算機能は，計算センタとして企業内組織から切り離され，独立するケースが多い。

その要因を列挙する。

① 設備の効率的使用から，オペレーション・シフト体制（２交替，３交替など）を採用する必要が生じ，他の組織の要員と比較しての勤務条件の特異性がある。

② 独立採算としての計算が容易である。

③ 要員の特殊性，すなわち，技術者的性格をもつ。

④ 業種・業界を超えての処理の共通性から，関連会社はもとより，他企業からの処理受託が容易である。

特に，サービス業，金融機関，商社などで技術者を多くかかえることは，他組織部門との要員とのローテーションが硬直化しやすい環境にあり，計算センタとして独立組織させる場合が多く見受けられる。また，計算センタとして独立した会社がシステム開発の機能をもつようになり，総合的な情報処理サービス会社へと成長していく傾向も見受けられる。

(4) 外部専門会社利用の容易性

コンピュータ部門は，専門的組織としての色合いが強く，多くの専門職としての要員をかかえている。

また，技術革新のはげしい職場であり一企業内ですべてのことを充足できないケースも発生してくる。

このことから，さまざまな形態で外部の情報処理サービス会社の利用が行なわれている。

利用形態として次のようなものがあるが，詳細は第６章でも学習する。

(a) 入力業者の利用

一般的によく利用されているのは入力業者であり，その業務内容は多岐にわたっている。

① パンチ・カード，紙テープ，磁気テープ入力

原票を外部に持ち出して穿孔を委託するケースと，時間的制約，機密保持から要員派遣を受けるケースとがある。

カード穿孔作業は，熟練を要する仕事であることも利用普及の要因となっている。

② 伝票等からのコーディング委託

手作業から，コンピュータ・システムへ移行するときなど，伝票項目のコード化の作業を委託するケースがある。

そのほか，定常的な仕事の中でも大量のコード化が必要な場合は，コーディングを委託し，入力作業の軽減に役立てているケースもある。

(b) 計算処理の委託

計算処理を専門会社へ委託する利用方式は，次のように大別できる。

① クローズド方式

処理すべきデータを渡し，コンピュータ処理は，専門会社が行い処理結果を受取る。

② オープン方式

コンピュータを時間借りして，オペレーションは自社要員が行う。

この方式を利用する理由は，

・新しいコンピュータが導入設置される以前にプログラムのテストを実施したいとき

・データ処理の機密保持の強い要求から自社要員がコンピュータを操作するとき

いずれの方式も一長一短はあるが，計算を委託する専門会社の管理運用体制の充実化の進展に伴なって，クローズド方式の利用が一般的である。

(c) システム開発作業の委託

委託する大部分は，プログラミング作業中心であるが，開発を全面的に委託するケースもある。

委託する作業としては，

① 調査ならびに分析

② システムの設計

③ プログラム設計

④ コーディング

⑤ プログラムのテスト

などがあり，各々の分野で得意とする専門会社と，一貫生産受託型の専門会社がある。

(d) コンピュータ・オペレーションの委託

コンピュータ・オペレーションは，シフト体制を要求されることが多く，特殊な勤務形態となる。

このため，専門会社からオペレーション専門要員の派遣を受けマシンオペレーションを担当してもらうケースがある。

また，コンピュータの運用を全面的に委託する方法もあり，この方式は（ファシリティ・マネジメント）と呼ばれている。

(e) その他の業務委託

そのほか，次のような業務の外部委託がある。

① 漢字処理のための業務

漢字入力，漢字処理と出力

② COM（Computer Output Microfilm）処理

・COMの作成全面委託

・COMの現像，カッティング

③ 通信端末機オペレーション委託

④ 磁気ファイルの保管

⑤ その他関連サービス

以上のような利用方法があるが，外部専門会社へ委託する要因は次のようなことに集約される。

① 内部要員の不足

・量的・質的不足の補充

・要員の勤務時間・ローテーションの問題解決

② 採算性の重視

・専門会社による合理的コストの利用

・設備投資負担の軽減

外部専門会社を利用する場合は，合理性に重点を置きがちであるが，処理上の機密保持，納入品の品質・納期などから信用のある会社の選定にも留意する。

(5) 要員の固定化傾向

コンピュータに関する技術上の特異性から要員の固定化傾向があり，他部門との人事交流を困難にしている。

固定化の要因をあげると次のようになる。

(a) 専門的要員と経験年数の長さ

① システム・エンジニア

システム・デザイナ

システム・プランナ

システム・アナリスト

などと呼ばれ，調査・分析・計画立案・設計などに，豊富な経験と専門的能力が要求される。

通常の場合は，プログラマ，電算機のオペレータとしての経験を積みシステム・エン

ジニアとなるため，長い年数が必要となる。

② シニア・プログラマ

プログラミング作業の中には，高度な技術を必要とするプログラム設計およびコーディング分野がある。

特に，アセンブラ言語を使用しての汎用（ユティリティ）ソフトウエア，オンライン・コントロール・プログラムの開発などは専門的技能を要求される。

③ 通信関係エンジニア

オンライン・リアルタイム・システムの運用は，通信関係の専門技術者を必要とする。

・データ通信システム

・データ通信機器

・電電公社のサービス内容，折衝

などの専門的知識が求められる。

④ データ・ベース管理者

コンピュータで利用するデータ・ファイル，すなわち，データ・ベースが増大してくると，一元的にデータ・ファイル内容を管理しておく必要が生じる。また，最近のコンピュータ利用方法の一つとしてデータ・マネジメント・システム（データ・ベース・システム）の導入が盛んになり，単に物理的なファイルの保存管理にとどまらず，データ内容を一元的に把握し管理する専門管理者の設置が要請される。

(b) 要員教育時間の長さ

コンピュータ技術は，たえざる進歩を伴っており，知識の陳腐化を防ぐために継続的な要員教育が行われる。教育は，社内教育・外部教育のほかにOJT(on-the-job training)の期間が最も長く必要である。

1.2 組織運営形態のいろいろ

コンピュータ部門は，企業の長い歴史の中で，比較的新しい部門である。

日本のビジネス・アプリケーションにおけるコンピュータ利用は，昭和30年，証券界へのユニバック120型電子計算機導入に始まる20数年間という短かい歴史でもある。

しかしながら，この部門は，コンピュータそのものの急速な進歩，それに伴う利用技術のめまぐるしい発展から技術革新の波を十分に受けており，また，コンピュータ利用規模の格差からさまざまな運営形態となっている。

形態としては，概略次のように分けられる。その特徴を実例をまじえて説明することが望ましい。

① 一貫生産型

② 部分的独立型

③ 本社集中型および支社分離型

④ その他の形態

(1) 一 貫 生 産 型

企画・開発・運用・評価に至る全機能をもち，一貫生産型とも呼ぶべき形態である。

この形態を採る場合は，コンピュータ部門の規模の大小はあまり関係してこないことを説明する。

組織のもつ機能については，単元D2で学んだ企画から評価に至る代表的なものを挙げる。

(a) 長期計画の作成

経営からのニーズを受けて，長期的なシステム計画を立案し，作成する。

長期計画は，新しいコンピュータの導入とそれに伴う新しいシステムの建設計画を含むケースが多い。

① 長期的視野にたった，現行システムの改善ならびに再開発

② 新しいソフトウエア技法の利用

③ MISの構築

などが考えられる。

(b) 各ユーザ部門のシステム化調査

システム化されているものの再評価ならびに新しいユーザ・ニーズにもとづく調査・分析作業

(c) システム化の立案から開発まで

ニーズの整理・評価にもとづくシステム開発計画を立案し，実施する。

この実施作業の中に，設計・プログラミング・テストなど人手を多く要する作業が含まれ，コンピュータ部門の活動の中心となる。

(d) オペレーション

① 入力処理

パンチ・カードの穿孔作業，磁気テープ直接入力作業などコンピュータ入力媒体を作成し，検証する作業。

② 電算機の操作（オペレーション）

電算機運行のためのスケジューリングを行い，それにもとづいてオペレーションを行う。

③ 出力処理

コンピュータから出力される諸帳票の検査，カッティング，製本，配布を行う。

(e) データ管理

磁気ファイルの保存，使用する帳票の在庫管理などを行う。

(f) 機器設備関係の導入と管理

機器の使用状況，運行状況ならびに保全を行う。

また新しい設備導入に関する計画とその実施を行う。

(g) コスト・コントロール

コンピュータ利用に関するすべてのコストを把握する。

企業によっては，利用部門へのコスト配賦を実施しているケースがある。

(2) 部分的独立型

(a) 前述(1)の諸機能を内部組織として分化させているケース

（例 示）

システム企画機能……システム企画部

システム開発機能……システム開発部

電算機処理機能……計 算 セ ン タ

また，システム開発は，プロジェクト制（タスク・フォース）を採用する例もあり，コンピュータ部門は，そのコントロールのみ受け持つケースもある。

(b) 組織機能の一部を外部情報処理サービス業者へ委託するケース

最も多い委託形態としては

① 入力データ作成

② システム開発のうちのプログラム作成

③ コンピュータ・オペレーション

があげられる。

外部委託の形態は，

① 部分的（一部）委託

② 複合的委託

③ 全面的委託

に分けられるが，その企業としての経営ポリシー，環境，企業形態などから，種々の利用形態となっている。

(3) 本社集中型と支社分散型

コンピュータの設置場所，システムの開発など，本社にすべて集中する場合と，支社に，それぞれ機能を分散する場合とがある。

日本の場合，一般的に金融機関等は集中し，製造業等は，各工場などに機能を分散してい

る傾向が見られる。

どちらのケースも一長一短があることを説明する。また，システム開発機能のみを本社に集中するなど中間の形態がある。

最近は，ミニ・コンピュータの性能向上，データ通信技術ならびにコンピュータ間のコミュニケーション・ネットワーク技術が急速に進歩してきており，コンピュータの地域分散化，負荷分散化が指向されはじめた。

(4) その他の形態

コンピュータ部門としての機能の分化以外に，組織形態が異なる要素があり，それらをあげる。この要素は(1)～(3)までの形態を採る条件ともなるが，どちらかというと部門の運営に関する要素であろう。

(a) 保有コンピュータの規模によるもの

これは種類，台数，大型コンピュータ，中小型コンピュータ，ミニ・コンピュータなどにより組織の形態に影響する。

(b) コンピュータ利用形態によるもの

① 運転時間の長さの程度

シフト体制をとるかとらないか

② オンライン・リアルタイムの利用

③ バッチ処理量の程度

次図に，EDP部門組織の一例をあげる。いくつかの組織の実例を学習し，比較してみること。

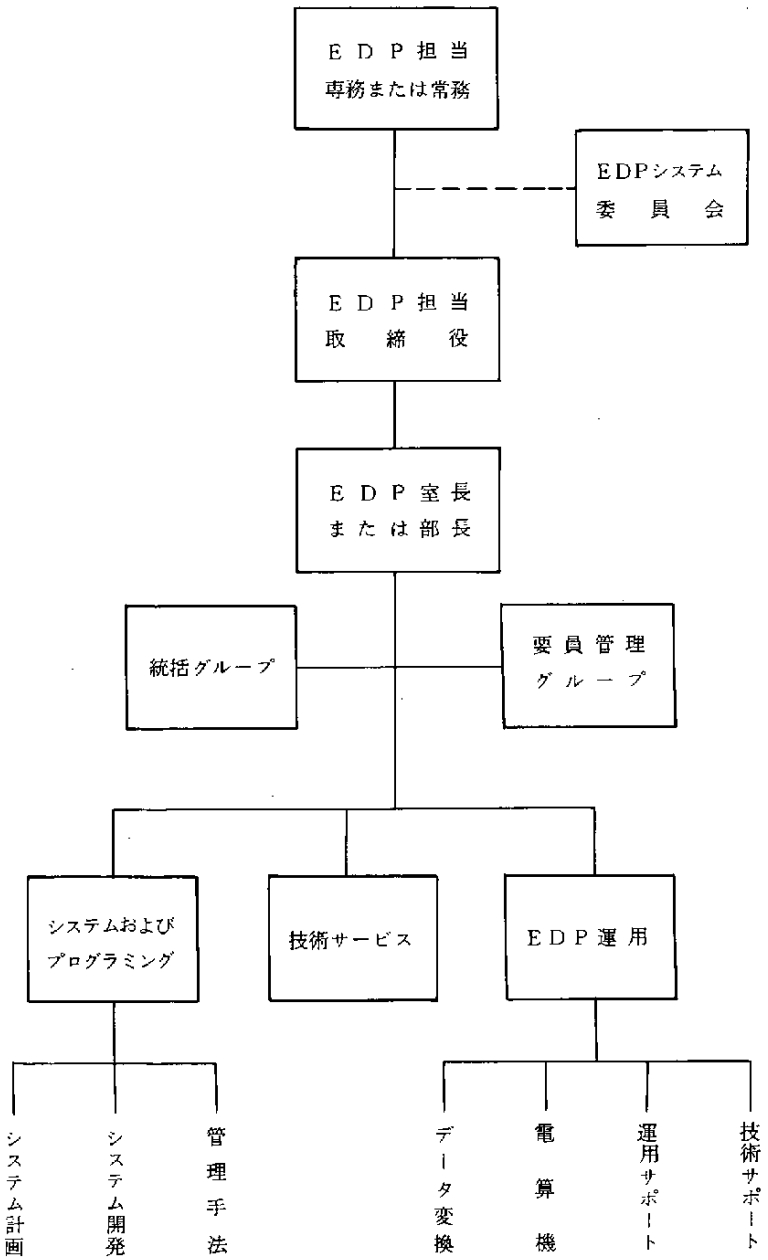


図1.1　EDP部門組織（例）

1.3 コンピュータ部門の義務と責任

この部門は，企業の生産・販売活動全般に亘る情報を集中的に把握管理しており，また経営合理化の先兵でもある。1.1組織の節でも述べた如く，経営の結びつきも強く，それだけ義務と責任は大きい。

コンピュータ部門として，特に留意すべき部門運営上のポイントを次のような視点で説明する。

・効率的活動の推進

・情報管理の重要性認識

・サービス性の保持・向上

・経営との結びつき

(1) 効率的な活動の推進

コンピュータ部門は，経営合理化の先兵的な役割りを持っている反面，組織内活動面の合理化が遅れ勝ちであることを学ぶ。

この点からも次のような諸活動を推進し効率化をはかっていく。

(a) 新技術の積極的導入

システム開発上の新しい技術について説明し，学習する。

① 最近は，システムズ・アプローチや，プログラミング手法など，システム開発技法に関する新しい考え方も生まれてきており，この面での新技術が次第に確立されてきた（第3章参照）。

・トップ・ダウン・アプローチ

・モジューラまたはフェーズド・アプローチ

・モジューラ・プログラミング

・ストラクチュアード・デザイン

・ストラクチュアード・プログラミング

・テーブル・ドリブン・エディター

などプログラミングの生産性向上に役立つと同時に保守性の向上もはかることができる。

② 「チーフ・プログラマ・チームによるプログラミング」などの新しい試みも提案されており，プログラム作成の効率化も工夫の余地が残されている。

③ データ・ベース・マネジメント・システムなどの支援ソフトウエアも完備されつつあり，メーカ提供の支援ソフトウエアはもとより，市販の支援ソフトウエアも検討すべきであろう。

これらのソフトウエアの使用により活動の効率化をはかっていくことができる。

(b) 要員の質的向上

要員の質的向上となるいくつかの施策をあげ説明する。

① 社 内 教 育

部門内の教育が主であり，教育育成計画をもっていることが望ましいが，新入部員教育以外は，あまり実施されていないのが実情である。

② 社 外 教 育

コンピュータ・メーカの講習は，詳細な教育研修スケジュールと研修機関を完備しており，多くのユーザが利用している。

一方，中立的な権威ある研修機関として日本情報処理開発協会，情報処理研修センタが通産省の要請を受けて設立されており，多くの人々に利用されている。

③ セミナー等の受講

メーカー主催のもの以外に情報処理関係団体等により，かなりの頻度でセミナーが開催されている。特に，情報化週間に多くの催しがなされており，積極的な参加による新知識の導入およびレベルアップの機会として利用すべきであろう。

④ 雑誌・図書等の充実

雑誌・図書等の充実をはかり，利用を促進する。

個人の知識向上のガイドラインとなるような図書数を揃えておくよう留意する。

情報処理関係の技術進歩は激しいのでその更新に留意する。

⑤ 情報処理技術者試験の受験

通産省が主催するこの試験は，毎年1回10月に実施されている。受験を目指しての勉強が，個人の知識向上となりコンピュータ部門全体のレベル・アップにもなるため，企業によっては，受験料の負担，講習会の実施などの支援を積極的に行なっている。

要員向上のためのいくつかの方法をあげたが，重要なことは，部門としてのポリシーあるいはフィロソフィー（哲学）の確立が第1であり，管理職またはそれに準ずる地位の人達が要員育成に対する積極的な姿勢と要員育成計画をもつことである。

(c) スケジューリングの重要性認識

コンピュータ部門の活動は定常的なものと計画に準拠して製作していく非定常的なものが混在している。

システム開発にみられるような計画立案からはじまる一貫生産的作業は，綿密なスケジューリングを要する。

システム開発に関するスケジューリングと，その管理のためのポイントをいくつかあげておく。

① スケジューリング管理専任者または，管理セクションを置く。

② 年間または半年間の部門スケジュールをもつ。

③ スケジュールのうち，クリティカル・パスを正確に把握し最も注意する。

④ スケジューリングのための基準（標準）をつくる。

⑤ スケジュールの実施状況は，週単位，月単位で把握でき，遅れについて早期発見と解決を迅速に行う。

⑥ 要員のスキル・レベルを詳細に把握する。

⑦ スケジューリングの手法としてCPM (Critical Path Method) を利用する。

⑧ テーマ（タスク）別および個人別スケジュール状況を把握する。

⑨ 活動報告を含めて，文書の標準化をはかる。

システム開発のスケジューリングのほかに，コンピュータ運用のためのスケジューリングもこの部門の一方の任務である。

コンピュータ運用管理は第5章で詳細を学習するが，コンピュータ運行スケジューリングのポイントをいくつかあげる。

① 運用する個々のプログラム（タスク）の処理時間をできるだけ正確に把握しておく。

② 対象とする電算機の性能について計数的に把握する。

③ 多量処理時の処理時間または処理能力について把握する。

④ データ量と処理時間の関係について計数的に把握する。

⑤ 障害等による再試行のための余裕時間を設けておく。

⑥ 月次処理等のピーク時のスケジューリングには，処理優先度を十分考慮する。

⑦ 定常的業務（デイリー業務）については，標準処理時間表を完備しておく。

⑧ 予防保守のための時間を考慮する。

このようなスケジューリング可能なコンピュータ処理以外に，タイムシェアリング処理，リモート・ジョブ・エントリ処理など，ユーザオリエンテッドなコンピュータ利用方法が発達してきており，従来の定常的なスケジューリングでは対処できなくなっていることも事実である。

(d) コスト対効果把握

コンピュータ処理におけるコスト対効果把握は，経営合理化の観点から，その必要性を学習する。

通常，省力化による効果は，直接的効果として算出可能であり，また，新しいシステムを実施後もトレースできる。

しかし，質的メリット，すなわち，間接的効果は計量不可能であり，システム推進者の

主観が入り勝ちであるが，できるだけ客観的な尺度を用意しておくことを説明する。

間接的効果のいくつかをあげる。

① ターンアラウンド時間の短縮

② コンピュータ印字による出力の見易さ

③ 企業イメージの向上

④ 計算の正確性による信用向上

⑤ 管理の質の向上

これらを別の角度からみると，①心理的効果と②管理的効果に分けられる。

これらの効果は，従来のものと比較して具体的に示すことがむずかしく，また複合的な効果として現出するため，その把握をより困難にしている。そのためややもすると過少評価される傾向がある。

効果をシステム計画時点で明確にする場合は，特に一次的なものについて考え，他はプラス・アルファとして見積るべきであろう。

コンピュータに関連するコストは，利用部門へ，その開発度合，利用度合によって配賦されるのが経営的にみて最も望ましい形である。配賦基準の設定は必ずしも標準となる一般的尺度はなく，各企業により種々の方法が採られている。

コンピュータ部門として把握すべき費用としては，次のようなものがある。

① 開 発 費 用

- 人件費
- 経　費（備品器具，消耗品，家賃など）
- 外部へのプログラム委託費
- テストのための電算機使用費

② 運 用 費 用

- 電算機器費用（償却費，レンタル料，リース料）
- 端末機器費用（　〃　）
- 附帯設備費用（　〃　）
- オペレータ人件費
- 備品・器具償却費
- 電力料・水道料
- 消耗品（磁気テープ，用紙など）
- その他一般経費

(2) 情報管理の重要性

この部門は情報を製造し，管理する以外に，情報の内容についての詳細な記録（ドキュメント）も整理されている。

また，情報に対する安全性，信頼性，機密保護など，情報の管理について慎重な計画と実施が要求される。

管理すべき情報の種類をあげる。

① 磁気テープ

② 磁気ディスク・パック

③ システムおよびプログラム・ドキュメント

- システム設計書
- プログラム仕様書
- オペレーション仕様書（指示書）
- プログラム・リスト
- その他システム・ドキュメント

④ 穿孔カード

⑤ 原始入力伝票

⑥ コンピュータ・オペレーション・シート（コンソール・ログ）

⑦ コンピュータ出力帳票

⑧ COM（フィルム）

これらの情報は，必ずしもコンピュータ部門で保存管理されているとは限らない。

コンピュータからの出力帳票の保存については，コンピュータ部門による保存期限・方法の決定などを行ないユーザ部門に指示することが望ましい。

(a) データの安全管理

情報の重要性に鑑み，その保存管理の安全性を確保しなければならない。

そのためには，

① 管理基準をつくる。

② 安全管理のための設備を充実する。

③ 専任の管理者を置く。

④ 定期的なチェックを行う。

などを実施する。

磁気ファイルの保管は，この部門の責任であり，安全管理のために考慮すべき点を列挙し説明する。

① システムおよびプログラム設計時に保存すべきファイルを明確にし，世代によるローテーションを行う。

② その企業にとって重要ないくつかのファイルは，別々の場所へ二重保存する。

③ 磁気ファイルのファイル保管室からの出し入れは，専任者により行うようにする。

④ 保管場所の条件，二重保存などから，ファイル保管専門会社の利用も検討する。

⑤ 磁気テープ，磁気ディスク・パックなど購入時に整理番号を付与し，台帳を完備しておく。

⑥ 定期的に在庫状態を検査する。

⑦ 保存場所の温湿度条件を良好に保つようにする。

⑧ 磁気テープの場合，長期保存に限界があるので，一定の周期でコピーすることを考える。

(b) 機密保護

機密保護は，前項(a)の安全管理とも密接な関係があり，まず，情報保存の安全が保持されることが前提となる。

機密保護は，コンピュータで利用するデータに関してだけでなく，システム活動において発生するいくつかの機密事項についても保護対策を講じておく。

① データに関する機密保護

データ・ファイルが安易に取扱うことができないよう，いくつかの配慮を施しておく。

・コンピュータのオペレーション指示は，特定の管理者によって行う。

・可能ならば，コンピュータ・オペレーションは専任オペレータが行い，他の者はコンピュータ・ルームへ入室不可能とする。

・プログラム・テスト等に本番データの使用は極力行わないよう工夫する。

・コンピュータ出力のうち，特に機密を要するものを出力するときは，責任者の立ち会いを行う。

・ミスとなった出力帳票は，裁断または焼却を責任者立合いのもとで行う。

② オンラインにおける機密保護

オンライン・リアルタイム，タイムシェアリング，リモート・ジョブエントリーなど遠隔地から通信回線を利用してのコンピュータ・ファイルからの出力は，コンピュータが中央のものだけでなく，極めて手近かなものとして活用されることになる。

そのため，不特定多数の人達が多くの端末機を利用して，情報を容易に得ることができる。

したがって，この面での情報の機密保護に対するいくつかの考慮が必要となる。

・端末機に保護機能を設ける

一般的なものとして鍵穴を端末機に作り，操作する人，それぞれに異なった鍵(Key)を与えておく。そのKeyがないと特定の情報が得られないようにする。

その他の方法として磁気カードがある。銀行におけるキャッシュ・ディスペンサの暗誦番号は，その一例であろう。

・ファイル（ランダム・ファイル）に保護キー（プロテクション・キー）を設ける。

このキーは暗号であり，この番号を用いないと該当するファイルからの操作はできないようにファイルを保護する。

・プログラムの保護キー（プロテクション・キー）

ファイルと同様，暗号を設けて，該当するプログラムを利用する場合，その利用を制限する。会話型のタイム・シェアリングなどで使用されることが多い。

③　文書に対する機密保護

システム設計からプログラミングにおける文書（ドキュメント）は，情報に関する機密事項を多く含んでおり，保存管理を厳重にすることによってのみ機密は保持される。その意味から，専任者としてライブラリアンを置いている場合がある。

④　要員のモラル向上

どのように厳重な機密保護対策を講じても，コンピュータ部門の要員のモラルが高く保たれていないと片手落ちである。

そのためには，個々の作業の重要性を認識することは当然のこととして，この部門の責任の重さを徹底するよう心掛ける。

日本の国民性から，機密保護については，ややもするとその重要性に対する認識がとだえ勝ちである。管理者は，定期点検などを通じて問題点なり改善策を見出し，着実に保護対策を施していかなければならない。

ケース・スタディとして諸規定の設定を試みる学習方法も有効であろう。

(c)　コンピュータ室の安全管理

災害などにより万一コンピュータが長時間停止することがあると企業にとって致命傷となりかねない。

そこで，コンピュータの安全管理のためには，さまざまな工夫が必要となる。

コンピュータ室の安全管理上，主要な事項をあげ説明する。

また，安全管理の観点から代表的なコンピュータ・センタの見学による学習が有効である。

①　コンピュータ設置場所

火災などの災害を考えた場合，つぎのような場所が望ましい。

・堅ろうな鉄筋コンクリート・ビルディングで耐火構造であること。

・場所として，周囲に民家等がなく類焼の恐れがないこと。

・地盤の比較的しっかりした場所であり，地震にたえられること。

・２台以上のコンピュータで地域分離が可能ならば，その考慮をする。

・厳しい条件のオンライン・リアルタイム・システムをとる場合，電力会社からの配電が２系統受けられること。

・浸水などの恐れがないこと。

以上のような点を満足することは，企業の立地条件などで困難な場合が多い。大規模なコンピュータ設置を行つているケース－特に金融機関に多くその例を見ることができる－は，コンピュータ専用ビルディングを設け，諸条件を具備している。

また，コンピュータ専用ビルを賃借し，多くの企業が共同入居してコンピュータを設置している例もある。

② コンピュータ室の保安

コンピュータ室の安全をはかるためには，部屋の設計に次のような配慮が必要となる。

・電算機および必要機器を設置する十分な面積を確保する。

・床に十分な強度をもたせる。

・関係各室との連絡をとりやすくする。

・振動・電気ノイズに問題がないこと。

・塵挨対策をとること。

・火災に対する予防処置をとること。

・容易に入室ができない設備にする。

これらの設備の安全性確保のために管理上の諸基準を設け，それを遵守していく。

このように基準を設けたり，厳重な管理を実施しても，かつ，災害に会うことが考えられる。そのための保険として「情報化保険」がある。

この保険については，保険会社とよく相談して利用すべきであろう。

また，通産省情報産業局・情報処理振興課がまとめた資料等を参考に学習する。

(d) 監査制度の採用

ＥＤＰ監査は，企業内の監査のみならず外部監査人に監査を委託する例もある。

日本公認会計士協会による「内部監査の手引書」，昭和52年３月に日本情報処理開発協会がまとめた「システム監査体制確立への道」などを参考にし，内部監査の手引として活用することにより，機密保護・安全管理が一層充実することが考えられる。

