

58-R009

ソフトウェア開発・運用の高度化・効率化方法に  
関する調査研究報告書(I)―開発計画―

昭和 59 年 3 月

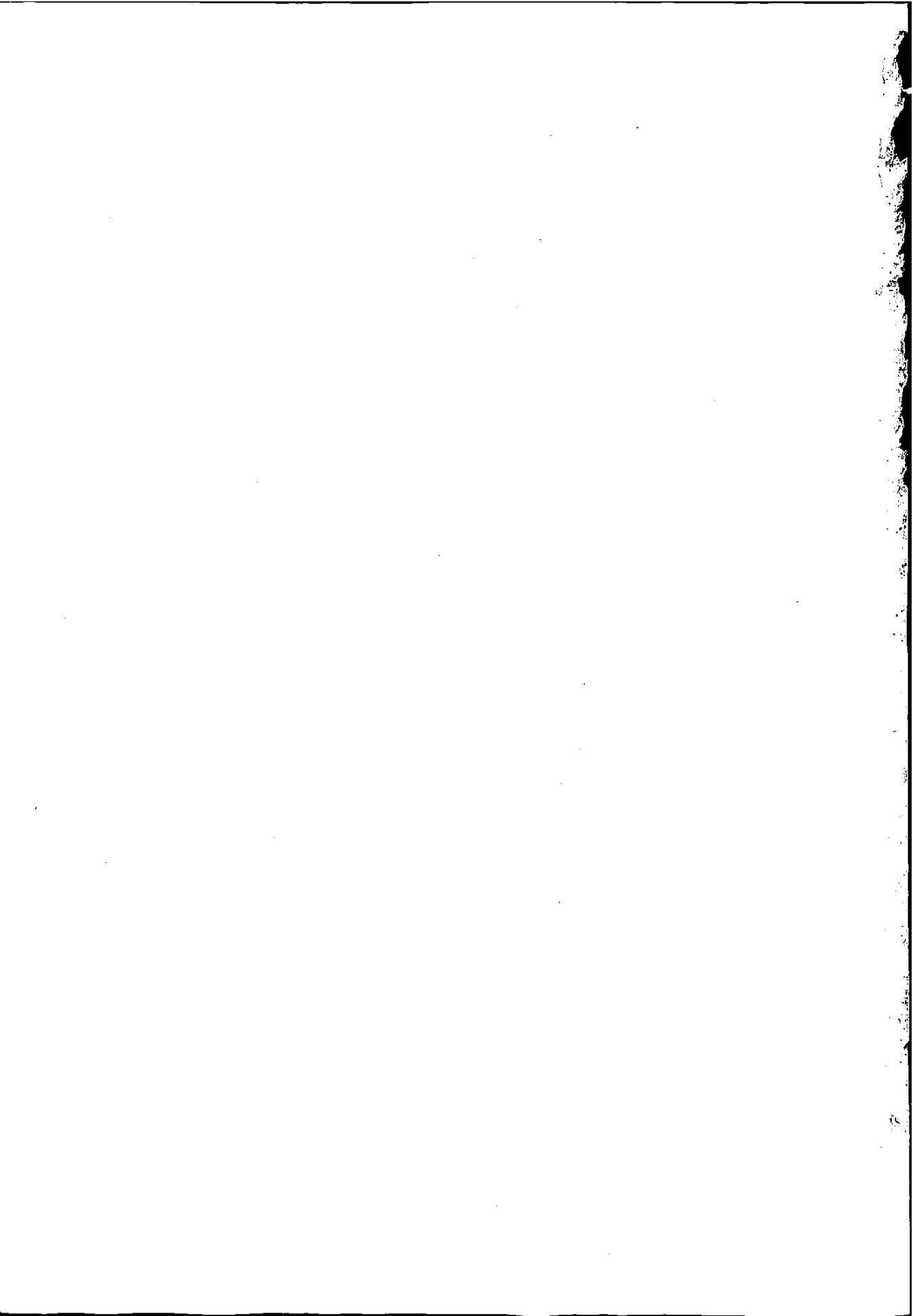
**JIPDEC**

財団法人 日本情報処理開発協会



この報告書は、日本自転車振興会から競輪収入の一部である機械工業振興資金の補助を受けて昭和58年度に実施した「ソフトウェア開発・運用の高度化・効率化方法に関する調査研究」の成果をとりまとめたものです。





## 序

近年における情報処理及び通信技術の急速な進歩に伴ない、現在では社会のさまざまな分野、局面において情報化の進展がみられる。しかし、この中核をなしているコンピュータ・システムそれ自身の問題点を考えてみると、コンピュータのハードウェアについては、最近の著しい技術革新と生産性の向上により、コストパフォーマンスの向上がみられるものの、ソフトウェアについては、今なお旧態然とした方法で開発されていること及びシステムの大規模化、複雑化、高度化さらにはニーズの多様化等によって、ソフトウェアコストのシステム全体に占める割合は一段と高まってきている。このため、信頼性が高く、保守のしやすいソフトウェアを、いかに効率的に、コストをかけずに開発するかが、コンピュータユーザの重要課題となっている。

しかしながら、企業等の情報処理部門の現状は、既存システムの運用・保守に多くの負荷がかかり、多くのシステムがバックログとして放置されることから、エンドユーザの情報処理部門に対する期待感がうすれ、パソコンを初めとするOA機器の導入によってユーザ自身が問題解決を図っていくなど、情報処理部門は、内外からさまざまなインパクトが強まりつつある。したがって、このような状況では、ソフトウェア開発・運用の効率化をどのようにして確立するかが急務であると考えられる。

本調査研究事業は、このような観点からソフトウェア開発経費・工数の削減（生産性の向上）、保守費用の削減、ソフトウェアの品質の向上を目的とし、①開発計画、②ソフトウェア開発、③運用・保守の3つの面からアプローチを行い、3か年計画でそれぞれの内容、運用できる手法・ツール、各種指標等について調査研究するものである。

今年度は、ソフトウェアの開発計画段階を中心にして、コンピュータユーザの開発実態を踏まえつつ、開発要求分析、開発計画の種類と内容及び技法、ツール等について調査研究を行った。

各企業等におけるコンピュータの利用は、企業等の実態に即して行われるべきものであり、ましてや最近の情報機器の進展と相いまって情報処理の形態も多様化しつつあるので、利用形態の如何によって一概にその是非を評価することはできない。しかし、ソフトウェアの開発・運用にかかる諸問題は、各企業等が共通して抱えている課題であろう。

本報告書が、この共通の課題を解決するうえでコンピュータユーザに何らかの示唆を与えるものとなり、ひいては、わが国の情報処理の発展に寄与することになれば幸いである。

なお、調査研究に当たってご協力をいただいた委員及び関係各位に深く感謝するものである。

昭和59年3月

# 目 次

## 概 要

1. 総 論 .....	1
(ソフトウェア開発・運用の高度化・効率化方法の現状と課題)	
1.1 ソフトウェア開発・運用の高度化・効率化の必要性 .....	1
1.1.1 ソフトウェアの規模に依る対応 .....	2
1.1.2 対象業務に依る対応 .....	4
1.1.3 処理形態に依る対応 .....	8
1.2 ソフトウェアの開発サイクル .....	13
1.2.1 ライフサイクルの考え方 .....	13
1.2.2 ライフサイクルの名称と作業概要 .....	14
1.2.3 ライフサイクルとドキュメント .....	19
1.3 高度化・効率化の概要 .....	23
1.3.1 基本的な考え方 .....	24
1.3.2 効率化のための具体的アプローチ .....	32
2. ソフトウェア開発計画の現状 .....	39
2.1 アンケート調査結果の概要 .....	39
2.2 システム化要求の現状 .....	43
2.3 要求分析の現状 .....	46
2.4 開発計画立案の現状 .....	53
2.5 そ の 他 .....	62
2.5.1 外注する場合の判断基準 .....	62
2.5.2 システム開発での失敗の原因 .....	63
2.5.3 システム化要求・分析時における問題点 .....	64

3. ソフトウェア開発の要求定義 .....	69
3.1 はじめに .....	69
3.1.1 要求定義の背景と必要性 .....	69
3.1.2 標準化の必要性と工程区分 .....	71
3.2 システム化要求 .....	73
3.2.1 目的・管理内容 .....	74
3.2.2 作業内容とワークドキュメント .....	74
3.2.3 体制 .....	79
3.3 要求分析 .....	79
3.3.1 目的・管理内容 .....	80
3.3.2 作業内容とワークドキュメント .....	80
3.3.3 体制 .....	83
3.4 システム分析 .....	83
3.4.1 目的・管理内容 .....	83
3.4.2 作業内容とワークドキュメント .....	83
3.4.3 体制 .....	89
3.5 要求定義の技法・ツール .....	89
3.5.1 要求定義技法の現状 .....	89
3.5.2 要求定義技法の概要 .....	92
3.5.3 要求定義技法の効果・留意点 .....	111
3.5.4 要求定義技法の有効度 .....	115
3.5.5 要求定義技法の適用範囲 .....	115
3.5.6 要求定義技法の選択要因 .....	116
3.5.7 要求定義技法の選定手順 .....	117
3.6 要求定義の留意点とチェックリスト .....	118



4.	ソフトウェア開発計画	133
4.1	開発計画の種類と内容	134
4.1.1	種類と概要	135
4.1.2	開発計画相互の関連	149
4.1.3	開発計画の変更	151
4.2	技法・手法・ツール	153
4.2.1	技法・手法の種類と概要	154
4.2.2	ツールの種類と概要	158
4.2.3	利用の実態と効果	162
4.3	見積方法	164
4.3.1	見積方法の概要	164
4.3.2	見積基準	171
4.3.3	見積の実態	172
4.4	開発計画の作成から承認まで	175
4.4.1	手続き	175
4.4.2	計画作成者と承認者の関係	177
4.5	外注の活用	178
4.5.1	外注利用の目的・判断基準	179
4.5.2	外注を利用する段階	180
4.5.3	発注の形態	183
4.5.4	外注先の評価	183
4.5.5	ソフトウェア・プロダクトの利用	184
4.6	開発計画の留意点とチェックリスト	185
5.	ソフトウェア開発計画立案の実際例	191
5.1	A社の事例	191
5.2	B社の事例	202

5.3 C社の事例 .....	208
6. ソフトウェア開発計画の今後の課題と将来動向 .....	219
6.1 ソフトウェア開発計画の今後の課題 .....	219
6.1.1 総合的かつ計画的なシステム開発 .....	219
6.1.2 マルチ・プロジェクト・マネージメントの推進 .....	220
6.1.3 要求仕様の明確化 .....	220
6.1.4 外注する場合の基準・手続きの制定 .....	222
6.1.5 各種計画の確実な立案 .....	223
6.1.6 品質に対する意識の向上 .....	224
6.2 ソフトウェア開発計画の将来動向 .....	224

## ソフトウェア開発・運用調査委員会

(五十音順敬称略)

委員長	道下 忠行	東海大学工学部経営工学科教授
委員	菅野 孝男	日本タイム・シェア(株)コンサルティング部 部長代理
〃	瀬野 浩	行政管理庁行政管理局副統計審査官
〃	玉井 正勝	(財)日本気象協会東京本部事業部主任技師
〃	東 基衛	日本電気(株)ソフトウェア生産技術研究所 管理技術開発部部長
〃	中島 栄	(社)行政情報システム研究所総務部部長
〃	永淵 寛幸	行政管理庁行政管理局副管理官
〃	原沢 正宏	ファコム・ハイタック(株)システム第7部次長
〃	前川 征弘	外務省大臣官房情報管理室
〃	宮路 三郎	(株)構造計画研究所総合技術本部 チーフコンサルタント
〃	小嶋 利文	(財)日本情報処理開発協会常務理事

## ソフトウェア開発・運用調査専門委員会

(五十音順敬称略)

主査	永 淵 寛 幸	行政管理庁行政管理局副管理官
委員	池 田 慶 二	日本電気㈱情報処理官庁システム事業部 第一システム部
”	菅 野 孝 男	日本タイム・シェア㈱コンサルティング部 部長代理
”	駒 勤	㈱行政情報システム研究所主任研究員
”	妹 尾 稔	三井情報開発㈱技術開発部部長
”	谷 口 雅 春	㈱シー・イー・シー技術管理部主任技師
”	徳 永 健 夫	行政管理庁行政管理局副統計審査官
”	野 崎 不二夫	㈱構造計画研究所総合技術本部課長
”	山 下 広 之	日立ソフトウェアエンジニアリング㈱ 企画室部長代理

## 概 要

本報告書は、ソフトウェア開発・運用の高度化・効率化方法に関する調査研究事業初年度として、ソフトウェアの開発計画段階に焦点を当てて実施した結果をとりまとめたものである。本報告書の構成は、次のとおりである。

### 1. 総 論

ソフトウェア開発・運用の高度化，効率化方法の現状と課題

### 2. ソフトウェア開発計画の現状

### 3. ソフトウェア開発の要求定義

### 4. ソフトウェア開発計画

### 5. ソフトウェア開発計画立案の実際例

### 6. ソフトウェア開発計画の今後の課題と将来動向

この報告書においては、ソフトウェア開発を的確に行うために、開発すべきソフトウェアの内容を特定して、その類別方法を明らかにするとともに、開発計画以前の作業として、ソフトウェア開発のための要求分析、システム分析の方法等についてとりまとめている。また、ソフトウェア開発を行うために必要な計画の体系とそこに盛り込むべき内容を明らかにするとともに、民間企業の主要ユーザに対するアンケート調査を行って、実際に行われている開発計画の立案状況と課題等を把握しとりまとめている。

さらに、本報告書では、民間の大手企業3社のソフトウェア開発計画立案の事例と効率化に係る具体的な問題等について紹介している。

また、これらを踏まえて、ソフトウェアの開発計画段階において取り組むべき今後の課題と将来動向について述べている。

本報告書の各章ごとの内容について要約すると、次のとおりである。

- (1) 総 論—ソフトウェア開発・運用の高度化・効率化方法の現状と課題—  
近年におけるコンピュータ部門のソフトウェアの運用・保守に要する経費、負荷は著しい増大をみせており、このことが、新規開発需要に対する

バグログの増大となっている。このため、ソフトウェア開発・運用の高度化、効率化方法が重要な課題としてクローズアップされてきている。そこで、この章では、総論として本調査研究事業の必要性について述べた後、ソフトウェア開発のライフサイクル及び高度化、効率化の方法等に関する現状と課題についてとりまとめている。

## (2) ソフトウェア開発計画の現状

ソフトウェア開発・運用の高度化・効率化に関する調査を民間企業ユーザに対し郵送調査（回答企業114社）した結果をもとに、システム化要求、要求分析、開発計画立案の現状に関しとりまとめている。

## (3) ソフトウェア開発要求の定義

コンピュータ部門がソフトウェア開発を推進する際に、開発すべき内容をユーザのニーズやトップマネジメントの意向及び企業の経営方針に即して要求分析する方法論とその技法に関し記述したものである。

## (4) ソフトウェア開発計画

システム開発における開発要求分析作業に続くものとして、開発計画立案作業のあり方について述べたものである。即ち、ソフトウェア開発を行うに当たって作成される計画の種類と内容、その場合に用いられる技法、手法、ツールの種類の概要と特徴、技法等の利用の実態と利用方法とその効果等について述べるとともに、開発コストの見積り方法と作業を進めるにあたってのチェックリストの紹介、及び開発計画の作成から承認に至るまでの手続関係についても言及している。また、外注利用も活発に行われていることから外注の活用に関しても一項目を設けてその考え方を述べている。

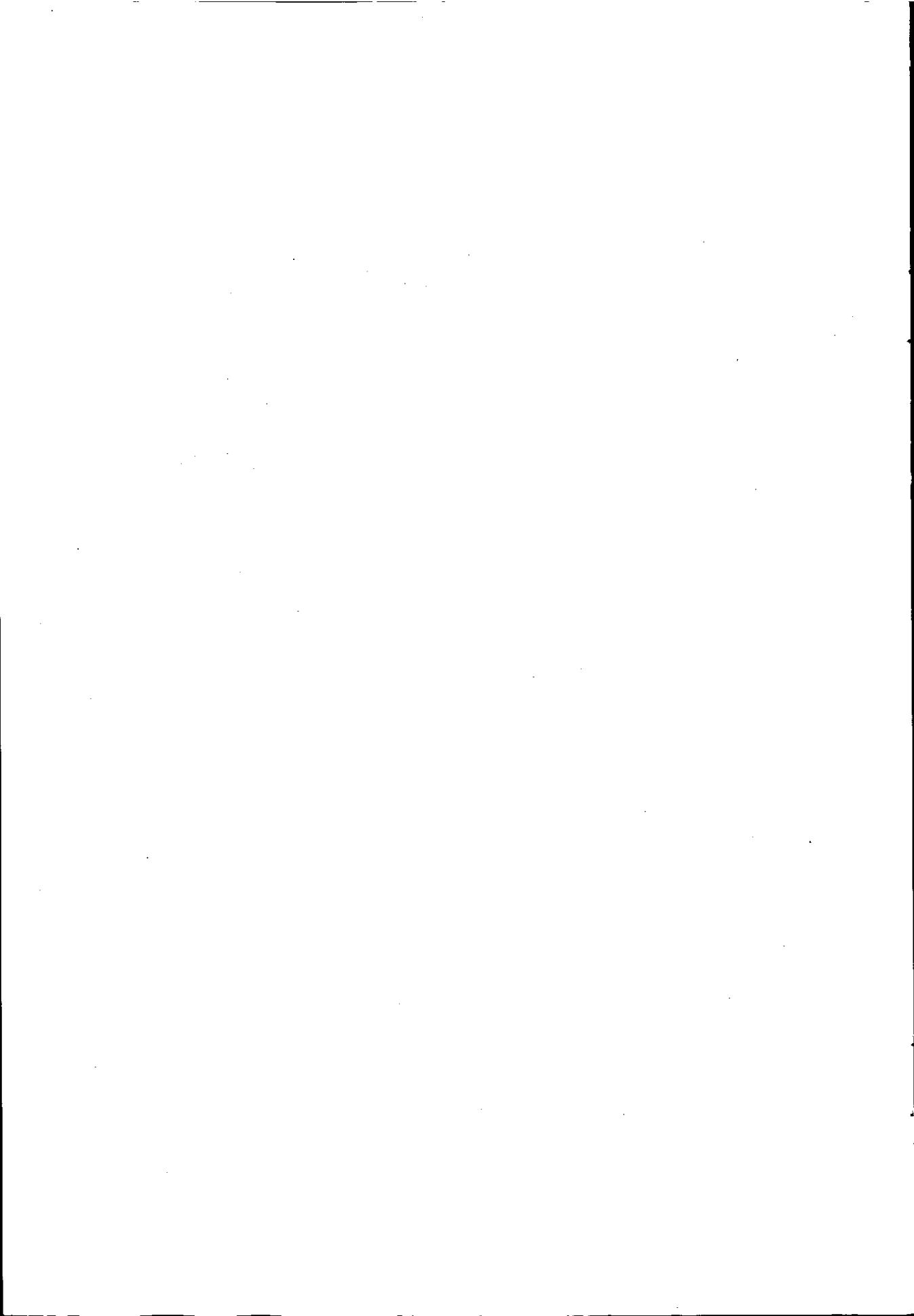
## (5) ソフトウェア開発計画立案の実例

ソフトウェア開発計画立案の実例として、3企業をとりあげこれらの企業におけるソフトウェア開発計画の特徴的な部分を中心に事例紹介を行ったものである。

## (6) ソフトウェア開発計画の今後の課題と将来動向

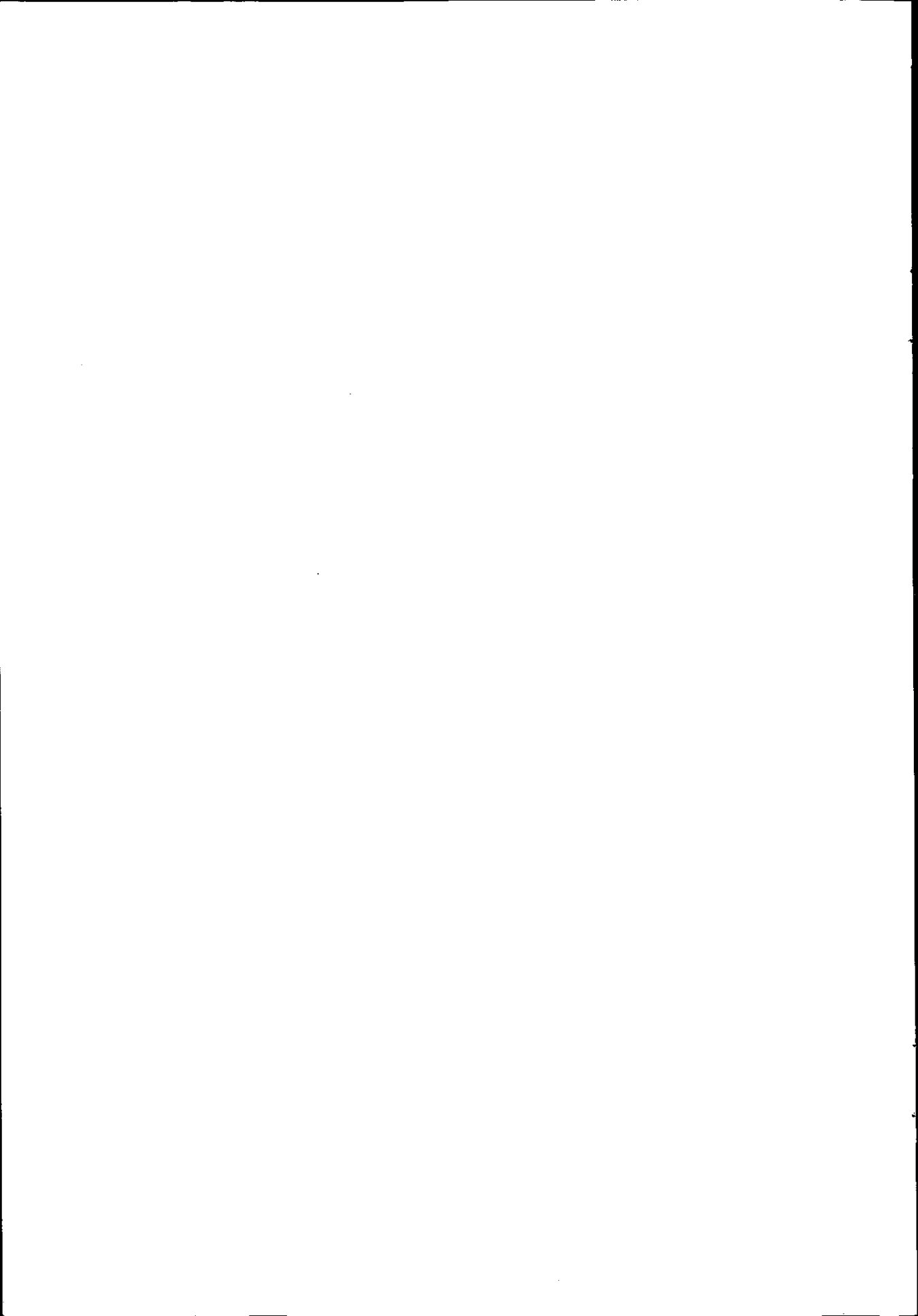
ソフトウェアの開発計画段階における今後の課題として、①総合的かつ計画的なシステム開発の推進、②マルチプロジェクトマネジメントの推進、③要求仕様の明確化、④外注の場合の基準・手続の制定、⑤各種計画の確実な立案、⑥品質に対する意識の向上等に対する取り組みが必要であることを述べている。

また、将来動向に関しては、ソフトウェア開発のための各種計画が、O A機器とくに最近著しく普及しつつあるパーソナルコンピュータの利用によって立案作成されるようになると展望している。





# 1. 総論



# 1 総 論

## — ソフトウェア開発・運用の高度化・効率化方法の現状と課題 —

### 1.1 ソフトウェア開発・運用の高度化・効率化の必要性

バックログ問題について日経コンピュータでは東商一部上場企業919社を対象にアンケート調査を実施し、回答結果を283社(回答率31%)から得た。その報告内容を日経コンピュータ(1982.12.27)に記載しているがそれによると回答企業の6割が1年以上のバックログを抱え込んで苦しんでいる現状が浮きぼりにされている。又何故このようにバックログが増加したのかの原因についての質問を同時に行なっているが、その原因の第1位に社内ユーザ部門のシステム化ニーズの急増、第2位が開発要員の不足、第3位がソフトウェア・メンテナンスの急増、第4位がコンピュータ部門のシステム化ニーズの急増などがあげられている。このアンケート結果でも分る通り、コンピュータ部門へのシステム化要求は増大の一途を辿っており、それに対して蓄積される一方のソフトウェアの運用・保守に多くの要員をとられ、開発作業に十分なる要員を割りふる事が出来ずに、次々にバックログとして蓄積されていくといった悪循環が繰り返されているのが実状であろう。しかもパソコン、オフコンといった各種のOA機器の低価格化、高性能化が一層システム化ニーズをかきたてており、其の上最近の高度情報化社会に対応すべく新しい通信ネットワークの構築やニューメディアに対する企業内への取り込み、経営戦略としての情報システムの活用などといった様々なインパクトがコンピュータ部門に押し寄せている。内部には多くのバックログを抱え、外部からは様々な技術革新やシステム化要請といった、内外にわたる様々なインパクトがコンピュータ部門に押し寄せ、その対応に苦慮しているのがコンピュータ部門の偽らない現実の姿であろう。一方エンドユーザ側にとってはシステム化要求をいくらコンピュータ部門に出しても一向に解決してもらえないといったいらだちと相まって、企業運営での重要な役割を担っている情報

システムの開発、運用に対して高度化、効率化が大きく叫ばれるようになってきた。このソフトウェアの開発運用の高度化、効率化問題には様々な要因が絡み合っていてなかなか同一の土俵で論じる事が難かしいので、ソフトウェア規模、業務処理分野、処理形態などの観点から述べることにする。

### 1.1.1 ソフトウェアの規模に依る対応

小規模から大規模のソフトウェア開発プロジェクトでの開発作業の構成要素は次の3つに分類している。イ：プロジェクトを遂行するにあたっての必要最小作業量、ロ：作業に依って引き起こされたエラーを取り除くのに必要な作業量、ハ：個人差に依って発生する作業量。この中で一番規模の大きなプロジェクトでは、エラーを取り除く作業が必要最小の作業量よりも多く、しかも個人差に依って発生する作業量が一番低い。この事は大規模のソフトウェア開発を行うにあたって多くの要員を投入する必要があるが、この参加要員間のコミュニケーション問題が重要な役割を帯びていることを意味する。更に、ソフトウェア開発で重要な要素を占めている個人的な能力が大規模ソフトウェアの場合には、それほど目立った存在にならずむしろプロジェクトのマネジメント問題がより重要な事を示している。一方小規模ソフトウェアの場合には、個人的な能力問題が作業量として大きなウェイトを占めているといった具合に、ソフトウェアの開発規模の大小によって、問題点が異なってくるのでその特質を十分考慮してソフトウェアの開発運用を行う必要がある。

#### (1) 小規模ソフトウェアの場合

小規模のソフトウェア開発を行う場合、個人或いは小人数で開発することになるが、そこに標準化に依る各種の基準といった外部的な枠をはめない限り、個人の能力に全く依存してしまい、納期や品質が全く個人或いはプロジェクト・リーダーの意識に頼らざるを得ないのが実状である。この様にして開発され蓄積されたソフトウェアは開発した

要員が対処しない限り、なかなか十分な即応体制がとれないといった問題も発生してきている。

この様に小規模ソフトウェアの場合、十分な開発体制、要員の教育、詳練なしでも開発要員がいる限りそれほどの問題は発生してこない。しかし企業は生き物であり、次々と発生するシステム化ニーズを解決する為に、次々と新しいソフトウェアが開発され、そのうちにいつしか蓄積されたソフトウェアも大量となり、何んの脈絡のないソフトウェアの集合体となり、大規模ソフトウェアが持っている問題点と同じような問題点を持つようになってくる。小規模ソフトウェアでも一度、開発されると長く利用されてるものであり、しかもどんどん社内に蓄積されていくものである。この様な認識を持って、事前の要員教育や将来の環境変化を想定し、運用・保守問題を十分勘案した最底限の高度化、効率化をはかる必要がある。

## (2) 大規模ソフトウェアの場合

最近のようにソフトウェアの大規模化、複雑化、広範囲化してくると好むと好まざるにかかわらず、我々の実生活に直接影響を持つソフトウェアが多数開発されてきている。この場合、勿論開発の効率向上をはかる必要があるが、それと同時に信頼性、安全性といった品質問題が今後一層重要な課題となってくるものと思われる。大規模ソフトウェアの開発過程を眺めてみると、開発計画段階では当該ソフトウェアのイメージがかならずしも明確でなく、しかもそこに要求される機能も不明確な状態でプロジェクトを開始するケースが多く、更に見積技術の不完全さといった悪条件が重なりプロジェクト規模が予想外に大きなものとなり、予算を大巾に超過したり、開発期間の遅延といったプロジェクト破産につながるケースが多い。其の上ソフトウェア規模が大きくなればなるほど多くの要員が参加して開発が行なわれる。この場合このプロジェクトに関係するエンド・ユーザも含くめてのコミュニケー

ションが重要な基礎となっており、「ささやきゲーム」のように、エンド・ユーザが目標としているソフトウェア機能が、SE、プログラマーに次々と伝達されていくうちに、変質してしまい、最初に意図したソフトウェア仕様と全くかけはなれたものになる可能性もある。従ってプロジェクトを管理する為には、事前の十分な標準化やプロジェクト運営ルールなどを策定する必要がある。又ソフトウェア規模が大きいとこれを実際に開発する部隊の他にこれを支援する部隊を設置してプロジェクトの進捗に対して各種の協力支援をはかり、スムーズな運営をはかる方式が良く採用されている。この様な事前の準備や標準化ルールが設定されていないと、運用・保守段階になっても多くの開発要員がそのまま保守作業にくぎ付けにされ、開発バックログの原因となってしまうので、大規模ソフトウェアでは如何にしてその品質を高め、より少ない要員で維持していくかが重要な課題である。或るコンピュータ・メーカーの調査に依ると、大型ユーザ61社のコンピュータ要員1人当たり保有プログラム本数の平均が1979年の48本/人から1982年の62本/人の約30%の伸びを示しており、確実にコンピュータ要員に運用、保守上の大きな負荷がかかってきていることが分る。この事は開発の効率化向上と同様に運用、保守段階での効率向上も今後の大きな課題であることを意味している。

### 1.1.2 対象業務に依る対応

対象となっているものの見方を整理する為に色々な分類が一般的には行なわれているが、ソフトウェアを分類する場合にはソフトウェアの持つ属性によって分類する方法がある。そこでここでは属性を対象業務ととらえて分類を行なうと次のように分類される。イ、事務処理分野、ロ、科学技術計算分野、ハ、経営管理分野

## (1) 事務処理分野

企業内の各組織のなかで発生する大量の分類、記録、計算、作表といった作業的な情報処理は企業、組織の維持の為に最底限の機能である。その為に始めてコンピュータを導入し、効率化をはかる事が可能な分野として事務処理分野が対象となる場合が多かった。つまり、情報処理分野での中核をなしている情報処理サービス業の業務内容比率を調査したレポートに依ると、受託計算業務の中で約86%が事務計算分野の業務、約12%が科学技術計算分野の業務となっており、この事は、オフィス内での合理化、効率化が重要な課題である事が分る。コンピュータがこの世に誕生して以来約30年が経過しており、コンピュータを利用したシステム化技術もそこそこの歴史を持つようになってきた。しかしながらこの分野でのソフトウェア開発の生産性は必ずしも高くないのが現状である。生産性の面だけをとらえるならば、同一業種の中で比較的共通的な事務処理、例えば会計処理のような事務処理に対しては汎用的なパッケージが開発販売されるものと思われるが、現実には、この様なソフトウェア・パッケージがあまり流通していない。この事は勿論、費用対効果的な面が非常に強いが、これ以上に企業の独自性という考え方が強く、しかもその企業に於いて永年培われてきた独自の事務処理ノウハウがあり、汎用化をより困難にしている点でもある。その為に企業内に蓄積されてくるソフトウェア規模は益々増大し、それに対処する為に多くの要員が必要となり、開発に十分な要員を投入出来ないといった悪循環をたどっている。しかも一度大規模な投資を行なってシステム化すると、これが新しい事務処理の中に組み込まれ、それによって新たな事務処理ノウハウを醸成していくといった処理サイクルを踏んで益々巨大化してきている。システムが巨大化するにつれて、手作業時に於ける柔軟性がうすれ益々硬直化してしまい外部環境の変化にも即時の対応が出来なくなって

きている。その為に十分なサービスが出来ず利用者側に不満足感を植え付ける結果になっている。又増大するバックログの為にシステム化の緊急性を要するシステムでもすぐには取りあげる事が出来ず、他社との営業競争上問題が発生するような事態も出てきている。その意味に於いて企業経営の重要な基盤をなしているこれらの事務処理分野での開発効率向上や高度化が企業の今後の発展の重要な鍵を握っているといっても過言ではない。

## (2) 科学技術分野

システム化要求の中で事務処理分野ほど規模は大きくないが、事務処理分野に次いでシステム化ニーズの高い分野である。この分野はまだまだ絶対量としてもそれほど多くない為に開発効率といった問題がそれほど大きな話題とはなっていない。しかもこの分野の業務を担当している部門が全社的に配置されているわけではなく設計部門とか技術部門といった部門に特定されている場合が多い。又ソフトウェアの開発もコンピュータ部門に委託せずにオープン・プログラマ的に自分で開発している場合が多い。その為に初期の事務処理分野と同様に問題点が顕在化せずに個人の中に埋没してしまう場合が多い。しかしこの分野では各種の技術計算のパッケージがかなり整備されており、自分で開発するよりはこのパッケージを利用した方が良策であるといった風潮があり、しかも権威のあるパッケージを利用した方がそこから得られる情報により高い信頼性があるので益々その利用頻度が高まっていくといった良い循環が生まれている。

又事務処理分野の場合にはそのソフトウェアがなくても何んらの代替手段が存在しているが、科学技術計算の場合にはその業務自体に他の代替手段では解決出来ない為に、納期問題があまり問題とならない場合が多い。しかもその問題解決の為に解法が判明する迄繰り返して行なわれる場合が多く、開発にあたっては事前に予算申請を行ない開



発を行なっているが、技術的にどうやって解決するかが先行して、費用や期間などが二次的に扱われる場合が多い。又自分達の省力化の為にプログラミングだけを外部に委託する方法をとる場合もあるが、自分達でも十分プログラミングする能力を持っているので運用・保守を自分で行なっている場合が多い。しかも最近パソコンの導入によって全くコンピュータ部門の力を借りずに非常にクローズドな世界で処理が行なわれているケースが多く、今後問題となってくるのは、技術者、研究者の横の連絡が十分でないために個人個人で作られ蓄積されているソフトウェア郡を二重投資防止といった観点からソフトウェアの管理とメンテナンスが重要な課題になってくるものと思われる。

更に最近では技術計算を基礎に業務全体をシステム化するといった大きなプロジェクトが開発されてきている。例えば見積計算システム、設計計算システムのようにFORTRANで10万ステップ以上にも及ぶようなプロジェクトも存在している。この場合エンドユーザ部門とコンピュータ部門と共同してプロジェクト・チームを作り、しっかりとした開発工程を踏んで開発が実施される場合も多数出ておりここでも事務処理分野と同様にプロジェクト管理や生産性向上が大きな課題となってきている。

### (3) 経営管理分野

この分野での大きな目的は経営管理や各種の経営計画策定にあたって意思決定者に判断材料を提供することにある。しかし、企業内にコンピュータが導入されて以来既に相当数の年月が経過しているが、企業のトップマネジメントに対して行なったアンケート結果に依ると必ずしもコンピュータ導入に十分な満足を得ていない状態である。この事は経営管理分野に於けるシステム化の遅れを如実に物語っているものと思われる。其の上この分野は事務処理分野のように定型化した業務でなく、非定型的な業務が多く、しかもモデル化がなかなか出来に

くい分野であり、又信頼性の高いデータが集めにくい分野でもある。更にこの分野での課題は経営管理資料である情報を経営者や上位の管理者が必要を感じなければ如何に質の高い有効な情報を提供する仕組を構築しても全く無意味となるので、如何にしてそれらの情報やシステムが意思決定時に必要であるかを認識させるトップ教育が重要な役割を帯びている。米国では、かなりのトップマネジメントが経営計画策定時にシミュレーション結果などを参照して意思決定の判断材料に利用しているという報告事例があるが、日本の場合にはまだまだそのような事例が少なく、今後ともトップマネジメントに対する意識改革をはかるべく地味な啓蒙運動を続ける必要がある。又従来の経験を土台にした勘だけに頼る経営でなく科学的な判断材料を活用することによって、より高度な経営が出来るような仕組を経営の場の中に融合させコンピュータに対する抵抗感を無くすことが今後の活用を促進させる重要なポイントでもある。其の上この分野での有効活用には経営管理情報のデータベース化が重要な基礎をなしており、これの早期の確立が望まれている。

### 1.1.3 処理形態による対応

企業内に於ける情報処理機能には、①情報の収集、②情報の処理、③情報の伝達といった3つの機能から成り立っているが、この情報処理機能を別の視点である時間という尺度によって眺めてみると、①情報を発生都度収集し即座に処理を行なって情報を伝達するリアルタイム処理形態と、②発生した情報を或る程度の時間を置いて収集し、一括して処理を行なった後に情報の伝達を行うバッチ処理形態に分類することが出来る。又空間的な視点からこの情報処理機能を眺めてみると情報が発生する地点、情報を処理する地点、情報を利用する地点の3ヶ所があり、これらの場所をどのように使い分けるかによって、①集中処理方式、②

分散処理方式に分類される。更に物理的な要素に依る分け方もある。つまり情報処理システムを構成するコンピュータのCPUと端末との接続状態によって分類する方式もある。①各種の端末装置がCPUの直接的なコントロール下にある場合をオンライン方式、②各種の端末装置がCPUの直接的な制御下でない場合をオフライン方式と呼んでいる。これらの処理形態の中で典型的なバッチ処理とオンライン形態の特徴、問題点について記述する。また、OAについても簡単に触れることとする。

#### (1) バッチ処理

企業に於いてコンピュータ化するのがこのバッチ処理形態からであるように思われる。この方式は大量データが発生し、しかも処理結果を手に入れるに利用者側にそれほどの緊急性を必要としないような業務処理に適していた。しかしこのような業務の中でも、以前では十分バッチ処理でエンド・ユーザが満足していたものが、より高度なしかもより早いターン・アラウンドを求めてオンライン処理に移行する場合も増えている。この場合ユーザの単なる欲望によってより早い処理を要求しているかを利便性と採算性とをよく見極めた上でどの処理形態にするかを決定する必要がある。事務処理分野に於けるバッチ処理のソフトウェアの開発には、過去に比較的多くの事例が存在していることからそれ等のノウハウを利用することによって、以前と比較するとかなり効率の向上が図れるようになっている。しかしこれとても大規模なソフトウェアになると見積精度が大巾に狂ってしまい大きな問題を引き起こすので、大規模ソフトウェアの場合、見積誤差もそれほど大きくなく、しかも高品質なソフトウェアを開発することが出来るよようにすることが今後の大きな課題である。又この処理形態で注意しなければならない点は既に蓄積されているソフトウェアとの整合性の問題である。比較的システム化ニーズの高いソフトウェアから順次開発を実施していくうちに、既存システムとの整合性を十分考慮しな

いと新たに構築するシステムとがオーバー・ラップしたり、エンド・ユーザ側では同じデータを2度も3度も投入するような事態が発生する場合もある。そこで新たなソフトウェアを開発する際には、既存システムを十分分析して、システム間の整合性をとりながら開発する必要がある。

## (2) オンライン処理

近年の情報処理技術の高度化によって産業全体や社会全体の情報化が促進され効率の高い活動が出来るような基盤が整備されつつあるが、その中でもオンライン処理は利用者ニーズに即応したより高いサービスをあらゆる分野に提供するようになってきている。このオンライン処理形態は、①データ収集システム、②メッセージ交換システム、③照会応答システム、④トランザクション処理システム、⑤遠隔ジョブ入力システム、⑥T、S、S処理システム、⑦コンピュータ・ネットワーク・システムのように更に細分化される。このオンライン・システムは、すべての処理が必ずしも複雑でなく単純なデータ収集システムもこの中に含まれている。その為に対象業務の特性並びに運用形態、費用などといった総合的な観点から最適な処理形態を選択する必要がある。又オンライン処理形態への変化を支えている情報処理技術は、バッチ処理段階に於いて培われた情報処理技術が新しいニーズをかん起し、そのニーズが新しい技術を産み出し、しかもその技術が更に人間の欲望をかきたてて、更に新しいニーズを醸成し、そしてより一層進んだ技術を産み出していくといったサイクルをたどりながら高度化されてきている。その意味でオンライン処理のソフトウェアを開発するにあたってはバッチ処理のソフトウェアを開発する時の基本的な事項を十分取り込んで開発する必要があるが、特にオンライン処理の場合は信頼性や安全性などを特に要求される場合が多く、しかも品質問題にも十分配慮して開発する必要がある。

又最近の第三次の銀行オンライン・システムのように開発に5000人月から6000人月を要する大規模なプロジェクトの場合多額な投資を必要とする為に、高級言語やパッケージを導入することによって効率向上をはかろうとしている。このように大規模なネットワークを利用したオンライン処理は大型化の傾向を示しており、これを円滑に開発する為には情報処理技術やプロジェクト管理技術などの高度化が必須要件となってきている。

又大規模なオンライン処理の場合には多様化してくる環境変化に対応する為に新機能の追加といった事態が発生するが、これに対してなかなか柔軟に対応する事が出来なかった。その為に今後開発する大規模のオンライン・システムが新規業務などの開始に対してどれだけ柔軟に対応出来るシステムを構築することが出来るかが今後の大きな課題でもある。

### (3) OAへの対応

最近では、一時の熱狂的なOAブームから落ち着きを取り戻し地に足がついたOA化を推進しようとする企業が増加してきた。従来のシステム化が大規模な定型業務を主体に進められてきたが、最近の分散処理用の端末や各種のパソコンの普及によって小規模の非定型的な業務にも適用範囲が拡大されておりOAが新しいアプリケーションの分野として位置づけられるようになってきた。しかし何故に最近になってこのようにOAブームがわき起きてきたかを考えてみる。この場合OAが意図する目的を「OA機器とソフトウェアを結びつける事によって効率的にオフィス機能を果たすこと」と定義すれば、従来のシステム化の考え方の延長線上で十分であり、又このOA化の目的を「企業体質の強化であり、企業の生き残り戦略である」と定義すれば、これだけでは不十分であり、より一層のOA化を推進する必要性を感じるように、OA化の意図する目的には様々なレベルや考え方があり必

ずしも一意的に規定出来ないのが現状である。最近のように企業競争の激しい時代には、OA化の目的を当然、「企業の生き残り戦略」としてとらえる必要がある。又オフィスに於ける生産性の低くさを製造部門に於ける高生産性と比較して云々しているが、この場合オフィスに於ける生産性を何を対象にして生産性を測定しているかが不明確であり、あまり比較の対象にならないと思われる。但しオフィス部門と製造部門とを比較するとあまりにも多くの要員を抱え事務処理機能を実施しているように考えられるのでより一層のOA化を推進していく引き金にもなっている。

其の上、エレクトロニクスの高度な発達によって革新的に進歩した高性能で、低価格のOA機器群がオフィス部門にどっと流れ込んでいくが、システム化の恩恵を受けていない積み残し領域のオフィス部門での活用が今後の大きな課題でもある。この活用こそが企業の持っている知恵の結集であり、しかもこの知恵の結集化の為の高度化、効率化が非常に待たれるゆえんでもある。其の意味に於いてOA化は全く新らしく発生してきた概念でなく、従来のコンピュータ化の延長線上にあり、これを支援するハードウェアの高度化がこれを一層容易にしているだけで概念的にはそれほど大きな変化は無いと思われる。つまりハードウェアの高度化、その利用技術、コスト面、環境条件などが丁度バランス良くマッチングした結果がこのようなOAブームを招来したものと思われる。

## 1.2 ソフトウェアの開発サイクル

### 1.2.1 ライフサイクルの考え方

コンピュータが実用化されて以来既に20年以上が経過しているが、なかでもハードウェアの進歩は著るしいものがあり、確実に高性能化、低価格化の方向へと発展してきている。一方ソフトウェアは旧態然たる環境の下で生産が続けられてきていた。ところが1968年に西ドイツで開催された国際会議に於いてソフトウェア・エンジニアリングという概念が始めて話題となり大きな注目をあびた。この概念は従来から行なわれていた生産方式、つまり職人芸的な技術者によって開発されるソフトウェア作りから脱皮して近代的な工業製品の製造と同じようにしっかりとした工程を経て、ソフトウェアも設計、製造されるべきであるという考え方を採っている。しかもこの様な概念での開発方式が最終的には開発費用の低減や品質の高いソフトウェア生産が可能となるものであるとしている。

この考え方が提唱されて以来ソフトウェアの開発の中に、ソフトウェア・エンジニアリングの概念が除々にとり入れられ、1975年の後半頃からソフトウェア・エンジニアリングという言葉が定着し始めた。

最近のようにソフトウェアの規模が大きくなり、しかも複雑化してくるとソフトウェアの持つバグによって引き起こされる問題は、非常に大きなものとなり、しかもそれが我々の社会生活に密着したソフトウェアとなると、一度バグによって問題が発生するとそれが及ぼす影響範囲は広範なものとなり、大きな社会問題をも引き起こす危険性を含んでいる。その為にもソフトウェアを開発するにあたっては、信頼性の向上や品質保証の問題は重要な課題となってきている。そこでソフトウェアの開発にあたってはソフトウェア・エンジニアリングで提唱しているライフサイクルの考え方を積極的に採り入れてライフサイクル全体の生産性に着

目した開発を行なう企業が増えてきている。このライフサイクルの基本的な考え方は、ソフトウェアの特徴である不可視性を如何にして可視性のあるものにして品質や進捗状況を管理するかにある。つまり工業製品を製造するのと同じ様に、ソフトウェアを開発する段階を幾つかの工程に分けており、そして各工程の終了を意味するドキュメントを作成して開発作業の確実な進捗を管理するのに利用している。しかも作成されたドキュメントをレビューすることによって、ソフトウェアの開発作業の中で潜在的に発生してくる種々のエラーを早期に発見するようにしている。この事によって手戻り工程をミニマイズする事が出来る。その結果品質の向上がはかれ、最終的に生産性の向上がはかれるといったソフトウェア開発での基礎技術の重要な考え方である。

### 1.2.2 ライフサイクルの名称と作業概要

ライフサイクルとは「発達の段階の一連の系列」と定義されているが、ソフトウェアのライフサイクルの場合は大きく分けると開発サイクルと運用サイクルから成り立っている。開発サイクルは問題提起からソフトウェア製品の誕生までをさしており、いわゆる受注から製品の出荷迄の工程をさしている。又運用サイクルは出荷後のアフタケアに相当している。ハードウェアの場合、長い間使用しているとハードウェア自体が劣化して寿命が過ぎてくるが、それに反してソフトウェアの場合、環境変化に応じた保守を効率良く行なっている限りは、いつまでも利用出来るといった特徴がありこの工程には運用、保守、評価といった工程も含まれている。このライフサイクルの概念をベースにした開発体制を必要とするのは、一般的には中規模以上の大型ソフトウェアの開発に特に必要とされる考え方である。しかし小規模のソフトウェア開発の場合といえどもソフトウェア・エンジニアリングの概念は十分に参考にする必要がある。但し大規模のソフトウェア開発に有効であって



も、大規模ソフトウェア開発での方法がそっくりそのまま小規模のソフトウェア開発にも十分有効であるとは言いきれないので、ソフトウェアを開発する上での開発環境、運用環境、ソフトウェアの特質などを十分勘案し、更には採算的な要素なども十分加味して如何なる有効な要素を取り込んで開発していくかを決定する必要がある。

ライフサイクルの工程名称には必ずしも統一的な定義はないが、NATOのソフトウェア・エンジニアリングの国際会議に於いて、ソフトウェアのライフサイクル工程名称を次のように定義している。

- ① システム設計の調査研究
- ② システム設計
- ③ 単体部品の設計
- ④ ユニット設計
- ⑤ ユニットの開発
- ⑥ ユニットのテスト
- ⑦ 単体部品のテスト
- ⑧ システム・テスト
- ⑨ 運用保守

又 B. W. Boehm はライフサイクルを次のように定義している。

- ① システム要求定義
- ② ソフトウェアの要求定義
- ③ 概要設計
- ④ 詳細設計
- ⑤ コーディングとデバッグ
- ⑥ テストと移行
- ⑦ 運用と保守

更に Munson 氏は SDC 社に於ける長年の経験をベースにしてライフサイクルの工程を次の 9 段階の工程に分ける事を主張している。

- ㊶ システムの要求定義
- ㊷ ソフトウェアの要求定義
- ㊸ ソフトウェアのシステム設計
- ㊹ 詳細設計
- ㊺ コーディングとモジュールテスト
- ㊻ モジュールの統合
- ㊼ ソフトウェアのシステム・テスト
- ㊽ 出荷手続き
- ㊾ 運 用

この様にライフサイクルの工程分けは個人や企業などによって独自の工程分けと工程名称を用いており、必ずしも統一がとれていないのが実状である。又日本に於いてもそれぞれのフレーム、メーカーが独自の名称、工程分けをしている。更に昭和 56 年に日本情報センター協会が会員各社に対して行なったアンケート調査結果のレポートにあるようにやはり様々な工程名称や工程定義が行なわれておりなかなか統一するのが困難であるように思える。その中でも特にバラツキの多いのが設計工程であるが、この工程に於ける名称として、機能設計、概念設計、概要設計、システム設計、基本設計などがあげられており、しかもそれぞれの定義領域もそれぞれ異なっていることが報告されている。ここでは、このライフサイクルを、㊶要求定義段階、㊷設計段階、㊸プログラミング、㊹テスト段階、㊺運用・保守段階と定義してそれぞれの工程に於いて如何なる作業が行なわれるかを以下に概説することとする。

#### (1) 要求定義段階

この工程での主要目的はシステム化ニーズを如何にエンドユーザからきっちりと引き出すかである。この工程で作業が着実に、しかも正確に引き出すことに成功すればシステム化ニーズ問題の半分が解決したといわれるように、エンドユーザとの対話を通して確定いくために

インタビュー技術などを持った高度な技術，経験が要請される工程である。しかもこの工程ではエンドユーザと開発者の相互理解の基礎資料となる各種のドキュメントを作成して相互の確認を得ておく事も重要な作業の一つである。このドキュメント上にはシステム概要，システム機能，データ仕様，入出力条件，システム構成，信頼性，スケジュール上のマイルストーン，納期などを十分検討して記述する必要がある。

## (2) 設 計 段 階

要求定義段階で確定されたシステム要求をプログラマに正確に伝える為にコンピュータが処理出来るように機能分けを行ない，それぞれの機能をどのようにするか決定を行なう。又データ仕様，システム構成，運用方法などについての設計を行なう。更にその設計書ベースにプログラマがプログラミング出来るようにそれぞれのプログラムの入力，出力，手順，ファイル・アクセス方法などを規定したプログラム仕様書を作成する。

## (3) プログラミング段階

設計段階で作成されたプログラム仕様書に沿って，入力されたデータが如何なる手順によって求められている出力が得られるかを図示するチャートを作成する。更にそのチャートなどを参照しながら，必要とする言語を得る為にコーディング作業を行なう。又そのコーディングしたプログラムが正しく作動するか否かを，種々のケースを想定したデータを利用してデバッグを行ないそのプログラムがプログラム仕様書通りの動作を行なうか否かのチェックを行なう。又保守段階で容易にプログラムを修正出来るように各種のプログラム・ドキュメントを作成する。

## (4) テスト段階

プログラミング段階でのテストは主にプログラマがプログラム作成上作りあげた論理構成が正しく動作するかが検証される。しかも対象

は単一のプログラム内だけに限定されるが、このテスト段階ではシステム設計書がシステム機能を正しく分割しているか、又それぞれに分割した機能間のインターフェイスが十分整合性がとれているか否か、更に実際の現場運用にそのシステムが十分耐える事が出来るか否かなどを、システム設計者を中心にして考え出したテストデータや実際の現場で使用されているデータを利用してテストを行なう段階である。この段階に於いてテストを十分実施しないで次の工程にパスしてしまうと、実際の現場で大きな混乱を発生し大きな問題となったりする場合もあるので、そのソフトウェアの品質には十分留意する必要がある。その為にはシステム設計者のみで検証テストを行なうのではなく、エンドユーザ部門にもこのテスト作業に参加してもらい、そのソフトウェアに期待される通りの機能を発揮するか否かのチェックを行なう。又そのソフトウェアの信頼性問題についても事前に十分チェックする必要がある。

#### (5) 運用、保守段階

テスト段階で十分そのソフトウェアが正常に稼動する状況が確認されると、この段階ではこのソフトウェアを実際の業務に活用していく段階である。又この工程に入る前のテスト工程段階で十分な教育・訓練をこのソフトウェアの関係者に実施しておかないと非常な混乱を引き起こすので大規模のソフトウェアになればなるほどこのソフトウェアの導入計画と要員計画を慎重に策定する必要があるが、この計画に沿って実際に運用を行ない必要とする情報をエンドユーザが得る工程である。この工程で特に大事なのはソフトウェアに内在している各種のバグが顕在化し運用上に障害を発生した時に、事前に設計してある方法によって迅速な対応をこのソフトウェアの運用を担当している要員によって行なう事である。つまりソフトウェアに障害が発生した時に迅速な対応がとれるか否かが、そのソフトウェアの評価を高くする

か低くするかの大きな影響を持っているので細心の注意を払って運用する工程である。更に大事なことは、いくら運用が正確に行なわれたり、立派なソフトウェアが構築されたとしても、外部環境の変化やソフトウェア構築上の前提条件の変化によってソフトウェアの修正作業が発生する工程でもある。この場合ソフトウェアのどの部分を修正し、最終的には単体プログラムのどの部分を修正する必要があるかなどを分析し、正確な修正作業を行なう。しかも修正した結果が他の要素に影響を与えていないかを十分検討し、修正作業結果が正しくなされ正常にそのソフトウェアが動作するか否かなどのチェックを行なう工程でもある。開発工程には新らしくソフトウェアを構築するといった創造の喜びが存在しているが、一方修正作業は比較的裏方的な性格があり、しかも修正に対して一度ミスを犯すとその影響が大きく細心の注意を払う必要がある。其の上現場作業の中に密着したソフトウェアの場合、エラーが発生すると完全に現場の作業がストップしてしまう場合もあるので、トラブル・シューティングや修正作業に対して緊急性が要求される場合も多い。

### 1.2.3 ライフサイクルとドキュメント

ドキュメンテーション技術はソフトウェアの生産性向上と品質を高める為の重要な技術をなしていると思われるが、しかし一般的には重要性の認識はそれほど高くなくドキュメント作成作業を余計な作業と感じたり、或いはドキュメントの作成作業を実施することによってかえって生産性が落ちるといった全く正反対の認識を持っている人々も多数存在している。その為になし十分なドキュメントがライフサイクルのかなりの部分を占めている運用、保守段階に多くの問題を投げかけている場合が多い。しかもライフサイクルの各工程で作成される各種のドキュメントがソフトウェアの品質管理や進捗管理に重要な役割を果たしているという