理想的には，内部監査の専任者またはグループを設定し，種々の点についての定期的な

監査を実施するとともに，外部監査人を並行的に活用することであろう。

(3) サービス性の保持・向上

コンピュータ部門は，利用部門（ユーザ）に対し，

・処理の正確性保持

・迅速なシステム開発

・迅速な情報提供

・利用部門の要請に対する適確な判断

・利用部門からのニーズ把握と整理

・運用に関する手続き訓練と指導

など，高いサービス性を保持すると同時にその向上をはかっていく義務がある。そのためには，新しい技法を積極的に導入して改善をはかっていきながら，利用部門とより良い接触を保持する必要がある。

(a) 利用部門（ユーザ）とのコミュニケーション

コンピュータ部門は，その性格から，ユーザ部門と遊離しやすい状態になることが考えられ，種々のコミュニケーションをとるよう工夫する。

① 定期的会合をもつ

職制ごとの定期的な会議により，両方の事情をよく理解し合う場を設ける。

② 重要事項決定機関を設ける

システム化の要請を全社的な視点から決定する第3者機関を設ける。

機関の名称としては，「EDP委員会」，「EDP推進会議」などと呼ばれ，EDP担当役員などが主催する企業レベルの最高意思決定機関もあれば，部長クラスで構成される場合もある。

③ 要員のローテーション

コンピュータ部門と利用部門との間でかなり意識的に要員の交替をはかっていき，コミュニケーション・ギャップを解消する。

④ 要員の社内留学制度を設ける

コンピュータ部門から利用部門へ，また，その逆の企業内留学制度を設け，人の交流により両方の意志疎通をはかる。

⑤ システムのPRを心掛ける。

実施中のシステムおよび新しく計画したシステムの内容について書きものとしてまとめ，コンピュータ部門はPR活動に心掛ける。

その他，パンフレット・スライド・映画などを用意して，利用部門にシステム内容や

コンピュータ部門についての理解を求めるためのＰＲ材料にする場合がある。

⑥　ユーザ部門とコンピュータ部門との間での手続きの明確化

運行しているシステムへの改善要請および新しいシステム開発への諸要求事項は，口頭による伝達だけではなく，一定の文書ならびにその手続きを制定して利用することが望ましい。

「改善要望書」，「開発要望書」などを設け，コンピュータ部門はその内容を検討し，回答欄に回答を記入したのち，迅速にユーザ部門にフィード・バックする。

このような文書を制定することにより，ニーズの整理が実施できる。

(b)　処理の正確性保持

正確性を保つためには，プログラム作成から運用に至るまで，いくつかのチェック・ポイントを設け，利用部門に対する信頼性確保につとめなければならない。

システム開発上の留意点は，単元Ｄ１，Ｄ２を復習してみること。

ここでは，特にテストについて復習し運用管理上の配慮点を学習する。

①　テスト計画と実施の徹底

テスト計画に折り込むべき事項には，つぎのような項目がある。

- テストのスケジュール
- テスト・データの作成方法
- テスト結果の検証方法
- テストに使用する支援プログラム
- ユーザ部門のテスト参画
- 所要物品の調達と取扱い
- 稼動に関する時間予測と計測実施

システム・テスト段階は，従来の手作業に代って十分な適用性があるかどうか検証すると同時に，利用部門における新システムへの習熟を併せ狙う意味をもっている。

システム・テスト段階では，システム移行の可否を判断するため，次のような点に留意する。

- インプットの到着状況
- アウトプットの利用部門への到着状況
- インプットのエラー発生状況
- プログラムの適応性
- オペレーション状況

システム・テストは，大規模なオンライン・リアルタイム・システムの場合，長い期間と大量のテスト要員およびマシン・タイムを必要とする。また，テストの結果によっては，かなりの部分の再開発を余儀なくされる場合もあるが，実施・移行後に問題が発生することを可能な限り事前に防止する意味をもっている。

② 運用管理の徹底

システム開発が完了し，コンピュータによる定常的な運行が開始されたのちも，処理の正確性を保持するために，運用管理上作業体系を明確化し，それぞれについての責任分野を定めておく。

システム運用に関する作業体系としては次のものがある。

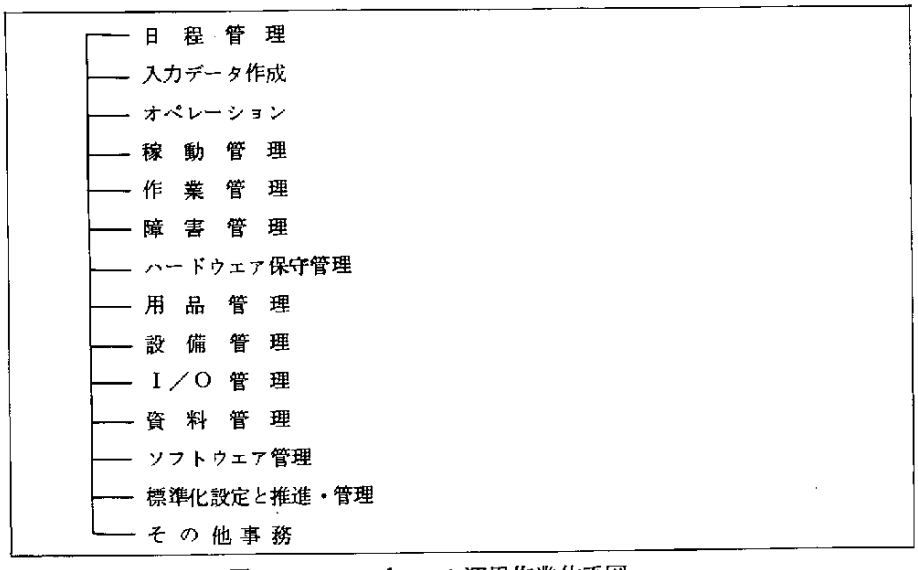


図1.2 コンピュータ運用作業体系図

正確性保持のための配慮点をあげ解説する。

〈日 程 管 理〉

定常的なスケジューリング，すなわち，日時・週間・月間のスケジューリングを立て，それを維持することが中心になるが，スケジューリングの阻害要因とその対策が必要であり，そのためには余裕を見込んだスケジューリングが望ましい。

スケジューリングの阻害要因は次のようなものがある。

・臨時の飛込み業務

・プログラム・テスト（デバック）

・再試行（リラン）

〈入力データの作成〉

入力データの種類は多様化してきており，いずれの場合も，正確なデータをコンピ

ュータに入力するための検査手続きを確立しておく。

入力データは，それぞれについてエラーを検出する方法は異なるが，本格的な処理を行なう前に，人手による検査，プログラムによるチェック強化をはかる。

〈オペレーション〉

コンソール・オペレーション，ファイルやプリンタのハンドリングなど機械操作，オペレーション準備，実績の記録などの作業がある。

ここでは，操作ミスの原因の一つとしてプログラムの質が問題となることから，プログラム設計およびプログラム作成時の配慮点を習得しておく。

- コンソール・メッセージを多用しない。
- まぎらわしいコンソール・メッセージは出力しない。
- 出力メッセージの標準化を行う。
- ジョブ・コントロールカードからの指示を用いる。
- 磁気ディスク，磁気ドラムなどの設備がある場合は，磁気テープの使用を極力少なくする。
- 標準出力帳票を多く活用し，帳票の種類を少なくする。
- オペレーション指図書の作成に留意する。

〈ファイル管理〉

コンピュータ処理のためのファイルの事前準備および処理後の保存管理であり，コンピュータ処理の中でもファイルの持つ重要性は大きい。

〈稼 動 管 理〉

機械の稼動状況，使用効率，作業の実施状況などを把握し，総合的に稼動に関する管理を行う。

〈ハードウェア保守管理〉

通常の場合，コンピュータおよび周辺機器などを導入した場合，メーカーないしは，販売各社と保守契約を締結し，保守を担当してもらうことになる。

事前保守のためには，十分なマシンタイムを提供することに留意する。

部品の点検，定期的な保守，障害記録にもとづく対策などにより，正確なデータ処理がより保証される。

〈Ｉ／Ｏ 管 理〉

入力データの受付窓口であるせん孔作業の手配，データのチェック，引渡しなどを行う。正確な入力データの確保および，コンピュータ出力の利用部門への引渡前のチェックなど，このセクションにおける処理の正確性はユーザとの接点であり，コン

ピュータ処理に対する信頼性を高めることになる。

〈標準化の設定・運用〉

システム設計やプログラムの標準化のみならず，運用管理面で標準化可能な部分は，可能な限り標準・基準を設定し，継続的な改良と運用を心掛ける。

(c) 迅速なシステム開発と情報提供

利用部門の要求を満足させるためには，正確な情報処理とともに，システム開発および情報の提供の迅速化である。

システム開発ニーズに対して迅速に答えるためには，開発要員とコンピュータ・タイムを増加させることも一つの方法である。

しかしながら，必要とするコストを考え合せれば，企業規模に見合った適切な規模があり，自ずと限界があるので種々の工夫が要る。

単元D2の復習の意味からシステム開発の効率化に役立つ方法をいくつかあげ解説する。

① 高級プログラミング言語の採用

アセンブラ言語の使用は，多用される共通サブルーチンに限定し，COBOL，FORTRAN. PL／1などのコンパイラを極力使用する。

② プログラム・ジェネレータの使用

レーポート・プログラム・ジェネレータなどの例にみられるように，コンパイラ言語をつくり出すプログラム・ジェネレータを使用することによりプログラミングの生産性を向上できる。

③ 汎用プログラムの利用促進

コンピュータ・メーカーの作成・提供した汎用プログラムを利用するだけでなく，使用頻度の高いプログラムを順次定型化し，汎用プログラムとして使用することは，生産性向上となる。

④ ユーザが利用できる簡易言語の提供

使用できるデータ・ファイルを限定して，ユーザ部門において簡単に利用できるデータ出力用の簡易言語を提供する。データ・ベース・マネジメント・システムを利用するような場合，このような簡易言語をもっているケースがある。

この言語をユーザが使用することにより開発部門の手をわずらわさずに，データ出力を直接得ることができる。

⑤ データ・ベース・マネジメント・システムの利用

最近の傾向として，データ・ベース・マネジメント・システム (DBMS)を活用する例が増加してきた。このシステムは，メーカーから提供されるものや市販されているも

のがある。DBMSの導入には慎重な検討が必要であるが，開発および保守の容易性を増加することができる。

⑥ タイム・シェアリング・システムの利用

タイム・シェアリング・システム（TSS）およびリモート・ジョブ・エントリー・システム（RJE）の利用は，コンピュータ・プログラムを端末機から投入することによって，処理結果を早く得ることができ，開発の迅速化をもたらす。

⑦ オープン・プログラマ制の採用

ユーザ部門でプログラミングを行い，コンピュータを利用する制度であり，前述⑥のタイム・シェアリングの利用とあいまって有効な制度である。

しかしながら，コンピュータ部門は，その運営管理に十分な配慮を行わないと利用上の混乱が発生する。

⑧ 保守の容易なソフトウエアの作成

ソフトウエア保守の増加は，保有プログラムの数および開発の時間的経過もあるが，最近のコンピュータのファミリ・シリーズ化も，プログラムの老朽化の一因となっている。

保守の容易なソフトウエア作成に留意することは，変更などに対する処理を容易にし，ニーズに迅速に答えることにもなる。

保守の容易性の面から，ソフトウエア作成上の留意点をあげると，それは開発の効率性向上のための手法とかなり一致する。このことは，ソフトウエア作成という一断面でなく，開発面での一貫性が必要なことを物語っている。

保守の容易性とソフトウエアの老朽化を防止するいくつかの点をあげる。

- 徹底したモジュール化をはかる
- 文書を完備する
- ユティリティ，サブ・ルチーンの多用
- 共通テーブルの活用
- 実行タスク（プログラム）を大きくしない

⑨ リアル・タイム化指向

オンライン・リアルタイム・システムの導入はコスト面で大きな負担となるが，また，大きな効果を得ることができる手段でもある。すでにリアルタイムを実施している場合においても，その規模を順次拡大することにより情報提供の迅速化が推進される。

オンライン・リアルタイムを導入する場合は次のような多くの困難さがあり，その打開が前提となる。

・コストが上昇する

オンライン可能なコンピュータの設置，通信端末機器の導入，通信回線料の発生などバッチ処理に比較して相当額のコンピュータ・コストの上昇となる。

・高水準の利用技術の要求

通常のバッチ処理と異なり，通信関係，障害復旧対策，プログラム構造設計などかなり高度の利用技術を習得する必要がある。

・障害対策への設備投資負荷

オンライン・システムの停止は，利用部門にとって致命的であり，可能な限り短時間の停止にとどめなければならない。そのため次のような特別の設備投資をもつ場合がある。

・切替え用予備コンピュータの用意

・予備電源装置の設置，自家発電，無停電装置など

・通信用線の２重化ないしは，公衆通信網への切替え利用

その他，無停止システムが必要な場合は，さらに，コンピュータ設備の完全な二重化をはかることになり，相当のコストアップを覚悟せざるを得ない。

1.4　経営とのつながり

1.1で述べたように，コンピュータ部門は，経営と強いつながりをもっている。また，経営との直属的所属形態をとることが多い。

このような経営との強い結びつきは，

①　経営のニーズに迅速に対応できる。

②　他部門との融合性をはかりやすい。

③　経営的視野に立った企画を立案しやすい。

などの利点からである。

コンピュータ部門の組織は，当初，従来から存在する経理部門，総務部門，人事部門，あるいは企画部門などの下部機能として出発した組織体が多く見受けられた。その後，コンピュータによる適用業務範囲が拡大するにつれ，独立した部門を形成するようになった。

このようにコンピュータ部門が独立性をもつに至った場合は，

①　予算規模が相対的に大きくなった。

②　業務処理の中枢となった。

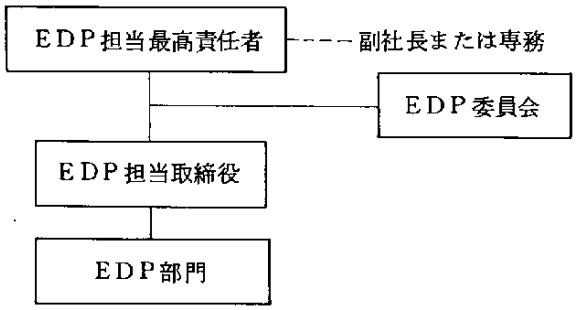
③　コンピュータ利用によるコスト・ダウンが大きな比重をしめるようになった。

などであり，これらのことからトップ・マネジメントは，コンピュータ部門の業務を十分把握し

ておくことが必要である。そしてこのような立場からコンピュータ部門は報告制度を確立し，トップ・マネジメントへ常に適正な報告がなされるよう努力すべきである。

(1) EDP 委員会の設置

EDP 計画の意思決定を行う最高機関として「EDP 委員会」または「EDP計画推進委員会」を設置している例がある。



組織上の位置づけは，図のような個所にあり，EDP部門以外の役員クラスで構成される。

EDP 委員会は全社的立場からコンピュータの有効活用を推進し，評価する委員会であり，主たる役割りは，次の通りである。

① コンピュータ・システム計画の承認

② コンピュータ活用の徹底

③ コンピュータ活用の評価

EDP 委員会によりコンピュータ部門の活動は公正に評価され，新しいシステム計画などがあった場合，推進が一層はかり易いものとなる。

(2) コンピュータ化による企業組織の変容

企業組織の効率化を考えると，コンピュータの高度利用が大きな要素となってくる。このことは，コンピュータ・ゼーションの進展による組織の変容が発生してくる可能性を秘めており，組織変更には経営トップの判断が必要となる。

コンピュータ化に伴う組織の変容について主な点をあげる。

(a) 現場作業の集約化

コンピュータ業務の適用拡大により単純業務は減少し，コンピュータ化できない，窓口的，監視的，営業的業務のみ現場に残ってくる。したがって，このような業務はあらたに集約化・簡略化され組織は変容する。

(b) ミドル・マネジメントの精鋭化

定型的な意思決定の多くはコンピュータにより処理可能となり，システム化され集権化されていく可能性がある。

ミドル・マネジメントは，従来より一層創造的職能に進んでいくであろうし，多くのミドル・マネジメントを要していた部門は，少数精鋭化していく。

(c) トップ・マネジメントの充実

情報伝達手段の充実により，多くの情報が迅速にトップ・マネジメントに集中することになる。また，戦略的意志決定も，数学的手法とシミュレーション手法などのコンピュータ化が進展する。

このことから，トップの意思決定のプロセスは質的に高められる。

(d) ゼネラル・スタッフの強化

組織内の情報のコンピュータによる集中管理により，企画，管理部門が集約化され強化される。このスタッフが，情報の収集・分析・立案を行い，この選択的情報にもとづいてトップの意思決定が行われる。

(e) トータル・システム化

トータル・システム化が推進されると，生産，営業，経理，資材，人事という各職能部門間で，システマティックな情報が共有され，各職能間が有機的につながるようになり，職能間の壁が除かれていく。

(3) プロジェクト・チームの推進

コンピュータ化が進展してくると，従来の個別業務のシステムをトータライズして，トータル・システム（総合システム）とする必要性が生じてくる。トータル・システムは，企業内の職能別組織を超えた，情報の流れにそった横断的サブ・システムの集合としてシステム化されるのが特徴である。

このようなトータル・システムの建設は，コンピュータ部門のみで実行することは困難であり，各利用部門から人員の参加を求めプロジェクト・チームを編成する。コンピュータ部門は，その推進場となる。

このようなプロジェクトは，経営の長期計画の一貫として位置づけられ，定められた期間，資金，人数により目的を完遂するのが特徴である。

指導上の留意点

(1) コンピュータ部門の組織的な位置づけ，特殊性を他部門と対比して説明する。

(2) コンピュータ部門の機能については．複合的機能面から説明し，独立企業体としての観点から指導していくことが望ましい。

(3) この章は，他の章の総集編的な構成になっており，コンピュータ部門の運営管理全般の概要が把握できるよう指導する。

(4) ケース・スタディとして，いくつかの企業の組織を採り上げて解説を加える指導が望ましい。

参考文献

〔1〕 Arnold, E.D., Shaw, J.C. and Atkins, W., "Managing the EDP Function" McGraw-Hill, 1971.

〔2〕 S.Kuehne, R.S., Lindberg, H.W., and Baron, W.F., "Manual of Computer Documentation Standards with Forms", Prentice-Hall, 1972.

〔3〕 岡本行二，「コンピュータのための機密保護と安全管理」 オーム社，1974.

〔4〕 P.W.メッガー，下田正昭，久保末沙（訳） 「プログラミング・プロジェクトの管理」近代科学社，1977.

〔5〕 「ソフトウェアの標準化」 日本経済新聞社，1977.

〔6〕 日本公認会計士協会 「監査資料集」 第一法規出版，1975.

〔7〕 Baker, F,T., "Chief Programmer Teams, Principles, and Procedures," IBM Federal Systems Division Report, 1971.

〔8〕 G.J.マイヤーズ，有沢誠（訳）「ソフトウエアの信頼性－ソフトウエア・エンジニアリング概説－」 近代科学社，1977.

〔9〕 高仲 顕，「ＥＤＰＳの実務」 朝倉書店，1972.

〔10〕 南条 優，保坂岩男共著，「システムの開発と運用」 オーム社，1972

〔11〕 長谷英夫，朝井幸洋共著，「情報処理センター その運営と効果的利用法」 ダイヤモンド社，1974.

〔12〕 ジェームス・H・グリーン，高井英造（訳） 「オペレーションの計画と管理」 東洋経済新報，1970.

〔13〕 吉谷竜一，「日程の計画と管理」 日刊工業新聞社，1970.

〔14〕 松本太郎，大芝 猛共著，「経営情報システム」 東洋経済新報社，1968.

〔15〕 社団法人日本能率協会出版事業部，「ＥＤＰスペシャル・リポート」1977

〔16〕 園谷 勇，「組織動態化の技法と実際」 ダイヤモンド社，1972.

〔17〕 ソフトウエア・リサーチ編，岸田孝一，三田守久，清水功老共著，「標準コンピュータ・プログラミング・全6巻」 日本生産性本部，1973.

〔18〕 南条 優，「コンピュータ室の運用と管理」 日本生産性本部，1971.

〔19〕 中江剛毅，「日本の情報システム管理」 泉文堂，1975.

〔20〕 岩城三郎，「オンライン・プログラム設計入門」 日刊工業新聞社，1973.

第2章　コンピュータ部門の要員管理

キーワード

要員の分類（classification of personnel），要員管理（managing computer personnel），システム分析者（systems analysis），アプリケーション・プログラマ（application programmers），データベース管理者（data base administrators），スーパーバイザ（supervisors），オペレータ（operators），入力処理要員（data entry personnel），通信関係専門家（telecommunication specialists），技術相談（technical service groups），ライブラリアン（librarians），スケジューラ（schedulers），インストラクタ（instructors），要員ローテーション（job rotation），リーダシップ（leadership styles），経験のレベル（experience levels, career paths），教育（education, training），要員の能力（personnel level），管理者（manager）

目　　標

コンピータ部門は，さまざまな職務をもった要員により構成されており，要員管理の優劣が開発から運用にわたる効率性を決定する重要な要素の一つとなる。

この章では，

- 要員の種類の認識
- 要員育成のあり方
- 要員管理のポイント
- 要員のローテーションのあり方

を学習し，各業務における管理職，管理職の補佐，あるいはリーダーとして身につけておくべき知識という勘点で説明するものとする。

内　　容

前章で学習したごとく，コンピュータ部門の組織形態が企業によってまちまちであり，また，企業組織全体の中でもこの部門の位置づけがまちまちである。

この章では，要員管理をより巾広くとらえる意味から，開発から運用までを含む一貫生産型で比較的規模の大きなコンピュータ部門を想定して学習する。

要員管理を検討する場合，つぎのようなことが前提となる。

① 企業におけるコンピュータ導入の経過および利用形態の差異。

② 情報処理要員に対する社会的評価が確立していない。

③ 企業内における要員の地位が不確定である。

また，コンピータ部門における要員管理に係る問題の特質として，次のようなことがいえる。

① 非常に高度な技術が必要であり，技術を身につけるための期間が長い。

② 慣性的な人員不足の傾向がある。

したがって，人的資源の有効利用と人材開発が情報処理業務の効率化を推進することになる。

2.1 要員の分類

(1) 要員の分類を説明する場合，次のような組織を前提として学習する。

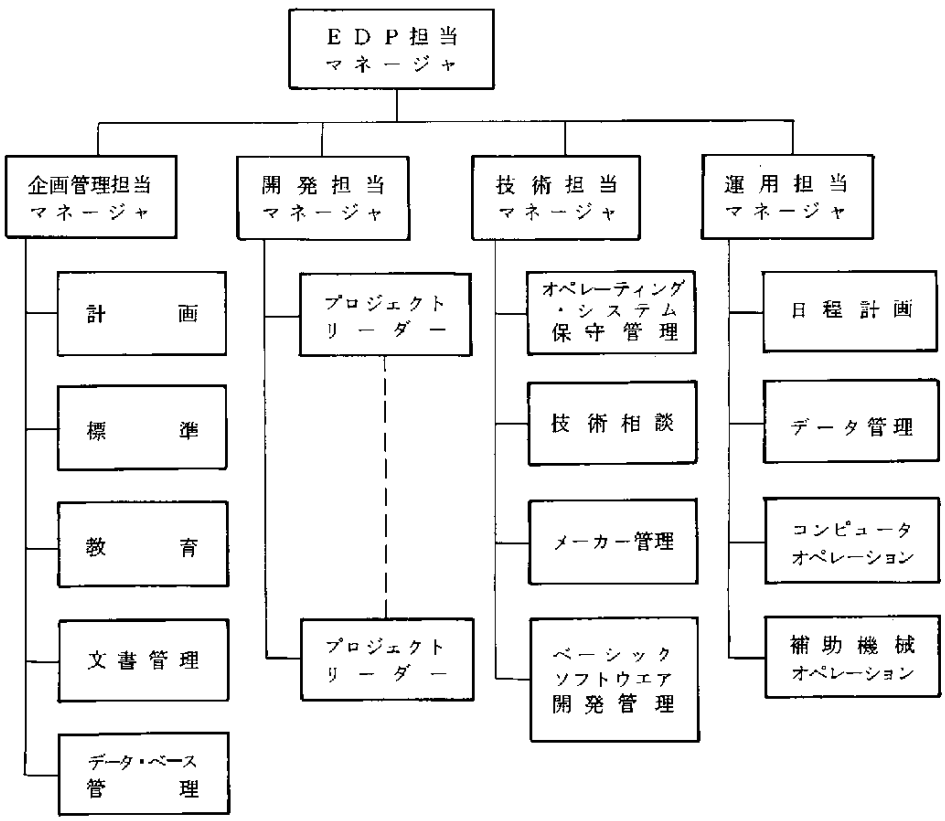


図２－１　コンピュータ部門組織の例

要員は，次のように分類し，説明する。（必ずしも一般的な定義はない）

表２・１　システム設計開発分野の要員

- システム・エンジニア
  - システム分析者（システム・アナリスト）
  - システム設計者（システム・デザイナー）
  - 技術専門家
  - ＯＲ分析者
- プログラマ
  - アプリケーション・システム・プログラマ
  - オペレーティング・システム・プログラマ
  - データ・ベース・システム・プログラマ
- そ　の　他
  - 通信関係専門家
  - データ・ベース管理者
  - 文書管理者

表２・２　オペレーション分野の要員

- スーパーバイザ（監督者）
- コンピュータ・スケジューラ
- コンピュータ・オペレータ
- エントリー・オペレータ（キーパンチャ）
- ライブラリアン（磁気ファイル管理者）
- データ管理者
- 通信機器管理専門家

その他の分類として，米国Infomation Systems誌の1976年度給与調査による職種は，つぎのようになっているので参考とする。

① 管理職の分類

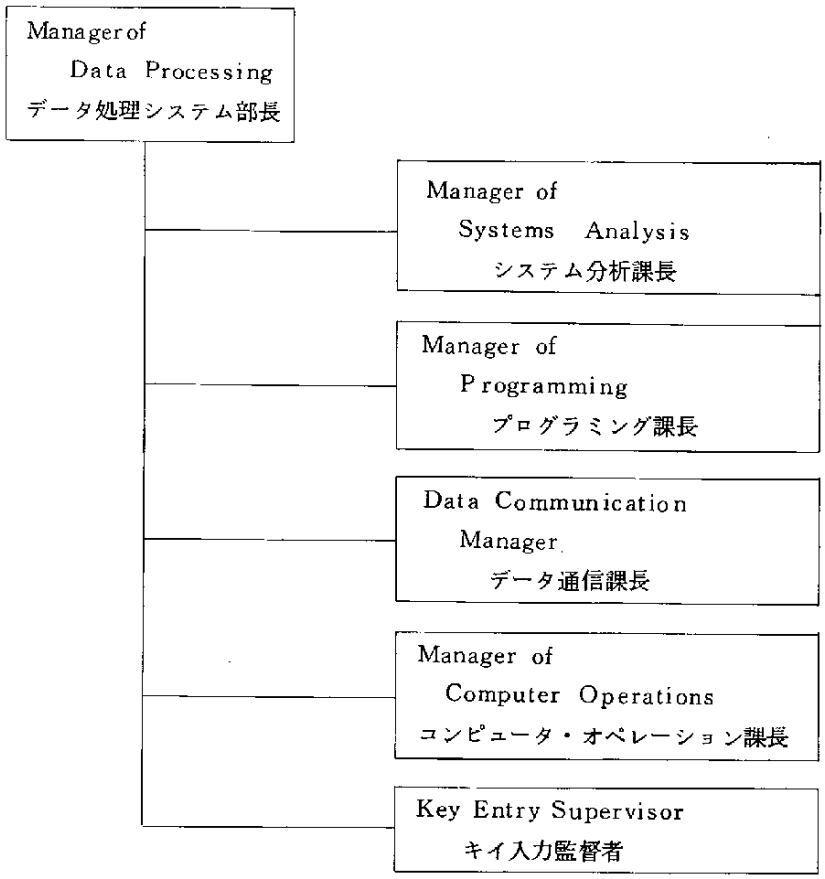


図2－2　EDP関係管理職の例

② 要員の分類

- Computer System Analyst
- Computer System Analyst and Programmer
- Programmer
- Data Communication Operator
- Computer Operator
- Computer I／O Controller
- Tape Librarian
- Key Entry Operator