認識が薄い為に十分なレビューをおこたったり、途中でシステム内容が変更したのにドキュメントを最新状態にしないまま運用、保守工程に引き渡してしまうといったケースが多い。その結果次々に蓄積されるソフトウェアを不十分なドキュメントでメンテナンスしていかなければならず、しかも効率化がなかなかはかれない為に多くの要員を抱えて人海戦術的にシステムの修正要求に対処しているのが現状である。

この事が増大しているシステム化ニーズに十分な開発要員を振り向けることが出来ない大きな要因でもある。勿論ドキュメントが十分揃っていれば、これらの問題がすべて解決出来る訳でなく、保守の生産性向上は要員の能力や過去に培ってきた経験とも大いに関係がある。

又開発段階で得た情報や作業結果が開発用のドキュメント上に時間の経過とともに具現化されており、開発が完了し次の工程である運用保守工程に移管するにあたって開発要員がそのまま残らない限り、運用、保守要員にとってこれらのドキュメントがソフトウェアの内容を知る唯一の情報源であり、かつ重要な引継ドキュメントである。つまり、これらのドキュメントを通して蓄積された情報の移転が可能となり、しかも作成されているドキュメント内容の良否が開発情報の移転や今後のエンドユーザの修正要求に対処する保守作業に大きな影響を及ぼすことになってくる。そこでこのドキュメント体系を運用、保守段階の作業の中で、必要とするドキュメントまでも含んだライフサイクル全般を通し生産性という観点から十分見直す必要がある。次にドキュメントが持つ主要な目的について記述する。

#### (1) ドキュメントの持つ目的

##### ① 関係者間の意志疎通をはかる。

最近のようにソフトウェアが大規模、複雑化してくるとこれにたずさわってくる要員も非常に多くなってきて、要員間の意志疎通をはかる重要なツールとしてこのドキュメントが活用されている。つま

り大規模ソフトウェアの開発には、システム・エンジニア、プログラマ、オペレータ、管理者、エンドユーザといった様々な人々が参加することになり、しかも人数が増加してくると、それぞれの要員が育ってきたバックグラウンドや経験も異なってくる。その為に必ずしも常識的な事が他の人にとって常識的でなくなってくる為にお互いが理解出来る分り易い言葉でドキュメントを作成してコミュニケーションの円滑化をはかり開発効率の向上をはかることが重要な課題となってきた。

② 多人数に依る同時併行開発を可能にする。

複雑で大規模なソフトウェアを多人数で同時に開発していく為にはドキュメントを通してソフトウェアを理解させる必要があり、しかもそのソフトウェアに要求されている機能を分割して開発するには、分割された機能の相互間の関連をドキュメント上に十分記載する必要がある。多くの要員が関係するプロジェクトを推進するにはこのドキュメントが重要な役割をになっており、これの記述方法に十分注意をする必要がある。

③ プロジェクト管理を容易にする。

前述した様に、ソフトウェアには目に見えないという特徴があるが、小規模のソフトウェアを開発する場合には十分なドキュメントなしで個人個人の頭脳の中にしまい込まれた各種の情報を利用する事によってソフトウェアを組立てることが出来る。しかしソフトウェア規模が大きくなると各種のドキュメントが重要な役割を果たすと同時にそのドキュメントが作成された事によって開発に必要とされる作業が確実に実行された事が分る。更に大きな作業の終了を意味する各種のドキュメント、例えば基本設計書が作成された事によって基本設計段階が完了したことが分る。つまりソフトウェア開発での進捗管理として、要員の作業結果をドキュメント上に表現する

事によってソフトウェアの不可視性を可視性のあるものに変換し、それがどの程度完成しているか否かで進捗状況を把握しようとする方法である。

(2) 各工程に於いて作成される代表的なドキュメント例

ドキュメントを作成する場合、特に注意しなければならないのは各工程に於いて作成するドキュメントの使用目的が異なる点である。そこでそのドキュメントが何の為に必要であるとか、誰れを対象にして作成するのかといった意味づけを十分認識しながらドキュメントを作成する必要がある。

例えば、エンドユーザとソフトウェアを開発する要員との間に於ける相互確認用として作成される要求定義書は、エンドユーザが要望しているソフトウェア内容の確認という意味からしてエンドユーザが十分理解出来るように書かれていなければならない。つまりコンピュータ用語などは出来るだけ避けて平易な言葉で誤解のないように明確に記述する必要がある。又ドキュメントを作成する上で常に注意しなければならないのは、このドキュメントを何の為に作成するのか、誰れに読ませるドキュメントであるのか、何時の時点に於いて作成するのかなどである。

それぞれの工程で現在一般的に作成されているドキュメントを工程対応毎に記述してみると次のようになる。

- ① 要求定義段階……ソフトウェアの開発提案書，外部仕様書
- ② 設計段階………基本設計書，詳細設計書，内部仕様書
- ③ プログラミング段階…プログラム仕様書，プログラム検取書
- ④ テスト段階………テスト仕様書，テスト結果報告書
- ⑤ 運用保守段階………運用作業指図書，運用マニュアル，保守マニュアル

(3) ドキュメント作成上の問題点



ドキュメントの重要性については、最近ではかなり多くの要員が認識を持つようになってきた。しかし現実には不完全なドキュメントが非常に多い。この理由として考えられる事は、ソフトウェアの開発、運用保守といった工程での作業の中にこのドキュメント作成作業が容れ込んでいない為と思われる。ドキュメント作成を特別の時間をとって作成するという意識でなく、自然な流れとして作業の中に組み込まれている事が重要である。理想を云えば、ソフトウェアの自動化システムのようにドキュメントを作成しなくてもソフトウェアが開発される事が望ましいが、現在の技術レベルではまだまだ不十分であるので、着実に、もれがなく正確なドキュメントを効率良く作成することが、開発効率向上をはかる大きな要因でもある。つまりドキュメントが十分作成されないという問題点を解決する為にはしっかりした体系と確実にドキュメントが作成され仕組みを考える必要がある。ドキュメント問題には一般的に次のような点がよく議論されているので、これらの問題点を一つでも解決する事が大いなる前進でこれらの問題に真剣に取り組む必要がある。

- ① 文書化が十分徹底しない。
- ② 文書化が体系化されておらず非効率である。
- ③ 標準化が不十分
- ④ ドキュメントに記載した内容が新らしく変更する必要が発生したにもかかわらず更新されない場合が多い

### 1.3 高度化、効率化の概要

景気が長期にわたって低迷を続けている為に、企業間に於ける競争は益々その激しさを増し経営者にとって企業経営の効率化や質的向上といった体質強化が重要な課題となってきている。その結果、設計製造部門、事務管理部門を始めとして研究開発部門迄をも含む全社的なシステム化ニーズを一

段と喚起する事になった。一方対外的にはエレクトロニクス技術の高度化によって急激なハードウェア価格の低廉化を招来し、コンピュータ利用の裾野を広げ新たな可能性を引き出してきている。

更にコンピュータ部門には、従来の情報処理技術の高度化に加えてニューメディアや回線解放によるネットワーク構築といった新技术を導入する事により、より高度な機能を持ちしかも利用者にとってユーザビリティの高いソフトウェアの提供が一段と期待されている。しかしこういったシステム化ニーズの増大にもかかわらずコンピュータ部門では十分対応が出来ずバックログとして数年間も開発待ちの状態の企業も出てきている。其の上開発段階での品質保証が十分でない為に問題点が後工程の運用、保守段階に持ち込まれてといった悪循環が十分断ち切れず運用、保守段階でのサービス・レベルを一層低下させている。又これに加えてソフトウェア・コストの高騰や生産性の伸び悩みが一層問題を深刻化させている。この為にここ数年開発、運用保守の高度化、効率化が大きく叫ばれているゆえんでもある。

### 1.3.1 基本的な考え方

ソフトウェアに対する需要は年率40%以上の伸びを示していると云われているが、この需要に対して十分答えらる抜本的な対策をとらない限りバックログを全面的に解消することは困難と思われる。其の上悪いことに日々、新しいソフトウェアが社内に蓄積され続けている。例えば富士通が行なったアンケート調査に依ると、大型ユーザ61社の中で、保有しているプログラムが1979年から1982年の4年間で約1.7倍の伸びを示している。ところが一方コンピュータ部門の要員数の伸びは1979年から1982年の4年間で1.3倍しか伸びておらず要員1人が面倒をみなければならないプログラムは確実に増大している。しかもコンピュータ部門の要員増はそれほど多くは望めないのが実状であり、要員1人当りの負荷は確実に増大傾向を示している。

そこで高度化，効率化の基本的な考え方として以下に述べることとする。

(1) 標準化の推進

① 手順について

工業製品が産業革命以降，飛躍的にその生産性を向上させ，大量生産が可能となってきたのは標準化の考え方と動力の導入によるものであるが，最近では其上消費者の多様化するニーズに合致させた高品質で低価格なものを提供するように求められてきている。そこで部品の標準化を進めその組合せによって消費者ニーズに合致したものを作って提供する方法をとっている。この方法の根底を支えているのが標準化の考え方であり，しかも多くの製品の間での部品の共通使用を可能とする高度な設計技術とそれを同一の生産ライン上に乗せ効率良く生産出来る生産技術とが確立されて始めて達成出来たものである。ところがソフトウェアの開発状況を眺めてみると，

- ㊶ 納期の遅延
- ㊷ 目標とした性能が十分発揮出来ない
- ㊸ 運用している最中でのエラーが多発し安定性に欠ける。
- ㊹ 高い費用を投じて開発したソフトウェアが利用者のニーズに合致しない。
- ㊺ 開発したソフトウェアが十分その効果を発揮しない

といった問題が多発している。しかも最近のソフトウェアは益々高度化，複雑化，大規模化の様相を帯びており，それらの問題点を一層顕著なものにしている。更にソフトウェア規模が大きくなればなるほど多くの要員が関係してくるので，しっかりとしたプロジェクトを推進する必要がある，しかも要員の間では，知識や経験がそれぞれ異なっており，コミュニケーション問題がしばしば大きな問題となってくる。こういった問題を解決する為にソフトウェアに関連のある人々に対して誰れが，何時，何を，どの様な手順で作業を実

行していくかを明確にした方法論を採用する必要がある。その為にはソフトウェアの開発，運用，保守といった作業に標準化が必須であり，しかもこの場合或る部分だけの効率化をはかっても必ずしも全体の効率化につながらない場合があるので，ライフサイクル全体を見通し，それぞれの工程内に如何なる作業があるかを明確にして，具体的な形での作業標準を設定する必要がある。又，標準化が十分確立されていると，或る程度の教育を受ける事によって，誰れでもが標準化の手順に沿って作業が実施出来，しかも抜けのない均質なソフトウェア生産が可能となってくる。

## ② ドキュメント

ソフトウェアを可視性のあるものにするツールとしてドキュメントがあるが，この場合でも無統一に書かれていたのでは，関係する要員の間でも十分な情報を効率的にパスすることが出来ない。又ドキュメントを書く事によってソフトウェアの可視性は実現出来たが，そこに記述するためのルールがないと個人的な特徴が発揮されてかえってコミュニケーション上の効率低下につながる所以このドキュメント作成にあたっては十分体系のとれたドキュメンテーションの標準化を導入する必要がある。この場合誰れが見てもこのドキュメント上に書かれている内容を正しく理解する為にフォーマットの統一化をはかる必要がある。又要員の個性があまり発揮されないように記入要領などを作成して記述方法も標準化する必要がある。更にこのドキュメントは工程内で実際に作業した結果をドキュメント上に具体化するのに利用されるので確実に作業が進行している事を目で確認することが出来る。その為のプロジェクト管理者がプロジェクトの進捗管理を行う重要な道具としての役割を持っている。別の視点から眺めてみるとエンドユーザとソフトウェア開発者との間で目的としているソフトウェアの仕様をドキュメント上に表現するこ

とによってお互いがその内容を検証することが出来る。しかもプロジェクトが長期にわたってくると、ソフトウェアの開発要員もその開発過程で得られた情報を忘れてくる事もあるのでその備忘録にも利用出来る。このようにドキュメントは管理者、システム・エンジニア、プログラマ、エンドユーザといった人々にそれぞれ異なった目的を持っているがプロジェクト運営には必要なものである。但しドキュメントを作成する場合、特に注意しなければならないのは、人間である以上ミスを犯す事があるので、ドキュメント中に間違いの発見都度タイムリーにドキュメントの内容を更新しておく必要がある。ドキュメントが最新状態にないと折角作成されたドキュメントも使用されなくなり最終的にはプログラム・ジャーナルが唯一の正しいドキュメントとなってしまうおそれがあるので更新のルール、保存方法、保存規定などを設定してドキュメントがいつでも利用出来るように環境条件などを整備しておく必要がある。又ソフトウェアを開発するにあたってドキュメントを作成する作業時間がかなりの割合を占めているのでこれを如何に効率良く行なうかがソフトウェアの生産性や品質向上での大きな鍵を握っているのでドキュメント問題が今後の重要な検討課題でもある。

## (2) ツールの導入

コンピュータ屋の紺屋の白袴と良く云われるようにプログラミング段階、テストラン段階を除くとほとんどコンピュータを利用しないで紙と鉛筆を利用してソフトウェアの開発を行なっているのが現状である。しかし人間の行なっている作業内容を十分分析してみると人間が直接自分の手で行うよりはるかに効率的で品質の良いものになる作業が存在していることが分る。そこでこれらの作業をツールに代替させることによって開発効率を向上させる事を目的に開発されているのが開発支援ツール群である。これらの個々のツールは基本的には、ソフ

トウェアの自動化を狙っているが、現時点での多くのツールは部分的な工程に於て効率化を狙っているのが多いので、それらのツールを導入して得られる効用は必ずしも目立ったものになっていない。しかしソフトウェアの規模が大きくなるにつれて開発の際には多くのツール群を導入して大きな効果を発揮している例も多数発表されている。但しこの場合でも工程に於ける部分的な作業分野での効率化促進には大きく貢献しているが、しかしこれらのツール群を単純に寄せ集めて利用するだけでは、操作性や一貫性の面から眺めてみると必ずしも全体的な効率向上に直接にはつながらない場面も出てきている。そこで今後ツールには新技術などをも十分取り込んだ個々の作業領域でのより高い生産性の追求と同時にライフサイクル全体を通した一貫性、管理性、使用性などを具備した機能のツールが望まれている。現在ライフサイクルの各工程で比較的効果を発揮しているツールとして次のようなものがあげられる。

- ① 要求分析段階…支援ツールを利用した要求仕様の定義活動
- ② 設計段階……ソフトウェアの自動設計、ドキュメントの自動化
- ③ プログラミング段階…プログラムの自動生成、テストデータの自動生成、プログラム・ドキュメントの自動生成
- ④ テスト段階……支援ツールを利用したテストデータの自動生成
- ⑤ 運用、保守段階…ドキュメントの自動生成、修正による波及分析の自動化

これらの自動化ツールすべてが実用化段階にあるわけではないが、今後のツールは個々の工程での自動化だけを指向するのではなく、これらを統合管理し、一貫したソフトウェアの開発・運用、保守作業を支援する環境を設定し、しかも統一性のとれた全体としても使い易く最適化がはかれる自動化ツールにする必要がある。

### (3) ソフトウェアの部品化の推進

ソフトウェアの品質を保証する為には全く新しいプログラムを作らないことだと極論する人もいるが、ソフトウェアの品質、生産性を向上させる為には出来るだけ、既に作られ現場で実用に供されている検証済のプログラムを利用することである。この考え方から発生してきたものにソフトウェアの部品化の考え方がある。ソフトウェアの部品化を考えるにあたって再利用出来る部品となるものに次のものが考えられる。

- ① システム要求の部品化
- ② 設計仕様の部品化
- ③ プログラムの部品化

などがあげられる。これらの中でシステム化要求を部品化するにあたっては開発方法論と非常に深い関係があり、この方法論がまだ十分確立していない為にまだそれほど成果として十分見るべきものがない。又部品として如何なるものを蓄積して再利用していくかが不明確である。ただ云えることは企業内で十分検討した方法論を継続的に採用し、蓄積することが必要である。又その蓄積された部品を実際に利用していく過程で何を部品化するのが適当であるかが判明してくるものと思われる。要は如何にして部品化していくかの思想をもって継続的に実施していくかが重要である。更に設計仕様の部品化はプログラムの部品化と一体となっている場合が多く、プログラムの部品化を推進させる為には前の工程である設計仕様の部品化を十分想定した上で考える必要がある。現在一番多く利用されているものにプログラムの部品化があるが、この場合注意すべき点は現在迄に既に蓄積されているプログラムの中から如何にして最適な部品を抽出するかである。又抽出して蓄積するプログラムの大きさをどうするかによっても部品の再利用率に大きく影響するので、企業の開発方式や環境に合致した部品分割基準を設定して蓄積する必要がある。又部品をあまり小さくし過ぎて

部品が非常に多く蓄積されてしまったために必要とする部品を探すのに非常に手間がかかってかえって手作りで作った方が早かったというような笑えない話もあるので、部品の蓄積方法や検索方法などには十分注意して部品化を推進する必要がある。又ハードウェアの製造に於いては部品化の考え方が進んでおりユニットとかパーツといったように階層的になっておりそれを必要なレベルで利用出来るようになってきている。そこでソフトウェアの場合も階層的な部品化を進めておき必要な時にどのレベルのどの部品を選択し、利用していくといった再利用化を推進する必要がある。この方法は開発環境の改善や方法論の採用といったものによって発揮される効果よりはるかに有効性を発揮出来る方法なので、今後は益々一層多く活用されていくものと思われる。但しこの場合、基礎になっているのが標準化の考え方であり、これが徹底されていないとなかなか必要とする部品が蓄積されないので標準化の推進と一体となっていくのが効果的な方法と思われる。

#### (4) ソフトウェアのライフサイクル全体に対しての品質管理の推進

ソフトウェアとハードウェアを比較してみるとソフトウェアには、  
①見えない、②個人の能力に極端に依存する。といったハードウェアにはない特徴を持っている。これ品質管理という観点から眺めてみると

- ① 目視検査が適用しにくい。
- ② ハードウェアのように物理的な計算手段が存在していない。
- ③ 個人の能力に依存する部分が多く出来上ってくる製品にバラツキが非常に多い。
- ④ 品質に対する評価尺度が確立していない。

といった特徴を持っており、この事がソフトウェアの品質管理を一層困難なものにしている。其の上ハードウェアと異なってソフトウェアには上記のような特徴があるので品質管理が導入出来ないのだという



きめつけが考え方の根底に存在している。そこで視点を変えて、工業生産されるハードウェアも始めて世の中に送り出された時には、確かに物として目視出来るが、品質に対する評価基準も何もなかったはずである。それが近代的な生産技術の確立や永年にわたる品質管理の追求によって世界でも第一級の品質の高い製品を作り出す事に成功したわけである。この考え方をソフトウェアの運用、生産の場に最近では多くの企業が導入して除々にではあるが効果を発揮してきている。しかし依然として品質管理を導入すると費用が増加するといった誤った考え方が存在しておりまだまだソフトウェアの分野に定着しているとは云い難い。しかもソフトウェアの分野でごく当たり前となつて行なわれている設計段階とプログラミング段階での手戻り作業などは、生産性の向上を大きくそ害するものであり、品質管理で良く使用されているチェック・リストなどを十分整備して設計上の詰め甘さなどをミニマイズし、かつ手戻り作業をなくすといった努力の積みかさねが全体的な生産性向上につながってくるものであるという認識を十分持つ必要がある。又ソフトウェアの品質を低下させている原因として

- ① コーディング・ミス
- ② 仕様固めのもれ、不備
- ③ 設計上のもれ
- ④ インターフェイスの誤り

といった技術力の不足や人に起因するものが多くあげられている。この場合、人に係る部分での品質を向上させる方法として、出来るだけ人が関与しないように作業をツールなどに代替させたり、管理目標などを設定して要員に品質の重要さを徹底させる方法などがある。つまり全社員に品質の重要性と品質向上が生産性の向上に直接的につながるものであるという認識を徹底させ総力をあげてこの問題に取り組むことが生産性を大いに向上させる重要な方法でもある。

### 1.3.2 効率化のための具体的アプローチ

前述した様に多くの企業が1～2年にわたるバックログを抱え悩んでいるが、其の上月日の経過とともに多くのシステム化ニーズが更にバックログ上に堆積してしまい、もはや従来の延長線上での効率化方法だけでは限界にきている感じがする。しかもコンピュータ要員の大巾な増員はほとんど期待出来ない状態である。そこで従来からとられてきた「コンピュータ関連の問題はすべてコンピュータ部門に一任する」という姿勢を一步前進させて、コンピュータ部門にふりかかっていた負荷を一部他の部門にも肩代りさせ、集中から分散化へといったコンピュータ関連業務の分業体制を推進させる必要がある。更にはエンドユーザから出てくるシステム化ニーズを単にエンドユーザが希望するから取り上げるといった姿勢でなく、企業内で持っている有限のリソース（人、物、金）を全社的な視点から有効に活用する為にソフトウェア問題に関して計画管理を強化する必要がある。

#### (1) 分業化の推進

##### ① 外部企業の積極的な活用

この方法は従来からも行なわれていた方法であるが、単なる力仕事を外部企業に委託するが多かった。しかし最近では外部の専門家集団も相当な経験を積んできており、自社で開発するよりは、はるかに高品質で安い費用で作成してくれる企業も出現してきており、ソフトウェアのライフサイクルのどの部分を分業化政策で委託するのだといった前向きな活用方法を考える必要がある。この場合委託会社の信頼性や質的なレベルなどを事前に十分調査しておく必要がある。又受入れ側の企業としても標準化やドキュメンテーションなどを十分整備しておく必要があり、しかも両者間の責任範囲や委託業務範囲などを明確にしておく必要がある。

##### ② パッケージの導入

外部の業務パッケージの導入は、コンピュータ化ニーズがあるにもかかわらず開発待ちしなければならないといった機会損失を削減するのみならず、コンピュータ部門の作業が全く無くなるのではないが、大巾な省力化効果が期待出来る。しかし日本でのソフトウェア・パッケージの流通状況を眺めてみると実際のところ米国に比べて流通状態は非常に遅れており、まだまだ企業側の受け入れ環境が十分でないように思える。しかし最近になって、海外からのパッケージも含めてかなりの種類のパッケージが市場に提供されてきており、しかも品質的にはかなり高いものが出廻ってきている。更に導入する側の意識もかなり変わってきており、良いパッケージの場合には、受け入れ企業の体制、組織などをも変更して、開発待ちの時間や開発期間に於ける機会損失をミニマイズする事を第一義にして導入するケースが増加してきている。パッケージを導入するにあたっては、パッケージの持つ要件（値段、実績、信頼性、機能、性能、使い易さ、ドキュメントの状況、カスタマイズにあたっての適応性、メンテナンス・サービス体制など）を事前に十分チェックする必要がある。又受け入れ側としては、受け入れる側の環境作りを十分準備しておかないと、パッケージとしては十分優秀であっても結局は利用されないといった問題も発生するので、受け入れ体制（受け入れ側のマネージャの意識改革、受け入れ現場の作業方法の変更に対する十分な根廻しと環境整備）を十分確立して、パッケージ販売会社との協同作業がパッケージ導入を成功させる大きな秘けつでもある。しかも最近のように企業間での競争が激化してくるとシステム化がより一層大きな武器となるので開発期間の短縮と開発コストのセイビングに大きな貢献を果たすパッケージの導入がかなり積極的に推進されてくるものと思われる。

### ③ エンドユーザ言語の導入

バックログの解消法として非常に有効な方法として前述したパッケージの導入があるが、それと同じ様に有効な方法として、コンピュータ部門に代ってシステム化ニーズを抱えている部門が独自の力で開発を進める方法がある。この場合、コンピュータ部門がなにもしないという訳でなくエンドユーザ部門が使える道具を提供したり、その開発方法の教育を行ったりして、その開発作業を支援する機能を果たす必要がある。この前提には全社的なデータベースの整備もコンピュータ部門によって事前に構築されていることが必要である。又バックログが増大している原因を詳しく分析してみると、従来の比較的定型的な業務のシステム化要求に加えて非定型的な業務も年々増大してきている。この事は今迄、コンピュータ化しにくいとあきらめていた人々が最近のハードウェア技術やオフコン、パソコンといったOA機器の低価格化によって身近かに、しかも比較的容易に利用出来る環境が出現してきたのをみて、非定型的な業務や個人レベルでの業務に積極的に活用に取り出したものと思われる。この場合パソコンを全社にばらまく代りに非定型的な業務でも簡単にユーザ言語を利用して自分の必要とする資料を作成出来る仕組みを全社に提供することによってバックログの解消をはかっている企業も多数出てきている。

#### ④ 企業内情報センターの確立

エンドユーザの性格上エンドユーザ言語のように比較的使い易い道具を用意してユーザ自身が開発出来る環境を提供してもなかなか自分の問題として利用しないのが通例である。そこで新らしく設立された情報センター内の要員によって、エンドユーザの開発サポート、コンサルティング、教育などを実施し、その結果ユーザ自身が自分の手で開発を推進させる事を目的にこの情報センターが設立されている。つまりコンピュータ部門が多くのバックログを抱えてど

うしようもない時に、コンピュータ部門に代ってユーザ自身が持っているシステム化ニーズを自分の手で開発出来るようにする支援部隊であって、社内の自給自足率を向上させる事が主な任務でもある。この考え方はカナダにあるIBMによって提案され現在世界中で多くの企業で採用され始めている。この方法は確かにエンドユーザのコンピュータ部門がなかなかシステム化してくれないといったいらだちの解消にも役立っている。又情報センターの設立によって次のようなメリットが得られるとしている。

- ① アプリケーションに十分経験のある要員自身が自分の問題としてシステム化に取り組むのでソフトウェア開発コストが削減出来る。
- ② バックログの解消と同時に今迄潜在化していたコンピュータ化ニーズを喚起し、しかもその開発を自身の手で行なう事が出来る。
- ③ エンドユーザ自身が情報センターのサポートを受けて自からが問題意識を持って考えるので企業のシステム化が一層促進される。

## (2) 計画、管理の強化

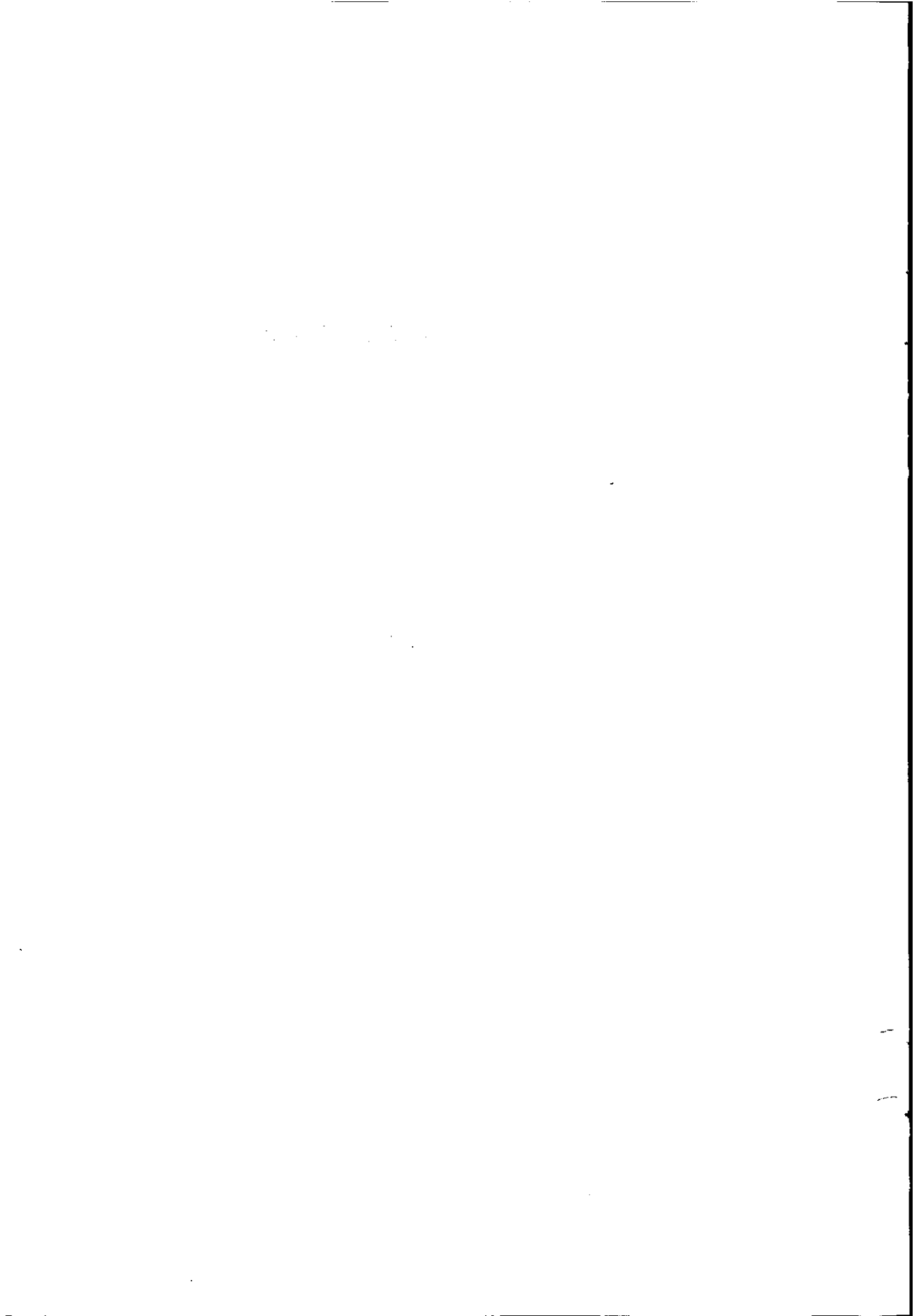
システム化ニーズがそれほど多くない時期にはニーズの発生都度それなりのプロジェクトチームを結成して対応してきたが、現在の様に多くのシステム化要求が一斉に発生してくると、従来のような、場当り的な方策では有限のリソースのもとでは、ほとんど限界にきている。そこで各部門から要請されているシステム化ニーズを全社的なレベルでその必要性を分析評価してニーズの解決順位を設定する必要がある。この場合コンピュータ部門のみで順位付けを決定するのではなくエンドユーザ部門をもまき込んで全社的なコンセンサスを作り上げる必要がある。それによって今後の開発計画と体制をエンドユーザ部門にも十分知らしめて全社的な協力体制が得られることが社内のシステム化を

推進していくにあたって非常に重要な事項でもある。つまり有限のリソース（人、物、金、時間）などを最高に効率良く活用する為には全社的なコンセンサスのとれた目標を設定して開発していくのが最良の方策でもある。従来の開発のやり方で行なわれた結果よく発生する問題点として

- ① トップや利用部門の協力が得られない。
- ② ニーズ間同志の重要性の是非が評価出来ない
- ③ ニーズのレベルが大小様々で多過ぎる為に体系的な整理分析が出来ない
- ④ 各部門の利害対立の調整が出来ない
- ⑤ 開発スケジュール、体制の合意がなかなか得られない。

などがあるが、これも開発計画段階に十分な全社的なコンセンサスが得られていない為に発生する問題でもある。そこで最近はこの計画段階に焦点を当てて、今後のシステム化が効率良く行なえ、しかも全社的なコンセンサスが得られる方法論を提供しているフレーム・メーカーもある。例えばIBMのBSP (Business System Planning)、富士通のEPG (End-user-oriented Planning Guideline) C-NAP (Custom-Needs Analysis Procedure)などが良く知られており、これらの方法を有効活用するのも一つの方法でもある。いずれにしても各企業の持っているリソースは有限でありこれを全社的なコンセンサスのもとで如何に有効に活用していくかが重要であり、折角のシステム化しようとする意欲の芽をつみかねないのでこのモチベーション問題を如何に上手にさばっていくかが今後の重要な課題でもあり、それを行なうにはこの計画、管理を上手に採り入れることである。但しあまりこの計画、管理を前面に押し出していくと折角のシステム化の高まりを妨害しかねないのでこのかね合いに十分注意する必要がある。

## 2. ソフトウェア開発計画の現状





## 2. ソフトウェア開発計画の現状

オンライン・ネットワーク・システムの進展やデータベース・システムの拡大にみられるように、コンピュータの利用はますます高度化しており、従来のシステム開発に比較して経費および作業負荷が増大しつつある。

他方、OA（オフィス・オートメーション）およびLAN（ローカル・エリア・ネットワーク）の波はパーソナル・コンピュータのセクション・マシン化を進め、システム開発および運用の急激な拡大・分散化をもたらしている。

このような状況にあって、全社的な整合性を見地に立ったシステム化および効率的なシステム化が急務となっており、加えてそこには信頼性・生産性も合わせ求められている。

高度情報社会の到来を考えた場合、システム開発は今後もさらに増大しつづけることが予想され、信頼性・生産性向上の観点からもシステム化の1つの基準となるような方法を確立する必要に迫られている。この点について、コンピュータ・ユーザのコンピュータ部門はどのような方法で対処し、乗りきろうとしているのであろうか。本章ではコンピュータ・ユーザのコンピュータ部門におけるソフトウェア開発計画の現状と課題について、調査結果に基づいて述べることにする。

### 2.1 アンケート調査結果の概要

調査はアンケート調査、事例研究、企業への訪問ヒヤリングによった。アンケート調査は9月1日から9月30日の期間に実施した。調査対象は大手コンピュータ・ユーザを中心に、無作為抽出法により600社を選出した。その結果、114社の回答を得た（回収率19%）。ただし、個々の項目については回答数が異なるので、その数字を各図表に示した。訪問ヒヤリングはアンケート調査終了後、その補足調査として、アンケート回答企業の中から5社、それ以外の企業から3社を選んで実施した。

詳細は次節以降で説明するが、本テーマの性格上、単純集計の回答よりも記述式の内容に興味深いものが見られるとともに、1つの方向性を引き出せるものが多かったように思われる。個々の設問の検討に入る前に、回答ユーザのコンピュータ部門の現状を概観し、分析の一助としたい。

(1) コンピュータ部門の人員構成

コンピュータ部門の人員は平均73.1人でシステム・エンジニア22.0人、プログラマー21.7人となっており全体の4割近くを占めている。しかし、回答のなかにはシステム・エンジニアとプログラマーの作業区分が明確でないものもいくつか見られた。

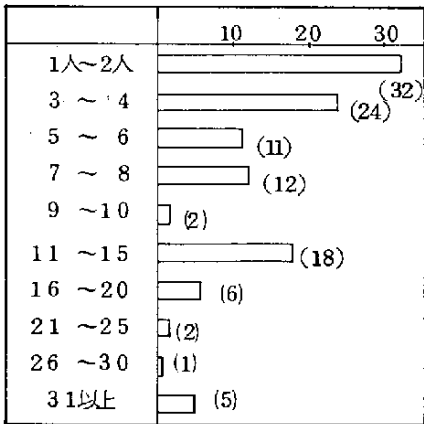
オペレータの数が9.4人と少ないのは、オンライン処理の増大と外部委託が主な理由だと考えられる。(表2.1)

各職種の人数別分布を図2.1に示す。平均値とは異なり、システム・エンジニア、プログラマーの人数が10名以下の企業がそれぞれ約50社と4割強を占めているのが特徴的である。

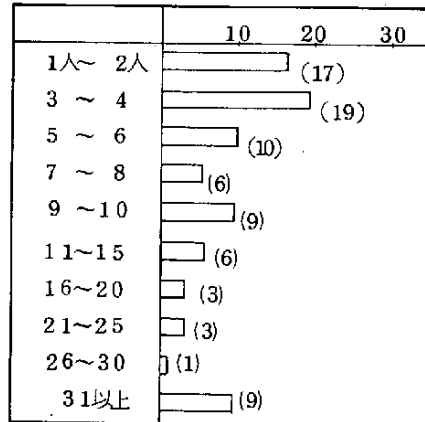
項 目	平均人数
管 理 者	8.1人
S.E	22.0
プ ロ グ ラ マ ー	21.7
オ ペ レ ー タ	9.4
キ ー パ ン チ ャ ー 等	11.9
計	73.1

表2・1 コンピュータ部門の人員構成(平均)

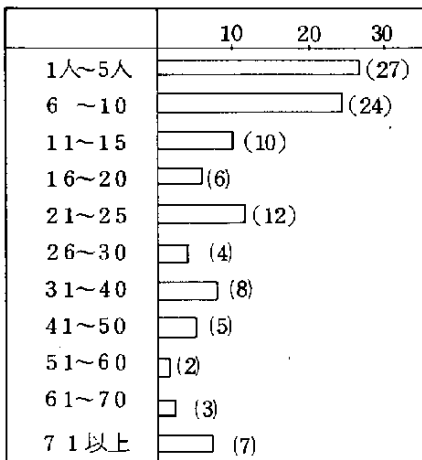
管理者



オペレータ



SE



プログラマ

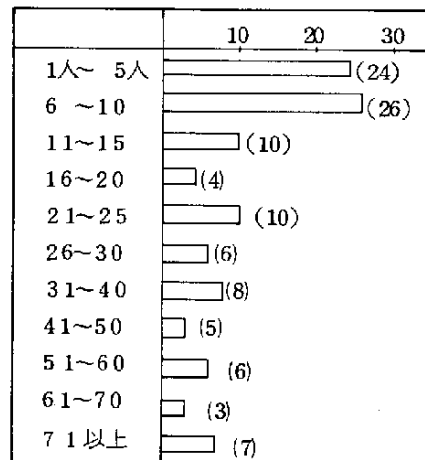


図 2.1 職種別の人数分布

(2) ソフトウェア開発予算

昭和58年度の平均は約3億6千万円で、最高は26億1300万円、最少は50万円とかなりのバラツキがある。また、コンピュータ部門の経費分担(経費負担元)の方法について、様々の考え方があることがうかがえる。

(3) 外部利用

外部利用しているという回答が88件(77.2%)あった。そのうち業務委託を行っているのが41件、外部要員を利用しているのが35件であった。

ヒアリング結果によれば、外部要員利用の中心はオペレータおよびパンチャーで、プログラマー派遣が次に多かった。ある企業では深刻なシステム・エンジニア不足に悩んでおり、システム化要求分析の段階から外部利用を実施しているケースもあった。

(4) 使用機種

主な使用機種をメーカー別にみると、IBMが48件(42.9%)と1位を占め、ついで日立26件(23.2%)、富士通24件(21.4%)、日本電気16件(14.3%)、ユニバック12件(10.7%)となっている。また、35件が複数のコンピュータ(表2・2)を利用しており、その比率は30.7%であった。そのうち19件が異なるメーカーの機種を利用していた。

	利用数	% (利用数/112)
I. B. M.	48	42.9
日立製作所	26	23.2
富士通	24	21.4
日本電気	16	14.3
ユニパック	12	10.7
その他	5	4.5

表2・2 メーカー別の利用機種

この設問は後で検討を加える技法・ツールの利用実態を知るうえで不可欠なものと考え、調査を実施した。

## 2.2 システム化要求の現状

本節ではシステム化要求について「要求部門」・「重点ニーズ」・「標準フォーマットの有無」の各項目をアンケート調査結果を中心に述べる。

### (1) システム化要求部門

システム化要求が最も多い部門はユーザ部門で、107件93.9%の高率に達している。(表2・3)

ユーザ部門	107 (93.9)
コンピュータ部門	5 (4.4)
トップ	1 (0.9)
顧客先	1 (0.9)

表2・3 システム化要求部門

N = 114

トップ(1件0.9%)およびコンピュータ部門(5件4.4%)が極端に少ない背景として、すでに第一段階のコンピュータ化が完了しトップ・ダウンの形での全社的なコンピュータ化の必要がなく、むしろユーザ部門が自らの業務改善等のためにボトム・アップ式の要求を出しているケースが

多いからである。ヒアリング調査においてシステム化要求の新規とメンテナンスの割合を、工数で比較して回答を求めたところ、メンテナンスが70~80%の割合を占めており、このことを裏づけている。

コンピュータ部門がシステム化要求をするケースとしては、「経営方針に基づくシステム化」や「全社的整合性の必要が生じた場合」および「システム化の不均衡を解決するため」などが考えられる。

## (2) システム化の重点ニーズ

「省力化が最大になるようにする」を1位にあげた回答が47件41.2%あり、3位までの合計でも109件95.6%と高率を示し、ほとんどの企業で「省力化」が重点課題となっていることがうかがえる。次に多いのは「費用対効果を最大にする」で1位にあげた回答が36件31.6%あり、3位までの合計でも96件84.2%となっている。

低成長経済・減量経営が呼ばれている現在、「省力化」にしても「費用対効果」にしても企業の最優先課題が組織の末端にまで反映している証左であると思われる。

「顧客サービスの向上が図れるもの」を1にあげた回答が19件16.7%、3位までの合計でも50件43.9%であった。また「処理効率を最大になるようにする」を1位にあげた回答が9件7.9%、3位までの合計でも55件48.2%となっており、上位2項とはかなりの開きがある。この理由として、1つには顧客サービスに関係のあるシステムが全体に占める割合は決して大きくはないこと、またもう1つにはオンライン・システムに比べバッチ・システムの場合にはシステムの処理効率がオンライン・リアルタイム処理のレスポンス・タイムのように目に見える部分が少なく、ユーザ部門にとってあまり意識にのぼってこないことなどが考えられる。

(表2・4, 図2・2)

## (3) システム化要求の標準化フォーマットの有無

51件44.7%が「ある」と答えている。(表2・5) この標準フォーマ

項目	1位		2位		3位		ウェイト付け
省力化	47件	41.2%	42件	36.8%	20件	17.5%	245点
費用対効果	36	31.6	39	34.2	21	18.4	207
顧客サービス	19	16.7	13	11.4	18	15.8	101
処理効率	9	7.9	13	11.4	33	28.9	86
他社より良いシステム	2	1.8	4	3.5	12	10.5	26
社員モラル	1	0.9	2	1.8	1	0.9	11
その他	0		1	0.9	9	7.9	8

表2・4 システム化の重点ニーズ

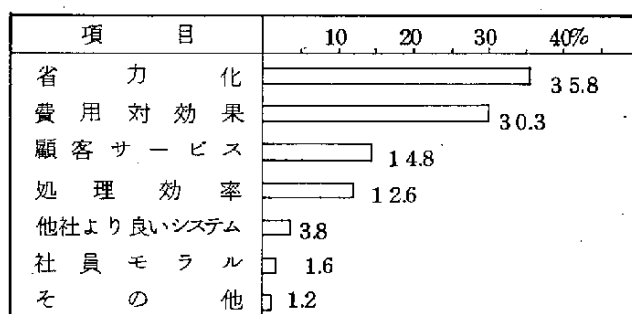


図2・2 システム化の重点ニーズ

ある	51 (44.7)
なし	63 (55.3)

N = 114

表2・5 標準化フォーマットの有無

ットは「機械化業務依頼書」・「システム開発依頼書」などと呼ばれ、システム化要求部門で記入する部分とコンピュータ部門で記入する部分とに分かれている。

システム化要求部門が記入する項目には、概ね次のようなものがある。

- ① システム化依頼部署名
- ② 業務名
- ③ 依頼区分（新規・変更等の区分）

- ④ 実施希望日（システム運用開始日）
- ⑤ システム化の目的
- ⑥ 現状の問題点
- ⑦ 効果
- ⑧ 処理サイクル
- ⑨ データ量
- ⑩ 予算額

コンピュータ部門で記入する項目には次のようなものがある。

- ① 作業着手の判定
- ② 本番実施見込時期
- ③ 作業期間
- ④ 開発費用見積
- ⑤ 運営費用見積

現実にはシステム化要求部門が依頼書を提出するまでに、コンピュータ部門をはじめ関連部署との打合せが行われており、その段階でシステム化の可否の検討は済んでいる。この依頼書が提出された場合にはシステム開発が実行されることになっており、これ以降の段階でシステム開発が取り止めになるケースはまれである。

システム化要求の詳細はほとんどの場合、依頼書に加えて添付書類で述べられている。システム化要求段階では、ツール・手法の利用はなく、ユーザ部門のシステム化の熱意および業務知識と経験、表現力に頼る面が多く、標準化できないのが現状のようである。

システム化要求の可否はシステム化要求部門の長が行い、コンピュータ部門に提出することになっている。

### 2.3 要求分析の現状

本節では要求分析段階での現状および問題点について、「要求分析の体制」



・「要求分析の技法・手法」・「ツール」および「積み残しとなる問題点」等の項目をアンケート結果より報告する。

(1) 要求分析の体制

要求分析の参加メンバーの内訳をみると、システム・エンジニアが107件93.9%、システム化要求部門が100件87.7%で、この両者を中心に構成されているケースがほとんどである。ついでコンピュータ部門が53件46.5%となっているが、これはコンピュータ部門の責任者が参加していると考えてよいであろう。プログラマーが参加メンバーとなっているケースは46件40.4%で、実作業担当者の理解度ひいては作業の円滑化を高める努力が試みられているものと思われる。トップの参加が8件7.0%となっているのはシステム化要求と同一の傾向を示しており、システム化の現状がトップ・ダウンの時期を過ぎたことの表われであろう。(表2・6)

(構成)

トップ	部門 コンピュータ	シ エ ン ジ ニ ア	プ ロ グ ラ マ	シ ス テ ム 化 部 門	外 部 コ ン サ ト	回答企業数
		○	○	○		26 (22.8)
		○		○		25 (21.9)
	○	○		○		25 (21.9)
	○	○	○	○		11 (9.6)
	○	○				4 (3.5)
○	○	○		○		3 (2.6)
						20 (17.5)

その他の組み合せ

N = 114

表2・6 要求分析の体制

(参加メンバー)

システム・エンジニア	107 (93.9)
システム化要求部門	100 (87.7)
コンピュータ部門	53 (46.5)
プログラマ	46 (40.4)
ト ッ プ	8 ( 7.0)
外部コンサルタント	5 ( 4.4)
そ の 他	4 ( 3.5)

N = 114

メンバー構成を各職種別に人数分布でみてみると、システム・エンジニアについては2～3名が59.8%と最も多くなっており、ついで1名が35.5%であった。システム化要求部門についても、2～3名が71%と最も多く、ついで1名が16%となっている。

プロジェクトによる作業も同様の傾向にあると思われ、大体5名前後の体制でシステム化が進められているようである。新規作業よりもメンテナンス作業の比率が高いことの表われであろう。(表2・7)

システムエンジニア		プログラマ		システム化要求部門		外部コンサルタント	
1人	38	1人	23	1人	16	1人	2
2～3人	64	2～3人	17	2～3人	71	2～3人	3
4～5人	3	4～5人	5	4～5人	12	4～5人	0
6人以上	2	6人以上	1	6人以上	1	6人以上	0

表2・7 メンバーの人数別分布

(2) 要求分析の技法・手法

要求分析に関連する技法・手法の知名度・使用頻度等についてのアンケート結果を表2・8に示す。

アンケートは連関図法・親和図法・系統図法・PDPC・YS法・ブレーンストーミング・面接調査・事務分析の各技法・手法について、「知らない」・「知っているが使っていない」・「分析の中心として使った」・「補助

連関図法 101	1.知っているが使ってない	35	親和図法 111	1.知っているが使ってない	58	系統図法 103	1.知らない	36
	2.知らない	25		2.補助的に使った	19		2.知っているが使ってない	27
	3.補助的に使った	23		3.試行したことがある	16		3.補助的に使った	15
	4.分析の中心とした	14		4.大変有効であった	10		4.分析の中心とした	12
	5.大変有効であった	9		5.知らない	7		5.大変有効であった	9
	6.一応の効果あった	8		6.一応の効果あった	6		6.試行したことがある	6
	7.試行したことがある	1		7.分析の中心とした	4		7.一応効果あった	6
	8.効果なし	1		8.効果なし	1		8.効果なし	1
P D P C 101	1.知らない	69	Y S 法 100	1.知らない	78	ブレインストーミング 109	1.知っているが使ってない	34
	2.知っているが使ってない	22		2.知っているが使ってない	21		2.補助的に使った	30
	3.分析の中心とした	3		3.補助的に使った	1		3.一応の効果あった	16
	4.一応の効果あった	3					4.試行したことがある	14
	5.補助的に使った	2					5.大変効果があった	13
	6.大変有効であった	2					6.分析の中心とした	11
	7.試行したことがある	1					7.知らない	2
	8.効果なし							
面接調査 109	1.分析の中心とした	60	事務分析 109	1.分析の中心とした	67	その他 109	1.分析の中心とした	
	2.大変効果があった	37		2.大変効果があった	30		BSP, CPS	3
	3.補助的に使った	17		3.補助的に使った	16		NUPS法	2
	4.一応の効果あった	14		4.一応の効果あった	11		PRIDE	2
	5.知っているが使ってない	9		5.知っているが使ってない	7		SDF	1
	6.試行したことがある	3		6.知らない	6		2.効果あり	
				7.試行したことがある	5		BSP, CPS	3
				NUPS法	1			
				3.一応の効果				
				ワークデザイン	1			
				BSP	1			

N = 111

表2・8 要求分析の技法・手法

的に使った」・「試行したことがある」・「大変有効であった」・「一応の効果があつた」・「効果はなかつた」の各項目に該当するかどうかをきいた。

記入があつた回答111件について知名度をみてみると、「面接調査」が109件98.2%と最も多く、つづいて「ブレンストーミング」107件96.4%、「親和図法」104件93.7%、「事務分析」103件92.8%の順となつた。「連関図法」は76件68.5%、「系統図法」は67件60.4%と過半数の知名度を獲得しているが、「PDPC」32件28.8%、「YS法」22件19.8%と極端に低くなつており、際だつた相違を見せている。

次に利用度についてみると、「事務分析」は67企業で分析の中心としてまた16企業で補助的に利用され、その効果についても「大変効果があつた」が30件、「一応の効果があつた」が11件とその評価も高い。「面接調査」は分析の中心として利用された件数が60企業と「事務分析」に劣るものの、その効果は「大変効果」が37件、「一応の効果」が14件と評価が最も高くなつてゐる。「ブレンストーミング」は知名度に比して利用される割合が低く、「分析の中心」が11件、「補助的」が30件となつてゐる。しかし、効果については「大変効果」が13件、「一応の効果」が16件と評価は得てゐる。「親和図法」はさらに極端で、「知つてゐるが使つてゐない」という回答が58件と過半数を超える企業が利用してゐない結果となつてゐる。その他の技法・手法についても利用度はかなり低く、「YS法」にいたつては補助的に利用されたケースが1件あつたにとどまる。しかし、「YS法」を除く技法について効果をみた場合、利用者からの評価は必ずしも低くはなくいづれの技法も利用者の40%近くから一定の評価を得てゐることがわかる。

これらの結果を総合して考えてみると従来からある「面接調査」・「事務分析」の技法がより多く利用されておゐり、同時に個別業務の分析によりフレキシブルに対応できているように思われるのに対し、「ブレンスト

ーミング」は補助的な技法として活用されているように思われる。これは「ブレインストーミング」が科学的な分析技法というよりも、むしろ問題解決技法として、目ざすべきシステムを考える場合の技法として利用されているものと考えられる。

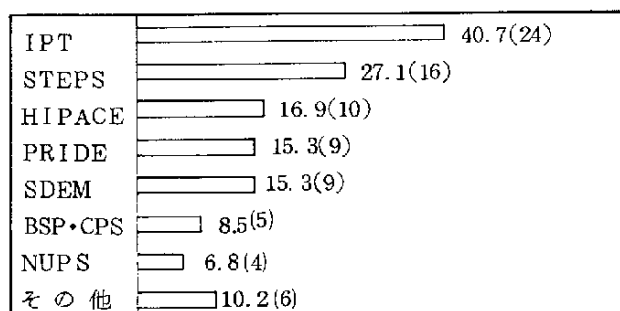
ヒアリング調査でも、機能分析を中心としたワークデザインの的手法による分析方法が多く利用されており、「ブレインストーミング」も補助的に利用し有効であったという回答を得た。

### (3) 要求分析のツール

要求分析で利用したツールはIPTが24件と最も多く、つづいてSTEPS 16件、HIPACE 10件の順となっている。これをメーカー別のユーザ比でみるとIPT (IBM) は50%、STEPS (日本電気) 100%、HIPACE (日立) 38.5%、SDEM (富士通) 37.5%となり、メーカーによる差異がかなり明瞭となっており興味深い。

また、各ツールの利用度が全体で59社51.8%と過半数を少し超えた程度で、標準的な方法があまり確立できていないことを表わしている。

(図2・3)



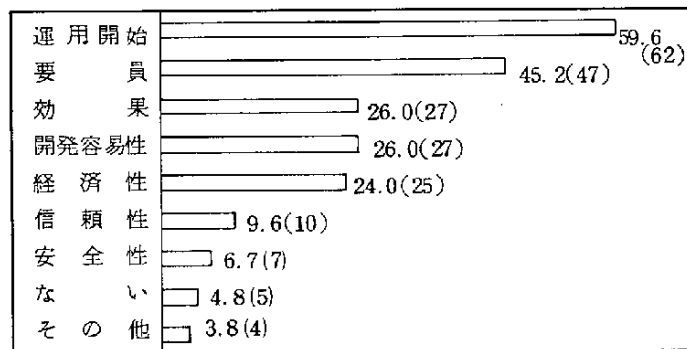
N=59

図2・3 要求分析で利用したツール

#### (4) 積み残しとなる問題点

要求分析の段階で積み残しとなる問題点は、「運用開始時期」が最も多く62件59.6%となっている。ついで「要員」が47件45.2%で第2位となっている。この2つは密接に関連しており、システムの開発規模や工程が、予算や要員によって左右されることおよびそれらが未解決のまま次のステップへ作業が進んでいることを表している。

「効果」・「開発容易性」がともに27件26.0%、「経済性」が25件24.0%となっているのは、システム化要求の重点ニーズ（1位「省力化」・2位「費用対効果」）が必ずしも解決されていないことを示している。これなどは、システム化要求および要求分析の各段階で、定量的効果を把握する方法を持ち合わせていないためにシステム化の実行可能性を比較検討できないことに由来しているように思われる。また、ゴーサインが出たからには何が何でもやり通すという日本的な対処方法もあるように思われる。（図2・4）



N=104

図2・4 要求分析時に残された問題点

問題解決のための代替案の検討は、類似システムの技術動向調査および汎用パッケージの調査を通じて行われている。しかし、そのまま適用することが困難だという理由から、採用されるケースはほとんどないようである。わが国におけるソフトウェア流通が進展しない現状が、ここからもうかがえる。従って、問題が発生したような場合には、種々の工夫をして切り抜けることが多い。このような場合には、システムを複雑なものにするケースが多く、開発の容易性に問題を残したまま開発計画立案に作業を進めることもある。

## 2.4 開発計画立案の現状

本節では、開発計画立案に際しての「計画の標準化」、 「開発計画の承認者」、 「開発計画立案に関連する技法・手法」および「ツール」、 「見積基準」等について、その実態を報告する。