図2－3　EDP要員の例

(2) 要員の業務遂行機能

要員を大きく

{ システム設計・開発分野の要員
{ オペレーション分野の要員

に分け，代表的な職種の機能をあげる。

(a) システム設計・開発分野の要員の機能

通産省が実施している「情報処理技術者試験」の案内書によれば，つぎのように定義しているので参考とする。

特種情報処理技術者……システムエンジニア

第1種情報処理技術者……上級プログラマ

第2種情報処理技術者……初中級プログラマ

一般に情報処理技術者は，コンピュータを利用した情報処理システムの開発，設計からその運用に至る幅広い分野で活躍する者の総称として用いられます。下図は，それらの分野と第１種，第２種情報処理技術者試験の対象者および特種情報処理技術者試験（システムエンジニア試験）の対象者の活動分野との関係を図示したものであります。

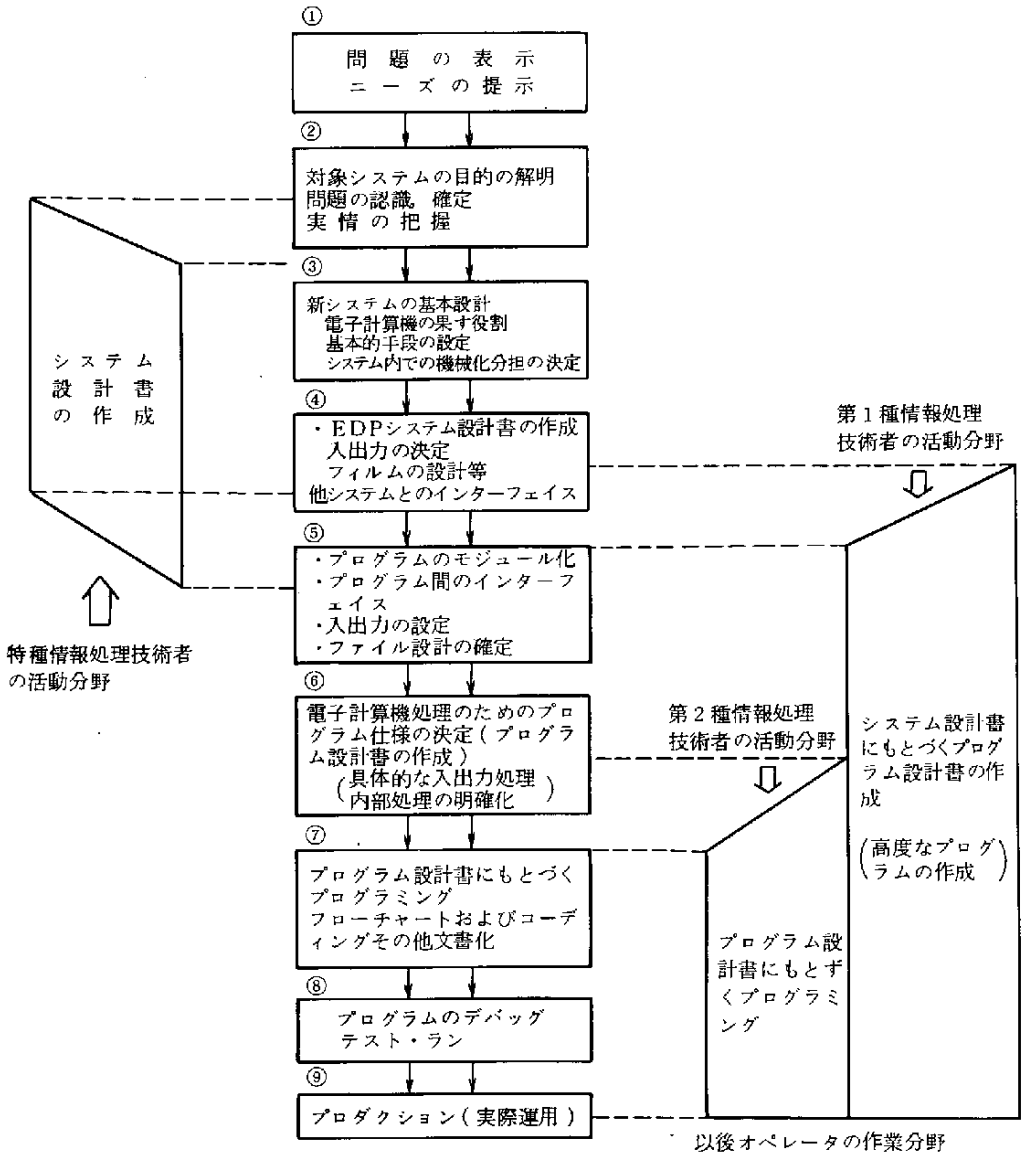


図２－４　情報処理技術者試験の対象者と活動分野（通商産業省）

(b) オペレーション分野の要員の機能

① コンピュータ・オペレータ

電子計算機要員のうち，最もコンピュータと密接して従事する職種であり，電算機をラン・ブック（オペレーション・マニュアル）にもとづいて操作する人達をいう。

・多重処理（オンライン，リモート・プロセス，バッチなど）

・オンライン処理（データ通信）

などの発達により，オペレーションも専門的要素を要求される。

② スケジューラ

コンピュータで運行するシステムが増大し，ユーザ部門に処理結果を配布するターン・アランド・タイムが厳しく要求される。

コンピュータの有効活用のためには，スケジューラによる厳格な運行予定表にもとづくオペレーションが必要となる。

スケジューラは，過去の運行実績にもとづいて，日々のコンピュータ・オペレーションスケジュールを決定し，オペレータに指示する機能をもつ。

③ 磁気ファイル・ライブラリアン

磁気テープ・ファイル，磁気ディスク・ファイルなど，外部記憶媒体の保管およびファイル庫からの入出庫を行う人達をいう。

コンピュータ利用の拡大は，磁気ファイルの量的増大につながる。

また，機密保持・安全管理の観点からも磁気ファイル管理の専任者であるライブラリアンを設ける必要が発生した。

④ キー・エントリ・オペレータ

別名キーパンチャとも呼ばれており，データ処理の入口を担当する職種である。

パンチ・カード作成のほか，データ・エントリー・マシンとして，

・キーツーテープ

・キーツーカセット

・キーツーディスク

・オンライン入力

など，パンチ・カードを媒体としない入力方式が多用されるようになり，入力作業が合理化されつつある。

この職種の特徴は，次の通りである。

① 女性中心の職場である。

② 作業は正確性とスピードが要求される。

③ 要員管理，コスト面から外注の利用が多い。（67.7％の企業が利用 …… コンピュータ白書 '76 ）

以上の要員の機能は，日本において職種としての細分化もはかられておらず，給与体系も年功序列型で，他の職員と明確に区分している例は少ない。

2.2 要員の育成

コンピュータを効率よく駆使し，情報システム化による成果を得るためには，コンピュータ部門の要員を計画的に育成することが必要となる。

ここでは，一般企業のコンピュータ部門における要員育成について学習する。

(1) 教育の形態

教育形態としては，企業内教育と企業外教育とに大別される。要員教育は，この両者を併用するのが大部分である。

(a) 企業内教育

企業内における情報処理教育は，その企業によるコンピュータ・リゼーションの度合と密接な関係をもち，さまざまな形態が存在する。

つぎのような発展形態が一般的に見受けられる。

コンピュータ担当部門（少数メンバ）
⇩
部課長などミドル・マネジメントへ
⇩
トップ・マネジメントへ
⇩
全社員を対象とした教育

また，つぎのような発展形態もある。

メーカーによる教育
⇩
コンピュータ部門内の計画による教育
⇩
全社レベルの一般教育

実際には，次のような問題点から，企業内教育は，十分行なわれていない傾向がある。

① 教育を担当するインストラクタが不足する傾向がある。

② 教材等の体制の整備が困難。

③ 慢性的な要員不足となるケースが多く，要員教育のために十分な時間をさけない。

これらの問題点を解決するためには，コンピュータ部門において，

① 教育担当専任者を設ける。

② 専任者は，公的機関（例えば情報処理研修センタ）で適切な研修を受ける。

③ 要員教育計画をもち，定期的に実施する。

④ 社内一般社員に対する知識向上のために，一般教育研修部門に対して情報処理のカリキュラムを組み入れるよう働きかける。

⑤ 一般社員に対する解説書（社内の業務適用例も折り込んだもの）を作成しＰＲする。

⑥ ミドル・マネジメント，トップ・マネジメントのための研修を企画する。

などの実践的な施策が必要である。

(b) 企業外教育

コンピュータ・メーカによる要員教育が企業外教育で最も多く採られている形態である。その他の教育として，

- 各種専門学校への派遣
- 情報処理研修センタへの派遣
- 各種講習会への参画

などがあり，これらを併用して知識向上につとめている。

コンピュータ・メーカによる教育は，どのメーカも力を入れており，カリキュラムやインストラクタが充実している。

教育コースは，およそ次のようになっている。

コース別 / 階層別	EDPS入門	プログラミング	経営科学	システムオペレーティング	システムデータ通信	システム設計
トップ・マネジメント	○		○			
ミドル・マネジメント	○	△	○		○	○
専門要員	○	○	○	○	○	○

コンピュータ・メーカの教育方法としては，次のような方法がある。

① 企業へインストラクタが出向いて教育するもの。

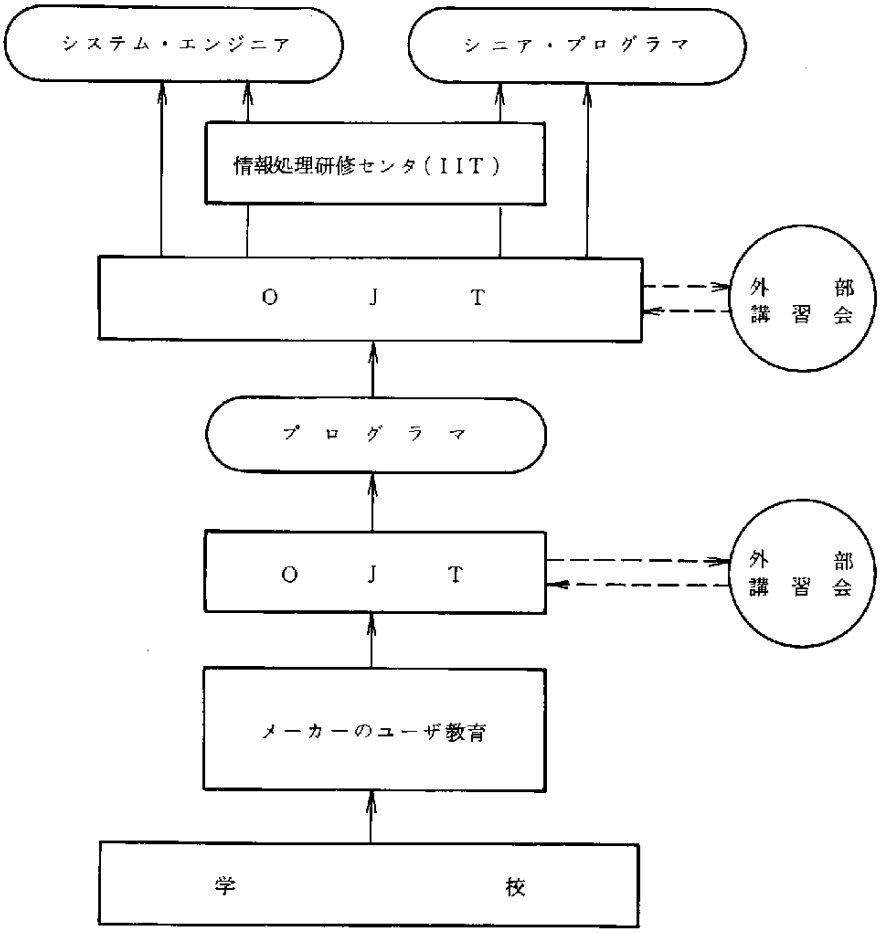
② メーカーの研修所において実施するもの。

③ 通信教育によるもの。

(2) 教 育 の 方 法

どのような要員教育の方法を採用するかは，各企業のコンピュータ部門により異なるが，要員のレベル・アップは，直接的にコンピュータ部門の生産性向上につながることもあり，さまざまな工夫が必要となる。

情報処理要員の養成は，つぎのように教育と実務訓練（ＯＪＴ）を併用して実施されるケースが多く見受けられる。



その他教育の一貫として実施されている方法を列挙する。

① 研究グループ活動

グループによりテーマを選定し，継続的に研究し，発表し合うことにより知識向上をはかる。

② 情報処理技術者試験の受験

企業が受験料の負担，受験のための勉強会の実施，合格者の優遇などを行ない，受験を積極的に支援することにより要員のレベルアップをはかる。

③ 技術動向等の伝達

雑誌・図書等の充実をはかることと併せて，外部講習会等への出席，部内における発表等を通じて，新しい技術動向の習得をはかる。

(3) 要員育成の環境づくり

要員育成のための環境づくりには，まず部門のマネージャレベルの人達に高い意識が必要なことはいうまでもない。

情報処理部門の要員育成の大部分は，仕事を通じて，いわゆるOJTによる育成であるといえる。

このことから，作業を実施する場合のチームの組み方と作業方式の標準化が重要である。

チームの組み方としては，システム・エンジニアを頂点として，プログラマに至るピラミッド型のチーム編成が望ましい。

その他，要員育成の環境づくりとして留意すべき事項を列挙する。

① 図書・雑誌などの充実

② 定期的な研修センタへの派遣

③ 海外視察等への派遣

④ 外部講習会への派遣

⑤ 部内での定期的講習会の実施

2.3 要員管理のポイント

(1) 要員の職業倫理（モラル）

コンピュータ部門は，効率化，機密保持の点から要員のモラル向上に配慮しなければならない。

要員に対しては，作業と教育を通じて情報処理に関する知識を取得させ，その重要性を認識させる。

情報処理作業に従事する要員に関する作業心得をいくつか挙げる。

① 協調性の保持による人間関係の円滑化

大部分の作業がチームによる作業であり，良いチームワークが必要である。

② 創造性の発揮と実行の勇気

システムづくりの多くは，すぐれた創造性の発揮により生まれる。また，新しいシステムづくりは，困難さが常につきまとうものであるだけに，実行にあたっては勇気ある行動が必要となる。

③ 健康の保持

コンピュータ部門の作業は，ルーチン・ワークが少なく，特に，システム開発要員は，勤務が不規則になりがちである。健康保持に十分留意しなければならない。

④ コスト意識の徹底

情報処理は，企業の合理化のために必要な手段であるだけに，システム開発，オペレーションなど，どの作業においても各人にコスト意識の徹底をはかる必要がある。

⑤ 機械の保全

コンピュータは高価なものであり，火気・電源に留意し，オペレーションの基本動作を守ることにより機械の保全をはかる。

⑥ 機密の厳守

情報処理の内容，出力された結果など企業にとって重要な情報がコンピュータ部門で取り扱われている。これらの情報の機密は厳守しなければならない。

(2) 各種要員の管理

作業を正しく遂行するための要員管理上の留意点をあげる。

① プログラマ管理

最近は，欧米においてプログラム犯罪が話題になっており，この面からの管理方法をあげてみる。

- プログラミングの分割
- プログラミングの担当変更
- リーダーによる管理・チェック
- 機種転換による変更
- 標準化の確立

日本においてほとんど意識されていないのが実情であろう。

② オペレーション要員管理

オペレーションを円滑に行うには，勤務内容と要員数がキーポイントであるが，稼動している諸システムのオペレーションが容易であることが前提であり，ミスが生じないシステムが最も望ましい。

オペレーション・ミスを少なくするためには，つぎのような方策を確立する。

・エラー・チェック方式の確立

ロジカル・コード・チェック，マスタ・ファイルによるチェック，合計によるチェック，日付によるチェックなど。

・作業表の準備とチェック

・作業完了時のチェック

・オペレーション体制の確立

スーパーバイザ，勤務時間，無理なオペレーションの排除など。

③ キーパンチャ管理

データ作成は，情報処理の入力準備作業として重要性をもっているが，次のような点で内部管理上の問題が多い。

・けんしょう炎などの職業病がある。

・単調作業が連続する。

・作業量が一定でない。

・若い女性がきらう……華かさがない

・腕の良し悪しが明確にわかる。

・退職が早く補充が困難。

これらの困難さをさける方法としては，採算は悪いが要員数を多くし，作業に余裕をもつことが唯一の解決策となる。

2.4 要員のローテーション

企業内の各セクションとの適度な人事交流は，コンピュータ部門とその他の部門とのコミュニケーションをより円滑化するとともに，情報処理に対する全社的な理解度を増すことによる効果も期待できる。

一方，コンピュータ部門は，とかく閉鎖的な部門となり，ユーザの真のニーズ把握が困難となることが考えられる。

部門内における要員ローテーションは，要員のモラル・アップ，生産性の向上から必要であり，外部部門および部門内部のローテーションの両方について，要員管理の円滑化の面から配慮していかなければならない。

ここでは，要員ローテーションの具体策を学習する。

(1) 外部部門とのローテーション

外部部門との人事交流は，通常人事異動の時期に行なわれることが多い。そのほか，研修的な意味でのコンピュータ部門と他の部門との交流も意味する。

外部とのローテーションとして行なわれている方法および人事的な面からの施策をいくつかあげる。

① 経営との人事政策の約束

定期的な人事異動において，ローテーションが計画的かつ継続的に行なわれるよう，コンピュータ部門から経営に対し要請し，経営との約束を得ておく。

・入社採用時の配慮

・新入社員の配属方法

例えば，他部門への配属をまず行ないコンピュータ部門にとっては，他部門から研修に派遣されて来た要員の中で，適性を見分け，定期的な人事異動により，コンピュータ部門の要員となるケースが見受けられる。

また，ユーザ部門へ派遣されたコンピュータ部門の要員は，情報処理の社内への普及と，ユーザ部門の業務知識吸収という，2つの局面をもっており，有効な手段である。

③ 企業内の情報処理普及の度合

企業における情報処理形態が，集約的に行われているか，分散的に行われているかにより，ローテーションの方式も相違する。

ここでいう分散的という意味は，製造工業等に多く見られるようなオープン・プログラマ形式による情報処理形態である。

分散処理形式では，必ずしもコンピュータ部門との人事交流によるユーザ部門と連携動作よりも，全社的な情報処理教育体制の確立が指向される。

(2) コンピュータ部門内のローテーション

コンピュータ部門内は，大きく，システム開発とコンピュータ運用（オペレーション）に分けられる。

システム開発においては，稼動システムの保守（システム・メインテナンス）と新規のシステム開発に区分される。

コンピュータ運用は，コンピュータのオペレーション，入出力データの管理，機器の保全データの保全など，さまざまな機能に分かれている。

大規模なコンピュータ部門においては，機能別に組織化されており，要員の種類も多く，内部における要員の計画的なローテーションが必要となる。

より良いシステムを建設するためには，システム開発から運用にわたる全般的な経験と知識が必要であり，この面からも内部のローテーションは，意識的かつ計画的に実施される。

内部ローテーションの必要事項のいくつかをあげる。

① 職種の区分を明確化する

開発・運用を問わず，業務範囲を明確にし組織化する。このことは，より専門的な業務の遂行による生産性向上と効率化を意味しており，要員の適性とローテーション・プランがたてやすい。

② 内部のプロジェクト・チーム化

その時々の経営のニーズを効率化よく消化すること，および要員のレベル・アップのために，プロジェクト・チームを弾力的に編成する方法がある。

この場合は，チーム・リーダーの能力にチームの効率が左右されること，および作業基準の明確化が前提となる。

③ 内部管理セクションの設定とローテーション

内部の開発スケジュール管理，予算管理，コンピュータ運行管理，文書管理などの機能を管理セクションとして設け，そのセクションと他のセクションとのローテーションを計画的に行うことにより，広範な知識の向上がはかられる。

④ 開発と運用のローテーション

開発と運用は，非常に密接な関係があるが，各々のセクションが独立していることが内部統制上良い結果をもたらす。

それだけに，要員の硬直化が生じやすく，意識的にローテーションをはかる必要がある。

最近の傾向として，開発の中のプログラムの外注。運用におけるデータ・エントリー，コンピュータ・オペレーションの外注などにより，従来，専門的であり人事交流の困難さを伴っていたセクションとの交流範囲が除外されつつある。

このことは，専門的な内部職種の外部情報処理専門会社の利用が，要員管理の困難性排除に役立つことを意味しており，情報処理コスト・パフォーマンスの面からも検討されるべきテーマである。

指導上の留意点

(1) コンピュータ部門の機能細分化が要員の種類増加となっていることから，機能と要員との関連づけの観点から指導していく。

(2) 要員の育成・管理に関する問題点は，各企業にとって多く存在している。問題を挙げ，その解決策はどうあるべきかを指導していく。

(3) 要員のローテーションは，部門内，他部門との間とを問わず，各企業によって，種々の工夫がなされている。実際のケースを調査し指導していくことが望ましい。

(4) 要員教育方法としての課題訓練は，単元Ｂ２第６章を復習してみる。

参　考

Functional Cost Accounting For D. P. Centers, by Paul Anthony, Management Accounting, Oct. '76

表2・3　米国のデータ処理センタの要員例

職種	給与（1カ月）	遂行機能
部長	$ 3,330	管理 — 間接費(注), プロジェクト計画／統制, システム調査, リポート作成, システム設計
次長	1,166	管理／一般管理
システム・アナリスト	2,166	システム調査, システム設計, システム仕様, プログラム・テスト, システム・テスト, システム変換／実施
システム・アナリスト補佐	1,330	フォーム設計, ドキュメンテーション, システム調査, システム仕様
データ課長	1,800	ファイル検査, ファイル訂正, リポート作成, 監督
データ課長補佐	830	ファイル検査, ファイル訂正, リポート作成
データ課長補助員	666	ファイル訂正
システム課長	1,500	補助プログラム, システム仕様, システム設計, システム・テスト, システム変換／実施
システム・プログラマー	1,500	プログラム・コーディング, プログラム・テスト, システム・テスト・ジョブ・ラン
プログラマー	1,666	プログラム・コーディング, プログラム・テスト, システム・テスト, ファイル訂正
プロダクション課長	1,666	ハードウエア据付け, スケジューリング, 監督
プロダクション課長補佐	1,190	ハードウエア設置, プロジェクト計画／統制, スケジューリング
補助プログラム・プログラマー	1,000	補助プログラム
プロジェクト課長	1,330	プロジェクト計画／統制
キイパンチ係長	1,000	キイパンチ, 監督
キイパンチ・オペレーター	916	キイパンチ
サービス調整係長	1,166	リクエスト準備, ジョブ準備, 監督
サービス調整補助員	666	ジョブ準備
コンピューター・オペレーター	830	ジョブ・ラン
コンピューター・オペレーター補助員	666	ジョブ・ラン, 照合／バースト
上級EAMオペレーター	1,330	ジョブ・ラン, 照合／バースト
速記者	836	ドキュメンテーション, システム仕様, 一般管理
タイピスト	750	ドキュメンテーション, システム仕様, 一般管理

○　これらの要員は，日本の企業においてどのような構成になっているか，コンピュータ白書1976年版の資料を載せた。

この表から，業種による要員数または要員の割合がかなり相違することが分る。

表２・４　コンピュータ利用状況およびオンライン化調査集計表

業種別・１社当り社内要員数平均

業種別＼職種		庶務その他			パンチャー			オペレータ			プログラマ			SE			管理職			合計人数			回答社数
		男	女	計	男	女	計	男	女	計	男	女	計	男	女	計	男	女	計	男	女	計	
一次産業計		.0	2.8	2.8	.0	5.3	5.3	.6	.8	1.5	4.3	.0	4.3	2.1	.1	2.3	3.1	.0	3.1	10.3	9.1	19.5	6
二次産業計		.7	2.1	2.9	.0	6.6	6.7	2.8	1.0	3.8	6.6	1.2	7.9	5.1	.1	5.2	2.2	.0	2.2	17.7	11.2	28.0	618
三次産業計		3.1	5.2	8.3	.0	7.9	8.0	5.3	1.2	6.5	8.6	1.6	10.2	6.1	.1	6.2	3.9	.0	4.0	27.2	16.1	43.4	645
公務計		1.4	.9	2.4	.1	8.6	8.8	2.2	.4	2.6	6.4	.7	7.2	3.5	.1	3.6	1.6	.4	2.0	15.4	11.4	26.8	128
全産業計		1.9	3.4	5.4	.0	7.4	7.5	3.9	1.0	4.9	7.5	1.3	8.9	5.4	.1	5.5	2.9	.0	3.0	21.9	13.5	35.4	1397
主な業種	繊維工業	1.0	2.6	3.7	.0	6.2	6.2	1.4	1.8	3.3	5.8	1.0	6.8	3.0	.0	3.0	1.5	.0	1.5	12.8	11.9	24.7	39
	化学工業	.3	2.3	2.7	.0	5.0	5.0	1.9	.9	2.8	5.1	1.5	6.7	5.2	.0	5.2	1.9	.0	1.9	14.7	9.8	24.6	78
	石油製品製造業	.6	4.8	5.5	.0	9.8	9.8	5.3	1.5	6.8	9.4	1.5	10.9	7.4	.0	7.4	2.8	.0	2.8	25.6	17.5	43.1	12
	鉄鋼業	2.4	6.2	8.7	.5	8.3	8.8	15.0	1.4	16.4	17.6	3.5	21.1	15.5	.3	15.9	4.9	.0	4.9	56.2	19.9	76.1	27
	電気機械器具製造業	2.1	3.3	5.5	.0	9.4	9.4	2.8	.6	3.5	12.0	1.5	13.5	9.8	.2	10.1	4.5	.0	4.5	31.4	15.2	46.6	66
	輸送用機械器具製造業	1.3	2.4	3.7	.3	13.3	13.6	4.4	1.0	5.5	8.8	1.2	10.1	6.0	.3	6.3	2.5	.9	2.7	23.8	18.4	42.2	57
	卸業・商社	.8	2.5	3.4	.0	5.1	5.1	1.3	.7	2.0	3.4	.9	4.4	2.8	.0	2.8	1.6	.0	1.6	10.1	9.4	19.5	107
	小売業	.3	1.5	1.8	.0	8.1	8.1	1.5	.6	2.2	3.4	.3	3.8	1.6	.0	1.7	1.4	.0	1.4	8.5	10.8	57.3	33
	金融業	3.2	8.3	11.6	.1	8.1	8.2	6.8	2.2	11.1	11.8	2.5	14.4	6.0	.0	6.1	5.4	.0	5.4	35.5	21.4	171.0	133
	生命保険業（含代理業・サービス業）	6.5	29.8	36.3	.0	43.8	43.8	5.0	3.1	8.1	28.0	7.3	35.3	32.8	.0	32.8	15.0	.0	15.0	87.3	84.1	133.5	6
	損害保険業（含代理業・サービス業）	10.8	68.1	79.0	.0	1.2	1.2	.1	.2	.4	12.1	7.1	19.2	19.4	.2	19.7	13.4	.4	13.8	56.0	77.5	85.5	7
	電力・ガス事業	4.9	2.8	7.7	.0	12.7	12.7	13.9	1.5	15.4	20.6	2.0	22.7	14.8	.0	14.8	12.3	.0	12.3	66.6	19.1	23.8	12
	広告・調査・情報提供サービス業	1.3	1.1	2.5	.0	7.3	7.3	1.2	.5	1.7	5.2	1.3	6.6	2.5	.0	2.5	2.5	.1	2.6	12.8	10.5	102.3	8
	情報処理サービス業・ソフトウエア業	8.1	8.7	16.8	.2	22.3	22.5	14.2	1.7	16.0	22.7	3.5	26.3	12.0	.3	12.4	7.8	.1	7.9	65.2	36.8	.1	79

表２・５　業種別・全従業員規模別・社内要員数比平均

社内要員数平均 ＼ 全従業員規模		無記入	五〇人未満	五〇～九九人	一〇〇～二九九人	二〇〇～三九九人	四〇〇～四九九人	五〇〇～九九九人	一、〇〇〇～二、九九九人	三、〇〇〇～四、九九九人	五、〇〇〇～九、九九九人	一〇、〇〇〇人以上	合計
一次産業計	全従業員数	.6	.0	.0	6.0	.0	.0	6.0	9.0	37.0	29.5	.0	19.5
	比平均(%)	.00	.00	.00	2.52	.00	.00	.94	.64	1.11	.43	.00	.60
二次産業計	全従業員数	61.8	3.5	6.3	6.3	8.5	7.9	10.6	19.7	38.1	65.8	234.8	29.0
	比平均(%)	.00	11.29	8.75	3.45	2.54	1.79	1.50	1.17	1.01	.92	1.17	1.16
三次産業計	全従業員数	64.3	14.1	22.0	28.6	25.6	19.5	47.9	41.9	58.2	31.4	279.9	43.4
	比平均(%)	.00	61.77	31.89	15.58	7.39	4.44	6.80	2.40	1.57	1.98	.59	1.85
公務計	全従業員数	12.6	15.2	22.4	17.6	6.8	14.6	23.1	28.6	21.8	38.2	45.5	25.5
	比平均(%)	.00	74.82	27.83	9.21	1.96	3.31	3.13	1.59	.59	.54	.16	.70
全産業計	全従業員数	139.3	14.0	19.6	22.0	17.4	13.4	25.6	72.4	42.5	77.0	217.1	35.3
	比平均(%)	.00	61.10	27.83	11.97	5.09	3.05	3.61	1.60	1.14	1.10	.71	1.39
主な業種 繊維工業	全従業員数	3.9	4.0	2.0	5.6	.0	6.0	7.3	24.0	25.0	23.0	121.2	24.7
	比平均(%)	.00	18.18	2.22	2.67	.00	1.41	1.16	1.36	.65	.31	.84	.85
化学工業	全従業員数	7.8	.0	3.0	14.1	5.6	6.6	11.3	21.0	35.3	76.0	63.5	24.6
	比平均(%)	.00	.00	3.61	9.64	1.78	1.53	1.58	1.21	.94	.97	.50	.98
石油製品製造業	全従業員数	1.2	.0	.0	7.0	.0	3.0	9.0	70.0	64.3	41.0	.0	43.3
	比平均(%)	.00	.00	.00	5.07	.00	.71	1.62	2.92	1.59	1.71	.00	1.86
鉄鋼業	全従業員数	2.7	.0	22.0	8.3	11.0	5.0	11.5	20.2	29.6	16.0	750.0	76.1
	比平均(%)	.00	.00	35.48	4.60	3.03	1.02	1.68	1.24	.84	1.33	2.19	1.81
電気機械器具製造業	全従業員数	6.9	.0	3.0	3.6	.0	6.7	8.6	18.3	34.0	77.7	438.5	46.6
	比平均(%)	.00	.00	3.75	2.29	.00	1.52	1.25	1.04	.98	1.05	1.47	1.25
輸送用機械器具製造業	全従業員数	5.7	.0	3.0	6.0	9.5	10.0	9.8	19.5	27.2	85.0	387.6	42.2
	比平均(%)	.00	.00	3.75	2.67	3.01	2.22	1.36	1.14	.78	1.14	1.47	1.11
卸業・商社	全従業員数	10.6	2.6	4.0	7.9	13.0	17.6	25.2	31.9	56.5	120.0	135.0	19.6
	比平均(%)	.00	7.96	4.46	4.23	3.66	4.02	3.66	2.03	1.39	1.50	1.35	2.32
小売業	全従業員数	3.3	.0	4.0	5.7	9.7	28.0	18.5	28.8	34.5	.0	.0	19.3
	比平均(%)	.00	.00	4.46	3.42	2.93	5.82	2.81	1.86	1.30	.00	.00	2.15
金融業	全従業員数	13.3	14.6	81.0	10.1	20.9	17.7	32.5	63.1	123.8	185.6	344.2	57.0
	比平均(%)	.00	48.88	100.00	5.10	5.77	4.01	4.19	3.40	3.47	2.88	2.25	3.16
生命保険業（含代理業・サービス業）	全従業員数	.6	.0	.0	.0	.0	.0	.0	88.5	.0	.0	213.0	171.5
	比平均(%)	.00	.00	.00	.00	.00	.00	.00	3.96	.00	.00	.43	.51
損害保険業（含代理業・サービス業）	全従業員数	.7	.0	.0	21.0	16.0	.0	.0	103.6	.0	293.5	.0	133.5
	比平均(%)	.00	.00	.00	9.01	5.33	.00	.00	4.82	.00	4.85	.00	4.89
電力・ガス事業	全従業員数	1.2	.0	.0	.0	.0	10.0	13.0	27.0	68.0	64.0	135.8	85.8
	比平均(%)	.00	.00	.00	.00	.00	2.22	2.06	3.13	2.24	1.02	.70	.86
広告・調査・情報提供サービス業	全従業員数	.8	.0	.0	12.6	14.5	.0	.0	10.0	.0	100.0	.0	23.3
	比平均(%)	.00	.00	.00	8.92	4.20	.00	.00	.60	.00	1.89	.00	1.92
情報処理サービス業・ソフトウエア業	全従業員数	7.9	22.8	50.5	142.4	170.3	.0	535.2	16.0	.0	.0	.0	101.5
	比平均(%)	.00	84.65	83.39	82.27	50.79	.00	75.70	1.27	.00	.00	.00	69.08

参考文献

〔1〕南条　優，保坂岩男共著，「システム開発と運用」　オーム社，1972.

〔2〕日本情報処理開発協会，「コンピュータ白書1977」コンピュータ・エイジ社，1977.

〔3〕服部幸英，「コンピュータ室の運用管理」　情報処理研修センタ，1977.

〔4〕S.Kuehne, R.S., Lindberg, H.W., and Baron, W.F., "Manual of Computer Documentation Standards with Forms"　Prentice - Hall, 1972.

〔5〕長谷英夫，朝井幸洋共著，「情報処理センターその運営と効果的利用法」　ダイヤモンド社，1974.

〔6〕中江剛毅，「日本の情報システム管理」　泉文堂，1975.

〔7〕額田　巌，「あなたもプログラマになれるーソフトウェア教育の手引」　日本経済新聞社，1971.

〔8〕中辻卯一，「経営管理とコンピュータ」　中央経済社，1971.

〔9〕松本太郎，大芝　猛共著，「経営情報システム」　東洋経済新報社，1968.

〔10〕レイトン・F・スミス，「経営幹部のためのコンピュータへの姿勢」　日本データ・プロセシング協会，1973.

〔11〕斉藤雅彦，「情報と組織」新管理者選書6，日本経営出版会，1976.

〔12〕経営科学研究所，「特集：コンピュータ要員の管理と教育」　Computer Report，9月号，1973.

〔13〕社団法人日本能率協会出版事業部，「ＥＤＰスペシャル・リポート」1977

〔14〕南条　優，「コンピュータ室の運用と管理」　日本生産性本部，1971.

〔15〕情報処理研修センタ，「情報処理要員確保に関する諸問題」　1973.

〔16〕江川　朗，「プロジェクト・チーム」　日本能率協会，1970.

〔17〕J.S.パウムカートナー，ビーコン・システム（訳）　「プロジェクト・マネジメント」技報堂，1971.

〔18〕能率大学（編）「管理者」　経林書房，1967.

〔19〕清水　勤，「ＯＪＴと組織開発」　総合労働研究所，1972.

第3章　システム開発管理

キーワード

企業システム部門(corporate systems division)，長期計画(long-range planning)，システム開発部(systems development)，データベース管理(data base administration)，システム設計(system design)，ファシリティ・マネジメント(facility management)，プロジェクト・マネジメント(project management)，システム・メンテナンス(system maintenance)，ドキュメンテーション(documentation)，システム・ライフサイクル(systems life cycle)，標準化(standards)，構造化プログラミング(structured programming)，IPT(improved programming technique)，HIPO(hierchy plus input-process-output)，トップダウン設計(top down design)

目　　標

コンピュータ部門の企業内における重要性が高まるにつれ，システム開発についての管理はその厳格さが要請される。

システムは，単に，要請されるがままに迅速に開発するだけでなく，メンテナンスの容易性運用の容易性をも十分に考慮されたものでなければならない。

最近の開発および運用環境の変化および新しい管理技法の発生した背景，その方法などを充分理解した上で，コンピュータ部門の開発管理に適用していかなければならない。

開発管理については単元D1第2章でも学習したが，ここでは，実例を極力折り込みながら実際的な知識を修得することを目標とする。

内　　容

コンピュータ部門の組織的な変遷と実体をよく理解し，その上での開発計画のあり方，部門運営上のポイントを把握する。

また，最近の傾向として，システムの標準化とそれに伴う文書化，文書管理の重要度が認識されてきている。

ソフトウエアの開発技法，ドキュメント技法の数々が提唱されてきており，これらを参考にしながら，その企業のシステム部門にフィットした方式を積極的に導入し，効率の向上をはかっている例も多い。

この章では，次のような項目について学習し，開発管理をする上での予備知識あるいは留意すべき事項とする。

① コンピュータ部門の組織的変化

② システム要員の職務分化

③ システム開発計画

④ 開発とメンテナンス

⑤ 大型プロジェクトの形成

⑥ 標準化とドキュメンテーション

⑦ 新しいシステム開発技法

3.1 コンピュタ部門の組織的変化

第1章を復習する意味から組織について再度とりあげる。

コンピュータによって処理される業務が多種多様になり，また，その内容もより経営の意思決定をサポートする資料の作成が多くなりつつある。したがって，コンピュータ部門の企業内における位置づけも変化をとげつつある。特にこの傾向は，コンピュータ利用に一日の長がある米国の大手ユーザに見られる。

米国に限らず日本においても，大手ユーザもしくは，先進的ユーザのコンピュータ部門は，企業内において「企業システム部門」（corporate systems division）ともいうべき位置づけをされることが多くなっている。即ち，その部門の機能として，単なるコンピュータのプログラム開発やメンテナンスおよびコンピュータ・オペレーションを担当するだけではなく企業内のシステム化ニーズの長期的視野に立っての管理・企画・調整を行うため，次のような広範な機能を包含しようとしている。

① 業務の処理方式

事務作業，生産作業のやり方，手順等の規定，管理。

② システム開発計画の企画・立案長期および短期の計画

③ システム化（EDP）委員会の開催

④ コンピュータ・システムの開発

⑤ コンピュータ・システムのメンテナンス

⑥ コンピュータ設備の運用管理

⑦ コンピュータ・センタ付帯設備の管理

⑧ 通信関係設備の管理

通信回線，ファクシミリ，テレックス，電話，国際通信関係など。

⑨ システム関係発送物，郵送物の受発信

⑩ 標準化，ドキュメント管理

⑪ 事務管理，事務指導

⑫ システム関係コストの統括

導入コスト，運用コスト

このような統合化された組織をもつ企業は，まだ少ないが，除々にこのような形態へ移行が行われるであろう。

その理由として，つぎのようなことがあげられる。

① 統合化による，オーバーヘッドを少なくし，コストを削減する。

② 意思統一と伝達のスピードアップにより，変化に迅速に対処する。

③ 経営と直結し，一体化した行動をとっていきやすい。

また，それは次のような傾向にもうかがい知ることができる。即ち，組織名称の変遷である。コンピュータ部門の名称は，昭和30年代初頭から最近までに相当変化してきている。

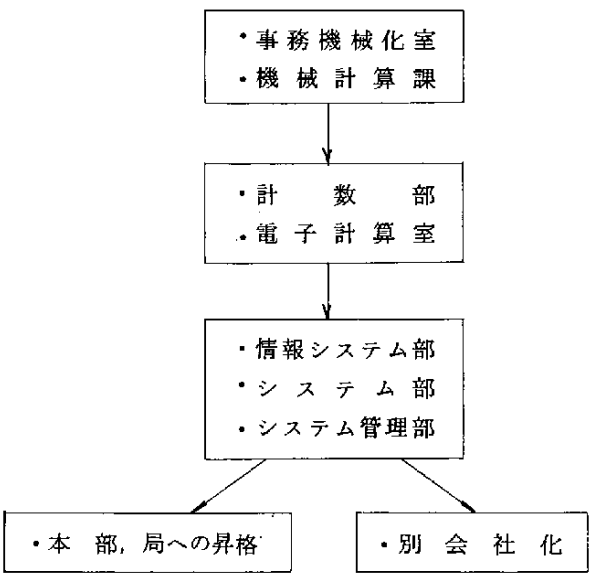


図３・１ 情報処理部門名称の変遷

組織上の変化は，コンピュータの導入年次の早い企業ほど，また，使用コンピュータの規模が大きいほど，強く表われている傾向が見受けられる。

また，システム部門を企業内のコスト部門として位置づけ，そのコストを経費と考える場合と，プロフィット部門としてとらえ，独立採算を目指す場合とにより変化の度合は異なってくる。

採算性の追求によって，独立別会社に分離した方が企業もシステム部門もメリットが大きい場合もあるからである。

組織の変遷は，当然のことながら，業務内容の推移に要因を求めることができる。業種およびコンピュータの導入年次により多少の差異はあるが，概して次のようなステップを踏んで拡大している。

① ステップー1

人事・給与・財務・経理・生産・販売などのうち，一つの業務だけを機械化処理。

② ステップー2

これら全部を日常定型業務として処理。

③ ステップー3

さらに予測など企画調査関係の質的に異った業務を加える。

④ ステップー4

各種業務をシステム化する。順次オンライン化へ進む。

⑤ ステップー5

別々に作られたシステムを統合化し，トータル・システムにする。

下図は，米国における大規模なコンピュータ・ユーザのシステム部門（年間総コスト100億円以上）の組織図である。統合化されたEDP部門の一例として参照されたい。

組　　織　　図

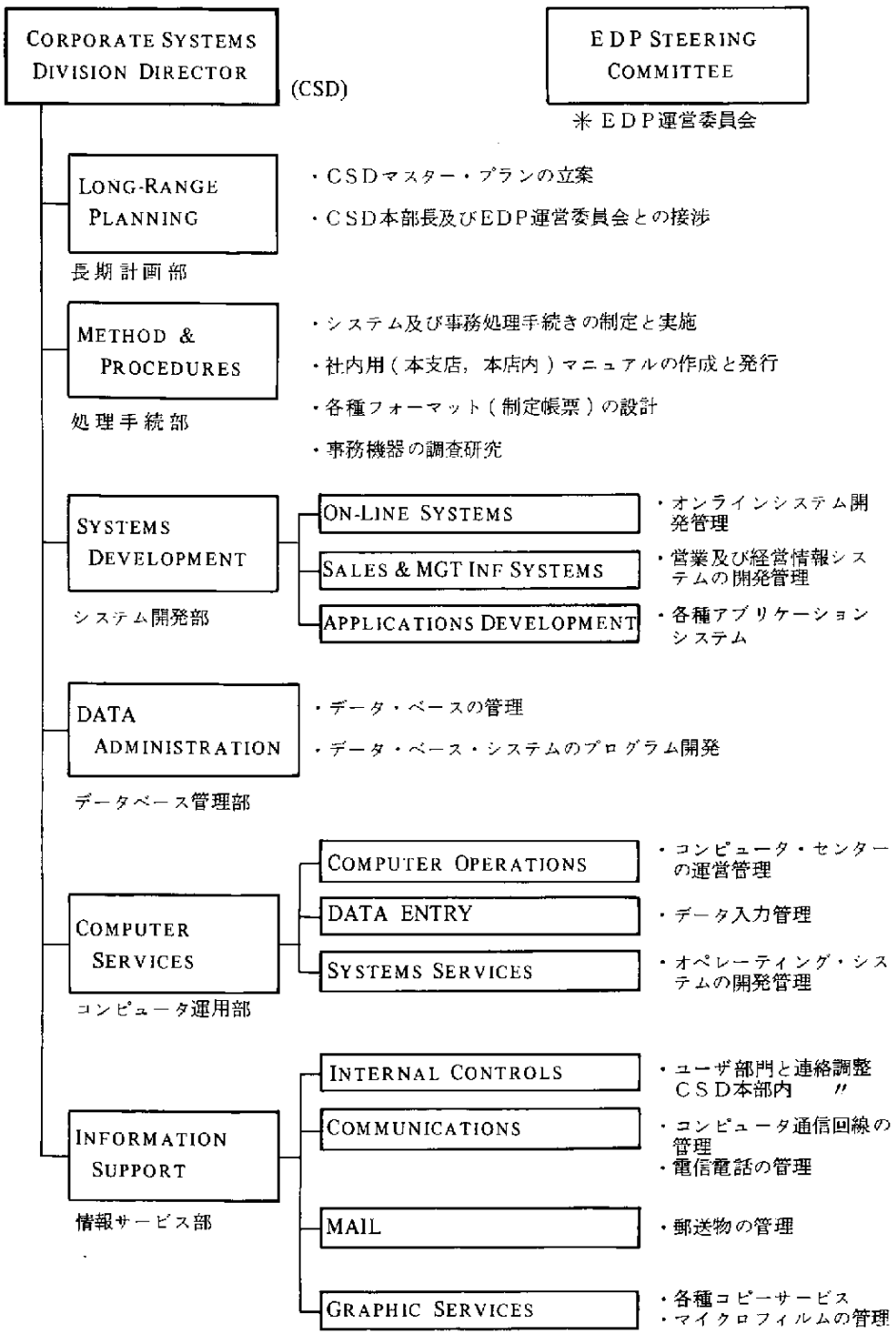


図３・２　米国における大規模コンピュータ・ユーザのシステム部門

3.2　システム要員の職務分化

組織の変化に伴って，当然システム部門要員の職務上の専門化も進行しつつある。一般には次のような形で職務分担が専門化してきているといってよい。

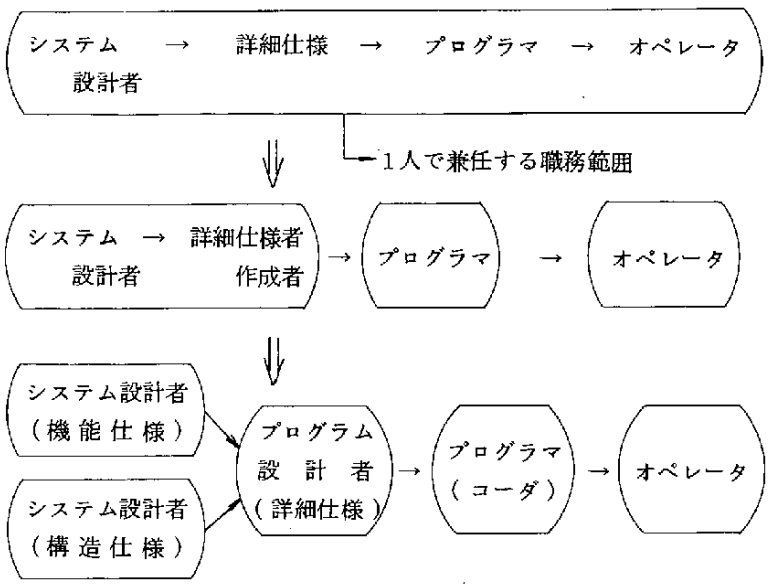


図３・３　システム要員の機能分化

さらに同じシステム設計者の中にも事務処理と科学技術系では別々のスキルが要求されるため，最近では，職務がより細分化，専門化される傾向が強い。職務分担の一例をあげると次のようなものがある。

① システム企画調査

② システム設計（事務処理）

③ システム設計（科学技術系）

④ プログラム設計

詳細仕様もしくはプログラム仕様設計

⑤ プログラム開発

コーダ，テスト要員など

⑥ コンピュータ運用管理

スケジュール作成，オペレータ管理，機器設備管理など

⑦ オペレーション

⑧ 入力データ作成

⑨ ファイル管理

⑩ オペレーションズ・リサーチ

⑪ 基本ソフトウエア管理

オペレーティング・システム，システム・ジェネレーション，ライブラリーなど

⑫ データ・ベース管理（DBA）

データ・ベース（ファイル）内のデータ項目の規定，管理，メンテナンスなど

これらの職務の遂行に当って，⑤，⑦，⑧などはメーカの要員やソフトウエア会社へ外注することも多いため，専門化，分化はより一層進化することになる。さらに，上記職務の大半を一括して外注する（ファシリティ・マネジメント）こともある。

ファシリティ・マネジメント・サービスの利用状況について，昭和52年度のユーザ団体連合会の調査によると表3.1のような結果となっているので参考とされたい。

職務分担が細分化すればするほど，システム部門全体の動きを把握し，各職務間の相互連絡手段を充分に設定しなければならない。

最近は，一定の要員で一定期間のうちでシステムを建設する。いわゆるタスク・フォース型のシステム開発組織体制をとる場合も多く，「プロジェクト・マネジメント」が重要な課題である。

タスク・フォースにせよ，定型的な組織にせよ内部のコミュニケーションが不十分な場合，次のような問題が発生する。

- ドキュメントの不備
- システム構造の不統一
- 責任体制のあいまいさ
- 管理不足によるスケジュールの遅れ
- システム・メンテナンスの困難さ
- システム・トラブルの多発
- システム稼動効率の低下

表３・１　ファシリティ・サービス利用状況

・パンチ業務	自社のみ	23％
	全面委託	26
	部分委託	47
・オペレーション業務	自社のみ	42
	委託	20
	要員派遣	38
・プログラミング業務	自社のみ	70
	自社半数以下	4
・システム設計業務	自社のみ	87
	自社半数以下	1

今後は，プログラミングおよびシステム設計分野の外部委託が増加していくものと予測される。また，委託している各社の動機については，次のようなものがあげられる。

表３・２　外部委託の要因

- 要員確保の困難さの解消
- 一時的業務負担の解消
- 労務管理問題の解消
- 経費節減
- 他部門との要員ローテーション容易化
- オペレーションの標準化

委託に際しての問題点については，次のものがあげられるが，委託するかどうかは，多分に経営的判断にかかっている。いづれにせよ，専門化の傾向およびコスト対効果の点から外部委託は，増大しつつある。

・機密保護上の難点

・運用・管理上の問題

・ノウハウの蓄積が減少する

・システム・メンテナンス上の問題

3.3　システム開発計画

通常，システムの開発計画は，ユーザ部門からのシステム開発要望もしくは，システム部門内からの要望にもとづいて，システム部内で作成される。

一般に，開発要望を満足させるために必要なファシリティ － 主として開発人員 －は，シス

テム部門のもつファシリティを越えることが多い。なぜならば，コンピュータのキャパシティは，余力をもたせており，それを上廻るシステム開発人員をシステム部門は保有していないことが多く，また，最近の傾向として，ユーザ部門からの依頼内容も稼動中の現行システムに対するメンテナンス要求の方が新規システム開発要求を上廻ることが多い。

そこで，システム開発計画の作成にあたっては，システム部門に専任の担当グループ － 企画調整機能 － を設置することが望ましい。

そのグループは，依頼された内容を充分検討し，条件の不備な点を補足し，次のような観点に照らして，開発を行うか当面保留するか，または依頼を断るかを決めなければならない。

① 開発依頼の緊急性，重要性

例えば，法律や制度の改正によって発生する開発依頼や経営政策の変更によるものは，緊急性，重要性が高いといえる。

② 経営上の長期システム化計画との関係

2～3年内に新しい体系のもとにシステムの再構築を行うことが決定されているような場合，依頼されたシステムが再度再開発の対象となるようなものは，その時期まで依頼内容を延期できないかどうかの調整が必要である。

③ 現行システムのライフ・サイクル

開発されてから7～8年経過するとその間のシステム・メンテナンスや環境条件の変化により，システムの考朽化，すなわち，システム・ライフの到来が目立つようになる。

そのようなシステムの再開発依頼は優先して採り上げる必要があろう。

また，最近のコンピュータの傾向として，ファミリ・シリーズ化が進んでおり，古いプログラムが相当の長期間使用されるようになり，システム部門の計画としても「計画的な再開発」が必要となってきている。

④ システム部門の開発能力との関係

一般に，充分すぎるほどの要員を保有しているシステム部門は稀であるから，限られた要員，特に，システム設計能力のある要員をどのように配置するかで開発計画の遂行が大きく左右される。

⑤ コンピュータ，端末，回線等の設備の使用状況と使用限界

設備の使用状況によっても開発計画は左右される。特に，コンピュータ容量の面から可用限界に近づいているような場合，より緊急性のある重要度の高いシステム要求が優先される。

以上のような条件を充分考慮してシステム開発計画を策定するわけであるが，計画自体は次のような区分ができる。

① 短 期 計 画

3～6カ月程度で終了する計画で，システム部門の正規のメンテナンスおよび開発要員の割り振りによって遂行できるようなもの。

② 中 期 計 画

実施することを前提に，システム部門の要員はもちろん，外部の要員をも含むプロジェクト組織によって対処する大規模な新規システム開発または再構築。

③ 長 期 計 画

実施するか否かにかかわらず，企業全体の立場からみて，長期的なシステム体系のあり方開発すべき新システムの指摘，再構築すべきシステムに関する計画。

これらのうち，短期計画においては，次のような内容を含む開発計画書を作成する必要がある。

- 開発するシステムの内容
- システムの目的と想定されるメリット
- 開発に必要な期間と作業工数
- 開発担当者
- 開発に要するコンピュータ・コスト，消耗品コスト等を含む費用見積り
- 作業スケジュール

また，中長期計画においては，概略次のような内容を開発計画書にもり込む。

- 経営からまたは，内部からのニーズ
- システム開発の目標あるいは目的
- 期待される効果，目標値
- システムの特徴
- システムの概要および内容
- システム構成 － ハードウエアおよびソフトウエア
- 開発スケジュールおよび開発体制
- 開発に要する費用概算

3.4 開発とメインテナンス

最近では，多くのシステム部門における共通の問題として，業務の主力が既存システムへの修正・改善，すなわち，メイテナンスとなってきている。企業によっては，要員の 40％以上を投入しているケースもある。

この傾向は，次のようなことが要因として考えられる。

① コンピュータ規模が大きくなり，多くのシステムを保有するようになった。

② コンピュータ・ハードウエア自体がファミリー・シリーズ化してきており，システムが長期使用されるようになった。

③ 経営変動がはげしく，環境の変化，経営の変化が著しい。

④ ユーザのシステムに対する理解が深まって行き，改善要望が多く出されるようになった。

一般に，開発とメンテナンスを厳密に区分することは困難であり，また，そのいずれが得策であるかを識別することもむずかしい。例えば，現在実施されていない業務のシステム化が開発であるが，既存システムのマスタ・ファイルのレイアウトを変更し，それに関連する相当数のプログラムに修正を加え，さらに新規のプログラムを追加するような場合は，メンテナンスではあるものの，新規開発とも言い得るような性格をもっている。

また，このような場合，新たなシステム設計方針のもとに関連するシステム全体の再構築を計ることによりシステム効率を上げる方が結果的には利益となることがある。

この観点から最近でてきているのが，定期的な再開発（scheduled scrap & build）という考えである。いかなるシステムでも開発後６～７年 － 計画から通算すると７～９年 － 経過すると，その間の社会的，制度的変化，企業内の変化によって，いわゆるメンテナンスに耐え得ないようなものになることから，システム部門の予定された業務として再開発作業を組み込んでいくものである。

定期的再開発のためには，常にシステム要員の何割かを確保しておくことは言うまでもないが，最も重要なことは，部門の運営方針として確立することである。

なぜならば，この計画は，外部ユーザからのニーズでなく，システム部門内部からのニーズであるだけに，とかく腰くだけになり，計画遂行が困難となりがちな問題である。

また，システム設計方針としても，再開発が容易になるように，サブ・システム毎のモジュール化，サブ・システム間のインタフェースの単純化・明確化に必掛けなければならない。