### (1) 計画の標準化

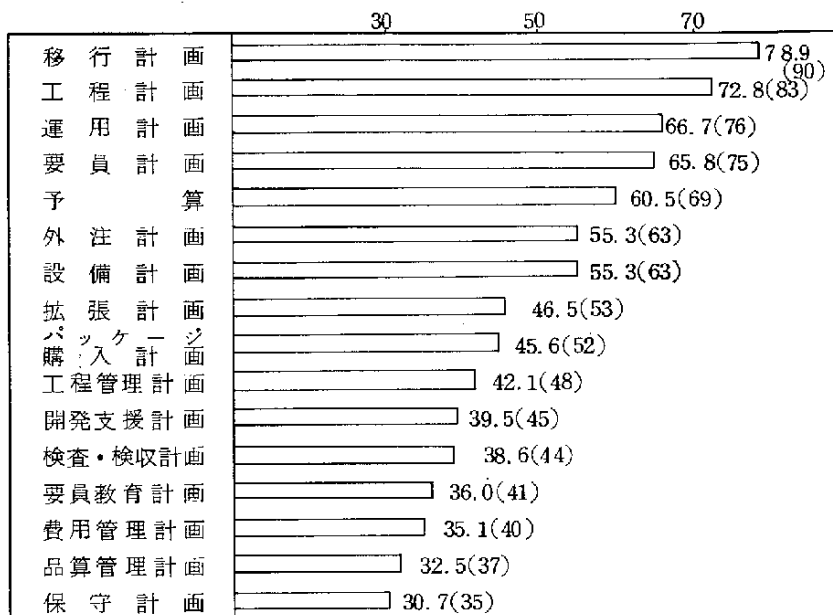
開発計画立案に際して、各種計画が「既に標準化されている」か、あるいは「その都度作成する」かについて設問をした。(図2・5)

全体的に見て、開発の都度作成されるケースが多く、すべての計画にわたって「標準化されている」を上回っており、この段階でも標準化できていないのが目につく。

開発の都度作成される率が一番高いのは「移行計画」で、90件78.9%であった。逆に「移行計画」が標準化されていると回答したのは15件13.2%にすぎなかった。その他開発の都度作成される割合が50%を超えたものとしては、「工程計画」83件72.8%、「運用計画」76件66.7%、「要員計画」75件65.8%、「予算」69件60.5%、「外注計画」および「設備計画」63件55.3%となっている。

一方、標準的に既に作成されているものについて見ると、1位が「要員教育計画」で38件33.3%、つづいて「要員計画」31件27.2%、「設

(開発の都度作成)



(標準的に既に作成)

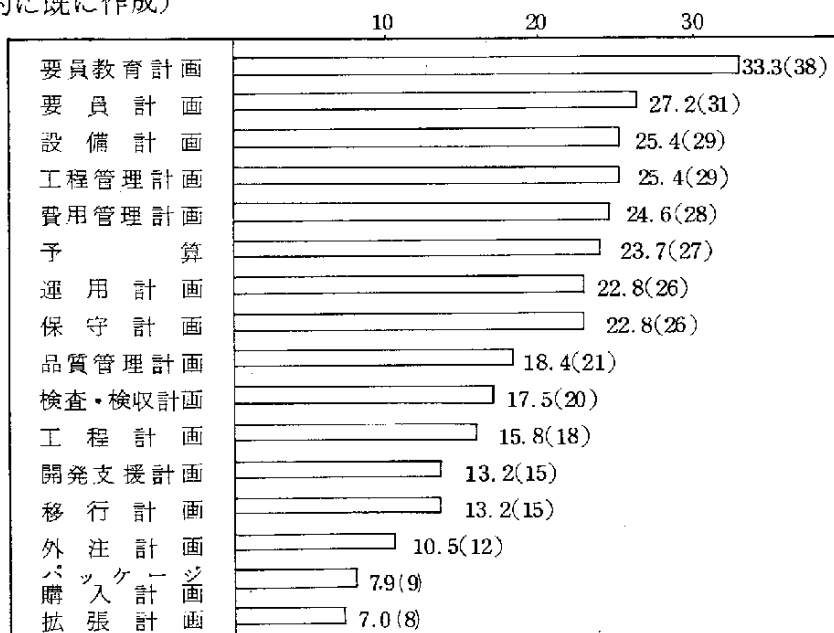
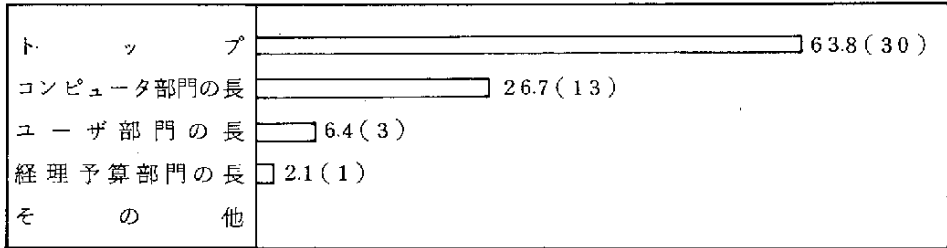


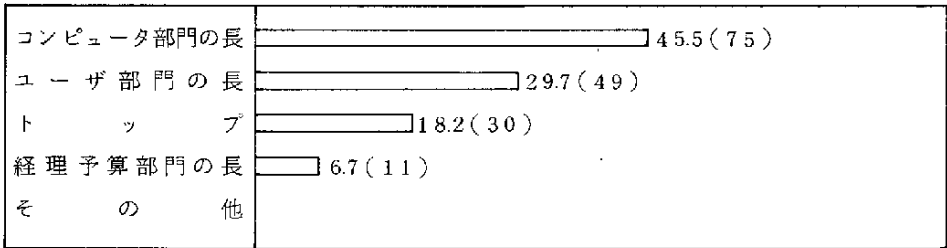
図 2.5 開発計画の標準化の状況



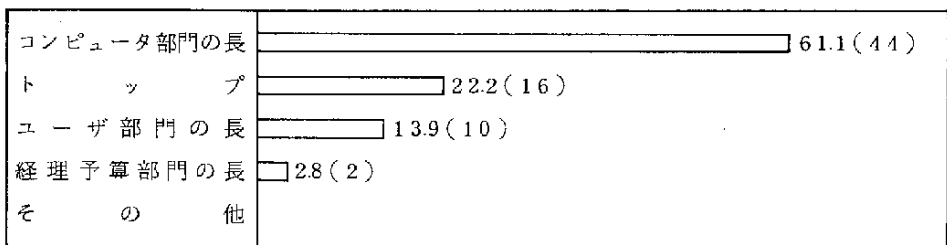
## トップが要求…… 47



## ユーザ部門が要求…… 165



## コンピュータ部門が要求…… 72



## 顧客先が要求…… 29

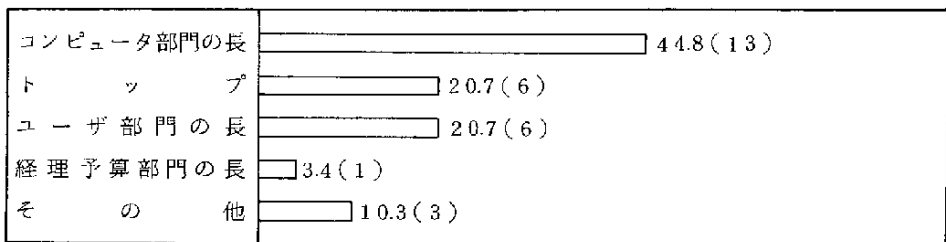


図 2.6 開発計画の承認者

備計画」および「工程管理計画」29件25.4%などとなっており、年度単位の計画が作成しやすいものが上位を占めているのが目についた。

## (2) 開発計画立案の承認者

誰が開発計画を承認するかを調査した結果、システム化要求部門がトップである場合に限り、トップ自らが承認者となるケースが最も多い(63.8%)。これはシステム化要求の内容が、経営方針等企业存立の基盤に深く関わっているからであろう。

システム化要求部門がユーザ部門、コンピュータ部門、顧客先のいずれの場合にも、コンピュータ部門の長が承認者となるケースが最も多い。これは、システム化要求分析から運用にいたるまでの各段階で、コンピュータ部門が中心となって作業を進めるため責任の所在を明らかにすることが主な理由だと考えられる。

一方でユーザ部門の長およびトップが承認者となるケースは、コンピュータ部門の作成した開発計画に対するユーザとしての承認と監視機能の発言が目的であると思われる。

ヒアリング調査では、システム開発の規模(主に金額)により承認者が異なるといった定量的で明確な判断基準を持つ企業もあった。(図2・6)

## (3) 開発計画立案の技法・手法

WBS・TRM・PERT・ハンディPERT・ガントチャート・マイルストーンチャート・トレンドチャートの各技法について、その「知名度」・「利用度」・「効果」等を調査した。(図2・7)

その結果、PERT・ガンチャートを除いて「知らない」という回答が非常に多く、これは全くの予想外であった。特に、TRM 73件67.6%を筆頭にWBS 63件58.3%、トレンドチャート 56件51.9%と回答者の過半数に知られていない状況である。

利用度についてしてみると、PERTは52件48.1%、ガンチャートは61件56.5%となっている。特に、ガンチャートは「良く用いている」が53

N = 108

W B S	知らない	58.3(63)
	知っているが使ってない	21.3(23)
	補助的に用いている	4.6(5)
	良く用いている	1.9(2)
	試行したことがある	0.9(1)
	大変有効であった	0.9(1)
	一応の効果があった	0.9(1)

T R M	知らない	67.6(73)
	知っているが使ってない	18.5(20)
	良く用いている	0.9(1)
	補助的に用いている	0.9(1)
	一応の効果があった	0.9(1)

P E R T	知っているが使ってない	39.8(43)
	補助的に用いている	19.4(21)
	試行したことがある	15.7(17)
	良く用いている	13.0(14)
	一応の効果があった	9.3(10)
	知らない	5.6(6)
	大変有効であった 効果はなかった	4.6(5) 0.9(1)

ハ ン デ ィ P E R T	知らない	46.3(50)
	知っているが使ってない	28.7(31)
	補助的に用いている	4.6(5)
	一応の効果があった	4.6(5)
	良く用いている	3.7(4)
試行したことがある	0.9(1)	

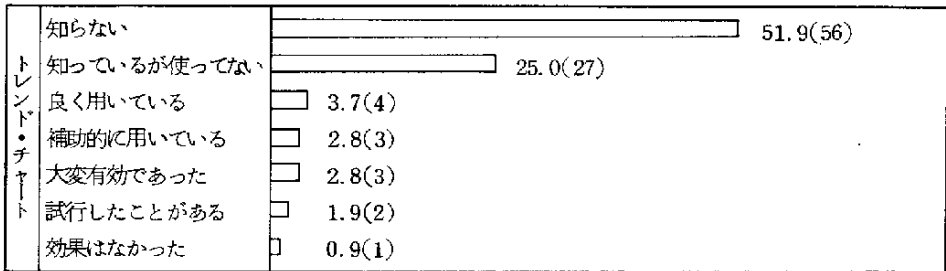
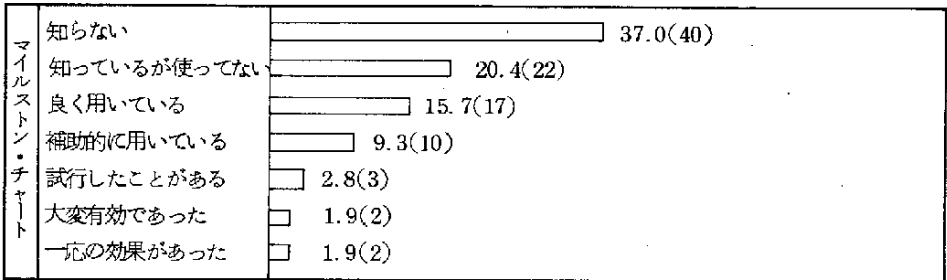
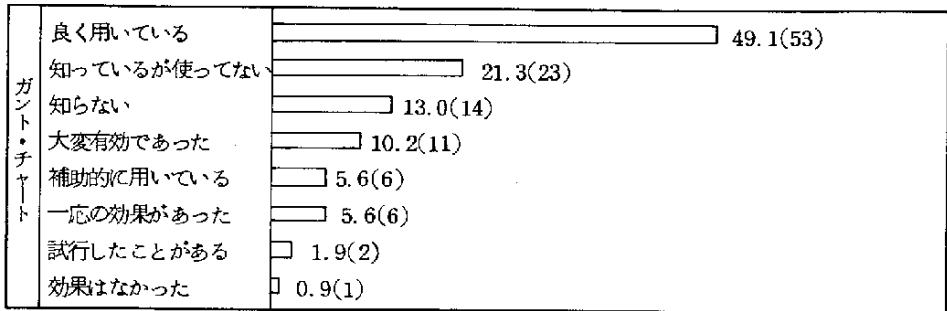


図 2・7 開発計画立案の技法・手法

件 49.1% とほぼ半数が利用している。

これは、ガンチャートが

- ① 計画が比較的簡単に行える
- ② 見やすく分りやすい

PERTが

- ① 作業の順序・相互関係が明らかになる
- ② 工程途中での将来の工数予測が簡単に行える

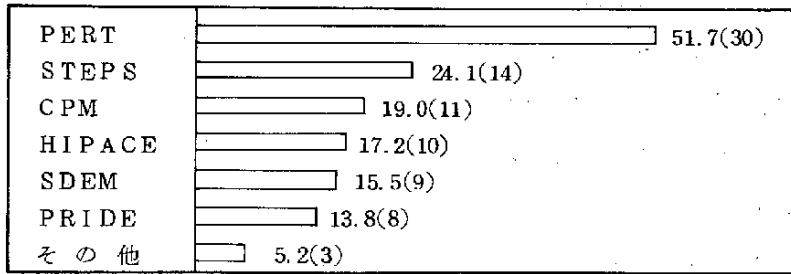
という特徴をそれぞれ有しており、利用者のニーズに適合しているからだと思います。

利用効率の面からとらえてみると、ガンチャートは17件15.7%から一応の評価を得ているが、利用者だけの評価は27.9%と上昇するもののそれ程高い数字を示していない。PERTについても同様の傾向があり、15件13.9%の評価に対し、利用者からは28.8%となっている。評価が低い理由として、ガンチャートでは作業相互の関連が不明確で工程管理上の要点がはっきりしないという欠点を、PERTでは作成・変更に必要な工数が大きく、また要員単位の管理には向かないという欠点を有しているからだと考えられる。

全般的に利用効果への評価が低い結果に終っており、このことが結果として、他の技法・手法の利用を試みることに消極的にさせているようである。しかし、ガンチャートの欠点を、全部ではないが一部を補完して作られたマイルストーンチャートの知名度および利用度は、今後は増加していてももらいたいものでありそこから更に良質の技法・手法が生み出されてくることを期待するものである。

#### (4) 開発計画立案で使用したツール

開発計画立案でのツールの使用実績をCPM・STEPS・RAMP・HIPACE・GERT・SDEM・PMP・PERT・PCMP・PRIDEについてきいてみた。(図2・8)



N=58

図 2・8 開発計画立案で使用したツール

「使用した」と回答した 58 件のうち 30 件 51.7% で PERT が使われており、ついで STEPS 14 件 24.1%、CPM 11 件 19.0%、HIPACE 10 件 17.2%、SDEM 9 件 15.5%、PRIDE 8 件 13.8% となっており、PERT の利用度が高いのが目につく。これは、PERT がメーカー色なしに入手・利用できるツールであること、および利用範囲が広いことを反映していると思われる。

(5) 見積基準値の有無

見積基準値については 66 社 57.9% が保有しており、またソフトウェアの種類により異った見積基準値を有していると答えたのは 66 社中 44 社 66.7% となっている。(表 2・9、表 2・10)

持っている	66 (57.9)
持っていない	48 (42.1)

表 2・9 見積基準値の有無

変えている	44 (66.7)
変えていない	22 (33.3)

表 2・10 見積基準値の変更の有無

これは、過去の数値の積み重ねの中から生み出されたものであり、各企業にとって、信憑性がきわめて高いと考えられる。ただ各企業での算出基礎（たとえば人件費）が異なるので、数値そのものを利用することは不可能である。

(6) 見積時の重点項目

工数等の見積りに際して重要視される項目を「バッチ・オンラインの区別」・「開発要員のレベル」・「処理内容」・「ユーザの緊急度」・「システムの規模」・「費用対効果」・「開発言語」・「システムの新規性」・「システムの難易度」・「システムの複雑度」について、3位までを選択してもらった。その結果「システムの規模」を1位にあげた回答が49件45.4%と約半数あり、3位までの合計でも77件71.3%とトップを占めた。この場合の「規模」はプログラムのステップ数が算出根拠となっている。つづいて「システムの難易度」が1位19件17.6%、「バッチ・オンラインの区別」が1位15件13.9%となった。「システムの複雑度」が1位は3件2.8%と低いにもかかわらず、2位にあげた回答が22件20.4%とかなりの重要性を示しており、システムの難易度・複雑度が工数に及ぼす影響の大きさを表している。(表2・11)

項 目	1 位		2 位		3 位		ウェイト付け
システムの規模	49	45.4	18	16.7	10	9.3	193
システムの難易度	19	17.6	26	24.1	19	17.6	128
バッチ・オンラインの区別	15	13.9	14	13.0	17	15.7	90
処 理 内 容	8	7.4	8	7.4	12	11.1	52
ユーザの緊急度	7	6.5	6	5.6	11	10.2	44
費 用 対 効 果	5	4.6	5	4.6	2	1.9	27
システムの複雑度	3	2.8	22	20.4	10	9.3	63
システムの新規性	2	1.9	2	1.9	8	7.4	18
開発要員のレベル	0		6	5.6	12	11.1	24
開 発 言 語	0		0		6	5.6	6
そ の 他	0		1	0.9	1	0.9	3

表2・11 見積時の重点項目

## 2.5 その他

本節では、記述式の回答をまとめ、コンピュータユーザがかかえている共通の問題点を探り出すことにしたい。

### 2.5.1 外注する場合の判断基準

75件の回答のうち、最も多かったのは「工程・工数が不足」が21件、ついで「要員不足」が18件、「開発能力不足」が11件とつづいている。工程・工数および要員の不足については、個々の開発計画に基づいて社内作業のオーバーフローする部分を外注するケースが多く基準値となるような数字は持っていないように思われる。外注の対象としてはプログラミングからデバッグ・テストの段階が多い。

「他システムとの関連がなく、独立性の強い自己完結的なシステム」の場合には一括外注すると回答したものが5件あった。これなどは、外注利用の基準が最も規格化された回答であるといえよう。「技術蓄積につながらない腕力だけの仕事（コーディング等）」という回答も、これと同じ考えで、システム設計までは社内で行い、それ以降の段階（プログラミング、デバッグ・テスト）は外注するという明確な基準を設定して、社内体制を確立している企業もあった。

他方、メーカーへの依存も根強く残っており、「ハードと一体となったソフト開発（オンライン・プログラム、フロントエンド・プロセッサ、コミュニケーション・プログラム等）」「OSの一部修正・追加」といったメーカーの標準サポートを超える部分について、メーカーへ外注するという回答もあった。

外注の範囲については、一括外注を行うケースはほとんどなく、「プログラム作成」・「デバック」を外注するケースが多い。特に、ソフトウェア開発計画案の作成までは、ほとんどの企業が内部処理を行っている。



外注先の決定基準については具体的な数字はなく、外注先の「信用」・「実績」・「技術力」・「見積金額」等で判断しているようである。特に、ソフトウェアの信頼性をチェックする技術手法が確立されていないため、外注先に対する評価は成果物の出来・不出来（バグの発生頻度）が基準となり、作業中の工程・工数管理等のチェックをなかなか行えないのが現状である。

## 2.5.2 システム開発での失敗の原因

41件の回答があり、失敗の原因や状況について、貴重な資料を得ることができた。

内訳をみると、「基本設計」に関する問題に起因しているものが8件と最も多かった。内容的には、この段階での詰めが充分でなかったために設計ミスを引き起こしたり、工数の大幅な増加につながってしまっているとのことであった。

「要求分析」に起因するケースとしては、要求分析の詰めが不十分であったため仕様変更を生じたり、コンピュータ部門が主導的に開発したためにシステムが理想的な絵作りとなり実際に利用するユーザ側として不満が残ってしまったり、ユーザ部門の意見が重視されていなかったため工程をさかのぼって変更の必要が生じたり、見積工数の増加については本稼働の遅延を発生させたという回答が6件と2位を占めた。

この逆のケースとして、「要求仕様があいまい」であったとする回答も4件あった。

両者の関係を表わすものとして、「協力関係・コーディネーションのまづさ」をあげたものが5件あった。協力関係やコミュニケーションが良好であっても、要求分析・システム分析については標準となる技法・手法がなかったり、活用されていなかったりするケースが多く、そのため開発が行きづまることも多々あるだけに、各部門が協力関係を密接に保つことは必要不可欠なことだと考えられる。

「技術力」に関する回答も多く合計10件にものぼった。大別すれば、技術力の「過言」と「不足」になる。「過信」したものとしては、メーカーの技術力からシステム・エンジニアやプログラマー（特に外注）およびコンピュータ処理能力にまでおよんでおり、結果として、「スケジュールの遅延」や「プログラムの再作成」をもたらしている。「技術力不足」は新技術（DB/DC，開発言語）に関するもので、理解不足のままシステム開発に入ったため、効率が悪くなったり見積りに相違をきたしたりしている。ここから、「パイロット・システムを開発して評価すべきであった」という反省が生まれている。

一方、単なる「技術力不足」だけでなく「要求分析」とも関連するものであるが、システム化の基本姿勢に誤りがあったケースもある。たとえば、「手作業の姿でシステム化し、コンピュータの有利さを十分に生かさなかった」・「人間系に与える業務量に対する認識の不足・不徹底」といった回答がこれに該当する。

「将来動向の見極めの甘さ・不徹底」という回答も5件あった。主に個別業務に対する見通しの甘さに起因し、そのため「1年あまりで全面的に見直しを迫られた」り、「対象範囲の拡大・処理方式の追加が発生した」りした。一方では、「外的要因の変化」ともいうべきものに起因した場合もある。たとえば、「全社的立場でのシステムへの変更」を強いられたり、「関連部署相互間（生産部門と販売部門）のシステム的な制約が大きく変わった（たとえば、バッチ・システムで開発していたが、オンライン処理が必要になった）ためのシステム変更」が発生している。

### 2.5.3 システム化要求・分析時における問題点

55件の回答のうち、最も多かったのは「要求の不明確・不完全さ」で12件あった。これは、ユーザ部門で要求を体系的な形にまとめられないことを示唆している。その原因としては、要求が断片的であったり、部門内で要求に差異があったり、コンピュータ化への認識不足といったことが

考えられる。その結果、要求内容が未消化となったり、時間がかかりすぎたりし、ひいては、システム開発途中での仕様変更や開発後のシステム・メンテナンスの発生という事態を生じている。

「効果の測定基準が不明である」という回答が6件あった。これは、個人の経験と勘に頼ったままで、標準化がなされていないことを表わしており、各種技法・手法の利用度の低さと一致している。これらの回答には、管理技法等を導入して作業の標準化を図りたいというコメントが見受けられたが、実際にはどの様な技法が最適であるのか判断しかねているようである。効果測定が可能になれば、予算計画との関係で作業範囲を決定できるので、この分野の定式化が望まれるところである。

「コストの測定が困難である」という回答も、上記と同じ状況を表わしているものと思われる。

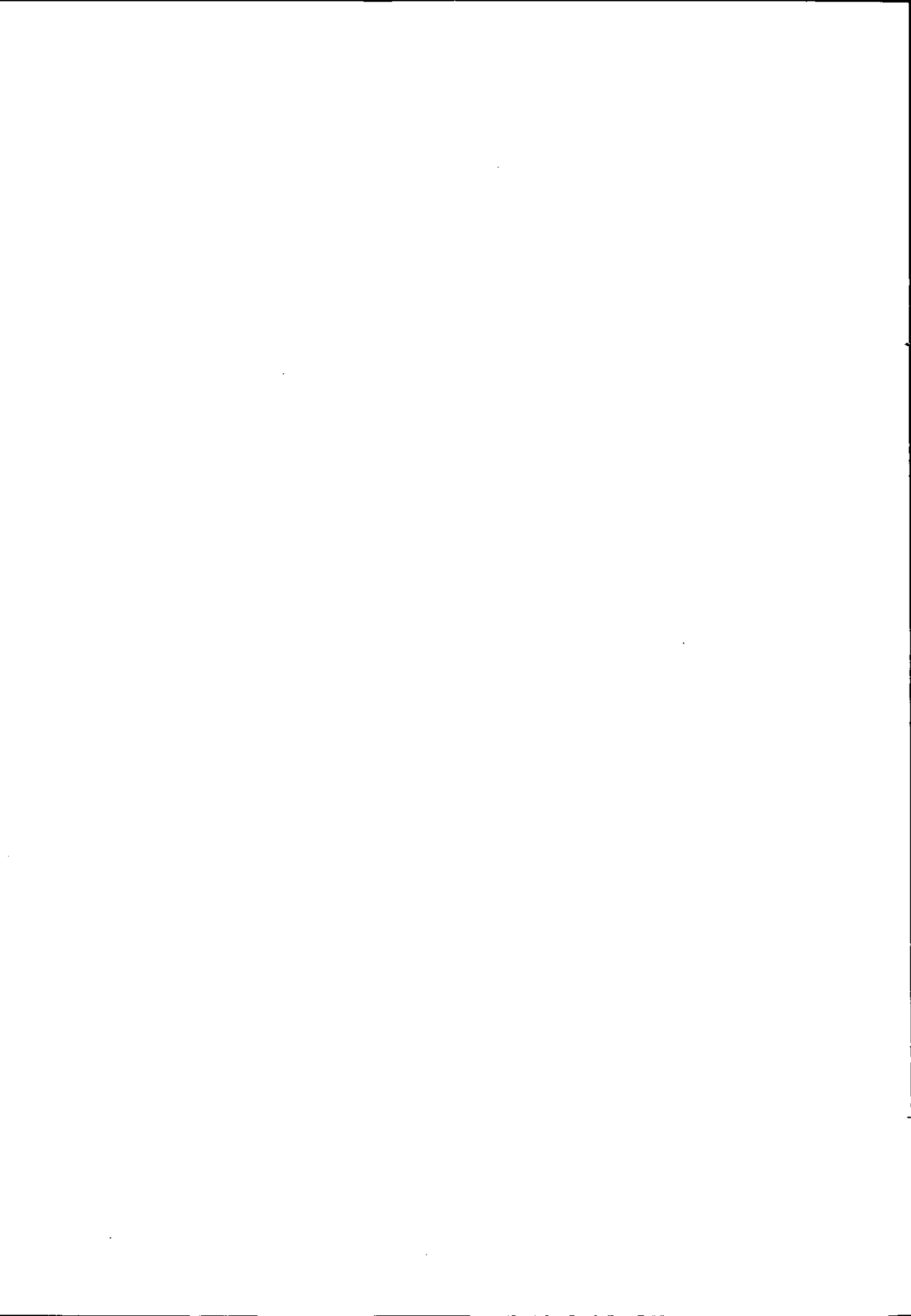
「機能分担に問題があった」という回答については、システム化要求段階ではユーザ部門も積極的に参加し、むしろ主導的であるのに対し、要求分析・基本設計と段階が進むにつれコンピュータ部門主導型となってしまう傾向を示しているものと考えられる。その結果、ユーザ部門の意見がシステムに反映されていなかったりコンピュータ部門に負荷がかかりすぎたりしてしまい、出来上がったシステムが予想された効果をあげないことにつながっている。

「ユーザ部門間の調整」・「既存システムとの整合性」・「全社的な視野」などに問題があったという回答は、類似した意味を持っていると考えられる。つまり、情報システムが企業戦略の遂行のうえで不可欠な要素となっている現在において、経営方針にそった全社的整合性のとれたシステム構築が求められているからである。オンライン・ネットワーク・システム、OA、LANなどによる分散処理化が進展すればするほど、1つのコンセプトに基づいたシステム開発が強く求められていくであろう。

コンピュータ部門は初期の「何でも屋」から脱皮し、システム化での自己の分担に目覚めつつあるのが、ヒアリング調査からもうかがえた。その1つのあらわれが、システム監査・セキュリティへの強い関心である。また、ユーザ部門の要求を実現するだけでは全社的に不均衡を生じること、そのためには意識の低いユーザ部門に対しては、いわゆる「足でかせぐ」形でコンピュータ化を図ろうと努力している。

ユーザ部門は、運用開始後の「システム化の効果」測定にはあまり熱心ではない。そのため、コンピュータ部門は反省材料のフィードバックが見込めない状態となっている。このような現状を打破するためにも、システム化の各段階で基準が設けられる必要があるのではないだろうか。