3.5 大型プロジェクトの形成

前項に述べた「定期的な再開発」によっても，

・オンライン化の大幅な拡充

・端末システムの入れ替え

・センタ・コンピュータの機種変更やグレード・アップ

といった大きな節目においては，充分な対処ができないことが多い。

このような場合には，中期計画の立案，大型プロジェクト・チームの形成，経営に対するオーソライズが必要となる。

プロジェクト・チームの形成に当っては，

・チーム組織構成

・作業計画

・運営方法

・プロジェクト・マネジャ，リーダの役割

に留意する必要がある。

システム要員の専門化に伴って，大型プロジェクト・チームにおいては，特に，共通テーマを効率よく処理するための専門グループが設定される。

① 業務処理設計グループ

② 事務処理方式，帳票設計グループ

③ データ・ベース，ファイル設計グループ

④ オンライン・コントロール・ソフトウエア設計グループ

⑤ 端末，回線設計グループ

⑥ 運用管理方式設定グループ

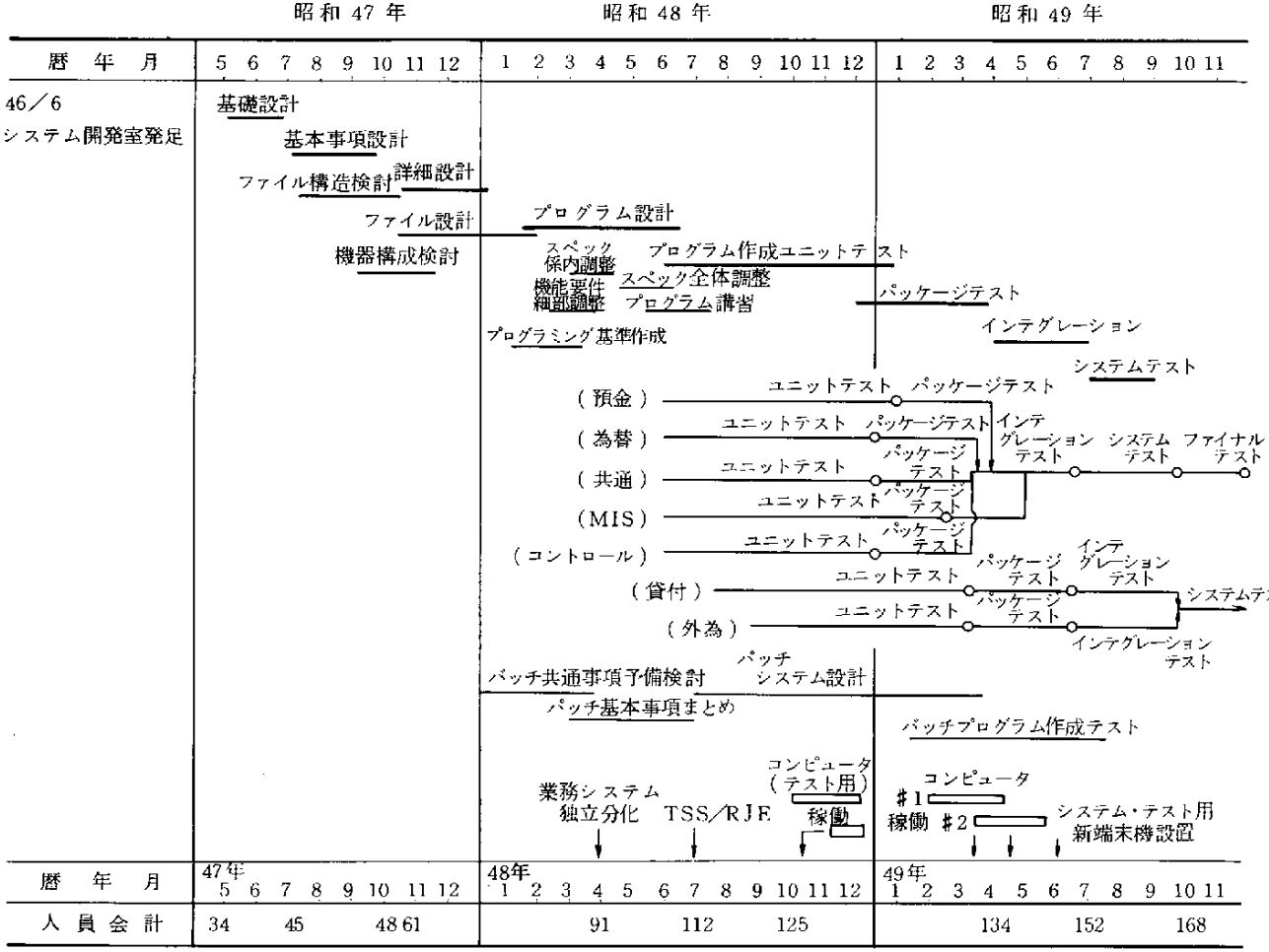
⑦ 共通プログラム設計グループ

⑧ システム・テスト・グループ

といったグループが，プロジェクトの進渉状況に応じて，ダイナミックに形成，再編成され，適切なプロジェクト進行が計れるようにされる。

次に大規模バンキング・オンライン・システム開発プロジェクトのマスタ・スケジュールと各開発過程に応じて変遷する専門グループ編成（組織体制）を表３・３，図３・４に示す。

表３・３　大規模バンキング・オンライン・システムの開発マスタ・スケジュール



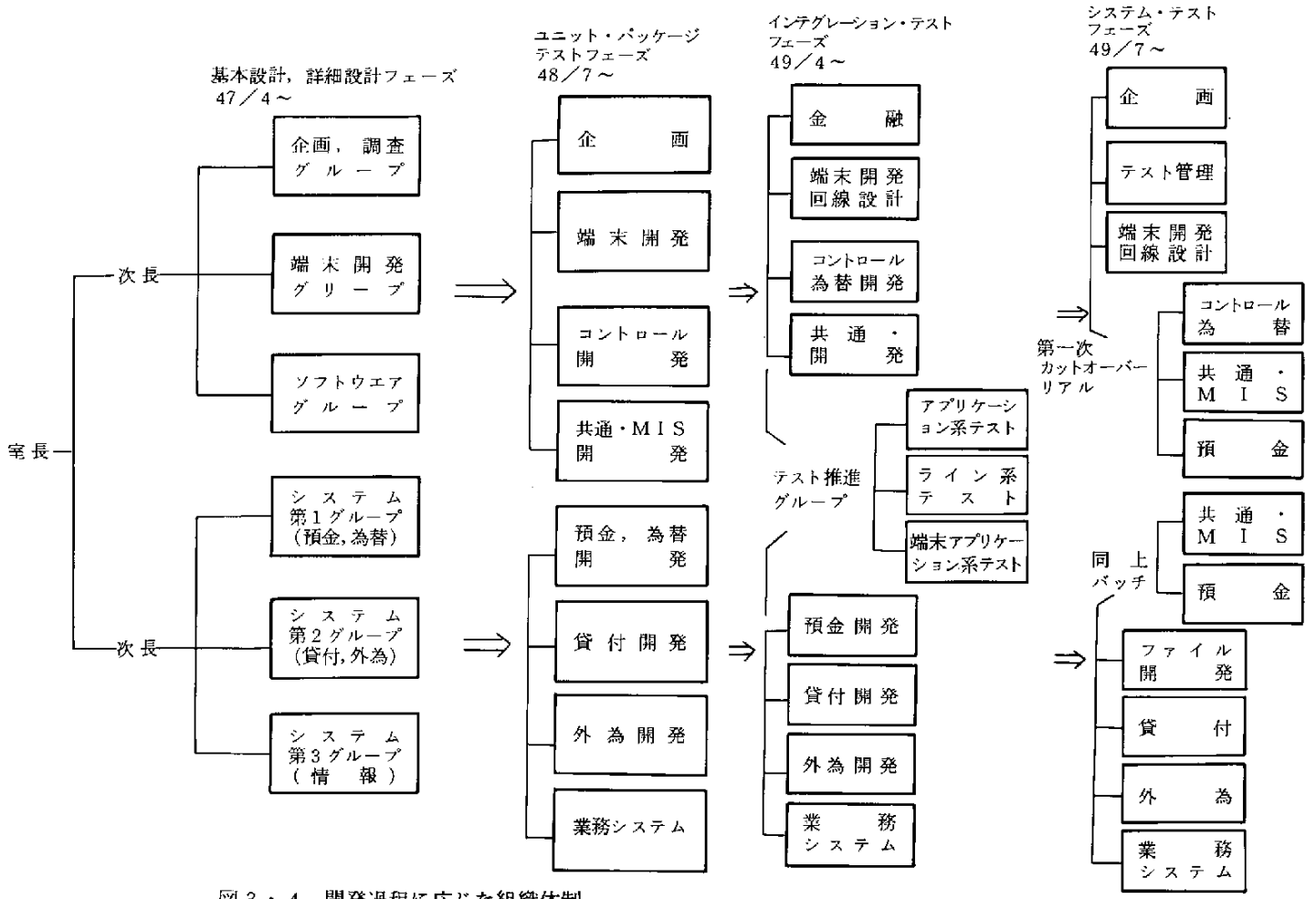


図３・４　開発過程に応じた組織体制

3.6 標準化とドキュメンテーション

標準化の目的は，結論的に言えば，システム開発の効率を向上させることであるが，より細分すると，次の2つに分けられよう。

(1) システム開発自体の効率化

① システム開発段階と作成ドキュメントおよびその内容の規定など。

② プログラミング・テクニックの適用。

(2) システム開発管理の効率化

① チーフ・プログラマ・チーム制などの開発体制の規定。

② プロジェクト管理基準，進捗状況管理基準など。

上記のうち，(2)については，規則としていったん設定したものは，それを遵守させる必要があるが，(1)については，基準を設定し，その運用にはある程度の幅をもたせることが必要であろう。また，(1)については，特に，適用した結果，都合の悪い部分は適時に手直ししながら，より良い基準にしていく方法をとる。

標準化を推めるためには，専任のグループ，もしくは管理調整グループ内に担当者を設置する必要がある。標準の設定のためには，次に示すような相当の作業が存在するためである。

- 現行システムの問題点と標準化すべき項目の抽出
- 標準体系の設定
- 標準設定・改訂手順の設定
- 標準化推進体制の確立
- 標準原案の作成
- 原案の各グループ・リーダへの内覧と意見の収集
- 原案の修正と修正案のオーソライズ機関（標準化委員会などを形成しておく）への提示と承認の取得
- 標準の公布（標準書の発行など）
- 説明会や教育の実施
- 各機関（標準承認・チェック，教育など）の運営

その実施の徹底を計るために最も重要なことは，システム部門の最高責任者が，標準化全般についての深い理解と推進に対する決意をもつことであろう。

また，具体的な実施の道具として，次のようなドキュメント様式についての設定が必要である。

① プログラム関連ドキュメント

- プログラム・コーディング用紙

・出力帳票設計用紙

・入力カード設計用紙

・ファイル設計用紙

② 新しい開発技法に基づくドキュメント

・階層的入力・処理・出力記述（HIPOチャート）用紙

・モジュール定義用紙

・モジュール間インタフェイス仕様定義用紙

・データ定義用紙

③ システム開発管理用ドキュメント

・週間スケジュール表

・プロジェクト進捗状況管理表

・活動報告書

・長期スケジュール表

④ 運用管理用ドキュメント

・ラン・チャート

・プログラム・オペレーション・マニュアル

・磁気ファイル入出力指示書

・ラン・スケジュール表

標準化の推進に当っては，それらのドキュメントをうまく組合せて用いることを教育することと，完成したドキュメント内容をチェックする常設機関を設けることがポイントである。チェック機関は，専任要員を確保することが望ましいが，それが不可能な場合は，システム部門内の各リーダを委員とした委員会形式によるのも一つの方法である。

また，教育の一環として，システム部門内で規定した全ての規準・基準・標準を集めた標準書を作成・配布する方法もある。

このように，ドキュメント体系が整備され実行にうつされると，つぎの2つのことを考慮しておかなければならない。

① 開発スケジュールへの折込み

開発終了時期にドキュメントの整備期間を見込み，スケジュール化する。

② ドキュメントの保存管理

保存とメンテナンス規定を設定し，誰でもが利用可能なよう体系的に保存し管理を行う。大量になる場合は，専任者を設ける。

先にも述べたように，設計の最終的な産物はドキュメントとして出力される。適切なドキュ

メンテーション技法および体系を整備することは，よい設計を行うための必要条件である。

ドキュメントについては，公的に決まった様式はなく，各企業ともそれぞれ工夫して実施しているのが現状である。

ドキュメントについては，各企業における実例を学習する方法をとる。

3.7　新しいシステム開発技法

新技法については，単元D 2第3章（ 3.6 ）でも学んだが，管理的視点で復習する。

構造化プログラミング(structured programming)という言葉が使われはじめてからすでに数年が経過している。その間にも，モジュラー・プログラミング，ワーニエ・メソッド，ストラクチャード・デザイン，IPT (improved programming technique)，HIPO (hirachy plus input - process - output)，トップ・ダウン設計といったシステム設計の新しい傾向を示すことばが多数登場した。

これらの新技法の登場する背景には，

① ハードウエアの急速な進歩に対して，ソフトウエアは遅れ気味である。

② ハードウエアのコスト低下傾向に対してソフトウエア開発コストは上昇傾向にある。

③ ソフトウエアのライフ・サイクルが長くなってきている。

④ ソフトウエアが巨大化しつつある。

などが考えられる。

これらの新技法発展のきっかけとなった構造化プログラミングは，E. W. Dijkstraが，1969年の第2回NATOソフトウエア・エンジニアリング会議で発表したワーキング・ペーパーの中で発表されたものである。その前年1968年にダイクストラは，有名な「GOTO statement considered harmful」というペーパーをCACMに発表している。

これらの考えの中で彼は，プログラムが大きくなるに従って増大する困難さに注目し，プログラムの正しさを証明する上で，よい構造をもったプログラムを作成するため制御構造の制限や抽象化の有用性，階層構造による段階的な精密化などについて述べている。

それを受けて，H D. Mills らは次のような研究を行った。

① すべての「通常のプログラム」 － 適正プログラム (proper program) － が3つの基本構造パターンのみによって表現できることを数学的に厳密に証明することで，理論的基礎を固めた。

② ①が理論的に証明できても実際にあるプログラミング言語においてそれを採用するには制約があることが多い。そのためそれぞれの言語に対する構造的記述の規約を作成し，具体的なテクニックの標準化を行った。

③ 部門内の教育活動を行い，開発援助ライブラリを基礎にソフトウエア開発チームを改革して実際の開発に適用した。

ミルズらの研究に，IBM社のOS開発中に考案されたHIPO（開発すべきソフトウエアシステムの機能を表現するための文書化技法）と，従来からいわれてきたモジューラ設計の方法論として展開されてきた，構造化設計（structured design）が結合して完成されたのが，IBM社のIPTである。

IPTとならんで，それ程の広範囲の作業をカバーしていないものの，各種の技法が発表されてきた。

そのいずれもが結果的には，

① コーディング作業自体にはほとんど労力を必要とせず，しかも質をあまり問わない労力でも遂行できる。

② 開発作業量の大手は，設計，それも機能の設定，構造の設定とテストに費やされる。

③ 設計段階での手抜きが結果的にコーデイング時の余分な，かつ，困難な作業を生じさせている。

といった点を一般に認識させ，システム開発における機能の設計と構造の設計の重要性を指適することとなった。

従って，各種技法の適用に当っては，その技法の背景となっている基本的考え方を充分理解し教育することが必要であり，表面的にまねをしても，思ったほどの効果を上げられないことが多い。

参　考

構造化プログラミングとIPT小史

① 1968 GO TO statement considered harmful（Dijkstra），CACM 1968・3

② 1969 ダイクストラ「構造化プログラミング」発表，NATO ソフトウエア・エンジニアリング会議

③ 1971 The programing language PASCAL（N. Wirth），Acta Information

④ 1972 構造化プログラミング（O. J. Dahl，E. W. Dijkstra，C. A. R. Hoare），Academic Press

⑤ 1972 Mathematical Foundations for structured Programming（H. D. Mills）IBM FSC72−6012

⑥ 1972 Chief Programmer team Management of Production Programming

(F. T. Baker), IBM Systems Journal

⑦ 1974 Structured Design (L. L. constrantine, G. J. Myers, W. P. sterens), IBM Systems Journal

⑧ 1975 Reliable Software Through Composite Design (G. J. Myers), Mason Carter Publishers

⑨ 1976 Structured Walk- Throughs and Inspections, IBM WTSC

指導上の留意点

(1) ソフトウエアの開発技法は，近年，新しい試みがいくつかなされており，実際に適用する場合を想定して，できるだけ具体的に指導する。

(2) システムの標準化，ドキュメントの標準化は，開発管理レベルを向上するための一つの手段であり，その定着化が問題である。いかにして定着させるかの観点からの指導を行う。

(3) 開発管理の実態は，各企業のコンピュータ部門によって異なる。この面から，できるだけ実例を折り込みながら指導すること。

参考文献

〔1〕 B.W.Boehm, "Software Engineering" IEEE Transaction on Computers, 1976.

〔2〕 G.J.MYERS，有沢 誠（訳）「ソフトウェアの信頼性ーソフトウェア・エンジニアリング概説」 近代科学社，1977.

〔3〕 P.W.メッガー，下田正昭．久保未沙（訳）「プログラミング・プロジェクトの管理」近代科学社，1977.

〔4〕 IBM Corporation, "Improved Programming Technologies -An Overview", GC 20－1850, 1975.

〔5〕 IBMマニュアル，「HIPOー設計の補助手段と文書化の手法」，GC 20－1851－1，1975.

〔6〕「ソフトウェアの標準化」 日本経済新聞社，1977.

〔7〕 F.P.Brooks, Jr. 山内正弥（訳） 「ソフトウェア開発の神話」 企画センタ，1977.

〔8〕 石崎純夫，藤田 猷共著，「トータル・バンキング・システム」 産業図書，1972.

〔9〕 HUGO, I.ST.J., "A Survey of Structured Programming Practice", AFIPS, 1977.

〔10〕Yourdon, E. and Constantine, L.L., "Structured Design", Youdon Inc, 1975.

〔11〕Warnier, J.D., 鈴木君子(訳) 「ワーニエ方式によるプログラミング学習」 日本能率協会, 1973.

〔12〕Warnier, J.D., 鈴木君子(訳) 「ワーニエ・プログラミング法規集」 近代科学社, 1975.

〔13〕江川 朗, 「プロジェクト・チーム」 日本能率協会, 1970.

〔14〕R.S.Kuehne, H.W.Lindberg, W.F.Baron "Manual of Computer Documentation Standards With Forms", Prentice-Hall, 1972.

〔15〕Max Gray, Keith London, "Documentation Standard", Auerbach Publishers, 1969.

〔16〕岩城三郎, 「オンラインプログラム設計入門」 日刊工業新聞社, 1973.

第4章　コンピュータの導入

キーワード

機種決定（computer selection），性能評価（performance evaluation），性能予測（performance prediction），性能計測（performance measurement），レンタル費（rental costs），買取費（purchase cost），ソフトウエア調達（software procurement），サービスの選択（selecting appropriate services），契約の観点（contractual aspects），設置手順（installation procedure），保守料（maintenance fee），コスト配賦（charge out），設備整備（equipment acquisition），電源（electric power resources），契約の観点(contractual aspects)，

目　　標

コンピュータの導入は，企業にとって，またコンピュータ部門においても，一つの転機であり，機種決定からテスト・検収にいたるまで特別の作業が発生する。

コンピュータ導入に伴って実施すべき種々の作業は，ルーチン的なものでなく，一時的な作業であることから，学習の方法として次の方法が望ましい。

① 代表的なコンピュータ・センタの見学による知識の習得。とくに，付帯設備の状況を知る。

② コンピュータ導入のケースを設定し導入作業のイメージ・アップをはかる。

ハードウエア設置に関しては，コンピュータ・メーカ，または販売会社において専門組織要員体制が充実されており，コンピュータ部門としては，それに対応する知識として，この章の項目を学習していく態度をとるものとする。

コンピュータ機種の決定については，従来から経営レベルの判断により実行されるケースが見受けられるが，コスト対効果（コスト・パフォーマンス）の面から，コンピュータ部門における専門的見地を通じての選択・評価が重視されつつある。

このような観点からも，新しい技術動向および新機種などの性能に関する知識取得などを折り込みながら指導していくものとする。

内　　容

コンピュータの導入は，買取制・リース制・レンタル制を問わず，企業にとっての事業計画

の一環であり，また，情報処理確立のための大きな武器である。

このため，特に，機種決定に際しては，慎重に対処しなければならない。

また，大型コンピュータの導入は，附帯設備面からの条件整備が要請され，この面での配慮が，導入後の効率的な運用の前提条件ともなる。

この章では，

① 機種決定までに行う，コンピュータの選択と評価。

② コンピュータすえつけのための設備条件。

③ ハードウエアの保守。

について，予備知識と管理上の留意点を学習する。

4.1 コンピュータの選定と評価

コンピュータの導入とその活用は，企業にとって大きな効果をもたらすものである。一方，コンピュータは，高価なものであり，限られた導入資金内において，使用目的に合致した機種を選定するための評価を可能な限り厳格に行う必要がある。

そのためには，次のことに留意する。

① 導入の目的・方針を明確化する。

② 導入推進組織を確立し良きプロモータを得る。

③ 導入にあたっての制約条件を明確化する。

④ コンピュータ評価の尺度を多くもち，色々な角度から検討する。

このように評価・選択されたコンピュータの導入は，経営最高責任者がその決裁を行なうのが通常である。

(1) 導入の前提条件

コンピータの選択にあたっては，次のような諸条件をよく検討し，その結果，ある程度の候補機種の範囲を絞り評価することになる。

- 導入の目的を明確化する。
- 社内の受け入れ体制を作る。
- 適用業務の作業量を十分検討する。
- 事前に採算性を十分検討する。
- 先進会社等の見学と意見を聞く。
- 適用業務に合った機種を選ぶ。
- メーカーのカタログだけに頼らない。
- 経営トップの意向を十分確認する。

コンピュータ導入計画は，まずシステムの要求があり，調査分析をし，概要設計，機種選定，適用性の検討というステップが説明され，図式化されている。しかしながら，現実には，コンピュータのハードウエア構成の計画が先行し，それに合せた適用業務を選定し，コンピュータ化をはかるという例が多く見受けられる。

(a) 使用目的の明確化

導入するコンピュータの使用目的によって選択する機種の範囲が決定する。したがってまず，使用目的を明確化することが大切である。

利用方法としてあげると，

例えば，

① 事 務 処 理

② 技 術 計 算

③ 工 程 管 理

④ バ ッ チ 処 理

⑤ オンライン・リアルタイム処理

⑥ タイム・シェアリング処理

これらの利用が，単独の場合もあるが，一般的には，複合的に組合され利用される。

また，最近の動向としては，

① データ・ベース指向

磁気ディスク装置を多用してのデータの蓄積と検索による処理の高度化。

② コンピュータ・ネットワーク指向

コンピュータ同志を通信回線で接続し，資源（リソース）の有効利用をはかる方法。

などにより，コンピュータの選択については，ハードウエアそのものの性能のみならず，その性能を十分引き出すベーシック・ソフトウエアの優劣は見逃せない。

どのような利用形態をとるにせよ，導入するコンピュータは，何のために利用するかという点が先決である。

利用目的を大別すると効果の面から，

- 省力化の達成（直接的コスト削減）
- 管理水準の向上・質の向上（間接的コスト削減）
- 機会損失の防止

また，システムの適用分野からみると，

- 業務分野 (operational system)
- 管理的分野 (administrative system)
- 戦略的分野 (strategic system)

への利用が考えられる。

また，次のような目的もある。

- 計算・集計 (operational)
- 解析・予測・計画 (planning)

これらの目的は，通常，単独なものではなく，複合したものとして計画される。

利用目的を適用業務という面からみると次のような分野がある。

- 生　　産
- 在　　庫
- 営　　業
- 経理・財務
- 人事・労務
- 技術・設計
- 企画・調査
- 広告・宣伝

(b) 導入推進体制の確立

経営からコンピュータ導入の方針が打ち出されたら，組織としての推進母体を設置しなければならない。

この組織は，企業の規模，導入目的，投下資金量・導入までの期間などにより異なるが計画の初期段階ではプロジェクト・チーム型で，トップと直結した「委員会」が設けられるケースが多い。準備段階も後半になると，コンピュータ担当部門へバトンタッチされ，より具体的な導入準備活動へと展開されていく。

コンピータ導入は，その企業にとってはじめての場合と増設。置き換え（リプレース）の場合では，導入推進体制およびその活動内容が相異するので，ここでは新規導入のケースとして学習するものとする。

導入組織が主として行うべき業務は，次の通りである。

① 社内ＰＲと全社的協力体制の確立

コンピュータ導入の目的とその効果を十分PRし，計画が成功するよう全社的なコンセンサスを得る。通常の場合，経営トップの強い決意により，行動が支援される。

② タイム・スケジュールの決定

目標とする稼動開始日までのマスタ・プランを決め，その実施とフォローアップを行う。

③ コンピュータ室の要員および設備の確保

導入するコンピュータを有効に活用する主体は要員であり，適正な見積りが必要である。また，設置のための諸設備の確保も併せ行う。

④ システムの計画立案

総合的な機械化，経営管理の高度化，経営を積極的に支援する情報処理という観点から実施するシステムの計画を立案する。

(c) 導入の資金計画の立案

コンピュータの導入とその運営には多額の経費と時間を要するものであり，採算性を十分検討する必要がある。また，導入にあたっては，経営から「投資限界」をなんらかのかたちで，具体的に明示してもらい，その範囲においてEDP化が計画されるのが通常である。

一般に，コンピュータ導入に関する費用は，つぎのように区分される。

イニシャル・コスト（導入前の費用）
- 導入準備費用
- 導入時の費用

ランニング・コスト（導入後の費用）

① 導入準備のための費用

つぎのような費用がある。

- 導入組織に要する費用

  大部分が人件費である。

- 機械室の設備費用

  電源，空調，床工事，間仕切り，防音装置，配線工事，備品など。

- 社内組織・職務手続き変更する費用

  会議室，帳票費，移行のための切替え費用など。

② ハードウエア導入時の費用

ハードウエアについては，買取制とレンタル制・リース制があり，買取制は一時的に資金は要するが，つぎのようなメリットがある。

＜買取制のメリット＞

- 使用時間（稼動時間）に制約がない

・法定償却年数を超えて使用する場合は，残存簿価がハードウエア・コストとなり安いコストで使用できる。

・資金的に余裕がある場合は，定率法を使用した原価償却により，投下資金を早期に回収できる。

このようなメリットを享受できる反面，導入したコンピュータは，長期間使用されることによるシステムの陳腐化・老朽化を招きやすい。最近のコンピュータの技術革新によるコスト・パフォーマンスの向上，利用技術の向上の面から取り残されるという危険も含んでいる。

＜レンタル制・リース制のメリット＞

・一時的な費用支出が発生しない。

・使用時間に合せた費用の支出ができる。

・新しい装置と置き換え（リプレス）しやすい。

買取制とレンタル制・リース制は，各々長短があり企業の環境・方針により異なる。

また，それぞれの制度を併用している企業も多くあり，その場合，周辺機器のみレンタル制を適用している例が比較的多い。

コンピュータ・ハードウエアの費用支出方法については，次の角度からも検討が必要である。

・経営上の資金ポジション

一時的な支払いが許されるかどうか，また，資金の見通し。

・コンピュータ使用期間の方針

導入したものを徹底して長期間使用するのか，新しい機種に早目にリプレスしていく方針かどうか。

・経費的観点からの諸計算

金利計算，買取の場合の償却計算，法人税納付に関係する税金計算などの諸計算を行ない。財務上の問題も検討する。

・売上高とコンピュータ費用の関係

一般的な尺度はないが，同業他社等との比較などから投資限界を見極めておく。

・メーカーの調査

コンピュータ・メーカーにより価格設定方式が異なるため，十分な調査が必要。

③ 導入後の費用

導入計画においては，とかく導入までの費用に重点が置かれ勝ちであるが，導入後5～6年先を見通しての導入後費用についても見積りが必要である。

導入後の費用には，次のようなものがある。

・組織の運営費用

要員種類別，すなわち，管理者，プランナ，プログラマ，オペレータ，パンチャなどの人件費と，その組織を維持するための諸経費。

・レンタル料・リース料

メーカー，およびリース会社により料金設定は異なる。国産機の場合，JECC（日本電子計算(株)）のレンタル料が参考となる。

・保　守　料

レンタルの場合は，保守料が含まれているのが通常である。また，保守料金は，ユーザの使用時間帯，故障の度合による部品の取扱い，保守員の常駐・非常駐などによって異なりメーカーとの細目についての折衝が必要である。通常，保守料は買取価格の４～５％程度である。