### 3. ソフトウェア開発の要求定義



### 3. ソフトウェア開発の要求定義

#### 3.1 はじめに

本章ではコンピュータ部門がシステム開発を推進する際に、“何を作るべきか”という事をユーザのニーズやトップマネジメントの意向及び企業経営方針に合わせて、要求分析する方法論と技法を検討する。

##### 3.1.1 要求定義の背景と必要性

コンピュータ利用部門の拡大、コンピュータ利用技術の高度化、複雑化、システム間関連の複雑化、環境条件の変動が大きくなるにつれ、多数の関与者が複雑にからみ、システム開発の目的が多様化している。

このようなことから、潜在する問題が十分に摘出されなかったり、問題間の相互関連がつかめず、問題範囲を決められない等のために、本当の要求は何であるのかを十分に理解しないうちにシステム開発してしまうので、その結果、開発要員、期間、コストがますます増大してしまう結果となる。

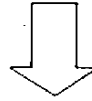
すなわちシステム建設の全体を通じて、生産性の向上、保守経費の削減、品質の向上が急務となっているのが現状である。

これらの問題はユーザニーズを的確に把握する方法が確立されていないことが大きな原因となっている。

そこでこれらの問題を解決するために多数かつ多方面にわたっているニーズを整理し、関連づけ、利用部門をふくめたコンセンサスを得ながら要求分析を行っていくステップと手法の確立が必要となってきた。

背景

- ・ 利用部門の拡大
- ・ 利用技術の高度化・複雑化
- ・ システム間関連の複雑化
- ・ 環境条件の変動大
- ・ システム開発規模の大型化



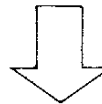
問題点

- ・ 潜在する問題が十分に摘出されない。
- ・ 問題間の相互関連がつかめない。
- ・ 問題の範囲を決められない。
- ・ システム開発のコストの増大
- ・ システム建設の手なおし、手もどりの発生  
の増大とコストの増加



対策

- ・ 的確なユーザーニーズの把握
- ・ 要求定義手法の確立
- ・ 多数関係者間のコンセンサスの確立



要求定義手法の標準化

図 3.1 要求定義の必要性と背景



### 3.1.2 標準化の必要性和工程区分

#### (1) 要求定義の問題

要求定義段階における問題は第一に正しくない要求が入ってしまうことである。真のニーズがどこにあるのかを正確に表現していないために、結果的に後になって仕様変更を必要としたり、効果のない要求仕様をまとめることとなってしまうことがある。

第二の問題は一貫していない要求、または互換性のない要求が入ってしまうことである。これは技術的な裏づけがないままに、要求そのものが現実とかけ離れてしまう場合で、現行システムと互換性のないシステム要求を作る場合が往々にして存在する。

第三の問題点は関連部門との打合せやコミュニケーションが不足しているために、要求のどの点が真に必要な問題点であり、どこが不要であるかの焦点がぼけて、結局真の要求がまとまっていない事があるものである。

#### (2) 標準化の必要性

以上の理由で要求定義の手法は個人独自の考え方でまとめるのではなく、標準化した共通の手法を用いることによって(a)正確な仕様化の実現・作業の均質化、(b)第三者が検査し得る仕様化、(c)保守性の向上：正確なコミュニケーションの達成、などが実現できる。

#### (3) 工程区分

ソフトウェアの開発段階は、規模、開発体制などや処理方式等によって細部では異なるが、基本的には同じような手順を踏んでいる。

近年、各メーカーでソフトウェアの開発工程、作業を体系化して発表しているが、共通している事はプログラミングを行う前の作業（企画、調査、設計等）に大きな力点を置いている事である。これは設計に開発時間の多くを費やしている現状からみて、設計段階の改善が結果的には信頼性、保守性の向上を図ることができ、ソフトウェア開発の期間短縮、

トータル・コストの低減が可能であるからに他ならない。

区分の方式は各種あるが、本調査研究では最も標準的な要求定義段階の工程ステップを、システム化要求、要求分析、およびシステム分析の三つに区分した。

① システム化要求

この段階はユーザ部門がシステム開発の要求を明確化する段階であり、トップの方針とエンド・ユーザの生の声を十分に摘出する。この段階ではユーザ部門が主体となってまとめる。

② 要求分析

この段階は摘出された開発要求が妥当性を持っているか、コンピュータ部門の専門家を中心に事務の流れ、帳票類の統廃合を調査分析し、妥当性をチェックする。

③ システム分析

チェックの結果ユーザ部門の要求が妥当であるとわかった場合、次に類似システムや最新の技術動向に照らし合わせてよりよい代替案を見出し、環境条件を勘案した上で比較検討を行い、システム案を作成する段階である。

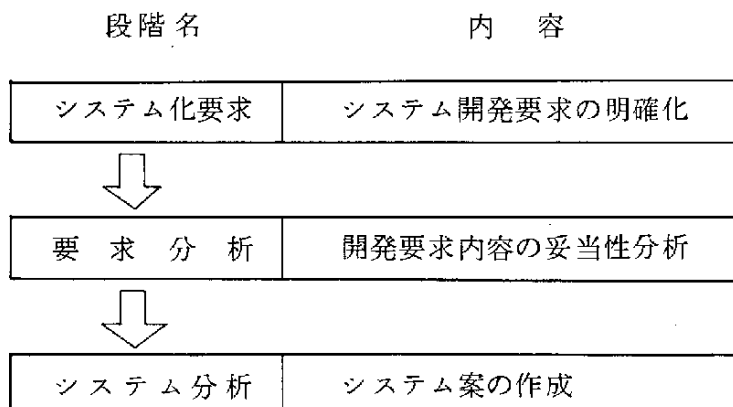


図 3.2 要求定義の工程

以上のように要求定義を3つの段階において検討を行うこととする。

#### (4) 検討項目の内容

検討する項目の内容は前記の3つの段階の工程区分にそって目的・管理内容、作業内容とワークドキュメントおよび体制の3点を中心に検討を行った。それぞれの検討のポイントは以下のとおりである。

##### ① 目的・管理内容について

ここでは各工程がどのような目的で、どのような作業内容を行うかを検討している。

##### ② 作業内容とワークドキュメント

ここでは作業の具体的なポイントをまず示している。次に作業成果を後続する開発作業や保守作業等のために記録を残しておくためのワークドキュメントとし、この体系を明確化するとともに、更にその内容項目を明確化し、その記述のポイントを検討している。

##### ③ 体制

体制においてはこれらの作業および記述を行うためにユーザ部門、コンピュータ部門、運用部門がそれぞれどのように係り合うべきかを検討している。

また次に技法・手法およびツールの項では、現在どのような技法および手法ツールがあるかを調査し、比較検討している。

最後に留意点とチェックリストの項では、これまでの集大成として、全体として留意すべき点をチェックリストとして検討し、総まとめを行っている。

### 3.2 システム化要求

システム化要求段階については問題発掘はまずトップの方針とエンドユーザの生の声が反映されていなければならない。この段階では問題、要求等が散在していたり、潜在する問題、要求等が十分に摘出されない等の現象が生

じやすい。そこで目的や背景を明確に定義し、これにそって全体のコンセンサスを得る必要がある。

### 3.2.1 目的・管理内容

ユーザ部門がシステム化して欲しい業務の内容・範囲を機能面から検討し、ユーザ部門におけるシステム化要求案を独自に作成する。すなわちシステム化要求を明確にし、全体のコンセンサスを得ることが目的となる。

### 3.2.2 作業内容とワークドキュメント(システム化要求書)

この段階の作業は最終的にシステム化要求書にまとめられる。このために作業をするわけであるが、この作業内容は次のようなものがある。

- (1) システム化の背景や必要性の整理を行なう。
- (2) 目的・効果を明確にする。
- (3) 対象業務範囲を明確にする。
- (4) 関連資料帳票類の洗い出しを行う。
- (5) 運用開始時期を明確化する。
- (6) 制限条件を明確にする。

以上の内容を含むものである。

これを具体的に記録するものがワークドキュメントである。ワークドキュメントは作業結果を記録しておくもので、後続する開発作業や保守作業等のために記録を残しておくドキュメントである。

ワークドキュメントの記述内容は次の通りである。

#### (1) ワークドキュメントの記述内容

##### ① 目的

システム開発要求の明確化

##### ② ワークドキュメント名

システム化要求書

③ 内 容

① システム化の内容

- \* 目 的
- \* 背 景
- \* 必 要 性
- \* 効 果
- \* 運 用 時 期

② 適用範囲

- \* 対象業務範囲
- \* 人手作業とのインターフェース
- \* 機能拡張方針

③ 関連情報一覧

- \* 入力帳票一覧
- \* 出力帳票一覧

④ 制約条件

シ ス テ ム 化 要 求 書	システム化の内容
	適用範囲
	関連情報一覧
	制約条件

図 3.3 システム化要求書の内容

(2) 記述上および作業上のポイント

上記の記述内容および作業上の内容を以下に説明する。

① システム化の内容

## ① 目 的

システム化の目的を明確に設定することがこの記述のポイントである。具体的には次のような内容を記述する必要がある。

- ・ 事務の改善，合理化……人件費，事務経費の節減
- ・ 管理の質の向上……正確性や迅速性の向上
- ・ 手作業事務の軽減……省力化

作業上のポイントとしては，エンドユーザ部門の目的意識を統一し，明確化する事がポイントとなる。

## ② 背 景

ここでは業務に関する現状と問題点及びシステム化に至る経緯を明確にする。記述上のポイントは次の点である。

- ・ 制度の改正
- ・ 事務サービス（帳票類の改善，迅速性，正確性）の改善要求
- ・ 業務量，担当職員の増加

ここでの作業上のポイントはシステムの規模，適用業務の範囲等にどらわれることなく，業務内容，事務の流れを把握し，エンドユーザが直面している背景と改善方法を明確にする。

## ③ 必 要 性

ここでは直面している背景に基づいて，システム化する必要性を明確に設定する。記述する内容のポイントとしては次の点である。

- ・ 問題点解決のための緊急性，優先度
- ・ 改善する内容の必要性，緊急度

作業上のポイントとしては，システム化の必要性を各部門で幅広く意識統一するとともに，システムの実施に際しての協力体制を確立することがポイントとなる。

## ④ 効 果

ここでは管理面に関する効果を明確にすることがポイントである。

すなわち以下の事項を主として記述する。

- ・ 業務処理……迅速性，正確性，簡易性，弾力性
- ・ データ利用，管理……容易性，適時性，安全性
- ・ 人事・組織……モラル，関連組織内の連携

次にコスト局面に関する効果を明確にする。このためここでは人手作業をコンピュータ処理システムに置換えた場合のコストの総比較を行う。

また質的な面から効果を明確化する。すなわち業務や研究面での質的向上やサービス面での向上，波及効果などを記述する。

効果の面では，システムの効果のみならず，システム化のもたらすデメリットも充分検討し，その対応策を盛り込むことが重要である。

#### ④ 運用時期

ここでは運用開始の希望時期を明確に設定するとともに，その必要性，理由を明確にする。

- ・ 業務の優先度，緊急度
- ・ 開始時期の制約（法・制度）

作業上のポイントとしてはシステムの規模，開発の難易度，要員の確保等とは関係なく，本当に必要な希望時期を設定することである。

### (3) 適用範囲

システム化の内容について上記したが，以下本項では適用範囲について記述する。

#### ① 対象業務範囲

本項ではシステムで対象とする業務内容，処理範囲を明らかにし，他業務との関連および新規業務に対する予測等を明確にする。

- ・ 組織および法・制度の新設，改訂による影響度

- ・ 対象業務の重要度，優先度，緊急度

ここでの作業ポイントは現状の事務処理の流れを正確に把握し，背景，必要性を検討した事項とともに，対象とする業務範囲を設定する。また関連部門および他業務に対する影響，波及度合を調査し，関連部門の合意を得ることが重要である。

## ② 人手作業とのインターフェース

ここでは設定された対象業務と人手作業との境界および情報の授受方法，手段等を明確にする。

- ・ 事務手続きと伝票，帳票類の対応
- ・ 人手作業が変更される内容

この段階での作業のポイントは事務処理の流れを帳票類の流れと対応させ，人手で行わなければならない範囲，改善できる範囲，内容を事前に十分に検討して，ユーザ部門としての統一見解を打ち出すことである。

## ③ 機能拡張方針

ここでは将来のシステム拡張，拡大および新規業務等による機能拡張の方針を明確にする。

- ・ 基礎データ（処理時間，処理件数）の予測
- ・ 組織および法・制度の新設，改訂

作業上のポイントとしては将来にシステム拡張が予測されている場合は，その拡張度合を充分検討して大巾に変更にも対処できるように考慮しておく事が大切である。

## (4) 関連情報一覧

### ① 入力帳票類一覧

対象業務に関連する入力帳票類を全て抽出し，その属性（情報名，作成目的，発生源，データ量，関連資料，機密保護の必要性等）を明確にする。



## ② 出力帳票類一覧

対象業務に関連する出力帳票類を全て抽出し、その属性（情報名、作成目的、利用部門、データ量、関連資料、機密保護の必要性等）を明確にする。

### 入力帳票と出力帳票の属性の関連

作業上のポイントとしては、帳票類の形式および制約を明らかにすること、業務特性に応じた各種帳票、図表の見直しを行うことが必要である。またメモ、電話等でのやりとりが定常化していれば、対応策を考慮する必要がある。またメモ、帳票類は実際に使用されている現物を収集する。

## (5) 制約条件

ここでは時間的条件（資料作成期限、応答・待ち時間の許容度等）を明確にする。また経済的条件（開発・運用経費等）を明確にする。更に技術的条件（信頼度、性能、帳票形式等）を明確にし、最後に組織的条件（地理的制約、分散管理状況、管理体系等）を明確にする。

ここでの作業上のポイントは必要条件および制約条件をエンドユーザ各部門から提出し、必要と考え得るものすべてを記述できるようにすることが必要である。

## 3.2.3 体制

システム化要求はユーザ部門からシステム化に関する要求を出すことが目的であるから、ユーザ部門だけでまとめる必要がある。従って体制としてはユーザ部門の代表者が関連ある部門と一緒に討論する体制が必要である。

## 3.3 要求分析

要求分析はユーザ要求がシステム化要求書としてまとめられた段階で、こ

の要求が妥当性を持っているかどうか、多角的な検討を行うところである。

またその結果としてシステム化の可否、優先順位等を検討する。

### 3.3.1 目的・管理内容

要求分析の段階ではユーザ部門で検討した要求内容について、事務処理の流れ、帳票類の統廃合、標準化等、システム化するための現状調査・分析を行い、要求事項の妥当性を総合的に分析する。その主な内容は次の通りである。

- ① 要求内容個々の妥当性を分析する。
- ② 事務処理の流れの調査と対象業務範囲、処理内容を検討する。
- ③ 帳票類の統廃合、標準化を行う。
- ④ 経済性、効果、技術面からシステム化を行うべきか否かを検討する。

### 3.3.2 作業内容とワークドキュメント

3.3.1の作業を行ない、その内容をワークドキュメントにまとめるわけであるが、このワークドキュメントは要求仕様書となる。この要求仕様書に記述する内容とその作業上のポイントを以下に説明する。

#### (1) システム化要求の内容分析

ここではシステム化要求の妥当性、一貫性および正確性の分析を行う。仕様書の主な項目は次の通りである。

- ① 目的・効果の分析
- ② 背景・必要性の分析
- ③ 運用時期の分析

#### (2) 適用範囲の分析

##### ① 対象業務範囲の分析

システム化の対象として設定した業務範囲が目的、必要性を具体化する内容となっているかを確認する。

- ① 業務処理の流れに対応する業務内容、処理範囲の妥当性
- ② 組織、法、制度の新設、改訂版および影響度の確認、対応

作業上のポイントは、まず対象作業内容、処理の流れは、従来の手作業等にとらわれず、効率化、合理化、および省力化等の観点から見直すことが必要である。一方人手作業による手順は文書化されていないことが多いので抜けがないか否か充分検討確認することが必要である。また目的、必要性等、ユーザ提起の問題点、改善事項を念頭に入れ、それを満足した業務範囲を設定しているかを検討確認する。

## ② 人手作業とのインターフェース分析

対象業務の作業分担、情報の授受方法等、マン・マシン・インターフェースの妥当性について分析・確認する。

- ① 事務処理の手続きおよび期限の妥当性
- ② 人と機械処理の対応手順

作業上のポイントは情報の流れ、授受方法の一元化、総合化の可能性および事務のスムーズな流れが実施できる体制が確立できるか否か、十分な分析と確認を行う。

## ③ 機能拡張分析

ここでは将来のシステムにおける機能拡張の必要性及びこれに対応する方法の内容を分析・確認する。

- ① 将来予測のデータについて妥当性を検討する。
- ② 機能拡張、新規業務の追加等の必要度確認
- ③ 関連部門への影響度の把握

作業上のポイントはユーザー提起のシステムおよび機能拡張、拡大方針を前提とし、背後にある必要性、データ類の伸び等を確認して、システム化に反映できるような方策を考慮する必要がある。

## (3) 関連情報の分析

① 入出力帳票類分析

ここでは業務内容と入力および出力帳票類の関連について、分析・確認する。

① 属性の内容と必要性

② 帳票類の統廃合，標準化，汎用化，簡素化を検討する。

③ 入力帳票と出力帳票の属性の関連性

作業のポイントは事務処理の内容および手順を検討し，必要最少限の帳票類の選定を行い，統廃合，標準化等の見地から関連情報を作成し，ユーザ部門へ提案できる内容にする。

(4) 制約条件の分析

ここでは制約条件で規定した内容および背景となるデータ，環境条件を分析，確認する。

① 必要不可欠条件と希望的条件の把握

② 制約条件を満足するための付帯事項

作業のポイントは条件の内容を充分把握分析し，システム化を行う場合のメリット，デメリットを明確にするとともに，コスト面から検討し，コストと効果の関連から最も効果のある方式を把握する。

(5) システムの妥当性・必要性

ここではエンド・ユーザ部門の要求内容を詳細に分析した結果に基づいて，システム開発の妥当性，必要性についての結論を明らかにする。記述する主な項目のポイントは次の通りである。

① システム化要求案としての可否

② 要求内容の妥当性

③ 改善内容の設定

などである。作業上のポイントは，企画，設計部門としての立場から技術的かつ客観的に分析を行い，その結果として，システム化妥当性の判断および次の段階への進展について，結論をつける。

### 3.3.3 体制

この段階はコンピュータ部門の企画設計担当者が中心となって作業し、調査・分析仕様書をまとめる。この場合ユーザー部門のシステム分析者もサブとして参加する。

運用時期分析を行う際には運用部門の担当者も参加させる。

### 3.4 システム分析

システム分析段階は要求分析で行った調査分析結果をもとにして、これに対応した類似システム、最新の技術動向をもとにして、ユーザ部門の要求案に対する代替案を作成し、環境条件を十分勘案して比較検討を行い、システム案を作成する。

#### 3.4.1 目的・管理内容

- (1) 類似システム、技術動向を調査する。
- (2) 種々の観点から代替案を作成する。
- (3) 環境条件（ハード、ソフト、制約条件）を明確にする。
- (4) 代替案を比較検討する。
- (5) システム案を作成する。

上記のステップで代替案を作成するのが本段階の目的である。

#### 3.4.2 作業内容とワークドキュメント

この段階の作業は類似システムを調査し、代替案を比較検討し、代替システム案の提案書を作成する事であるが、ワークドキュメントの項目は次の通りである。

- (1) 類似システム及び技術動向調査
- (2) 環境条件
  - ① ハードウェア

② ソフトウェア

③ 制約条件

(3) 代替案の作成及び比較検討

① 機能・性能

② 経済性

③ 効果

④ 技術分析

⑤ 類似システムとの対比

⑥ 体制・工期

⑦ 開発難易度

(4) システム案

① システムの目的と効果

② システムの範囲

③ システムの概要

④ 経費・工期

以上の項目が含まれるが、これを記述上のポイントおよび作業上のポイントの両面を説明する。

(1) 類似システム及び技術動向調査

ここでは類似システムの特質（対象範囲、機能、効果、規模、経費、期間、使用技法など）を横断的にとらえ、その長所、短所を調査・分析し、結果を当該システムに反映できるように明確にする。

またシステム化に関連する各種技術の調査結果に基づいて、その特徴、動向等を明確にする。

記述する主な項目は次の通りである。

① ハードウェア装置について、機能、性能、価格を明らかにする。

② システム開発、運用の方法、技法等を明らかにする。

作業上のポイントは先進的なユーザの事例、各種文献、海外資料等

を参考にし、広範囲に当該システムの設計に有効となる技術を吸収し、システムに反映する。

## (2) 環境条件

### ① ハードウェア

システムの目的及び必要条件を実現するための機器構成を明らかにする。

#### ㊦ システムの拡張計画への対応

#### ㊧ 業務の規模、処理内容に適した機器の選択

作業上のポイントは類似システムを参考とするとともに、適用実績等から当該業務に適用する機器を選択し、機器構成の目安を設定する。

### ② ソフトウェア

ここではシステムの目的及び特質に応じたソフトウェアの選択方針を明確にする。記述の主な項目は次の通りである。

#### ㊦ プログラム言語……機械語、高級言語（性能）

#### ㊧ 既存、既成プログラムの利用……DE/DC

#### ㊨ 専用、汎用プログラムの選択

作業上のポイントは、類似システムを参考とするとともに、運用推移の実績等から当該業務に使用するソフトウェアの選択を設計方針、経費、保守等の面と対応させて検討する。

### ③ 制約条件

ここでは時間的、技術的、組織的条件としての最終的な制約条件を明確に設定する。

作業上のポイントはユーザ部門と企画・設計部門で十分な意見の統一をはかり、制約条件としての決定案を作成する。

## (3) 代替案の作成及び比較検討

### ① 機能・性能

ここではシステム化の要求案と代替案の機能・性能を対応させ、横断的な比較検討評価を行い、結果を明確にする。記述する主な項目は次の通りである。

- ① 対象とする業務範囲、機能
- ② 処理効果
- ③ 関連部門とのインターフェース

## ② 経 済 性

ここでは開発経費、運用経費を主体として経済性の比較検討評価を行い、結果を明確にする。記述の主な項目は次の事項である。

- ① 機能・効果と経費
- ② 環境条件と経費

作業上のポイントは、まず要求内容に基づいてシステムの理想形態、必要最小限の構成、現状に応じた形態等を考慮し、機能、範囲を設定する。

また上記システムの機能・範囲に応じた開発・運用に必要な経費及び工期の見積りを行う。この場合総枠での算出ではなく、個々の機能、範囲等に分割して明細を作成するのが望ましい。

## ③ 効 果

システム化の効果を管理、経費、業務、研究面から比較検討し、評価を行い、その結果を明確にする。

## ④ 技術分析

適用しようとする各種技術及びその導入法等を比較検討する。記述する主な事項は次の通りである。

- ① ハードウェアの技術動向
- ② ソフトウェアの技術動向

## ⑤ 類似システムとの対比

この段階では類似システムの調査結果をどのように反映し、システ



ムに対応させているかを比較検討し、評価を行ない、その結果を明確にする。記述の主な項目は、次の通りである。

- ① 相違点，メリット，デメリット
- ② 機能，範囲，経済性，採用技法

作業上のポイントは、企画・設計部門で提起した複数の代替案を横断的にとらえ、機能・性能、経済性、効果、技術分析、類似システムとの対比、体制、工期、開発難易度の諸観点から比較検討を行う。

比較検討に際しては単にどの代替案がすぐれているかの判定ではなく各々の代替案の長所、欠点を明確にして、システム案に対応し、反映できるように考慮する。

また予算および要員、体制等を考慮し、代替案の内容の実現程度を充分認識・把握することが重要である。

#### ⑥ 体制・工期

開発完了までの作業工期と所要人員についての比較検討評価を行ない、その結果を明確にする。従ってここで記述するポイントは業務範囲、諸機能に対する開発・運用の工数、体制、工期である。

#### ⑦ 開発難易度

ここでは開発作業の難易度を十分予測し、その対策としてもり込む内容を比較検討し、評価を行い、その結果を明確にする。主な記述項目は次の通りである。

- ① 作業内容，作業量
- ② 技術的対応
- ③ スケジュールの作成

#### (4) システム案

##### ① システムの目的及び効果

代替案の比較検討結果に基づいて、総合的に最も有利なシステム案を作成し、その目的及び効果を明確に設定する。

## ② システムの範囲

目的及び効果を実現するためのシステム化の業務範囲を明確に設定する。

記述上のポイントは、

- ① 対象業務名、概要
- ② 対象業務のシステム化の範囲
- ③ 人手作業とのインターフェース

である。

ここでシステムの目的及び効果とシステムの範囲の作業上のポイントは以下の通りである。

- ① システム化要求段階、要求分析段階及びシステム分析段階の最終的な結論が本システム案となる。
  - ② ユーザ部門、企画・設計部門及び運用部門の最終的な合意を得られた内容としてとりまとめられる。
- ## ③ システム概要

システムの基本的な処理機能(サブシステム)を明確にする。

主な記述項目としては次があげられる。

- ① 入出力情報
- ② 基本機能の構成と処理概要
- ③ システムの規模・機能

があげられる。

## ④ 経費・工期

この段階ではシステム開発・運用に必要な経費・工期及び要員等を明確にする。

上記③システム概要および④経費・工期の作業上のポイントは、予算、要員等で大幅な変更がない限り、目的、効果、範囲、概要、経費、工期等の内容は、システム化の最終案として、システム設計に反映で

きるようにする必要がある。

### 3.4.3 体制

この段階では代替案を作成し、システムの概要設計にまとめるので、比較的コンピュータ部門の専門家が行がなければできない分野である。従ってコンピュータ部門が加わって設計をまとめる。運用に関するところは運用部門のメンバーも参画する。

要 求 仕 様 書	システム化の内容分析
	適用範囲の分析
	関連情報の分析
	制約条件の分析
	システム化の妥当性・必要性

図 3.4 要求仕様書の内容

## 3.5 要求定義の技法・ツール

要求定義の効率化を図り生産性、保守性、信頼性等を高めるためには、これまで設定した工程区分、作業内容、ドキュメント項目等に基づく標準化のほか、各種の技法を導入し活用する必要がある。

本節では、要求定義の技法の現状、概要利用の効果・留意点、有効度等について分析し、適用範囲、選択要因、適用手順等について述べることとする。

### 3.5.1 要求定義技法の現状

本項ではアンケート項目「2.2 要求分析技法」の調査結果に基づいて、

技法の知名度，利用度，有効度の現状について簡単に述べる。

(1) 知名度 (図 3.5)

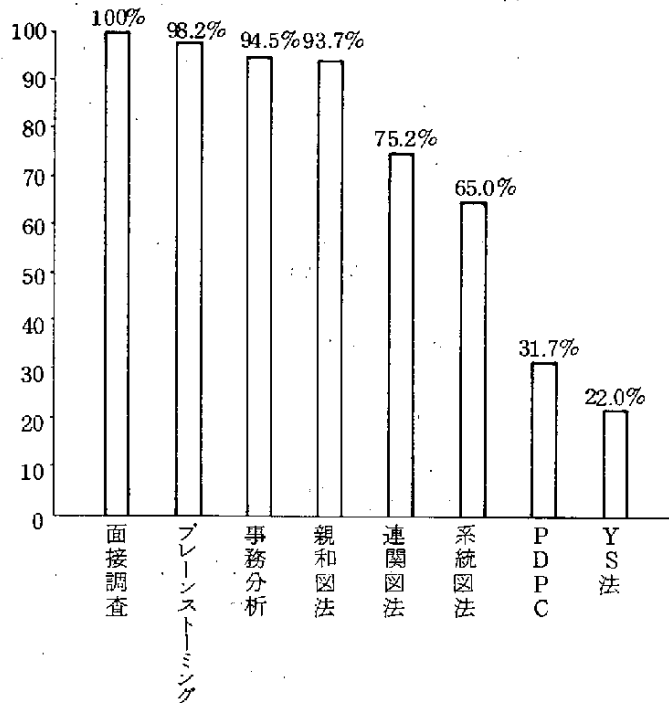


図 3.5 要求定義技法の知名度

各技法についての知名度を図 3.5 に示す。面接調査，ブレーンストーミング事務分析および親和図法については 9 割以上が知っていると回答している。

(2) 利用度 (図 3.6)