・データ伝送費用

オンライン・システムを利用する場合は，データ伝送費用として専用回線費用，専用権設定費用，公衆回線費用などが必要。

・電力料金・水道料金

・消　耗　品　費

カード，磁気テープ，紙テープなど。

・帳票印刷費

制定用紙の場合の版代，印刷費など。

・スペース費

貸ビル，自社ビルを問わず家賃，共益費など，使用するスペースに係る費用。

・保　険　料

レンタルの場合，保険料が含まれているのが通常である。最近は，「コンピュータ総合保険」「情報処理業者賠償責任保険」などの新種の保険があり，従来の火災保険でカバーできなかった分野，例えば，情報メディアの損害についても保証するような保険が出現している。その導入の可否も設備状況と併せて検討する必要がある。

・減価償却費

コンピュータ機器，端末機，固定設備，備品器具などにより法定償却年数が異なり，また，定率法・定額法により償却額も相違する。

・固定資産税

・外　注　費

プログラム作成費，パンチ費，切替作業費など，外部業者へ作業依頼する場合に必要。

これらのランニング・コストは，経済環境の変化により変動するものが多い。したがって，計画段階での見通しが必要である。また，導入実施後も計画との対比を継続して行っておくことは，次期の導入計画策定にも役立つので，資料整備と要員育成を心掛ける。

(d) 採算性の検討

電子計算機システムの導入は，企業の合理化をはかり，企業にメリットを与えることを目的とする以上，その採算性，すなわち，経済性について検討する。

システム導入の採算性の計算は，かなり複雑であるため，その計算が十分行なわれておらず，経営者のカンと度胸だけに依存しているケースも見受けられるが，コンピュータ部門は，常に，コストと効果に対する意識をもち続けなければならない。

EDP化の経済性をみる場合，短期的・目先的な効果を狙うことよりも，長期的な展望に立った総合的なコスト・セービングの考え方をとる。

① 経済性を表現する場合，次のような費用対効果の表現が一般的であろう。

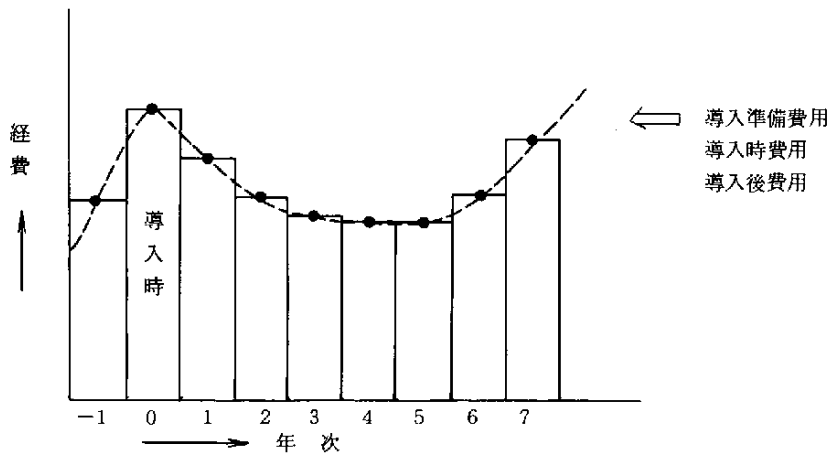


図4・1 費用の推移

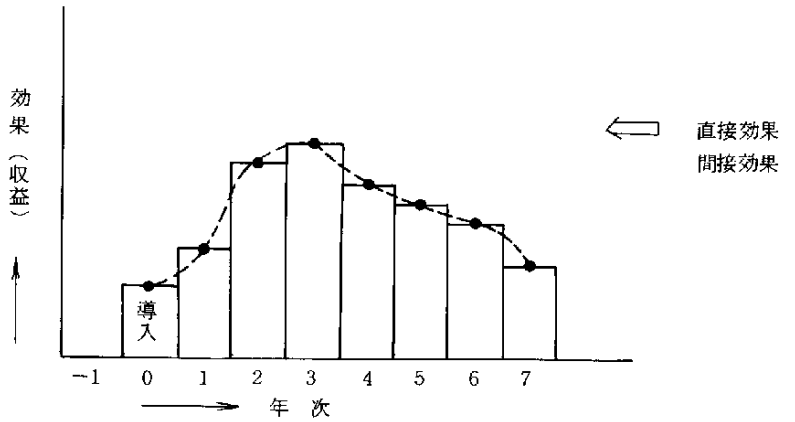


図4・2 効果の推移

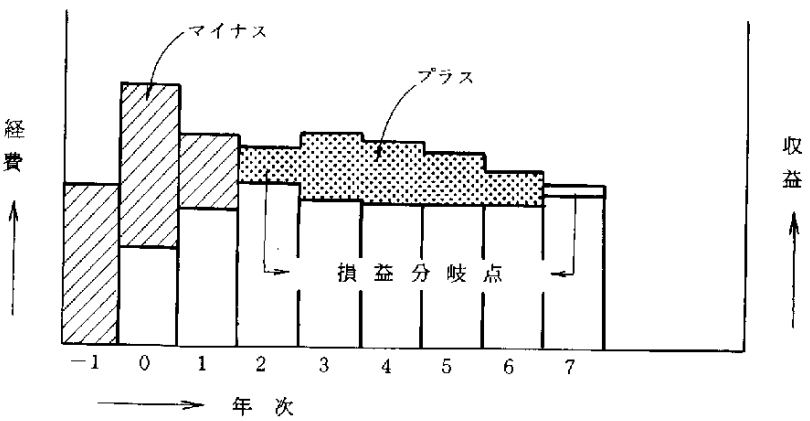


図4・3 費用対効果の経済性

このグラフからみると，初期段階は収益が得られないが，2～3年目からプラスに転じ，6～7年目から老朽化によるディメリットとなり，新しい設備更新が必要となってくることを示している。

しかし，このグラフは，あくまでも一般的な傾向であり，既存の設備との総合的な収支を見極める必要があろう。

② 効果の分類

前節(c)では，費用の分類と留意点を述べたが，ここでは，効果の内容を示す。効果については，

・直接的効果……量的な効果

・間接的効果……質的な効果

に大別される。

<直　接　効　果>

・人件費の節減

直接人件費，間接人件費など企業の実体により計算する。

・事務費用の節減

帳票，事務所のスペース，機器，事務用品などの低減。

・在庫仕掛の低減

管理レベルの向上による在庫量，仕掛量の低減，製造原価の引き下げ。

・機械の稼動率向上

日程管理の合理化による固定資産回転率の向上

・スピードアップ，能率向上

資金面から金利計算を，また，人件費の換算を行う。

<間　接　効　果>

・計画・判断の迅速化

利益の最適化，販売の迅速化

・管理水準の向上

管理の精度が向上し，損失を少なくすることが期待できる。

・予測の適正化

市場の需要予測，適正な生産量による合理化効果。

・サービスの向上

納期の適正化，得意先への連絡の迅速化。

・量的拡大への対処

マス処理によるメリットの享受。

これらの効果は，可能な限り数値にして具体的に表現されることが望ましい。コンピュータの導入には多額の経費を必要とするので，単なる人員削減をねらうと失敗するケースが多く見られる。むしろ，

・総合的長期的な効果をねらう。

・企業の体質改善に寄与よる積極的効果をねらう。

などの目標を設定し，それに見合ったシステム建設計画を立案する。

効果については，各企業のコンピュータ化の度合，企業の性格，環境によってどのポイントを重視するかが異なる。したがって，導入の採算性について評価することは，か

なりの困難を伴うので，安易なメリット計算は慎むべきである。

(e) 導入の動機の明確化

導入の動機については，その企業体質の向上である。新規の導入とレベルアップにより動機は異なるが，大別すると次のようになる。

- 経費の節減

  労務費，経費，材料費など。

- 資料提供の迅速化
- 直接的利益の増加
- 間接的効果向上

また，レベルアップについての動機をアンケートからピックアップすると次のような答があげられる。

- 処理量の増大
- 処理時間の迅速化
- トータル・システム指向
- コンピュータ部門の生産性向上
- 新規業務の発生
- サービス向上
- システムの弾力性欠如
- 労働力不足
- マン・マシン・システム指向
- 情報の価値の向上
- 本社と事業所の連絡強化
- データバンクの整備

(2) 選定のための評価

評価については，単元D２第２章において学習したので，ここでは次の点を復習し，導入事例を調査する場合のチェックポイントとするよう指導することが望ましい。コンピュータ導入においては，ハードウエアの特質を十分理解することは必要であるが，ソフトウエアの重要性も見逃せない。

特に，新機種発表の場合は，ハード的な性能面に目をうばわれ勝ちであるが，できるだけ多角的に総合評価すべきであろう。

(a) 性 能 評 価

性能の評価は，システムの構成，メーカーの特色的な差異などにより単純に行うことは

困難である。しかし導入目的と負担可能なコストの限界から評価範囲は絞られてくる。評価は，あくまでも相対的なものであり，絶対的なものではないことを知っておく必要がある。

性能評価は，大別すると次の2つとなる。

・静的（スタティック）な評価

主としてカタログ値に基づくもの。

・動的（ダイナミック）な評価

ギブソン・ミックス（Gibson Mix），ベンチマーク・テスト（Bench Mark Test），などによる処理能力係数（Processing Activity Factor）の比較。

(b) その他の評価

メーカーからの説明書，導入ユーザからの事情聴取などを加味し，最後にコストとの対比を行ない結論を出す。

前項の性能評価のほかに，次のような点についても評価しておく。

① 保有機種との互換性

リプレースするような場合は，プログラムの互換性が最も要求される。しかし，新規増設の場合は必ずしもプログラムの互換性は要求されず，多くの場合，磁気テープあるいは，通信回線によるデータの受渡しが可能であれば良い。

プログラムの互換性の場合も互換性の程度を慎重に検討しておく必要がある。

② オペレーションの容易性

容易性は，オペレーティング・システムの機能に負うところが大きく，また，サポート・ソフトウエアの完備状況も調査しておく。

③ 導入に要する期間

メーカーまたは機種により納期が異なるので調査が必要。

④ 技術的な角度

技術的な先進性とその信頼性について評価する。

⑤ 信頼性・安全性

納入実績とユーザにおける評価が主たる尺度となるが，初期ユーザにとっては，評価しづらく，メーカーの出荷前のテスト状況，開発期間などから判断せざるを得ない。

⑥ 設 備 条 件

空調の条件，設置面積，消費電力，電源条件，調整期間などを評価する。

⑦ 保 守 性

保守体制，保守の容易性，保守ツールの完備度合など，ハードウエア・ソフトウエア両面から評価する。

⑧ メーカーのサービス

導入に対するサポート，教育サービス，システム建設の支援など，メーカーのサービス状況について評価する。

4.2 コンピュータの設置条件

コンピュータ導入に際しては，運用面や将来の拡張性を十分考慮して，設置に関する諸条件が検討されなければならない。

通常，メーカーに設備計画の専門セクションとその要員が配置されており，導入側の負担はかなり軽減されている。

導入のための設備を整えるためには，多額の費用が必要であり，導入機種，適用業務，設置場所との関係などから，どこまで完備すれば良いかの見極めが大切である。

設備については，代表的ユーザの見学が望ましい。

(1) コンピュータ関連の室

自社内のどこかのフロアーを利用する場合とコンピュータ専用の貸ビルにより条件が異なるが，概略，次のような室が必要である。

(a) コンピュータ室

- 将来の拡張性を可能な限り折込む。
- オペレーションやハードウエアの保守が容易なスペース
- 磁気ファイル，帳票などのキャビネットなどのスペース

(b) パ ン チ 室

防音，休憩場所，コンピュータ室との連絡を考慮する。

(c) 電 源 室

定周波電圧装置，自家発電装置，無停電装置など設置する場合は，独立した室が必要。

(d) 空 調 室

空調する量が多く冷凍機を設置する場合は必要である。空調機をコンピュータ室に設置する場合は，必ずしも必要でない。

(e) 磁気ファイル，帳票，保管室

媒体の性質から温湿度を一定に保つことが必要である。

(f) 保 守 室

保守員のための備品・器具および机などを置く室。

(g) プログラマ室

オープン・デバッグなどのための独立した室が望ましい。

(2) コンピュータ室の設計

(a) レイアウトの設計

・オペレーションの容易性

・オペレータへの配慮

オペレータの健康に留意して，中央処理装置を別の部屋にしているケースもある。

・拡張性の配慮

・保守の容易性

(b) 床 構 造

大別すると3つの方法がある。

① 床上ダクト方式（ケーブルカバー方式）

装置間のケーブルにカバーをつけて保護し，床面の加工を施さない方法であり最も簡単な工事である。ただし，床面に凸凹が生じ，オペレータの作業に支障をおよぼす。

② ダクト方式（レースウエイ方式）

床面に溝を掘り，その溝にケーブルを通しカバーをする方式である。設備のレイアウト変更が困難であるが経済的。

③ フリーアクセス方式（上床式）

床を二重構造にし，上床と下床の間にケーブルを敷設する。上床は自由に取りはずしと穴あけが可能で設置変更は容易である。また，空調等も床下を利用できて効果的であり，最も多く利用されている方式である。

床構造については，次の点も併せ留意しなければならない。

・アースの取り付け

・床の強度

一般のビルの強度は，通常コンピュータを設置する床荷重となっていない場合が多く，注意を要する。

・床の表面材料

(c) 空 調 方 式

空調方式には，導入するコンピュータの規模により，次の3つの方式がある。

・パッケージ空調……中・小型機

・ダクト空調……中・大型機

・床下空調……大・超大型機

空調について考慮すべき事項には，次のようなものがある。

① 室内の温・湿度の許容限度

機種によって相違する。

② 空調装置能力の計算

発熱量，侵入熱量，室の広さなどから容量を計算する。

③ 塵埃除去装置の設置

④ 温・湿度の自動制御

(3) 電 源 装 置

電源装置の必要な理由を説明し理解させる。

(a) 電源装置の必要性

① コンピュータの入力規格（通常 208 V）と商用電源の電圧が不一致。

② 外国機種の場合，周波数が不一致のものがある。

③ 安定した入力を得る。

④ 停電対策。

(b) 電源装置の種類

下記のような装置が必ずしも必要なわけではなく，コンピュータの利用目的や機種によって異なる。

① 変圧装置（トランス）

コンピュータの規格に合せて変圧するもの。

② 自動電圧調整装置（AVR）

次の3つのタイプがあり，現在では，商用電源の質が向上しているので，中型機以下は，この装置が多く使用されている。

- 静止型AVR
- 誘導型AVR
- 電動発電型AVR

各々に長・短があり，メーカーによって異なる。

③ 定電圧定周波装置（CVCF）

②と比較してコストが高いが，大型機以上の設置に良く使われている。

静止型と回転型がある。

④ 無停止装置

シビアなオンライン・システムを実施するような場合に使用する。

・蓄電池……5分間程度のしゅん断をカバーする。

・自家発電装置……ジーゼル・エンジンを使用。

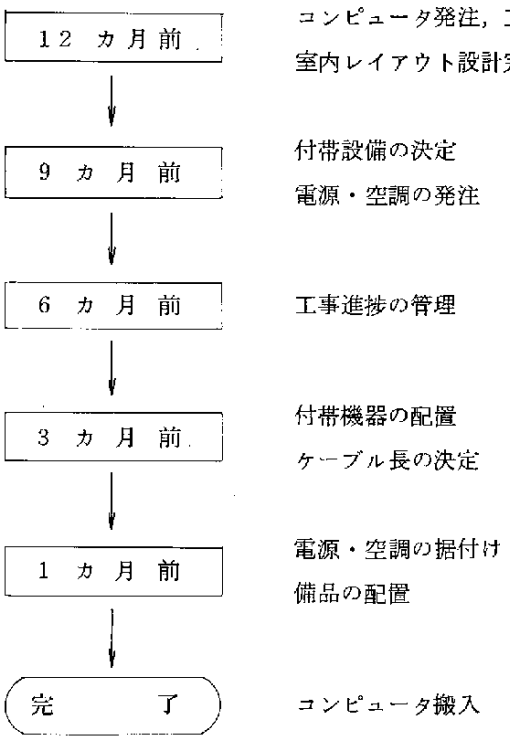
・フライング・ホイール型発電機

円盤を常時回転させて置き，停止の際発電機にかえる。蓄電池の代替的な意味をもつ。

(4) 設備の日程計画

導入機種決定から据付完了までの間は，前項までの諸設備を完成させなければならないので，PERTチャートを利用して，詳細な日程計画をたて，スケジュールにそったチェックが必要である。

計画が1年と仮定すると概略次のようになる。



4.3 ハードウェアの保守

保守については，ユーザ側の立場から説明する。

(1) 保 守 契 約

コンピュータ導入後1年間は，保守料金が無償であるのが通常である。ただし，部品交換については，有償のことが多い。

契約については，購入契約と同時に交わされるのが通例である。

契約には，次のような点が含まれる。

・料　　金

・期　　間

・シフトのあり方（時間）

・部品交換の取扱い

・常駐する場合の体制

・定期保守

(2) 予防保守と事後保守

① コンピータ停止を減少させるためには，予防保守が大切であり，そのためのコンピュータ時間を定期的にアサインしなければならない。最近は，予防保守のためのソフトウエアが各機種とも完備されつつあり，故障個所を事前に発見することが多くなった。

② 事後保守は，いかに早く復旧させるかに主題がある。したがって，部品の完備と保守員の質が問われる。

障害に対して迅速な復旧を要求するユーザは，機器構成上の余力（リダンダンシー）を持たせ，故障個所の切り離しによるリカバリーをとっている例が多い。

(3) 保　守　管　理

保守による障害対策のヒストリカル・ドキュメントを分析し，故障の傾向・修復方法など整理し，予防保守強化につとめる。

また，故障が多い個所については，メーカーと交渉し抜本的な改善策を講じるよう要請を行うことも必要である。

指導上の留意点

(1) コンピュータ機器の技術進歩は著しいので，新しい技術動向を極力折り込みながら指導すること。

(2) コンピュータ導入のケースを設定し，ケース・スタディによる導入作業の知識を取得させるよう指導する。

(3) 設備の条件および状況を知るために，代表的なコンピュータ・センタの見学を含めた実地指導が望ましい。

参考文献

〔1〕岡本行二，「コンピュータのための機密保護と安全管理」　オーム社，1974.

〔2〕南条　優，保坂岩男共著，「システム開発と運用」　オーム社，1972.

〔3〕産業能率短大システムデザイン研究会，「電子計算機導入ガイドブック」　1969.

〔4〕南条　優，「コンピュータ室の運用と管理」　日本生産性本部，1971.

〔5〕斉藤正治，「ＴＳＳの計画と運営」　ＴＢＳ出版会，1977.

〔6〕日本事務能率協会，「ＭＩＳハンドブック」　日刊工業新聞社，1969.

〔7〕日本情報処理開発センタ，「コンピュータ導入活用指針」　ＪＩＰＤＥＣ，1972.

〔8〕鈴木光彦，「図設電子計算機システム」　日刊工業新聞社，1970.

〔9〕長谷英夫，朝井幸洋共著，「情報処理センター」　ダイヤモンド社，1974.

〔10〕村川武雄，「設備投資の経済計算とその理論」　日科技連，1961.

〔11〕S.ステイムラ，大野　豊（訳）「コスト／パフォーマンス」　日本経営出版会，1972.

〔12〕岩城三郎（編）　「コンピュータシステムの拡張・変換の手引」　企画センタ，1973.

〔13〕「新版電子計算機ハンドブック」　コロナ社，1976.

〔14〕「情報処理ハンドブック」　オーム社，1972.

〔15〕社団法人行政情報システム研究所，「コンピュータ・システムの効率評価に関する研究」1975.

〔16〕Freiberger, W.Ed "Statistical Computer Performance Evaluation", Academic Press, 1972.

〔17〕Drummond, JR., M.E. "Evaluation and Measurement Techniques for Digital Computer Systems" Prentice-Hall, 1973.

〔18〕Lucas, JR., H.C "Performance Evaluation and Monitoring" Computing Serveys, Sept, 1971.

〔19〕Karl Seiler, Ⅲ. "Introduction to Systems Cost Effectiveness" John Wiley & Sons, Inc, 1969.

〔20〕J.Morley English (Editor) "Cost Effectiveness" John Wiley & Sons, Inc, 1968.

第5章　コンピュータ室の運用管理

キーワード

コンピュータ室の運営（computer center management），バッチ処理（batch processing），デマンド処理（demand processing），リアルタイム処理（real-time processing），単一方式（simplex），二重化方式（dual），待機方式（duplex），共有ファイル（shared file），負荷分担（load sharing），資源分配（resource sharing），オープン・ショップ（open shop），クローズド・ショップ（closed shop），データ管理（data management），使用時間管理（time schedule），シフト制（shifts），定期保守（scheduled maintenance, routine maintenance），監査システム（auditing system），使用料金の設定（pricing）

目　　標

ソフトウエア開発部門を除いた，いわゆる，オペレーション部門をコンピュータ室とし，その周辺を含めて必要な知識を修得させる。

コンピュータ室を効率的に運用するためには，コンピュータ・ハードウエアを中心とした管理だけではなく，コンピュータを利用する立場からのシステム運用を検討することが運用の効率化につながることも理解させる。

コンピュータ室の運用方式は，コンピュータ・システムの発展とともに大きく変化してきており，規模や適用業務によっても異なる。この章ではできるだけ共通な事項をとり出して学習し，考え方を理解させる。

内　　容

コンピュータ室には，多種類の機器設備があり順調に稼動するような管理が必要である。また，これらの機器を操作しコンピュータ運用に万全を期すことが要請されている。

コンピュータ室は，次のような機能を受け持っており，これらの管理ポイントについて修得することをねらいとする。

① コンピュータ設備の管理

② コンピュータ・オペレーションの管理

③ 付帯設備の管理

④ データ・エントリーの管理

⑤ コンピュータ出力の管理

⑥ 安全管理

⑦ 保守管理

⑧ ファイルの管理

⑨ 帳票の管理

⑩ 用度品の管理

⑪ 稼動の管理

最近の傾向としては，オペレーション部門をシステム開発部門と切り離したり，コンピュータ・センタとして独立（別会社）させたりするケースが多くなってきた。

コンピュータ室の業務のうち，次のような機能を部分的に外注するケースが増加しつつある。

① データ・エントリー業務……パンチ業者

② 付帯設備（電源・空調）管理……サービス会社

③ オペレーション業務……ファンリティ・マネジメント会社

④ 磁気ファイルの保管……保管会社

⑤ コンピュータ室入出管理……警備会社

⑥ 稼動分析評価……コンサルティング会社

学習の方法としては，コンピュータ・センタの見学が最も望ましい。

5.1 コンピュータ室の運用方式

コンピュータ室の運用方式は，業務処理の形態やコンピュータ設備・組織のあり方などによりさまざまな方式があることを説明する。

大別すると次のような方式がある。

- バッチ方式
- リモート・バッチ方式
- デマンド方式
- オンライン・リアルタイム方式

操作・処理，システム構成によって，次のように分けることができることも説明する。

(1) 操作方式

- オープン方式

  専任のオペレータを用意せず，自由にオペレーションを行う。これは，定型的な業務に

コンピュータを使用しない科学技術計算，テストなどで多く採られている方式である。

・クローズド方式

専任のオペレータを用意し，すべての操作は，オペレータによって行われる方式。

(2) システム構成によるもの

・シングル・プロセシング

・マルチ・プロセシング

・パラレル・プロセシング

・ホスト・プロセシング

(3) オンラインの方式

障害に対する考え方により分類される。

・単一方式（Simplex）

・二重化方式（Dual）

・待機方式（Duplex）

(4) 新しい運用方式

次のような，新しい方式について触れておくことが望ましい。

・コンピュータ・ネットワーク

・コンピュータ・ハイアラーキ

・コンピュータ・フロントエンド

・ファイル・シェア

・ロード・シェア

(5) コンピュータ・ネットワーク

最近のコンピュータ利用形態の一つとしてコンピュータ・ネットワークが注目されているので，予備知識として学習する。

(a) 歴史

・1966年MITリンカーン研究所で，コンピュータ結合の実験。

・1970年IEEE国際会議で言葉が使用される。

ARPAネットワーク（米国）がスタート

・サブネットの形成が通信会社で広がる。

アメリカ……Telenet，Timenet，ATT（Bell）

カナダ……DATAPAC…… 1977～

日本…… DDX…… 1979予定

ヨーロッパ…… EPSS(英国)，TRANSPAC（フランス），EDS（西独）など。

(b) 目的と効果

・リソース・シェアリング

資源の共同利用が最大の目的

・拡張性の高い分散処理システムの実現

・リモート・バッチとTSSの同時利用

・ユーザ間の情報交換，共同利用の推進

・同一機種のロードシェア

(c) 問題点

多くの長所を持つが次のような難点もある。

・プロトコル（接続方式）の設計，複雑なプログラムが必要

・コンピュータ呼出しに時間が掛かる

・保守管理が面倒

・ネットワーク監視センタのような附属システムが必要。

(d) 今後の課題

新しい方式のため，問題点も多く，次のような課題がある。

・プロトコルの設計の世界的標準化がない。

・ソフトウエアの完備

・データ通信コストの低減

・障害箇所の発見

・コンピュータの利用体制・管理体制

5.2 設備管理

コンピュータを運用していくための設備の管理として次のことを説明する。

① コンピュータ装置の管理

② 付帯設備の管理

(1) コンピュータ装置の管理

コンピュータおよびその周辺機器は，常に正しく動作しなければならない。そのためには，日常どのような管理が必要か，故障時にはどのように対処したら良いか学習する。

(a) コンピュータ装置の種類

CPU，メモリ，磁気テープ装置，磁気ディスク装置，ライン・プリンタ・カード・リーダなどの特性を説明する。

(b) 付属装置の種類

・データエントリ装置

カードパンチ，キーツーテープ，キーツーディスク，OCR，OMR，MICR，紙テープ装置など

・データ出力装置

オフライン装置として説明する。

漢字プリンタ，ライン・プリンタ，プロッタ，ノンインパクトプリンタなど。

・端末装置

キャラクタ・ディスプレィ，グラフィック・ディスプレィ，テレタイプ，OMR，OCR，カード・リーダ，プリンタ，バンキング・ターミナルなど

・その他装置

用紙切断機，あて名印刷機，封入封かん機，メール・シーラなど

これらの装置の概要と特性を説明する。

(c) 操作と管理

次の点を説明する。

・操作マニュアルの整備

・教育と訓練

・異常時の処理

・日常の点検・手入れ

・責任者の明確化

操作マニュアル，保守点検教育については，メーカで完備しており，ユーザ側は管理責任者を明確にし，管理業務を遂行することになる。

保守については第4章参照のこと。

(2) 付帯設備の管理

コンピュータ装置を正常に動作させるためには，付帯設備の管理も併せて学習する。

(a) 付帯設備の種類

第4章を参照のこと

・電源装置

・空調装置

(b) 保守点検

通常，サービス会社に委託している。

次のような点を説明する。

・サービス会社の体制

・定期点検の方式

・装置の一覧と履歴

・故障時の記録と報告

・異常時の処置方法

・温・湿度の異常

・停電対策

・夜間の故障対策

・災害時の対策

5.3 データ管理

次のようなデータの管理およびそれに付随する消耗品管理について学習する。

・入力データの管理

・出力データの管理

・ファイルの管理

・帳票の管理

また，各々の説明については，データ生成の装置とその特性についても折り込みながら指導する。

(1) 入力データの管理

コンピュータへのデータ入力については，新しい傾向が見受けられるので，最新の機器とその利用方法について修得する。

(a) データ入力装置

最近では，デンタ・エントリ機器の種類も豊富になり機能面・操作面が一段と向上し，アプリケーションに適合した機器も出現している。この点から，最新の動向を折り込みながら説明する。

① 最近の頃向

・マイクロ・プロセッサの普及により安価な端末機にもインテリジェント機能が付与されている。

・集中型のデータ入力から，原票の発生現場で入力する分散型に移行する傾向がある。

② データ・エントリ・システムの分類

・ダイレクト・データ入力方式

OMR，OCRなど原票の直接読取り。

・キー・デンタ・エントリ方式

キーパンチ，キーツーディスク，キーツーカセット，キャラクタ・ディスプレィ端末機などのオペレータが介入しキー・インによる入力

直接入力……オンライン

媒体を介しての入力……カード，磁気テープ

③ 集中と分散の比較

（集中型）

- バック・アップが容易
- ターン・アラウンド・タイムが大きい
- エラー修正がむずかしい
- ピーク処理の配慮

（分散型）

中間媒体を用いるものとオンラインによる直接入力がある。

- タンアランド・タイムの短縮
- エラー発見・修正が容易
- ピーク処理がない
- 必要情報の入手が可能
- 機器台数が増加する
- 端末機が高価

経済性については，技術革新により低価格化が進んでいる。

④ 導入検討事項

業務の実態に合せて，分散と集中および機器のあり方を決定する。必ずしも標準的なパターンは存在しないことも知って置くこと。

検討事項を列挙する

- 入力目的の明確化
- データ量と変動の様子……データの性質
- 入力から処理までの時間的余裕
- 投入可能なコスト
- 要員
- 期待効果
- 機器にもたせる機能要求
- コンピュータとの接続方式

(b) データの管理方式

次のような点について説明する

① データ受渡し時の管理

- 受付票，受付簿の記帳
- 原票の枚数チェック
- 原票返還の有無
- 読取り（人間・機械）不能時の処置

② 入力データ・チェック方式

プログラム・デザインまたは，原票のデザインにも関係する。

- バランス・チェック
- チェック・ディジット方式
- バッチ・トータル・チェック
- 検孔（ベリファイ）

③ データ作成の管理

- 不良原票の発生率
- データ作成誤り発生率
- データ作成方法
- データ作成者の作業管理

入力データ作成と管理については，次のことを留意する。

表 5.1　入力データ管理上の要点

• 外注業者の選定と評価 外注の利用が一般的であり，コスト面も併せて評価すること。
• 原票のデザイン フォーム・色・サイズについての配慮がデータ作成の効率化に影響する。
• データ入力室の環境 部屋の設計，オペレータの休憩・交替制
• 入力機器の選定 業務への適合性・操作性
• 原票持込み方法 集中化の廃除・枚数の正確性

(2) 出力データの管理

コンピュータ化が進展するにつれて，出力の量の増大はもとより，出力方式も多様化してきている。

ここでは，コンピュータ室での出力の集中的な管理について説明する。

(a) データ出力装置の種類

次のような装置についての機能と特性を説明する。

① ライン・プリンタ装置

最も多く使用されている装置であり，一行当り 132 字～ 160 字の英・数・カナ文字を印字し，分当り 500 ～ 10,000 行以上と速度も多様である。

② 紙テープさん孔装置

出力した紙テープをテレックス，テレタイプ，または，別のオンライン系への入力用に使用する。

- この装置も，コンピュータ直結式のものとオフ・ライン式のものとがある。
- オンライン処理方式の発達により，紙テープを媒体としない傾向がある。

③ COM装置（Computer Output Microfilm）

紙の代替手段としてマイクロ・フィルムに出力するもので，出力を見る場合，フィルム・リーダ装置が必要。

- 大量のもので使用頻度が高くないもの
- 客先への報告の控えなどへの利用
- 金融機関などの元帳への利用

装置・消耗品の費用などから，かなり大量のものを適用しないと割高となる。

④ 漢字プリンタ

最近，その利用が拡大しつつあり，出力の方式も多様である。

- 出力の質と処理速度
- 適用業務とその効果
- 漢字入力方式
- コンピュータの処理方式
- 消耗品のコスト
- 安定性・操作性

などの点を比較検討する。学習方法としては，何社かの装置を比較検討するケース・スタディが望ましい。

表5.2　導入時の考慮事項

・使用する媒体（用紙・紙テープなど）の品質
・消耗品の使用率とコスト
・保守点検の方式
・操作性・安定性
・出力量と処理時間・処理日の分布
・複写と単票の量
・保有コンピュータとの接続関係・能力関係

(b)　データの管理方式

次のような点について説明する

①　データの受渡し

・受渡し方法の決定・ルール

・送付表・納品表の添付

・出力チェック表の完備

②　出力データのチェック

・品質の検査

・出力ページの検査

・コンピュータによるサポート

宛先の出力・コンピュータ出力一覧表の作成。

③　出力装置の管理

・保守点検

・故障の発生率

・故障時の代替手段

表5.3　出力管理上の留意点

・機密保護
重要な出力についての特別な対策とミス出力の処置方法
・外注の取扱い
・用紙不足を生じないための配慮
出力スケジュール・量の把握・連絡
・装置の使用状況の管理
・対外部への出力（レポート）のチェック
コンピュータ室から直接郵送される出力についての検査方法

・出力量の時系列把握

・出力コストの管理

(3) ファイル管理

円滑な運用と保存の安全性確保がポイントであり，次のようなことを説明する。

(a) ファイルの種類

磁気テープ，磁気ディスク・パックを主体とする

・原始データ・ファイル

・スクラッチ・データ・ファイル

・ヒストリカル・データ・ファイル

(b) ファイルの管理方式

自社または保管会社を問わず，一定の保管場所へ整理して保存管理する。専任の係を置くことが望ましい。

① 台帳による管理

・整理番号方式

・業務別方式

・ラベル方式

② ローテーション管理方式

・世代ローテーション

・週間・日間・年間などの期日ローテーション

③ 異常時のバックアップ

重要ファイルについて，異なる場所へ二重に保管する。

④ ファイルの受渡し

受渡しのルールを明確にする。

・入出庫伝票

・入出庫台帳

・保管庫への入出者制限

⑤ 機密保持と安全対策

・保管庫を設け，防火・耐熱構造にする。

・入出者を一定にする

・保管場所の温湿度を一定に保つ

・重要ファイルは，コピーを作り，別の場所（保管会社の利用が多い）へ保管する。

⑥ その他

・媒体の品質管理

・磁気テープのクリーニング，エラー発生度や使用頻度による新しい品との交換。

(4) 帳票の管理

適用業務の増加に比例して使用する帳票の種類と量が増大する。大規模ユーザにおいては，帳票保管室を持ち，専任者が搬入・搬出・整理に従事している。

次のような点を説明する。

① 用紙の管理

・空調の整備

ＯＣＲ用紙・パンチ・カードなど機械で直接読取るものは，保存状態を良好に保つ必要がある。

・帳票フォーム台帳の整備

統一した番号体系をもつこと

② 保管場所と保管ルール

・重要帳票の管理

・帳票入出の方法

・管理台帳とたな卸し

③ 印刷会社との関連

・発注ルール

・緊急的な搬入

・印刷会社の保管

④ その他

・コンピュータ室との関連

・用紙の品質と操作性

・集中保管の床荷重

・管理者の設置

(5) 消耗品の管理

前項(3)，(4)の中でも述べているが，次のことを説明する。

(a) 消耗品の種類

・プリンタ用紙

・紙テープ

・パンチ用カード

・磁気テープ，ディスケット，カセット

・マイクロ・フィルム

・各種リボン

(b) 消耗品の管理

消耗品コスト削減については，業者の選定，在庫管理などに留意する。

・保存方法

・在庫管理（入出庫，棚卸，最低保有量）

・コスト管理

5.4 コンピュータ使用管理

ここでは，狭義の運用管理について次の点を説明する。

・オペレーション管理

・多重処理のオペレーション

・コンピュータ稼動評価

・安全管理

(1) オペレーション管理

(a) 使用計画

コンピュータの運用方式によりオペレーションは異なるので，各方式について次のことを説明する。（運用方式は5.1参照）

① スケジューリング

② 処理方法

・定期的な処理

・臨時的な処理

・時間の割り当て

・優先順位のつけ方

③ コンピュータ使用の申込みと受付け

④ 多重処理に対する考慮

⑤ コンピュータ故障時の処置

(b) 時間管理

次の管理方法を説明する

① オペレーション記録簿による管理

・運行指示者

・運行日誌

・運行予定表

② アカウンティング・プログラムによる管理

各プログラムの実行稼動時間の記録（コンピュータ・ログ・ファイル）をもとにジョブの稼動状態を管理できる。

・稼動時間（Elapsed Time）

・CPU— Time

・I／O使用時間

・入力データ量　　など

③ その他の時間管理

次のような時間についても管理する。

・電源投入時間

・コンピュータ稼動時間

・アイドル時間

・ハンドリング時間

・故障時間

・保守時間

・運転日数

(c) オペレータ管理

専任のオペレータを置く場合は，次の点について計画・管理を行う。

① シフトの方式

・シフトの回数

・シフト当りの人数

・交替時の引き継ぎ方法

② オペレータの教育

・操作手順書の整備

・教育方法の確立

・他部門とのローテーション

③ 障害時の対処

・発見方法

・保守員への連絡

・ユーザへの連絡

・復旧（リカバリ）手続き

(d) 文書化・標準化

次の点を説明する。

- 作業間のコミュニケーション
- 作業の記録保存
- 品質管理
- 操作手順書，指示書
- コンピュータ運用に関する標準化すべきドキュメントとしては，次のようなものがある。
  - ＥＤＰ作業計画表（標準処理時間表）
  - ＥＤＰ作業月次予定表
  - ＥＤＰ作業依頼票（申請書）
  - ＥＤＰ作業月報
  - ＥＤＰ作業管理表
  - インプット（入力データ）送付票
  - せん孔作業指示票
  - アウトプット送付票
  - オペレーション指示票（オペレーション・マニュアル）
  - 障害記録表，報告書
  - ラン・プロセスチャート
  - デバッグ（クローズ）作業依頼書
  - デバッグ受付簿
  - 磁気ファイル取扱い基準
  - ライブラリ取扱い基準
  - 制御カード取扱い基準
  - コンピュータ稼動状況記録
  - 機器操作法規定
  - 機械室環境管理基準
  - 用品管理基準
  - 障害対策手続き規定

(e) オペレーションの効率化

オペレーションを円滑に遂行するためには，管理上の問題もあるが，アプリケーション・ソフトウエアの作り方，コンピュータ・ハードウエア構成が大きく関係する。

表5.4　オペレーション効率化への考慮事項

① ソフトウエア作成面
- コンソールによる指示を少なくする。
- 磁気テープへのオート・ラベリング機能の採用
- ラン（ジョブ）におけるタスクの連結
- 磁気テープ使用台数の制限
- メモリ量の使用制限

② オペレーション管理ソフトウエアの使用
- ラン・ジョブ管理ソフトウエア
- リソース管理ソフトウエア
- 磁気ファイル入出庫指示票（表）作成ソフトウエア
- 出力宛先管理ソフトウエア　　など

③ ハードウエア構成面
- 磁気ディスク・ドラムの多用
- 高密度磁気テープ装置の採用
- 機器レイアウト

(2) 多重処理の運用

多重処理（multiprocessing）は，コンピュータ資源の有効利用方法として，一般的な方式となっている。

大型コンピュータでは，バッチ処理，リモート・バッチ処理，タイム・シェアリング処理，オンライン・リアルタイム処理を同時並行的に，かつ，それぞれの処理を多重的に運用している例が多く見受けられる。

ここでは，定常的・定量的なバッチ処理運用は除き，オンライン・リアルタイムおよびタイム・シェアリング処理についての運用管理上の留意点を中心に学習する。

(a) オンライン・リアルタイム処理の運用

オンライン・リアルタイムの特性を中心に次のような点を説明する
- データのピークの傾向
- ＣＰＵのボトル・ネックと最大処理能力
- 端末機の状況監視と管理
- 回線状況の把握
- 並行的に運行するバッチ処理への影響度
- 障害時の対処

・操作ミスの防止策

・ドキュメンテーション

(b) タイム・シェアリング処理の運用

タイム・シェアリングの運用の前提として，次のような特性を把握する。

① ユーザが任意に使用するため，コンピュータへの負荷が流動的である。

② ユーザが不特定多数である。

このようなことから，次のような問題が発生する。

・並行して処理しているバッチ処理の時間が，2～3倍掛かることがある。

・約束のターン・アランド・タイムが守れなくなり，クレームがつくことがある。

・保存すべきデータやプログラムの消失が発生することがある。

これらのことから，安定運用のための基本事項として次のようなことを設定しておく。

① 利用範囲の設定

② 運用体制と諸手続設定方法の明確化

③ 運用と管理に必要な諸手続業務の役割分担

④ 利用コストに対する考え方

タイム・シェアリング処理のためのリモート使用開放のための注意事項としては，次のことがあげられる。

・プログラムとデータ使用の事前連絡

・ジョブ名（プログラム，アカウント）の規約を守る。

・定常業務の事前連絡

・リソースの利用限度の決定と事前通告

・実行優先度

・時間帯の指定

・プログラム，データ保管のルール

(3) コンピュータ稼動分析評価

単元D2，第2章および第4章で述べたコンピュータの性能評価は，導入に際してのコンピュータ・システムの能力評価であったが，ここでは，導入後の稼動評価について説明する。

評価の目的としては，コンピュータ資源（リソース）の利用状況を知ることによって

・ハードウエア構成の改善勧告

・プログラムの改善勧告

・使用スケジューリングの改善

が可能となる。

稼動評価は，一時的に実施しても効果は少なく，継続的に実施することに意味がある。

バッチ・システムとリアルタイム・システムの評価について分けて説明する。

(a) バッチ・システムの評価

① 評価のポイント

- 単位時間当りのジョブ処理件数
- ターン・アラウンド・タイム
- 機器構成や運用形態で変化する
- 処理能力は多重プログラミングの多重度にかなり依存する。

② 評価の方法

- 実行計測を行う

  ハードウエア測定機によるデータ，ソフトウエア・モニタによるデータ

- シミュレーション・モデルを作り予測する方法。

  新しく投入するランの稼動予測のツールとして用意し，大きなシステムを建設する場合に利用する。

(b) オンライン・リアルタイム・システムの評価

バッチ処理と異なる点を説明する。

- データのピーク時の処理能力
- ターン・アランド・タイムの予測
- トランザクション別およびトランザクション・ミックス（標準的組合せ）の処理能力
- トランザクション待ち行列（Quing）状況
- ファイル使用状況

(c) 処理能力向上のポイント

概略，次のようなことが言える。

- システム構成要素のバランスがとれていること。
- ＣＰＵの能力が特に重要であること。
- バッチ処理は，入力・処理・出力の各プロセスのバランスがとれていること。

結論的には，システム構成の改善をはかっていく場合，コスト対効果の観点から

- Ｉ／Ｏの改善
- チャネルの改善
- メモリの改善

を行う。

またＣＰＵについては，年々低コスト化の傾向があり，機種のリプレースによるコ

スト・パフォーマンス向上も考えられる。

(4) 安全管理

コンピュータ室の安全管理については，コンピュータの普及により問題意識が高まりつつある。また，一方において，コンピュータによるトラブルが社会的に大きな混乱を引き起すケースもある。

ここでは，安全管理のポイントとして次のような点を説明する。

(a) 災害の分類

人的なものとそれ以外に大別される。

① 人的災害

日本では発生率は低い（JECC調査では15％），米国では50％である。

- ソフトウエアによる事故防止
- 出入室管理の徹底

カラー識別バッチ

ガード・マンによるチェック

ＩＤカードによる扉の自動開閉

② 非人的災害

―火災によるもの―

- コンピュータ本体からの出火は少ない
- フリーアクセス床にする
- 用紙類はコンピュータ室に置かない
- 電源装置はコンピュータ本体とできるだけ離して別室に置く。
- 喫煙室を別室にする。

―火災以外の二次被害―

- 水もれ
- 消化剤
- 腐食性ガス

対策としては，次のようなものがあげられる。

- 自動火災探知機・煙探知機
- ハロン消火システム・二酸化炭素消化システム
- ビル建材の選定

③ 地震

- ビルの構造

- 耐震性床材
- コンピュータ機器の据え付け方
- コンピュータの分散設置

(b) 安全対策の実施状況

- 安全対策については，限界がないので，企業によって，その取り組み方がまちまちである。
- 大規模なコンピュータ・センタ，または，情報処理専門のセンタなどは，安全対策面の配慮が行き届いているので参考になろう。
- 実施状況については，ＥＤＰユーザ団体連合会による「ＥＤＰユーザの電算室実態調査」が昭和52年にまとまっており，この調査内容が参考になる。

以下，そのポイントについて列挙する。

① 火災

警備員の巡回，消火器の設置をはじめとする基本的対策は90％以上が実施している。完全実施は70％と高率。

② 出水

排水孔の設置は，必要性があまり重視されていない。

③ 空調

独立空調，定期点検は実施率が高いが予備空調，クーリング・タワーへの防災対策への関心は低い。

④ 電源

日本の場合，電気の安定供給が維持されているため実施率は低い。

⑤ 天災

地震対策としての建築への関心は高いが機器の移動・転倒防止は軽視されている。

⑥ 入室管理

不十分な答えが多く，データ管理場所への入室は66％が不十分であるとしている。

⑦ 記録媒体の保管

- 非常持出基準の設定
- コピー，バックアップ・ファイルの保持・分散
- 防火・磁気遮蔽容器の使用など完全実施は10％と低い。

⑧ 組織・訓練

組織化と命令系統の明確化はほぼ実施されている。

⑨ 保険

情報化保険の加入について設問しており，半数以上が必要性を認めながら未着手であり，最も実施率が低い。

⑩ バック・アップ計画

- バックアップ用コンピュータありが12％
- バックアップ契約ありが11％と低く未着手の企業が多い。

⑪ その他

次のような点についての対策は，ほとんど実施されていない。

- ドキュメントの副本保持
- 安全対策方針の確立
- 火災時手順書の常備
- 棚・キャビネットの移動・落下防止施策
- コンピュータ室内緊急用電灯の設置
- 非常持出の設定

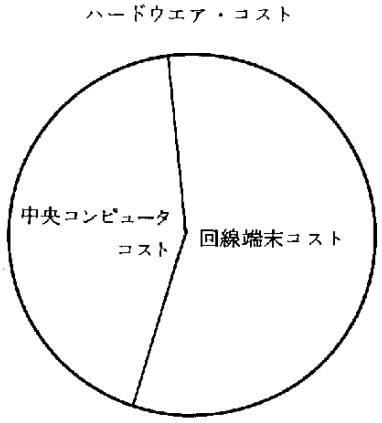
5.5 コンピュータ・コスト賦課

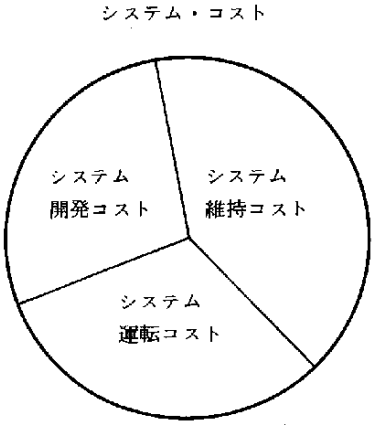
コンピュータ・サービス・コストのユーザ部門への配分は，コンピュータ部門のコスト回収と，コンピュータ・リソースの使用量の割当とを目的として行なわれる。

費用配分には，決定的な回答はなく，いくつかの方法はあっても，それぞれに欠陥があることを知っておく。

(1) コストの内訳

コンピュータ・コストは，ハードウエア・コストとシステム・コストに大別される。





次のような費用があげられる。

① プログラム開発費

・基本分析料，附加分析料，基本プログラム料，附加プログラム料，準備作業料または，人件費，コンピュータ使用料，パンチ料，消耗品

② コンピュータ費（直接費）

コンピュータ・ハードウエアについては，買取制，リース制，レンタル制によって費用が異なるが，通常レンタル換算によって計算される。

CPU，メモリ，磁気ファイル，I／Oデバイス，端末，回線，消耗品

③ コンピュータ費（間接費）

・設備コスト

電源，空調，建物

・運転費用

消耗品，電力料，水道料

・運用管理の経費

人件費および経費

(2) 使用料金の設定方法

ここでは，

・専用機械の使用

・特定のプロジェクトに割り当てた費用を除き，コンピュータ，リソースの共用についてのコスト賦課を学習する。

学習の方法としては，計算センタでジョブを共同利用すると仮定しての，コストに係る問題をクリアする学習も有効であろう。

料金の設定方法のいくつかをあげる。

① ジョブ優先度による料金設定

ジョブの優先度によって標準料金を中心にして，料金差をつける。

特急は30％増，ゆっくりは20％引きなどの格差。

このケースは，共同計算センタをタイムシェアリングで使用するような場合に有効である。効果としては，

・夜間，週末などの利用が促進される。

・ユーザがプログラム作成上，効率を考える。

② 時間帯によっての料金設定

センタが供給できる時間と需要とによって価格を決定する。

・マルチ・プログラミング・モード

・優先度の低い処理

は低い料金を設定する。

③ 均一料金制

全経費を，そのリソースが使用されると予想される時間数，あるいは実際に使用された時間数で割り振る。

④ シフト料金制

③の変型として，シフトによって料金を変える。この方式は，ピークをある程度平準化できる作用をする。

(3) コスト賦課の長所・短所

コスト賦課を考える場合，次のような長所と短所を把握しておく。

(a) 長所

① データ処理サービスのコストと質を知ることにより費用低減の検討ができる。

② コンピュータ部門は，外部の情報処理サービス会社との費用比較により，コスト低減の刺激を与えられる。

③ リソースのより有効な利用を推進する働きがある。

④ コンピュータ使用の無制限な増大に歯止めがかかる。

(b) 短所

① コスト賦課システムの管理それ自体にコストがかかる。

② 費用を公正に分担させることが困難。とくに，同一コンピュータ出力が多数のユーザに使用される場合の問題

③ システムによる質的な効果をコスト対比しにくい。

④ ユーザからの革新的または実務的なシステム開発ニーズ意欲が減少する。

(4) 平均コストによる配賦による問題

比較的多くとられている方式として，総コストを請求可能な時間数で割る平均コストによる配賦がある。しかし，次のような欠点があることに注意する。

・時間による価格差がないので，ピーク負荷を大きくする。

・サービスの需要と供給によって価格が一定しない。

・過剰な能力を持つリソースによるユーザ負担の増大

・コスト面からのシステム（プログラム）改善の基準がない。

5.6 ＥＤＰ監査

監査を大別すると次のようになる。

① 内部監査

- 経理手続きの公正さの証明
- 企業の社会的責任
- 株主に対する責任
- 内部統制のための業務監査

② 外部監査

- 経理手続きに不正はないかの監査
- 監査結果の財務報告書への記載
- 通常，内部監査人にコンピュータ監査人が存在する。最近は，金融機関（全部ではない）のＥＤＰ監査がその公共性から法により義務づけられている。
- 米国においては，1973年Equity Funding社の事件が契機となり，コンピュータ内部処理の公正性の監査（Auditing Through Computer方式）の必要性が認識され，コンピュータ監査人が登場することとなった。

(1) 監査のポイント

① 稼動前監査……最も重要視

業務システムの設計における統制・監査上の欠陥を見出す。

- 予備設計のレビュ

  目的，機能，実施計画，内容の概要

- 詳細設計のレビュ

  セキュリティ（安全対策・機密保護），リカバリの仕様

- 監査証跡
- セキュリティ
- リカバリ

  内部統制，監査面からの設計仕様の承認

② 稼動後の監査

仕様書通りになっているかの確認

③ コンピュータ部門の監査

ＥＤＰ部門の内部統制の評価

- 統制手順
- 安全性

- 機密保護対策
- リカバリ対策
- 文書化（ドキュメンテーション）
- プログラミング技法

表5.5　ＥＤＰ監査の対象項目

項目		特記事項
システム開発	(1) フィージビリティスタディ	① もっともらしい仮定について，その内容をチェックする。 ② コスト・ベネフィット分析。便益をしっかり把握したうえで当該プロジェクトのシステム開発をスタートさせる。 ③ リソース，ツールについて過大，過小評価はないかチェックする。
	(2) システムデザイン	④ コントロール，監査可能性が組み込まれているか。これらの事項を組み込むことによりシステム完成の時期が延びるのであれば，そのことについてトップを説得する。
	(3) プログラムデザイン	① プログラムの分離（テスト・バーション／プロダクション・バーション）
	(4) プログラムテスト	データの分離（テスト・データ／ライブ・データ）
	(5) システムテスト	① 設計目標と対比して，システムの効果をチェックする。 ② オペレーションに入り得る確信を得るまでパラレル・ランを行なう。
	(6) 移行	① 旧システムのライブ・データが滅失しないようにする。 ② コンバージョン・システムとして，そのデザインを十分コントロールする。
組織		① ＤＰ機能を他の部門から独立して組織する。 ② 責任の分離 ・プログラマ，システム・アナリストとオペレータの分離 ・ユーザとオペレータの間に立ちＩ－Ｏをコントロールし，例外，ルール外れを解決するコントロール・グループを設ける。 ③ 教育，訓練を通して，個々人の誠実性を強調する。 ④ プログラム修正，システム変更にたいして正式な承認制度を作り文書化する。
入力		① データ・インプットに関する現行手続が守られているか検証する。 ② インプット内容についてユーザが責任をもつようにエラーの内容をフィードバックする。 ③ 重要項目について dual field を考慮する。 ④ オンラインでは，一切のインプットをシーケンスにテープ・ログする。
出力		① 一切のエラーを報告させ，その理由を検閲する。 ② 正当な担当者のみにレポートが配付されるようにする。 ③ 未承認の者がコード内容を知り得ないようにする。
プログラム		① 件数チェック，データの論理チェック，レーンジチェック ② エラー処理用のアウトプットとともに，エラー・コントロール・リストをアウト・プットし，ミス対策に役立てる。 ③ 再作業を考慮してプログラムのユニットを決める。

項　　目	特　　記　　事　　項
オペレーション	④ プログラムの複製を保有し，現に使用されているプログラムと時々比較する。 ⑤ アウトプットにジョブ制御ステートメントを含ませ，未承認プログラムの使用を回避する。
データ機密保護	① ソフトおよびハードのメモリー保護を明確にする。 ② ライブラリー・システムを作る。消磁したのちワークに供するファイルを決める。 ③ 端末からキーインされたパスワードは端末，センターともアウトプットしないようにする。
安全対策	① ガラス張りのショーケース的コンピュータ室は再考する。 ② 災害時の行動計画は，定期的に予行演習を行なう。 ③ 重要なものを見極わめる。 ④ 規定用紙，運搬具など，付属物件視されがちなものはないか。
ハードウエア	① パリティ・ビット，リード・アフタ・ライト・チェック，サーキュラー診断ルーチン
通　　信	① ターミナルとユーザの識別，パスワード，ログ ② 暗号パッケージの採用 ③ 専用回線に対して，ダイアル回線をバックアップ

(2) 監査の現状

監査の必要性は，どの企業でも十分認めてはいるものの，内部監査としては内部組織の確立，専門要員の育成，プログラムへの配慮，監査ツールの整備など必要であり，企業によってバラツキが見受けられる。

この面を補足するため，外部監査会社の利用が促進されつつあるものの，ＥＤＰ部門自らの経営への問題提起，問題意識と現実的な解決が先行すべき点である。

監査についての学習は，銀行のＥＤＰ監査の考え方，実施方法などをモデルとして行うことが望ましい。

指導上の留意点

(1) コンピュータ室の運用管理は，さまざまな方法がとられているので，できるだけ具体例を示しながら考え方や方法を指導するとよい。

(2) 設備管理については，実例を紹介し，可能であれば，代表的な計算センタの見学を折り込み具体的な管理の仕方を学ぶよう指導することが有効である。

(3) コンピュータ室の安全管理，ＥＤＰ監査は，比較的新しいテーマであり，現状と照らし合せながら今後のあり方として指導していくとよい。

参考文献

〔１〕南条　優，保坂岩男共著，「システム開発と運用」　オーム社，1972.