各技法についての利用度を図 3.6 に示す。面接調査，事務分析は約 9 割が利用していると回答している。

このうち，分析の中心として使用したものは事務分析，面接調査の順

で利用度の順位と入れ代っている。

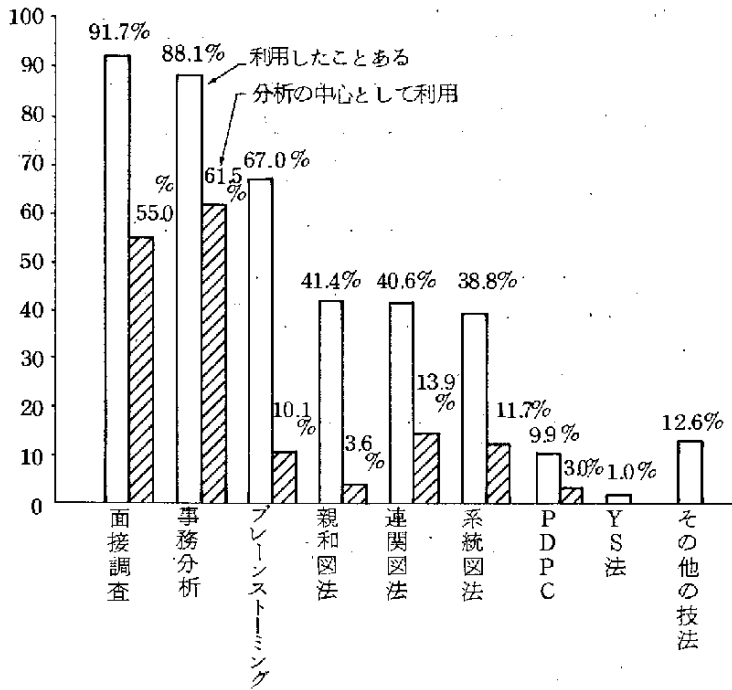


図 3.6 要求定義技法の利用度

(3) 有効度 (図 3.7)

各技法の効果に関する 3つの質問の比率を図 3.7 に示す。各技法の有効度は利用度の場合と同様に有効と認識されているが、プレゼンテーションは他の技法に比べて、やや低くなっている。

以上知名度、利用度及び有効度からみて要求定義の技法は面接調査、および事務分析技法を中心に利用されており、他の技法は必要に応じて補助的に利用されているのが現状である。

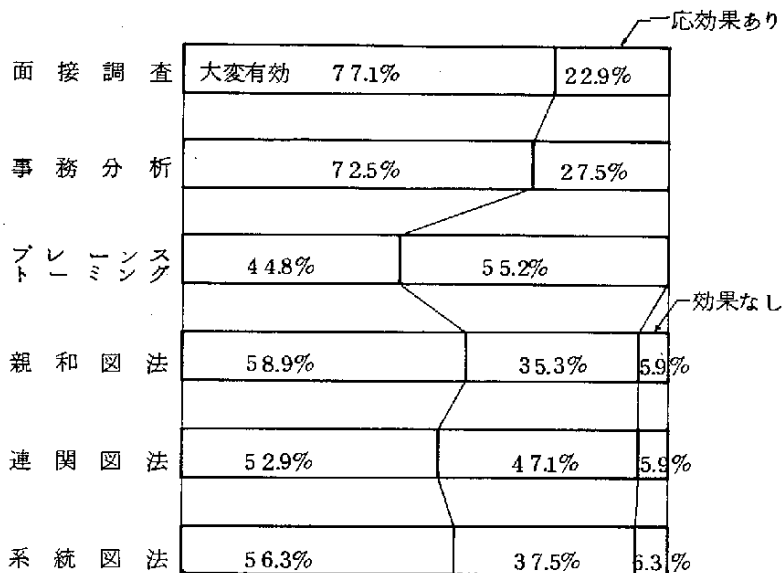


図 3.7 要求定義技法の有効度

### 3.5.2 要求定義技法の概要

本項では要求定義段階で最も重要と思われる分析技法について、その概要を述べる。分析技法の主なものは、次のようなものである。

- (1) ブレインストーミング
- (2) 親和図法
- (3) 面接調査
- (4) 事務分析
- (5) 関係図法
- (6) 系統図法

以下に各技法の概要を述べる。

- (1) ブレインストーミング

ブレインストーミングは、既成観念から解放された自由な発想、斬新なアイデアが欲しい場合に有効な技法である。従って、いっさいの束縛的要素を排除して会議を行う必要がある。

① ブレインストーミングの4原則

- ① 他人の発言を批判しない。
- ② 質よりも量（全員対等，全員発言）
- ③ 飛び入り歓迎（他人の発言を受け更に結合，改善，発展させる）
- ④ 奔放に発想し，自由に発言する。

なお、要求定義段階では主に、システムの目標の体系化（業務活動への目標のブレイクダウン）を行う場合や、問題点の解決策の提案などの合意の形成に用いられることが多い。

② アプローチ

- ① 問題志向のアプローチ
  - ① なぜそうなるのか（真の原因）
  - ② どのようにすればよいか（解決策）
- ② 目的志向のアプローチ
  - ① 何が必要か
  - ② どのようにすればよいか

(2) 親和図法

この技法は、問題解決の技法で、カードを用いて次の手順で行う。

① カードづくり

カードには1つの意見を簡潔明瞭に記入する。

② カードのグループ化

カードづくりが終わったら、内容が似ているものを集めグループ化する。できない孤立化したカードはそのままにしておく。

③ 表札カードづくり

グループ化されたカードの束ごとに、それにふさわしい表現を表札

カード（色を変えると良い）に記入し表札カードを上にして、グループをクリップか輪ごむでたばねておく。

④ グループ編成

②，③を必要な回数だけ実施する。

⑤ 図解または空間配置

大きな紙（模造紙など）を用意して，その上に紙きれをグループごとに配置し，枠で囲い，見出しをつける。

各グループの間は矢線などで結んで関係づけを行う。

1枚の紙に配置しきれない場合は，索引図解と細部図解に分ける。

図には，テーマ，作成日，作成場所，作成者などを記入する。

⑥ 文章化もしくは時間配置

図解により得られた情報の要旨を，シナリオ化する。口頭発表でもよい。

空間配置だけでは，相互関係のみしかつかめないで，シナリオ化により前後関係を明確にする。

親和図法は個々の調査事実をグループ化体系化して行く技法として開発されたが，問題解決技法としても有効に活用されている。

要求定義段階の利用分野としては，ブレインストーミングと同様である。



### (3) 面接調査法

面接調査法は経営者、管理者、利害関係者および作業担当者等と直接会って質問をし、必要な事項を聞き出す方法であり、個別面接と集団面接による方法がある。

#### ① 個別面接

手間はかかるが、本音を聞くことができる。

- ① 相手の立場、利害関係を理解して、相手の気持を察する。
- ② 教えてもらうという姿勢で聞く。
- ③ アドバイスはそれとなく少しずつヒントを与え相手の口からいわせ、相手の手柄にする。
- ④ 常に本音を聞くようにする。

#### ② 集団面接

一般的な情報の入手には効率的であるが聞きづらい。できるだけ同格のグループに分けた層別面接法を採用した方がよい。

集団面接の場合にはとくに、事前に主旨、日程、質問内容などを連絡し了解を得た上で実施し、事後速やかに整理した結果をフィードバックする。

面接調査での質問事項は資料調査等で漏れたことを中心に調査するが、一般的には次のようなものがある。

- ① 用語の意味と用法
  - ② 作業手順、注意点、コツ
  - ③ 事務量（平均とピーク）、発生の周期や頻度
  - ④ データの精度や信頼度
  - ⑤ 現行システムの問題点、不平、不満、トラブル、エラー等
  - ⑥ 改善案、アイデア、希望、期待や理想等
- #### ③ 業界他社の事例や技術動向など

その他に、流通、製造等の業務調査では現場で作業を観察あるいは

実習すること（現場観察法）により、資料・面接で得られなかったことが体験できる。仕事の流れに沿って順番に観察する作業追跡法と一定の場所で時間をずらして観察する定点観察法がある。

面接法も観察法も他の調査でのもれを補足する方法として必要である。

#### (4) 事務分析技法

事務分析技法は現状の業務の内容、工程（フロー）、入出力帳票等を把握する技法である。

事務分析技法は、本来は事務改善のための手法であり、コンピュータ化を前提としたものではないが、現行又は新システムの理解には極めて有効である。

##### ① 業務の内容調査

業務の内容、手順、使用する情報を一般文章形式で記述する。

##### ① 調査手順

##### ㉑ 資料調査

職務分掌規程、業務手順書、入出力帳票、業務知識、組織図などを集め、整理する。

##### ㉒ 面接調査

資料調査を補足し確認する。

##### ㉓ 調査の内容

##### ㉑ 業務概要（図 3.8(1)）

対象業務の概要を記述する。

##### ㉒ 業務組織（図 3.8(2)）

業務の遂行に関係のある部署を記述し、組織図を書く。

##### ㉓ 業務関連（図 3.8(3)）

組織と業務の関係を記述し、説明図を付ける。



#### ④ 業務の詳細

業務内の個々の処理（作業内容）を中心に記述し、情報との関連を図示する。（図 3.8(4)）

#### ② 事務工程（フロー分析）

事務フローは、物の流れに適用した工程分析フローを、情報の流れを中心とする事務処理フローに修正したものである。現在日本では、日能方式（日本能率協会方式）、能大方式（産業能率大学方式）、NOMA方式などがあり、それぞれ事務分析図表と称する図表と記号が若干異なっている。

事務工程分析を行うにあたっては前述の業務内容調査を行っておくことが大切である。

#### ① 事務工程分析の記号とフロー

##### ① 日能（JMA）方式

JMA方式の事務工程分析の記号および事務フローチャートの例は図 3.9 および図 3.10 の通りである。

書き方は、帳票の流れに従って記号を適用して、必要な記事は記号の横に記入しておく。

フローの流れは上から下へ発生順に書く。

分類	記号	説明
作 業		判断する、計算する、分類する
		○から●に転記する
送 達		書類を運ぶ（相当運くへ送ぶ場合、部門から部門へ送付）
		質の検査、誤字、誤算、記入もれ、段ちがひ、検印など
査 査		量の検査、伝票枚数など
		二つの書類の内容照合
		質、量の検査を同時に行う
停 留		未記入帳票
		用済み、最終保管
		廃票
		一時保管
そ の 他		伝票、複写枚数だけ書く、名称を記入する
		台帳、名称を書く
		報告書、計算用紙、中に名称を書く
		一括する、帳票種類が多いときは一括後の工程線を一本で表わす
		一括した帳票を分離して流す (①、②などと帳票の種別を付記する)
		個人と個人のおかれ目
	課と課のおかれ目	

図 3.9 JMA方式事務工程分析記号

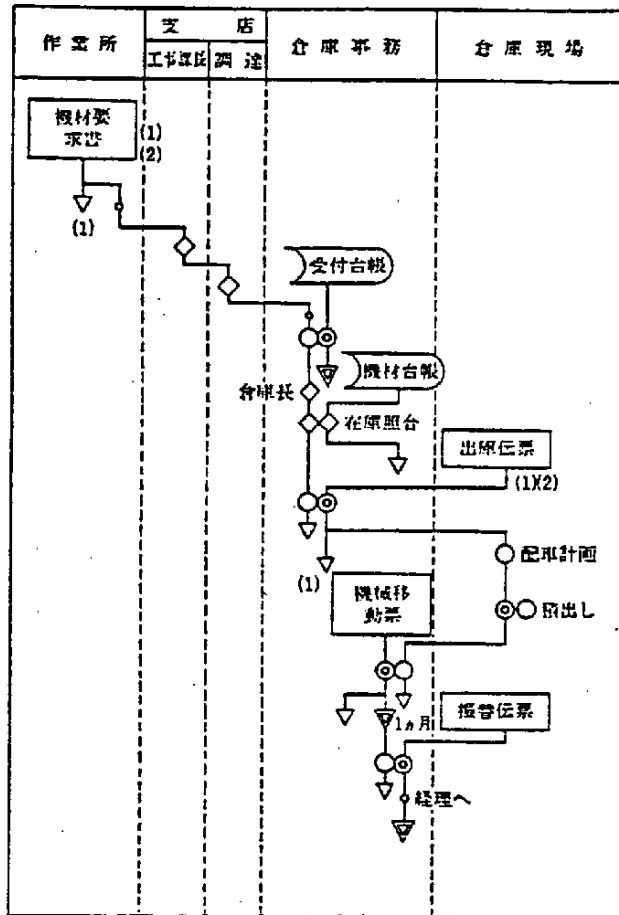


図 3.1 0 機材出庫手続例

⑥ 能大方式

能大方式の分析記号および事務フローチャートの例は図 3.1 1 および図 3.1 2 の通りである。

書き方は、作業を○で示し、その内容はその中に記号で明示するものである。

能大方式は全体の流れが左から右に記入するフローである。



㊦ 日能方式と能大方式との比較

㊦ 日能方式の特徴

記号の数や約束が少なく、物の流れが工程記号に近い。

(i) 長 所：すぐ書けて読める。作業の単位を大きくも小さくもできる。

(ii) 短 所：記号不足分は文字で記入する。

㊦ 能大方式の特徴

記号が多く、約束もやや多い

(i) 長 所：フローチャートに慣れれば読み易い。

(ii) 短 所：記号を覚えにくい。記号が複雑なので、素人への説明がしづらい。

㊦ 事務工程分析表作成上の注意点

フローチャートは対象業務の現状を正確に把握する側面（現状分析用）と分析・改善後の新しいシステムの手順を関係者に示す情報伝達用（事務規程など）との側面があるがここでは前者の現状分析の段階の要点を示す。

㊦ 対象業務の経営機能の明確化

目的と手続の関係を明らかにする（アウトプットの把握）。

㊦ 経営機能の実態の把握

フロー・チャートは帳票の流れを中心としているので、無形の方法手順は調査もれになりやすい。帳票以外の入力情報や決定プロセス等も把握する。

㊦ 目的とする機能が果たされている程度を判定

経営の特性や諸制約条件をマクロに把握しておき、何が行われているかを明らかにする。

㊦ 作業の合理性の検討

目的をもっとも合理的に達成しているか、方法手順自体がもっ



とも経済的に遂行されているかを検討する。

⑥ 時間的空間的条件の検討

⑦ 各工程に必要な資格条件の検討

③ 入出力帳票調査

事務フロー分析等で明らかになった入出力帳票について、現物（記入例のあるもの）を入手し、整理する。

要求定義の各段階ごとに要求される内容は異なるが、図 3.1 3，図 3.1 4 に例を示す。

(1) 入力帳票類一覧

情報名	作成目的	発生源	データ量	関連資料	機密保護	備考
出勤簿	各職員の出勤状況を把握する。	庶務	1,000	人事台帳 貸付金台帳	有	各職員の出勤状況を記録する帳票を新たに設定する。
超過勤務等命令簿	各職員残業状況を把握する。	庶務	1,000		有	
異動通知票	各職員異動に伴って変更のあった情報を把握する。	人事	異動数 (最大1,000)		有	
徴収依頼票	各職員当月の、返済額を把握する。	共済	貸付者数 (最大1,000)		有	

(2) 出力帳票類一覧

情報名	作成目的	利用部門	データ量	関連資料	機密保護	入力帳票との関連	備考
給与支給明細書	各職員給与支給額とその明細を明確にする。	給与各職員	1,000		有 有	出勤簿、超過勤務等命令簿、徴収依頼票	
金種別内訳表	各職員支給額をもとに支給に必要な金種別の現金必要量を明確にする。		1				
給与等	各職員給与支給実績の履歴を記録する。	給与 給与	1,000				

(各省庁電子計算機利用効率化共同研究会「ソフトウェア開発の効率化

研究結果報告書(昭和55年度)」行政管理庁 S. 56. 3 P. 132, 133)

図 3.1 3 入出力調票調査の例 1

a. 用途中心の調査

例 1. (入出力調査表 1)

帳票名	各票名	最終保管	使用区分	摘要
売上伝票	1. 売上伝票	顧客	売上案内	商品に添付
	2. 受領書	経理課	売上計上	受領印を受け回収
	3. 売上出庫伝票	商品課	入・出庫管理	出庫確認記録
	4. 売上伝票控	営業課	控・集計用	日報・商品統計
仕入伝票	1. 送品伝票	支店倉庫	送品案内	商品に添付
	2. 仕入伝票	支店経理課	仕入計上	商品に添付
	3. 受領書	本店経理課	支店売上計上	本支店勘定
	4. 送品伝票控	本社倉庫	出庫管理	出庫確認記録
仕入返品伝票	1. 仕入返品伝票	本社営業課	仕入返品仕切	仕切値
	2. 仕入返品出庫伝票	商品課	入・出庫管理	出庫記録
	3. 仕入返品伝票控	経理課	控	

(「FACOMシステム分析と設計」UASOOLIN\*1 P, 39~40)

b. データ量中心の調査

例 2. (入出力調査表 2)

帳票名	発生 サイクル	月間部数	平均 行数	月間 レコード数	桁 数	ピーク時	ピーク時の レコード数	作成 時間	既存 新規	摘要
売上伝票	毎日	80,000	5	400,000	桁	1~3日 18~20日	170,000	3分	既	
返品伝票	毎日	8,000	5	40,000		13~15日	8,000	3分	既	
入金伝票	25日~ 末日	15,000	1	15,000		25~ 末日	15,000	1分	既	
仕入伝票	週1回	4,000	5	20,000		なし	なし	5分	既	本社で起 票される。
仕入返品 伝票	月2回	2,000	5	10,000		なし	なし	3分	既	

(注) 桁数は、現行のものを入れてもよいし、コード化などが決定した後の概要的な入出力設計にもとづいたものを、後で入れてもよい。

図 3.1 4 入出力帳票調査の例

c. 帳票項目中心の調査

例 3. (入出力調査表 3)

①は入力として決めかねるもの

項目名	桁数	コード化の状況	最大桁数	入・出力	使用文字	特 記	摘 要
【売上传票】 伝票 係	3	—	3	I	数字	毎日 001 から始まる 連番	支店別に、001 から付 ける。
伝票日付	4～ 6	—	6	I	数字	伝票日付が売上日となる。	
住 所	10～ 20	なし	20	①	漢字・カ ナ・数字 ・その他		住所、顧客名をコードで 入力する方法を考える。
顧 客 名	5～ 15	一部	15	①	"	支店別に連番を利用	全社的に使えるコードが 必要
商 品 名	10～ 25	なし	25	①	カナ・英 字・数字	事務ではコードを使用 7桁の英数字コードを 使用	事務で使用しているコー ドは、コンピュータに転 用できるか。
数 量	3～ 5	—	5	I	数字		
単 価	3～ 6	—	6	①	数字		単価と金額の入力は必要 か。マスターとして持たせる。
金 額	4～ 7	—	7	①	数字		
合計金額	4～ 7		7	I	数字		

(「AFCOMシステム分析と設計」UASOOLIN\*1 P. 39～40)

図 3.1 4 入出力帳票調査の例

(5) 連関図法

連関図法は、原因—結果、目的—手段など関係が複雑にからみ合っている問題について、図 3.1 5 のような連関図を作成して、問題の適切な解決策を導くのに有効な技法である。

問題解決に当っては会議方式がとられるが、親和図法が情念的なのに対し、論理的な手法といわれる。

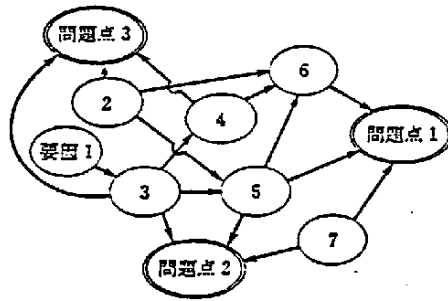




図 3.1 5 連関図の概念図

① 連関図の基本構成

問題点や要因を表す短い文章や結果を  や  で囲み、それらの因果関係を矢線で示す。

矢印の方向は原則的に、原因から結果へ、または手段から目的に向ける。

言葉や文章は短く的確に表現する。

② 連関図法の作成手順

ここでは多目的な問題の解決の場合の手順を挙げる。

① 目的に応じたチームの編成

② チームミーティングによる要因、重要項目の洗い出し

ブレインストーミングなどにより目的達成に必要な手段などを自由にあげて、記録する。

③ 連関図の作成

同類項目の統合や整理をして、各項目間の因果関係を矢印でつなぐ。

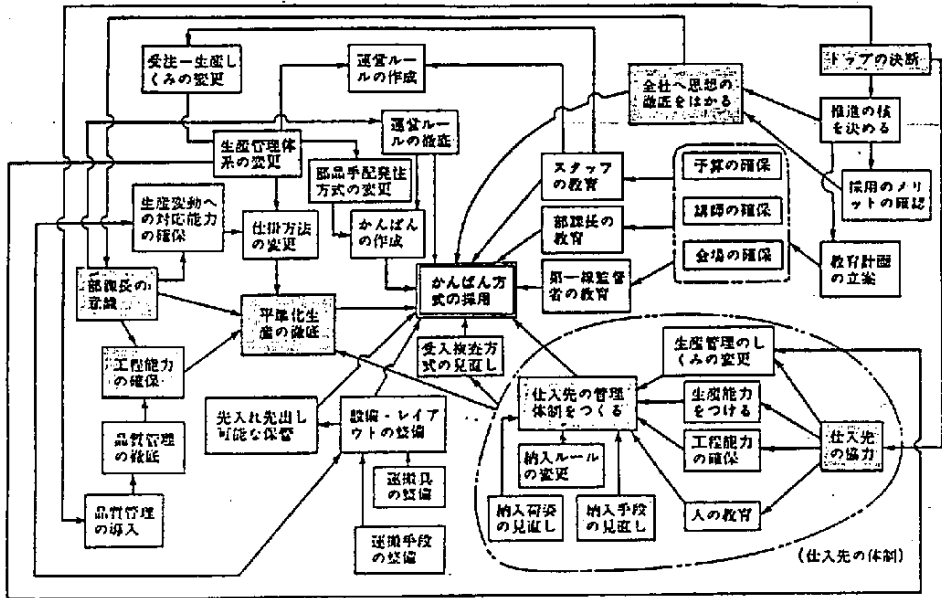
④ 図の修正……全体的立場で見直す

⑤ 重要項目の絞りこみ

重要項目の枠は二重線にしたり、ハッチングする。

⑥ 重要項目の具体的実施計画

事例を図 3.16 に示す。



出典：「新QC七つ道具研究会宿題集」小島晃治氏作成より

図 3.16 新生産方式導入のための連関図

### (6) 系統図法

系統図法とは、目的・目標を達成するために必要な手段・方策を系統的に展開した系統図を作成して、問題の全貌に一覧性を与えて、問題の

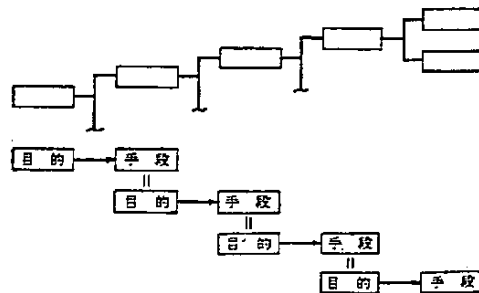


図 3.17 系統図の概念

重点を明確にし、最適の手段・方策を追求する方法である。(図 3.17)

① 系統図の種類

① 構成要素展開型

要素を目的-手段の関係に展開するもの。

② 方策展開型

問題の解決や目的・目標を果たすための手段・方策を系統的に展開するもの。

② 系統図の作成手順(主に方策展開型)

① 目的・目標の設定

対象(業務)について、その究極的な目的を明確にしてカードに記載する。目的は原則として「～を～するためには」と簡潔明瞭に表現する。

目的の達成のための制約事項を明確しておく。

設定した目的に対して「何のためにその目的を果すのか」と質問して一段上の目的を確認し、適切な目的を設定する。

② 手段・方策の抽出

設定した目的を達成するために必要な手段を模造紙に順次記載していく。抽出のしかたとしては、

① レベルの高い手段から順次抽出する。(トップダウン)

② 最末端レベルの手段・方策を抽出し、これをグループ作りしながら、順次上位レベルの手段を抽出する。(ボトムアップ)

③ 思いつくままに手段を抽出していくなどがある。

手段は「～を～する」のかたちで表現する。制約事項も記述する。

③ 手段・方策の評価

制約事項も含めて実行可能性を評価する。実行不可能に見えてもアイデアを追加して実行可能になることもある。

㊦ 手段・方策カードの作成

㊦で実行可能なものを記入する。

㊧ 手段・方策の系統づけ

㊦の目的カードを模造紙左端中央におく。(制約事項は下に記入)

その目的達成には、まずどの手段が必要かを考え、㊦のカードから探し、目的カードの右側におく。2枚以上あれば、縦に並列に配置する。

上記の方策を目的と考えたとき、この目的を果たすためにどのような手段が必要かを考え、㊦のカードから探し、目的カードの右側におく。これをくり返す。

③ 目的の確認

最下位の手段から始めて、順次上位の手段(目的)が下位の手段の全部又は一部で達成できるかを確認する。no のときは手段を追加する。

④ 系統図法の用途

系統図は、トップの方針なり、部門の目標を末端まで浸透させ実践させるために展開されたり、要求定義段階で業務機能を明確して行くのに便利である。また品質改善などの特性要因図としても使用できる。

(7) その他

技法ではないが、事例研究的調査、技術動向調査がある。それ等について述べる。

① パッケージ等調査

下記調査法により、システム名、販売会社、対象業務の範囲、機能、効果、規模(ステップ等)、価額、ユーザ数、プログラム言語、適用機種等を調査する。調査方法は、

① 文献調査

通産省の「ソフトウェア・プロダクト年鑑」、ディーラーのパン



フレット，利用者マニュアル等

④ 実地見聞

ディーラーのデモンストレーション，ユーザの見学等を行い文献以外の情報を確認する。

② 技術動向調査

下記の調査により，システム名，開発会社，対象業務の範囲，機能，効果，規模（ステップ数等），経費，開発工数，使用技法等を調査する。

① 文献調査

他社の事例より，機械化のヒントを得，機械化の技術動向・普及状況を知り，機械化に必要なシステム技法を調査する。

ハード的な情報についてはメーカーの資料も収集する。

④ 実地見聞

ユーザや開発者を訪問し，実際に見聞する。

④ 講習会への参加，コンサルタントの指導を受ける。

### 3.5.3 要求定義技法の効果・留意点

機能分類した要求定義技法の適用に当っては各々の技法の特徴，効果，留意点等について事前に十分認識した上で，最適な技法の選択，組み合わせを設定する必要がある。

そこで，本項では各技法の効果的な選択と利用を図るうえで必要となる次の点を明らかにした（表 3.1）。

(1) 効果・留意点

技法の利用によってもたらされる効果と導入上の問題点，デメリット等を留意点としてとりあげた。

(2) 適用方針

技法の特徴，留意点等をもとに円滑な導入を行うための検討課題とし

て組織体制，作業方法，作業分担，意識，他技法との関連等について検討した。

実際の技法の選定に当っては，これらの特徴，効果，留意点のほかに後述の有効度（システムの規模，重点方針等），適用範囲等を考慮して技法の組合わせを総合的に実施する必要がある。