〔２〕岡本行二，「コンピュータのための機密保護と安全管理」　オーム社，1974.

〔３〕南条　優，「コンピュータ室の運用と管理」　日本生産性本部，1971.

〔４〕服部幸英，「コンピュータ室の運用管理」　情報処理研修センタ，1977.

〔５〕日本公認会計士協会，「監査資料集」　第１法規出版，1975.

〔６〕財団法人日本情報処理開発協会，「システム監査体制確立への道」　1977.

〔７〕「コンピュータ・システムの効率評価に関する研究」　社団法人行政情報システム研究所，1975.

〔８〕ロバート，C．コールドスタイン，佐野芳男（訳）「プライバシー保護－費用とその分析－」　企画センタ，1977

〔９〕斉藤正治，「ＴＳＳの計画と運営」　ＴＢＳ出版会，1977.

〔10〕藤井　一，「誰にでもわかるーオフィス・コンピュータ」　日本経営出版，1977.

〔11〕谷村外志男，「コンピュータの運用管理」　日刊工業新聞社，1972.

〔12〕日本情報処理開発センタ，「コンピュータ導入活用指針」　ＪＩＰＤＥＣ，1972.

〔13〕土岐秀雄，「電子計算機／機械室管理」　日本経営出版会，1967.

〔14〕日本ＩＢＭ，「ＡＣＣＥＳＳ」，1978.3

〔15〕情報処理学会，「情報処理」　Vol. 16，№12，（1975）
Vol. 19，№ 2，（1978）

〔16〕N．エブラムソン，F.F．クオー（編），猪瀬　博（訳）「コンピュータ・ネットワーク」　産業図書，1975.

〔17〕工藤章次，「ＯＣＲの導入と活用」　日刊工業新聞社，1972.

〔18〕岩尾達男，「コンピュータ・エラー」　ＥＤＰ入門双書(7)，創元社，1974.

〔19〕S．スティムラー，大野　豊（訳）「コスト／パフオーマンス」　日本経営出版会，1972.

〔20〕J．マーチン，北原安定（訳）「リアルタイム」　日本経営出版会，1968

〔21〕太田文平，「経営の現代化とコンピュータリゼーション」　丸善，1972.

〔22〕杉山　光，加藤仁史，松崎　務（共著）「ＣＯＭシステム」　日刊工業新聞社，1971.

〔23〕松平　誠，「データ通信入門」　日本経営出版会，1970.

〔24〕ハリレ，L．ブラウン，富島一夫（訳）「ＥＤＰ監査」　中央経済社，1971.

〔25〕伏見　章（編著）　「ＥＤＰ監査の実際」　中央経済社，1971.

〔26〕森　　実，大矢知浩司，「コンピュータ監査の実際」　白桃書房，1974.

〔27〕通産省機械情報産業局　情報処理振興課，「電子計算機システム安全対策基準」 1977.

第6章　情報処理サービス業の利用

キーワード

情報提供サービス（informing services），情報処理サービス（information processing services），情報検索サービス（information retrieval services），ファシリティ・マネジメント・サービス（facilities managemant services），コンサルタント(consultant)，ソフトウエア・パッケージ（software package），データ・エントリ・サービス（data entry services），情報通信サービス（telecommunication services）

目　　標

この章では，情報処理サービス業の概要を説明することによって，その効果的利用が企業にとって利益になり，また，情報処理部門がかかえている顕在的，潜在的な諸問題を解決する手段であることを学習する。

情報処理サービス業の利用形態は，情報処理機能の部分的な外注から全面的利用まで多岐にわたっている。各企業のコンピュータ処理が拡大と複雑化の進展により，情報処理部門と経営組織体との不適合性の問題が発生しつつあり，かつ，機能が細分化する傾向にある。このことは，情報処理サービス業利用増加の要因ともなっている。

この章では，サービス業利用の必要性・有効性と現状について学習することを目標とする。

内　　容

情報処理サービス業の業務内容は，概略次の通りである。

- システムの設計
- ソフトウエアの開発・販売
- 情報処理・情報提供
- ファシリティ・マネジメント
- マシンタイム販売
- 入出力データの作成
- 教育・講習等
- コンサルティング

ユーザの利用形態は，全面委託するものから，部分的または組合せがあり多様である。

この章では，次のことを学習する。

① サービス業利用の必要性と効果

② サービス業の種類と業務内容

③ サービス業の現状

④ サービス業の活用と今後の方向

6.1 情報処理サービスの必要性

第1章において，コンピュータ部門の形態はまちまちであり，また，部門におけるいくつかの問題をかかえていることを学習した。

わが国のコンピュータ導入組織体のほとんどは，システム開発，プログラム作成，オペレーションおよび付帯作業など機能的に異なるものを自社内で実施してきた。しかしながら，次のような問題が顕在化してきている。

(1) コンピュータ部門における問題点

① 経営組織上，部門の地位が確立されておらず，同部門が企業の生命部門であるという認識が少ない。

② 要員に特別なスキルが要求され，長い教育期間を必要とするため，他部内とのローテーションが困難となる。

③ コンピュータ室，いわゆるオペレーション部門を担当している要員のモラル低下と不満の存在。

④ コンピュータの稼動時間延長による労働条件の悪化，要員確保の困難性。

⑤ 付帯的業務に従事している要員は，他部門へローテーションした場合，コンピュータ部門の経験が生かせない。

⑥ 安全対策等への配慮がコスト面などから十分に実施できない。

⑦ 経営からの情報処理コスト引下げ要請に対し，自社内では限界がある。

(2) 問題解決策

これらの問題点を解決する方法として，情報処理サービス業の部分的または，全面的利用であるが，その他，次のような解決方法法があることも説明する。

・情報処理部門を独立会社として発足させる。

・外部の専門業者と合弁会社を設立する。

・関連会社数社のコンピュータ室をまとめて，グループ計算センタとして発足させる。

(3) 情報処理サービス業利用の効果

情報処理サービス業は，コンピュータ資源および技術力など共同有効利用の場をユーザ

に提供することによって，コンピュータ業務の効率的な運営をはかることができる。

利用の効果を列挙すると，次のようになるので，利用の場合は事前にチエックするポイントともいえる。

(a) 信頼性の確保

データと処理機能の安全確保のためには，設備面，要員に関する配慮を慎重に検討し，かつ，実施しなければならない。また，そのためには，高いコストを必要とすることは，前章で学んだ。

サービス業は，専門的公的立場から，安全確保について

- センターへの入出館チェック
- データの保護設備
- 管理体制
- 災害時のバック・アップ
- 防災・防犯

などに十分な対策を講じているため，安心して利用できる。

(b) 技術力の活用

専門企業として蓄積されているノウハウを利用できる。

- 高度に専門化した各分野の情報技術者の経験と知識の利用
- 高レベルのソフトウエアの利用
- コンピュータ室運営管理のノウハウの利用

(c) 経済性の向上

利用する場合，最も多い理由となっているもので，次の理由があげられる。

- コンピュータの規格にもよるが，自社導入の場合，設備のための初期費用と運転費用などの経費がレンタル料の約3倍必要であるといわれている。
- 要員育成のための費用が不要
- 高性能のコンピュータと豊富な汎用ソフトウエアの共同利用による経費削減

(d) 迅速性の確保

- 自社導入と比較して，コンピュータを導入して稼動させるまでの時間的制約の排除
- 通信回線ネットワークサービスおよびそれに付随するタイムシェアリング・サービスを迅速化
- 量的・質的なオーバーフローの対処の迅速化

上記のような，種々のメリットがあげられ，その利用も拡大しているものの，

- 情報処理業に対する認識不足

・情報処理業界自身ＰＲ不足

・ユーザとなり得る経営およびコンピュータ部門の意識

・情報処理業自体の歴史の浅さと経営基盤の確立

などから，その活用は，十分とは言えない。このような点から利用されない理由についても討議し，情報処理サービス業について理解を深める学習方法も有効であろう。

6.2 情報処理サービス業の種類と業務内容

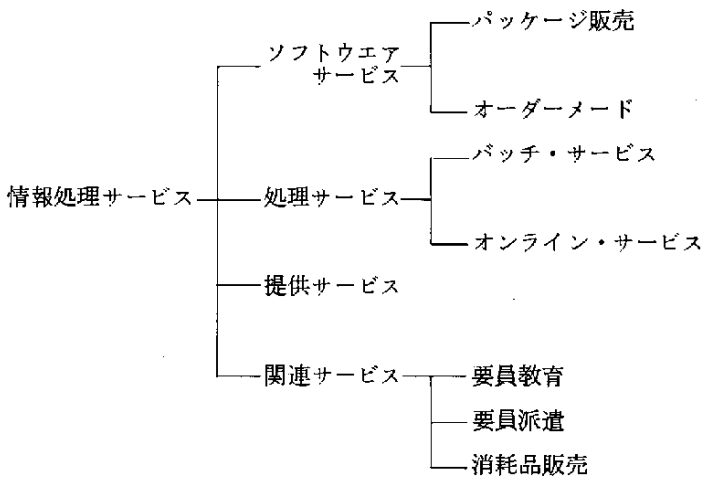
情報産業は大別すると，コンピュータ産業と情報処理産業に分類される。

本章においては，情報処理産業を広義にとらえ，情報処理サービス業として説明する。

(1) 情報処理サービス業の分類

ここでは，次のように分けて説明する。

表 6.1　情報処理サービス業の構成



行政管理庁「日本標準産業分類」によれば，次のように分類されているので参考とする。

情報サービス業（851）

・ソフトウエア業（8511）

・情報処理サービス業（8512）

・情報提供サービス業（8513）

・その他の情報サービス業（8519）

(2) 情報処理サービス業の業務内容

表 6.1 にそって説明する。

(a) ソフトウエア・サービス

コンピュータ・ソフトウエアの開発および開発に関し，調査，分析，助言などを行う。次のようなサービスに分けられる。

- 情報システムの開発
- プログラムの作成
- ソフトウエア作成のコンサルタント
- ソフトウエア要員の派遣
- ソフトウエア・パッケージの開発・販売

(b) 処理サービス

コンピュータなどを用いて委託された計算サービス，データ・エントリサービスなどを行う。

- 委託計算サービス
- 計算センタ
- タイムシェアリング・サービス
- マシンタイム・サービス
- データエントリ・サービス
- 出力処理サービス

この処理サービスが，狭義の情報処理サービス業と呼ばれている。

(c) 情報提供サービス

各種のデータを収集加工・蓄積し，情報として提供するサービスである。

日本標準産業分類によれば，次のようになっている。

- 不動産情報提供サービス
- 交通運輸情報提供サービス
- 気象情報提供サービス
- 科学技術情報提供サービス

(d) 関連サービス

情報処理に関連したサービスとして，次のようなものがある。

- 要員派遣

ソフトウエア開発，オペレーション，データ・エントリーなど，情報処理要員の派遣

- 要員教育
- 磁気ファイルの保管
- データ配送サービス
- 消耗品・備品販売

磁気ファイル，帳票，プリンタ・リボン，キャビネット類など。

・磁気テープ・クリーニング・サービス

・マイクロ・フィルム作成

6.3 情報処理サービス業の現状

わが国の情報処理サービス業の現状をいくつかの角度から学習する。

情報処理サービス業は，ソフト企業と委託計算企業に分けられ，また，前節6.2のように分類される。しかし，企業の実態をみると，次のように多角的である。

・専業的形態から兼業的形態へ

・本格的ＥＤＰＳを設置して，ソフトウエア開発から受託計算までを総合的に取扱っている企業

・ミニコンや端末機の販売をシステム込みで行うもの。

・ＥＤＰＳを持たずにソフトウエア開発を行うもの。

・オペレーションのみ委託する企業

・磁気ファイルの保管を専業とする企業

(1) 情報処理サービス業設立の動機

サービス業の設立動機と経緯を分類すると，次のように種々のものがある。

・大企業の計算部門が分離されたもので，一部，余力で一般企業からも委託しているもの

・独立の計算委託，パンチ作業の委託企業，ソフトウエア開発など

・会計上，税理士が顧問会社を主体に，専門知識を活用して組織化したもの

・電子計算メーカ，販売会社の一部門としてのセンタ

・同業種の中小企業が集まって，共同利用を目的としたもの

・地方自治体，公企業，大学，研究団体等の計算センタ

・企業グループの共同利用を目的としたもの

(2) 業界の概況

昭和52年１月日本電子計算機（株），－ＪＥＣＣ－の調査をもとに概況を説明する。

日本の情報処理産業の成立は，まだ日が浅いが，昭和48年に比べて，企業数が541企業から900企業に増加しているごとく急速に発展し，その業務内容も整備されつつある。

① 経営主体別センタ分布

表6.2のごとく，900事業体，1,505センタとなっている。

表 6.2　経営主体別センタ分布状況

	事業体数	センタ数
独立計算委託企業	788	1, 212
公　共　機　関	55	63
メーカーおよび関係販売会社	57	230
計	900	1, 505

また，業務別内訳は，次の通り

・ＥＤＰＳを設置　　24.9％

・ソフトウエア　　12.3％

・関連作業　　62.8％

② 規模による分布状況

従業員，資本金，売上高の規模別の分布状況を独立計算委託企業601社の範囲でみると次の通りである。

表 6.3　従業員規模別企業分布

1 ～ 9人	12.3％
10 ～ 19	14.0
20 ～ 49	29.6
50 ～ 99	23.6
100～ 199	11.8
200～ 299	3.3
300以上	5.3

表 6.4　資本金規模別企業数分布

100 ～ 499万円	23.6％
500 ～ 999	22.6
1,000 ～ 4,999	40.1
4,999 ～ 1億未満	6.7
1億円以上	7.0

表 6.5　売上高規模別分布

5,000万円未満	20.4％
5,000 ～ 1億円	18.1
1億～ 5億円未満	43.8
5億～ 10億円未満	8.5
10億円以上	9.2

③ 主要企業の売上げ

わが国の情報処理サービス企業は，通産省が発行している「情報処理サービス企業等台帳総覧」により，その内容は，各地方通産局管内ごとに分冊にまとめられており，広く閲覧に供されている。下記の表は，その中からわが国の主要情報処理サービス企業として，売上げ高の10位までをピック・アップしてみた。台帳は，申告制になっており，昭和52年10月1日現在で申告されたものである。

表 6.6　主要企業の売上げ　　(52. 10. 1現在)

企　業　名	売上高（百万円）	技術者数（人）
日本ビジネスコンサルタント	16,100	1,487
三井情報開発	7,516	365
野村コンピュータシステム	6,700	329
日本電子計算	6,444	819
日本電気情報サービス	5,940	230
日本情報サービス	4,800	494
日立ソフトウエアエンジニアリング	4,795	1,090
ティ・ケイ・シー	4,650	314
インテック	4,550	830
センチュリリサーチセンター	4,440	600

④ 売上げ高の構成比

ＪＥＣＣの調査によれば，1,276事業所数についての昭和51年度の構成は，表6.7のようになっている。

表 6.7　年間売上高構成比

事業所数	1,276
売上高合計	306,969百万円
事務計算	34.5%
その他計算	4.1
ソフト開発・プログラム作成	15.3
カードパンチ	13.9
マシンタイム販売	5.3
要員派遣	10.6
情報提供	3.9
各種調査	8.2
その他	4.2

(3) 情報提供サービスの状況

情報提供サービスは，データの収集，蓄積に膨大な投資を必要とするうえに，

・技術的な困難性がある

・ユーザ側の対価支払意識不足

などから，米国に比較して，まだ十分な発展をみるに至っていない。

わが国の提供サービス機関を列挙するので参考とすること。

① 公的サービス

情報の検査および印刷物としての提供サービス，コンピュータ出力サービスがある。

・日本科学技術情報センタ

・労働省，労働市場センタ

・日本特許情報センタ

・加工情報センタ

・日本貿易振興会

・世界経済情報サービス

・日本科学技術振興財団

・物性データバンク

② 民間のサービス

サービス企業の数は少ないが，次のようなサービスが提供されている。

・NEEDS

日本経済新聞社による総合経済情報提供システムであり，海外の情報提供会社のデータも取り入れている。

・野村総合研究所

国内主要企業の財務データや株価情報の提供。

・市況情報センタ

Quickという呼び名で，オンラインリアルタイムによる証券取引所の株価情報提供を中心としたシステムであり，1976年9月現在でCRTディスプレイは4,700台に達した大規模システム。

・帝国興信所

企業調査にもとづいた企業情報ファイルのサービス

・国際医学情報センタ

医学・医療に関する資料提供サービス

(4) 情報通信サービスの状況

情報処理または情報提供をデータ通信により行う情報通信サービスは，

・電電公社

・国際電電

・民間企業

により営まれている。

① 電電公社のサービス

・販売在庫管理システム（DRESS）

1976年末現在で，ユーザ数 972，端末数 3,190 となっている。主として中小企業が利用している。

・科学技術計算サービス（DEMOS）

商用ＴＳＳによる計算サービスで，1976年末ユーザ数は 921，端末数 1,202 となっている。

・各種システム・サービス

ユーザの求めに応じて行うサービスで，オーダーメイド型といえる。1976年末で6システムのサービスがある。

② 国際電電のサービス

国際オートメックスサービスと呼ばれる国際間通信が中心であり，1976年末のユーザ数は14，端末数は19となっている。

③ 民間のサービス

1976年末の郵政省の調査によれば，企業数は54社となっており，80システムのサービスで，端末数は，1システム当り平均128台となっている。

適用業務の内容は，次のようなものがあげられる（郵政省調査昭52. 3）

サービスの多い順に並らべる。

・販売管理……49.4%

・在庫管理……48.1

・科学技術計算……36.1

・経理業務……35.4

・統計計算……31.6

・生産管理……30.4

・給与計算……30.4

・財務計算……30.4

・資料作成……24.1

・運送管理……21.5

・原価計算……20.3

・メッセッジ……19.0

・人事情報検査……16.5

・預金業務……15.2

6.4 情報処理サービス業の活用

情報処理サービス業の業務内容は多様であり，利用方法も自社にコンピュータの導入およびコンピュータ部門を保持しないで全面委託をするようなケースから，パンチの一部分を委託するようなケースまで，バラエティに富んでいる。

ここでは，どのような面で利用し，また利用上どんな点を留意したら良いかを学習する。

(1) サービスの利用形態

概略，次のような業務を情報処理サービスとして利用している。

(a) 業務内容による利用

・要員派遣を受ける

オペレータ，ＳＥ，プログラマ，キーパンチャ

・ファシリティ・マネジメント・サービスの委託

・ソフトウエア製作

定常的またはスポット的なもの

・パンチ業務委託

・特殊出力処理の委託

漢字処理，ＣＯＭなどの作成

・ソフトウエア・パッケージの購入，借入

・コンピュータの時間借り

・事務処理計算の委託

・科学技術計算の委託

・オンラインまたはＴＳＳの利用

・磁気ファイルの保管委託

・情報提供サービスを受ける

・コンサルティングを受ける

・要員教育を委託する

・情報データファイルの購入

(b) 適用分野による利用

情報処理センタは，巾広く適用業務分野を拡大しつつあり，共同で利用できる各種のシステムおよびソフトウエアを保有している。

そのため，業務のシステム化のノウハウおよびソフトウエアの共同利用によるコスト・ダウンが期待できる。

適用分野の概略は，次の通り。

① 経営・事務分野

- 人事関連……給与，社内預金，社会保険等
- 販売……顧客管理，販売統計，在庫管理
- 財務・経理……一般会計，予算管理，原価管理，借入金管理，資金管理
- 購買……発注管理，資材管理，買掛管理
- 総務……株式業務，固定資産管理
- 生産……生産管理，工程管理，品質管理
- 経営……経営管理，経営分析，投資分析，需要予測
- 自治体業務……税務業務，住民記録
- 社会システム……医療，交通，教育

② 科学・技術分野

- 土木・建築・造船……構造解析，設計計算，応力解析
- 石油・化学・原子力……流通分析，結晶分析，炉心解析
- 交通・輸送……交通網解析，輸送量予測
- 航空・宇宙……飛行計画，性能計算，軌道計算
- 環境・資源……汚染予測，日照計算，騒音解析
- 都市・地域……地域分析・人口推計，都市計画

(2) 利用の状況

情報処理関係に要した諸経費の状況から，外部委託の状況を見ると，表6.8のようになっており，全体の中にしめる外部委託の比率は低い。

表6.8 情報処理関係経費の状況

	ＪＥＣＣ調査 昭51年
調査対象企業数：4,432社	
電子計算機費用	単位：百万円
・導入諸掛り	4,614
・減価償却費	54,463
・レンタル料	253,247

・保守料	12, 967
・回線料	17, 466
電算機室費用	
・人件数	240, 629
・その他	101, 726
・ソフトウエア購入費	2, 042
外部処理費用	
・ソフトウエア委託料	5, 670
・パンチ委託料	27, 924
・マシンタイム借料	4, 203
・計算委託料	22, 066
・要員人件費	23, 166
・その他	4, 840
合　計	775, 023

(3) 外部委託先の選定

委託先の決定にあたっては，長期的な経済性を考慮し，技術力，信用力も加味して考えなければならない。

決定の条件をあげると次のようになる。

① 総合的な企業の信用度

財務的な安定性，経営方針，設立の経緯と背景，経験年数など総合的に判断する。

② 業務処理実績

・過去および現在の顧客層について

・固定業務と不定業務の比率

・得意な業務分野

③ 技術力と指導力

・設備能力と規模

・要員の数，質の程度，定着性，年令

・管理のレベルとノウハウ

・保有ソフトウエアの質

④ 安全性・信頼性

安全保護設備，管理体制，バックアップ体制，機密保持

⑤ データ送運の便利性，迅速性

距離的な便利性，オンライン・サービス，

⑥ 価　　格

・業務の重要性，正確性，技術力

・業務の量，拡張性，

一時的な料金の安さにとらわれないで，質の高いサービスを適正価格で受ける。

ユーザにおける，選択理由をアンケートをもとに調査すると，次のようになっている。

表 6.9　センターの選定理由

	（JIPDEC　昭和 48.3 調査）
・信頼できる	40.2 %
・同一機種	27.5
・価格が安い	39.5
・技術水準が高い	21.6
・専門技術を持っている	21.0
・紹介	16.8
・企業系列関係	15.5
・広告	2.4
・その他	19.2

(4) 適用にあたっての留意点

人手のかかる単純事務計算，アンケート集計などの一時的作業，数値計算，技術計算など臨時的に発生する業務は，外部委託し易いものと言える。

当初，利用する場合は，次の点に留意するとよい。

① 採算計算のしやすいもの

直接的効果が明確で短期間にメリットが生ずるもの。

② メイン業務でないサブ・システムから適用すること。

③ 日常処理よりも，月間処理，週間処理などから着手する。

④ 委託者側は条件整備を十分行ない，自社導入と同じ手続を要する。

(5) 委託料金の体系

委託料金については，情報処理サービス業とユーザの間における契約および交渉によって決定されており，標準的なものはない。しかしながら，その時々の経済的，社会的な状況およびサービスの意合度合によって変化していることも実情であり，利用する場合，数社を比較する必要も生じる。

委託料の内容は次のとおりである。

① システム分析料

システム設計に要する費用であり，システム・エンジニアの占有度合とレベルによっ

て決定される。

② プログラミング料

コーデング，デバッグを一括していうことが多い。ステップ数でステップ当りの場合と，プログラマの占有時間で計算する場合とあり，使用言語や難易度によって異なる。

③ インプット・データ作成料

パンチ料金は，通常，1タッチ何銭という基準で計算される。数字，英数字，カナ文字の割合，原票の見易さ，納期などからも算定される。

④ 消耗品費

実費請求されるものがほとんどであるが，管理，保管のための費用がプラスされる。

⑤ 電算機使用料

CPU，I／Oなどの使用時間，占有度によって計算される。また，時間貸しの占有料金もある。

⑥ その他の費用

- コンサルティング料
- ファシリティ・マネジメント料
- 要員派遣料
- プログラム変更・修正料
- ソフトウエア使用料

これらの料金の原価要素としては，

- 機械レンタル料または購入償却費
- 人件費
- 賃借料
- 備品器具・消耗品費
- 外注費
- 販売・管理費

などで構成され，委託料金の基礎となっている。

6.5 外部利用の今後の方向

今後のコンピュータ室の方向は，ますます機能分化していき，また，自社のみで処理する場合の問題解決のため外部サービスの利用は拡大していくであろう。

コンピュータ室の今後の重点施策の方向と考えられるものを列挙するが，その中にも多くの解決すべきテーマが存在する。

(1) コンピュータ室の今後の重点施策

- 部門の社内における位置づけの強化
- コンピュータ・システムに対するトップの理解度の高揚
- 社内ユーザとの円滑なユミュニケーション
- 他部門との人事交流の活発化
- 要員の充足と質の向上
- 入力媒体作成のための外部サービス業の確保
- 外部サービス業の有効な活用
- 総合システムの構築
- データベース体制の確立
- 標準化の確立
- 情報処理コストの低減

(2) 今後のコンピュータ室のあり方

ユーザの意識調査（1974.5）による今後のコンピュータ室のあり方は，次のようになっており，外部サービスの利用が活発化する傾向がうかがえる。

表6.10 今後のコンピュータ室のあり方

• 自社内で一切の処理を行う	18.4%
• 一時的，不定的に外部を利用する	38.1
• 技術水準の高い外部を利用する	11.4
• 全面的に外部に委託する	3.9
• コンピュータ室を独立企業にする	12.2
• 外部機関と合弁会社を設立する	0.4
• 関係会社と協同のセンタを設立する	15.4

指導上の留意点

(1) この章では，日本における情報処理サービス業の現状を知ることにより，どのように利用していったら良いかを指導する。

(2) 情報処理サービス業の効果的活用という観点から指導する。

(3) 可能であれば，代表的な総合情報処理サービス会社の業務内容等について，見学を含めて紹介することが望ましい。

(4) 情報処理サービス業についての参考文献としては，（財）日本情報処理開発協会が「コンピュータ白書」を含めて，いくつかの報告書をまとめている。

参考文献

〔１〕中江剛毅，「日本の情報システム管理」　泉文堂，1975.

〔２〕中江剛毅，「情報処理産業の経営戦略」　中央経済社，1973.

〔３〕日本情報処理開発協会，「コンピュータ白書1977」　コンピュータ・エージ社，1977.

〔４〕長谷英夫，朝井幸洋共著，「情報処理センター（その運営と効果的利用法）」　ダイヤモンド社，1974.

〔５〕日本情報処理開発協会，「システム監査体制確立への道」，1977.

〔６〕日本電子計算機株式会社（編）　「全国計算センタおよびソフトウエア開発企業便覧」1976

〔７〕行政管理庁，「日本標準産業分類」，1972.

〔８〕日本情報開発協会，日本情報センタ協会，「情報処理サービス業需要構造調査」1973

本書は，日本自転車振興会の機械工業振興資金による補助金の交付を受けて実施した「昭和 53 年度情報処理教育に関する調査研究等補助事業」の一環として作成したものです。

昭和54年３月発行

財団法人　日本情報処理開発協会
情報処理研修センター

〒１０５　東京都港区浜松町２丁目４番１号
（世界貿易センタービル　７階）

TEL　０３（４３５）６５１１（代）

許可なしに転載，複製することを禁じます。

返　却　期　限　票

○　最後にある日付があなたの返却期限です。
○　遅れないように期限内に返却しましょう。
○　続いて借りたいときは届け出てください。

返却期限票　　MARUZEN