表 3.1 要求定義技法の効果・留意点および適用方針

開発技法		効果(○)・留意点(△)	適用方針	備考
分析技法	ニーズ分析システム 分析技法	<ul style="list-style-type: none"> <li>○エンド・ユーザの参加により、各種ニーズ問題点を幅広く抽出し、体系的に整理可能</li> <li>○エンド・ユーザの参加意識が高まり、共通の問題意識が生まれ、開発・運用がやりやすくなる。</li> <li>○重要なニーズ、問題点を選択しながら体系的に目的を設定し、機能・範囲の明確化が図れる。</li> <li>△リーダーのリーダーシップ、役割が重要</li> </ul>	<ul style="list-style-type: none"> <li>○ニーズの抽出に際しては関係するエンド・ユーザの参加を求め、コンピュータ知識の有無に関係なく、現状業務についての経験知識および問題意識を持つ社員を選任する。</li> <li>○皆でアイデアを寄せ合い、ニーズ、問題を定義し、解決案を作り、代替案をテストし、実行に移せるものを選ぶプロセスが全員の積極的参加のもとで進められるような雰囲気、組織体制を作る。</li> </ul>	親和図法、ブレーストローミング、目的樹木法、連関図法、特性要因分析図法等
	事務分析技法	<ul style="list-style-type: none"> <li>○事務処理を主として帳票の流れの面からとらえて図示することにより、全体としての流れを把握でき処理工程の改善点を分析できる。組織の形態と機能、情報の流れ、作業時間等の把握に良い。</li> <li>△記号や記述方法などのルールを明確にしてエンド・ユーザでも理解しやすくしておく必要がある。</li> </ul>	<ul style="list-style-type: none"> <li>○組織間の物と情報の流れを分析し、業務内容、入出力帳票を調査し、問題点、改善点を明確にして新システムの処理手順、処理範囲を設定する。</li> <li>○既存の分析記号だけでなく独自のわかり易い記号を工夫して実際の分析に利用すると良い。</li> </ul>	能大式 日能式 NOMA式

表 3.2 要求定義技法の有効度一覧

有効度等 技法		システムの規模別有効度			開発の重点方針別有効度							技術的要因	
		大規模 (20,000以上)	中規模 (5,000以上)	小規模 (5,000未満)	信頼性	生産性	保守性	機能	性能	操作性	経済性	要員の 作業量	要員の 技術的 荷
分 析 技 法	ニーズ分析システム分析技法	○			○			◎				大	中
	事務分析技法	○	○		○			○		○		中	低
	費用対効果分析技法	○									○	中	中
	パッケージ等調査	○	○	○			◎				○	小	低
	技術動向調査	○							○	○		小	高

表 3.3 要求定義技法の適用範囲

要件定義技法		工程区分	システム 化要求	要求分析	システム 分析	開発計画	基本設計	詳細設計	プログラム 設計	プログラ ミング	テスト
分 析 技 法	ニーズ分析システム分析技法										
	事務分析技法										
	費用対効果分析技法										
	パッケージ調査										
	技術動向調査										

#### 3.5.4 要求定義技法の有効度

要求定義技法を機能別に分類し、それを単独に用いる場合の有効度等について分析したのが表3.2である。

有効度を分析する要因として、ここではシステムの規模、開発の重点方針に関して検討した。一方、適用する場合に問題となる技術的要因についても検討した。

システムの規模は大・中・小に分類し、大規模は20,000ステップ以上、中規模は5,000ステップ以上、小規模は5,000ステップ未満を一応の目安として設定した。

ソフトウェア開発の重点方針は信頼性（品質）、生産性、保守性、機能、性能、操作性、経済性の7つに分類した。

技術的要因は、技法を適用する場合の課題である要員の作業量（大・中小）および要員の技術的負荷（技術レベルの高・中・低）の2つについて分析した。

作業量の大・中・小や技術レベルの高・中・低などは相対的なものである。

#### 3.5.5 要求定義技法の適用範囲

本節の前項までは、要求定義技法の分類を行い、個々の技法の特徴、効果と留意点、有効度および技術的要因等について分析してきた。

本項以降では、これらの分析結果に基づいて、どのような方法で最適な技法を選択し、要求定義段階のどの工程でどの技法を適用すれば、効果的に正確で、一貫性があり、完全な要求定義ができ、かつ全工程に亘って、品質および保守性が高いソフトウェアの開発を効率的に行えるかについての検討を行う。

個々の技法にはその目的（特徴）により各工程のうちでも適用して最も効果のある範囲がある。各技法がどの範囲に適用できるかを示したのが表

3.3である。要求定義はシステム化要求，要求分析，システム分析の3工程であるが，後続工程でも適用できる技法もあるために全工程を示した。

### 3.5.6 要求定義技法の選択要因

要求定義技法を効果的に適用していくためには，効果と留意点(表3.1)，および適用範囲(表3.3)など，技法自体の要因を考慮するのはもちろんのこと，表3.2に示した技術的要因(要員のレベル)やシステムの規模と開発の重点方針等のシステム特性を考えて選択する必要がある。

本項では主に，要員レベルとシステム特性による要求定義技法の選択について述べる。

#### (1) システムの規模

システムの規模は技法の有効性と大きな関連がある(表3.2)。通常小規模のシステムでは基本設計に直ちに入ってしまう場合が多い。大・中・小規模の区分は大体的目安であり，境界に近い規模の場合には両方の規模に適用できる技法が選択可能である。

#### (2) 開発の重点方針

開発の重点方針と技法の有効度の間にも大きな関連がある(表3.2)。

一方，個々のシステムの特長により開発の重点方針は決まる。例えば軍事システムでは信頼性，性能，操作性等が重視され，経済性等は殆んど無視される。

また，重点方針間には矛盾がある場合もある。

#### (3) 技術的要因

要求定義法の適用に際しては，その効果，有効度は重要な要因であるが，一方では要員がその技法を使いこなせるかという問題がある。

技法の導入に大きな訓練を必要とするとか，作業量が膨大になる場合には有効な技法でも適用ができない場合がある。

このため技術的要因として①要員の技術的負荷，②要員の作業量を挙

げ、各々3段階に区分して目安とした。システム開発を行う場合には開発体制も検討して技法を選択する必要がある。

### 3.5.7 要求定義技法の選定手順

要求定義段階での各種の技法を導入する場合には、開発体制の現状を充分認識した上で最適な技法の適用を図る必要がある。本項では、最適な技法を選択するための方法を検討し、図3.18のような手順を設定した。

この手順は全工程に及ぶ技法の選択にも適用可能と思われる。

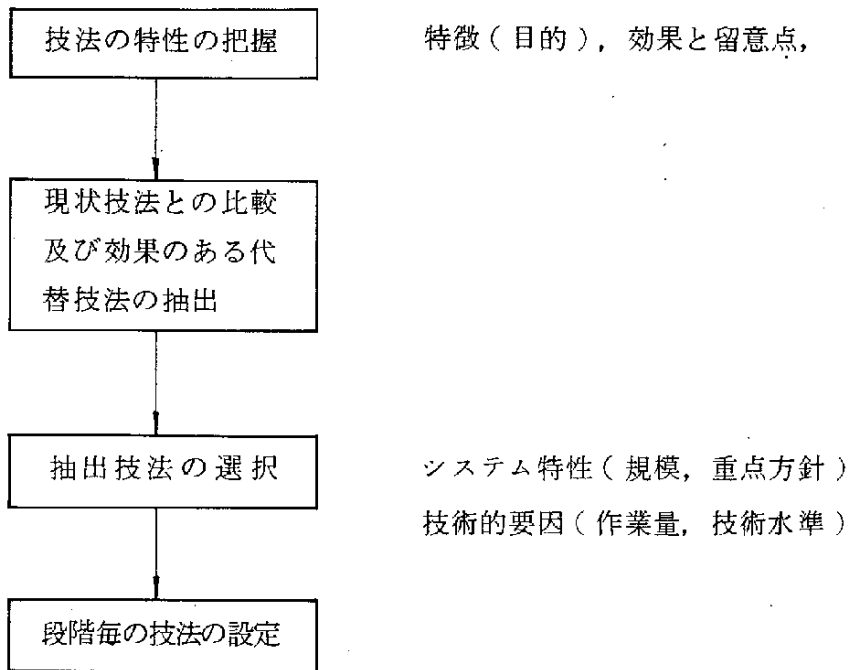


図 3.18 技法の適用手順

(1) 技法の特性の把握

要求定義技法について、それぞれの特性の把握を行う。

(2) 代替技法の抽出

対象業務の実態に即してそれぞれの技法を比較して、より効果のあると思われる技法をいくつか抽出する。

### (3) 抽出技法の選択

前項に基づいて、システムの規模、開発の重点方針、技術的要因を分析し、最終的に導入する技法を選ぶ。

## 3.6 要求定義の留意点とチェックリスト

3.2節から3.4節においては、作業内容とワークドキュメントの項目が設定され、的確な作業および記述を行うための具体的指針が示された。

また、3.5節では各種の要求定義技法の特徴が示され、その最適な選択を行うことにより、要求定義段階の作業の生産性を高め、かつ信頼性および保守性が向上することが示された。

本節においては、要求定義段階における留意点とチェックリストを、ワークドキュメントの記述上のポイント及び作業上のポイントとしてまとめた。

これは3.2節から3.5節の内容を要約したものであり、要求定義作業の生産性、信頼性向上につながれば幸いである。

チェックリストはシステム化要求段階、要求分析段階、システム分析段階の3つの工程区分ごとに、表3.4に示されているが、その工程区分毎の表の構成は以下の通りとなっている。

### (1) 記述項目

ワークドキュメントの記述項目の名前であり、主項目と詳細項目に分けた。

### (2) 記述上のポイント

これは、記述項目の詳細項目毎に、記述する場合の留意点とチェックリストを挙げた。

### (3) 作業上のポイント

これは、要求定義の作業を行う場合の視点、作業方法等についての留意



点とチェックリストを記載した。

(4) 担当部門

ユーザ，企画・設計，運用の3つの部門に区分して，作業の主要担当部門には○印を，必要に応じて協力する部門には△印を記入した。

(5) 特記事項

適用可能な技法・ツール，その他を記載した。

表 3.4 要求定義の工程別記述上及び作業上のチェックリスト

凡例 ○：主要担当部門

△：必要に応じ担当する部門

システム化要求段階

記述項目		記述上のポイント	作業上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
1.システム化の内容	(1)目的	<ul style="list-style-type: none"> <li>*システム化の目的を明確に設定する。                             <ul style="list-style-type: none"> <li>◦業務の改善，合理化…人件費，事務経費の節減；生産性，経済性，安全性の向上</li> <li>◦管理の質の向上…正確性，迅速性，タイムリー，使い易さ</li> <li>◦手作業業務の軽減…省力化</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>*エンド・ユーザ部門の目的意識を統一し，明確にする。</li> </ul>	○			目的樹図法，親和図法，ブレーンストーミング等
	(2)背景	<ul style="list-style-type: none"> <li>*当該システムの現状を明確にする。                             <ul style="list-style-type: none"> <li>◦現行システムの業務内容，事務の流れ，手続等資料</li> <li>◦ユーザ部門の組織，システム化要求の責任者，参加者等</li> </ul> </li> <li>*当該業務の問題点とシステム化要求の経緯・理由を明確にする。                             <ul style="list-style-type: none"> <li>◦主な問題点（障害状況）と原因と改善策制度の改正，データ管理の一元化，業務サービスの改善，業務量・人員増加</li> <li>◦あれば同業他社の現状</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>*現在の業務内容，事務の流れを明らかにする。</li> <li>*エンド・ユーザが直面している問題点を明確にし（改善意欲・参加意識を高め，アイデアを出しやすくして）改善案をまとめる。</li> </ul>	○			事務分析技法，ニーズシステム分析技法等 ブレーンストーミング，特性要因分析図法，親和図法等

記述項目		記述上のポイント	作業上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
	(3)必要性	<ul style="list-style-type: none"> <li>* 当面している背景に基づいてシステム化する必要性を明確に設定する。 <ul style="list-style-type: none"> <li>◦ 問題解決のための緊急度，優先度</li> <li>◦ 改善内容の必要性，緊急度</li> <li>◦ 段階的開発か，一度に開発か</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>* システム化の必要性を関連部門で幅広く意識統一するとともに新システムの実施に際しての協力体制を確立する。</li> </ul>	○			
	(4)効果	<ul style="list-style-type: none"> <li>* 管理局面の効果や影響 <ul style="list-style-type: none"> <li>◦ 業務処理…迅速性，正確性，簡易性，弾力性等</li> <li>◦ データ利用・管理…容易性，適時性，安全性，プライバシー等</li> <li>◦ 人事・組織…モラル，関連組織間の連携，人事異動，組織変更等</li> </ul> </li> <li>* コスト局面に関する効果・費用 <ul style="list-style-type: none"> <li>◦ 現行システムのコスト</li> <li>◦ 新システムのコスト（開発，運用，保守）ユーザ部門の協力体制も含む。</li> </ul> </li> <li>* 業務・経営局面における効果・影響 <ul style="list-style-type: none"> <li>◦ 業務・経営の質・レベル，既存システムへの影響度</li> <li>◦ 顧客サービス，波及効果</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>* 効果および新システムのもたらすデメリット等も充分検討し，その対応策も盛り込むことが重要である。</li> </ul>	○			費用対効果分析技法等

記述項目		記述上のポイント	作業上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
	(5)運用時期	<ul style="list-style-type: none"> <li>* 運用開始の希望時期を明確に設定し、その必然性、理由を明確にする。</li> <li>◦ 段階的な開発の場合には優先度に応じて</li> <li>◦ 開始時期の制約（法・制度・業務上）</li> </ul>	<ul style="list-style-type: none"> <li>* システムの規模、難易度、要員の確保等とは関係なく希望時期を設定する。</li> </ul>	○			
2.適用範囲	(1)対象業務範囲	<ul style="list-style-type: none"> <li>* 新システムの範囲の決定</li> <li>◦ 新システムの業務内容、事務の流れ等を明らかにする。</li> <li>◦ 改善ヶ所を明記する。</li> <li>* 他業務との関連、影響度と対策</li> <li>◦ 管理局面…組織変更、人事移動、情報伝達法・サイクル等</li> <li>◦ コスト局面…原価等の増減</li> <li>◦ 業務・経営局面…既存システム・他業務への影響等</li> </ul>	<ul style="list-style-type: none"> <li>* 背景で検討した事項とともに新システムの対象業務の範囲を設定するとともに、関連部門及び他業務に対する影響、波及度合を調査し、関連部門の合意を得ることが重要である。（効果では主に良い影響について検討したが、ここでは主に悪い影響がある場合の対策と合意が重要となる）</li> </ul>	○			事務分析技法、ニーズ分析システム分析技法等
	(2)人手作業とのインタフェース	<ul style="list-style-type: none"> <li>* 人手作業とコンピュータ処理との境界及び情報の授受方法、手段等を明確にする。</li> <li>◦ 事務手続と伝票、帳票類の対応（新システム用のもの）</li> <li>◦ 人手作業の変更内容</li> </ul>	<ul style="list-style-type: none"> <li>* 事務処理の流れを伝票類のデータの流れと対応させ、人手作業の範囲、コンピュータ化の範囲を明確にする。</li> </ul>	○			

記述項目		記述上のポイント	作業上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
	(3)機能拡張方針	<p>* 将来のシステム拡張、拡大及び新規業務等による機能拡張、変更の方針を明確にする。</p> <ul style="list-style-type: none"> <li>◦ 拡張機能内容、基礎データ（処理時間、処理件数など）の予測</li> <li>◦ 組織及び法・制度・規則及び業務の新設、改訂</li> </ul>	<p>* 将来にシステム拡張が予測されている場合には、その拡張機能を充分検討して大幅な変更にも対処できるように考慮する事が重要である。</p>	○			
3.関連情報一覧	(1)入力帳票一覧	<p>* 対象業務に関連する入力帳票類を全て抽出し、その属性（情報名、作成目的、発生源、データ量、関連資料、機密保護の必要性等）を明確にする。</p>	<p>* 帳票類の形式及び制約を明らかにする。 * 業務特性に応じた各種帳票、図表の見直しを行う。 * メモ、電話等でのやりとりが定常化していれば対応策を考慮する必要がある。</p>	○			
	(2)出力帳票一覧	<p>* 対象業務に関連する出力帳票類を全て抽出し、その属性（情報名、作成目的、利用部門、データ量、関連資料、機密保護の必要性等）を明確にする。</p> <ul style="list-style-type: none"> <li>◦ 入力帳票と出力帳票の属性の関連</li> </ul>	<p>* 帳票、メモ類は実際に使用されている現物をまず収集する。</p>				

記述項目		記述上のポイント	作業上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
4.制約条件		<ul style="list-style-type: none"> <li>* 時間的条件（資料作成期限，応答，待ち時間の許容度等）を明確にする。</li> <li>* 経済的条件（開発・運用経費等）を明確にする。</li> <li>* 技術的条件（信頼度，規格，性能，帳票類形式，操作性，要員の技術レベル等）を明確にする。</li> <li>* 組織的条件（地理的制約，分散管理状況，管理体系等）を明確にする。</li> <li>* 法的条件（業務を規制している法律・制度・規則・手続き等の名前・条項等）</li> </ul>	<ul style="list-style-type: none"> <li>* 必要条列及び制約条件をエンド・ユーザ各部門から提出し，必要と考え得るもの全てを記述するようにする。</li> </ul>	○			

要求分析段階

記述項目		記述上のポイント	作成上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
1. システム化の内容分析	(1)目的・効果の分析	<ul style="list-style-type: none"> <li>* 目的・効果の的確性を確認する。</li> <li>◦ 効果に関する判定基準の妥当性</li> <li>◦ 目的と効果の関連性のチェック</li> </ul>	<p>* ユーザ部門の提起した目的意識を充分尊重すると共に、背景・必要性から見て、それが的確か、一貫しているか、完全かについて、客観的に検討・確認する。</p>		○		目的樹木法，親和図法，面接調査，費用対効果分析
	(2)背景・必要性の分析	<ul style="list-style-type: none"> <li>* 背景・必要性を的確なものにする。</li> <li>◦ 現行システムの確認と記載もれ等のチェック</li> <li>◦ 現行業務の問題点・原因を検討・確認し，真のニーズを把握すると共に，改善策の妥当性，一貫性のチェック</li> <li>◦ 必要性（優先度等）の妥当性</li> </ul>	<p>* 背景・必要性で提起された内容により，現行システムを把握し，問題点と改善策の妥当性を検討し，的確なものにする。</p>		○		事務分析技法，HIPO，SADT，特性要因分析技法，面接調査等
	(3)運用時期の分析	<ul style="list-style-type: none"> <li>* 運用時期を確認し，設定する。</li> <li>◦ 開発スケジュール（優先度含む）。</li> <li>◦ 開発要員，体制，経費等の大まかな開発計画の設定。</li> </ul>	<p>* 運用時期に対して，開発要員，体制，経費，スケジュール等の目安を算出し，実現可能な時期かどうかを充分検討する。</p>	△	○	△	親規業務，大規模システムではユーザ等と充分な検討を行うこと。

記述項目		記述上のポイント	作業上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
2.適用範囲の分析	(1)対象業務範囲の分析	<ul style="list-style-type: none"> <li>* 対象業務の範囲が目的、必要性を具体化する内容となっているかを確認する。 <ul style="list-style-type: none"> <li>○ 事務処理の流れに対応する業務内容、処理範囲の妥当性</li> <li>○ 目的、必要性と業務範囲の対応</li> </ul> </li> <li>* 他業務との関連、影響度と対策の妥当性の確認 <ul style="list-style-type: none"> <li>○ 既存システム、データベースへの影響と対応策は妥当かの確認</li> <li>○ 組織、法・制度の新設・改訂等による影響度と対応策の確認</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>* 対象業務の内容、処理の流れを改善策をもとに効率化、合理化、省力化等の観点から見直し、新システムの適用範囲を決める。</li> <li>* 人手作業による手順は文書化されていないことが多いので抜けがないか充分検討・確認する。</li> <li>* 目的・必要性等ユーザ提起の問題点、改善事項を念頭に入れ、それを満足した業務範囲を設定しているかを検討・確認する。</li> </ul>	△	○		事務分析技法、ニース分析システム分析技法、HIPO、SADT等
	(2)人手作業とのインタ・フェース分析	<ul style="list-style-type: none"> <li>* 対象業務の作業分担、情報の授受方法等のマンマシンインタ・フェースの妥当性 <ul style="list-style-type: none"> <li>○ 事務処理の手続き及び期限の妥当性</li> <li>○ 人と機械処理の対応手順</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>* 情報の流れ、授受方法の一元化、統合化の可能性及び事務のスムーズな流れが実施できる体制が確立できるか否か十分な分析と確認を行う。</li> </ul>	△	○		同上
	(3)機能拡張分析	<ul style="list-style-type: none"> <li>* 機能拡張の必要性及び対策を分析・確認 <ul style="list-style-type: none"> <li>○ 機能拡張・追加の必要度確認</li> <li>○ データ量、対応策の分析・確認</li> <li>○ 関連部門への影響度の把握</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>* ユーザ提起のシステム及び機能拡張拡大方針を前提とし、背後にある必要性、データ類の伸び等を確認してシステム化に反映できるような方策を考慮する。</li> </ul>	△	○		



記述項目		記述上のポイント	作業上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
3. 関連情報の分析	(1)入力帳票分析 (2)出力帳票分析	<p>* 業務内容と入出力帳票類の関連について分析・確認する。</p> <ul style="list-style-type: none"> <li>データ項目の内容・属性・必要性</li> <li>帳票類の統廃合，標準化，簡素化</li> <li>入力帳票と出力帳票の属性の関連性</li> </ul>	<p>* 事務処理の内容及び手順を検討し，必要最小限の帳票類の選定を行い，統廃合，標準化の見地から関連情報を作成（構造化）し，ユーザ部門に提案する。</p>	△	○		
4. 制約条件の分析		<p>* 制約条件で規定した内容及び背景となったデータ，環境状況を分析・確認する。</p> <ul style="list-style-type: none"> <li>必要不可決条件と希望的条件の区分</li> <li>制約条件を満たすために特に必要となる事項</li> </ul>	<p>* 条件の内容を充分把握分析し，システム化を行う場合のメリット・デメリットを明確にするとともに必要性，コスト面から検討する。</p>	△	○		
5. システムの妥当性・必要性		<p>* エンド・ユーザ部門の要求内容を詳細に分析した結果に基づいて，システム化の妥当性，必要性の結論を明確にする。</p> <ul style="list-style-type: none"> <li>システム化要求案としての可否</li> <li>要求内容の妥当性</li> <li>改善内容の設定</li> </ul>	<p>* 企画・設計部門としての立場から，技術的かつ客観的に分析を行い，システム化の妥当性の判断及び次段階への進展の可否について結論を出す。</p>		○		

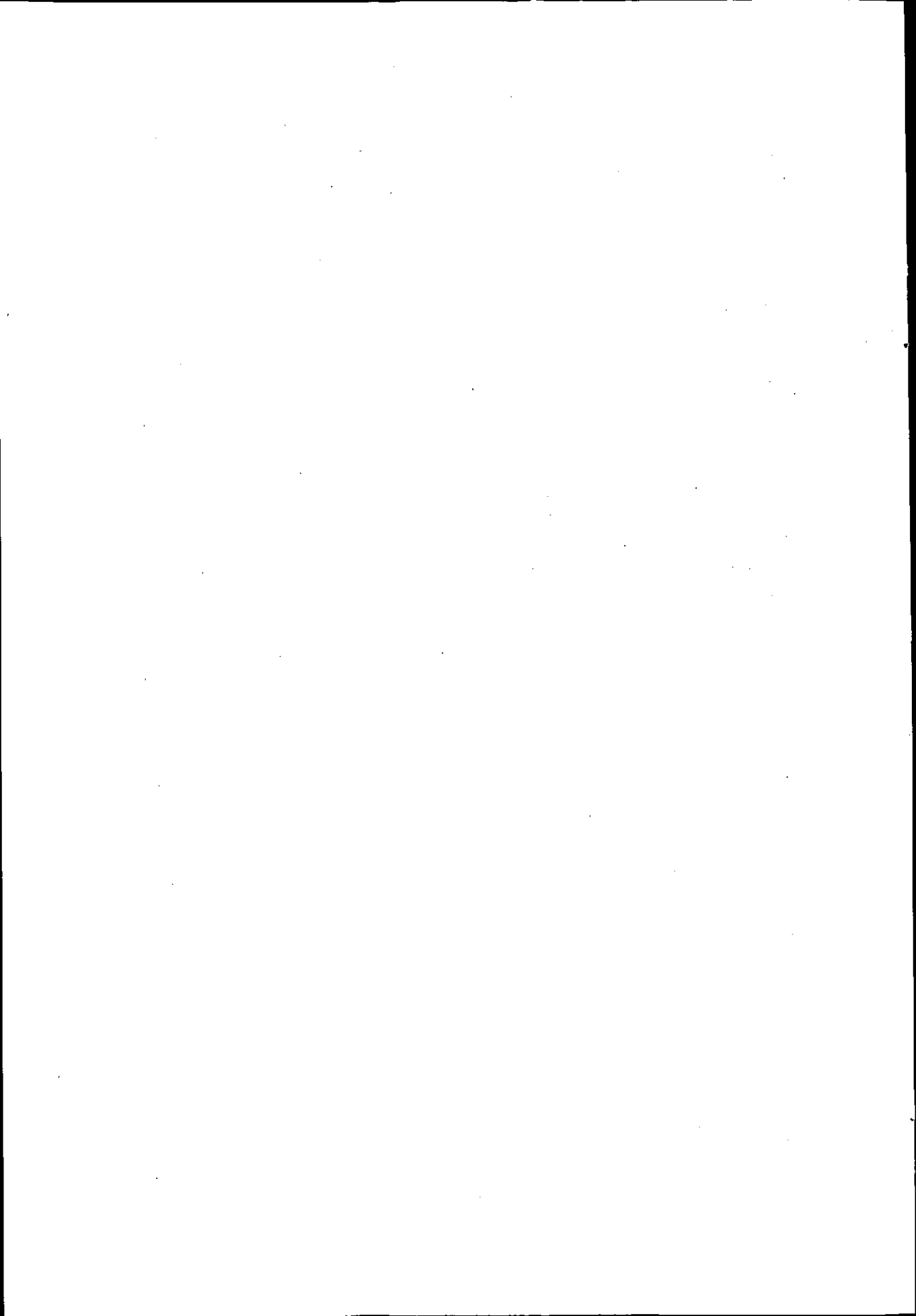
システム分析段階

記述項目		記述上のポイント	作業上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
1.類似システム及び技術動向調査		<p>*類似システムの特質(対象範囲, 技能, 効果, 規模, 経費, 期間, 使用技法等)を的確にとらえての長所・短所を分析して, 当該システムに反映できるようにまとめる。</p> <p>*システム化に関連する各種技術の調査を行い, 特徴, 動向等をまとめる。</p> <ul style="list-style-type: none"> <li>○ハードウェア機能, 性能, 価格</li> <li>○システム開発, 運用の方法・技法等</li> </ul>	<p>*同業他社, その他の先進事例, 各種文献, 海外資料等を参考にし, 広範囲に当該システムに有効となる技術を吸収し, システムに反映できるように対処する。</p> <p>特に制約条件で対応策に問題がある所は良く調査する。</p>		○		パッケージ等調査, 技術動向調査等
2.環境条件	(1)ハードウェア	<p>*システムの目的及び制約条件を満たすための機器構成を明確にする。</p> <ul style="list-style-type: none"> <li>○業務の規模, 処理内容等に応じた機器の選択と構成</li> <li>○開発・運用上の問題点と対策</li> </ul>	<p>*類似システムを参考とするとともに, 対象業務の機器構成を決め, 開発・運用上の問題があれば対応策(導入, レベルアップ等)を検討する。</p>	△	○	△	
	(2)ソフトウェア	<p>*システムの目的, 特質に応じたソフトウェアの選択を行う。</p> <ul style="list-style-type: none"> <li>○使用言語等</li> <li>○パッケージ, 既存プログラムの利用</li> <li>○データ・ベースの利用・変更等</li> </ul>	<p>*類似システムを参考とすると共に, 当該業務に使用するソフトウェアの特性を, 経費, 工期, 保守性等の面から検討する。</p>	△	○	△	パッケージ等調査

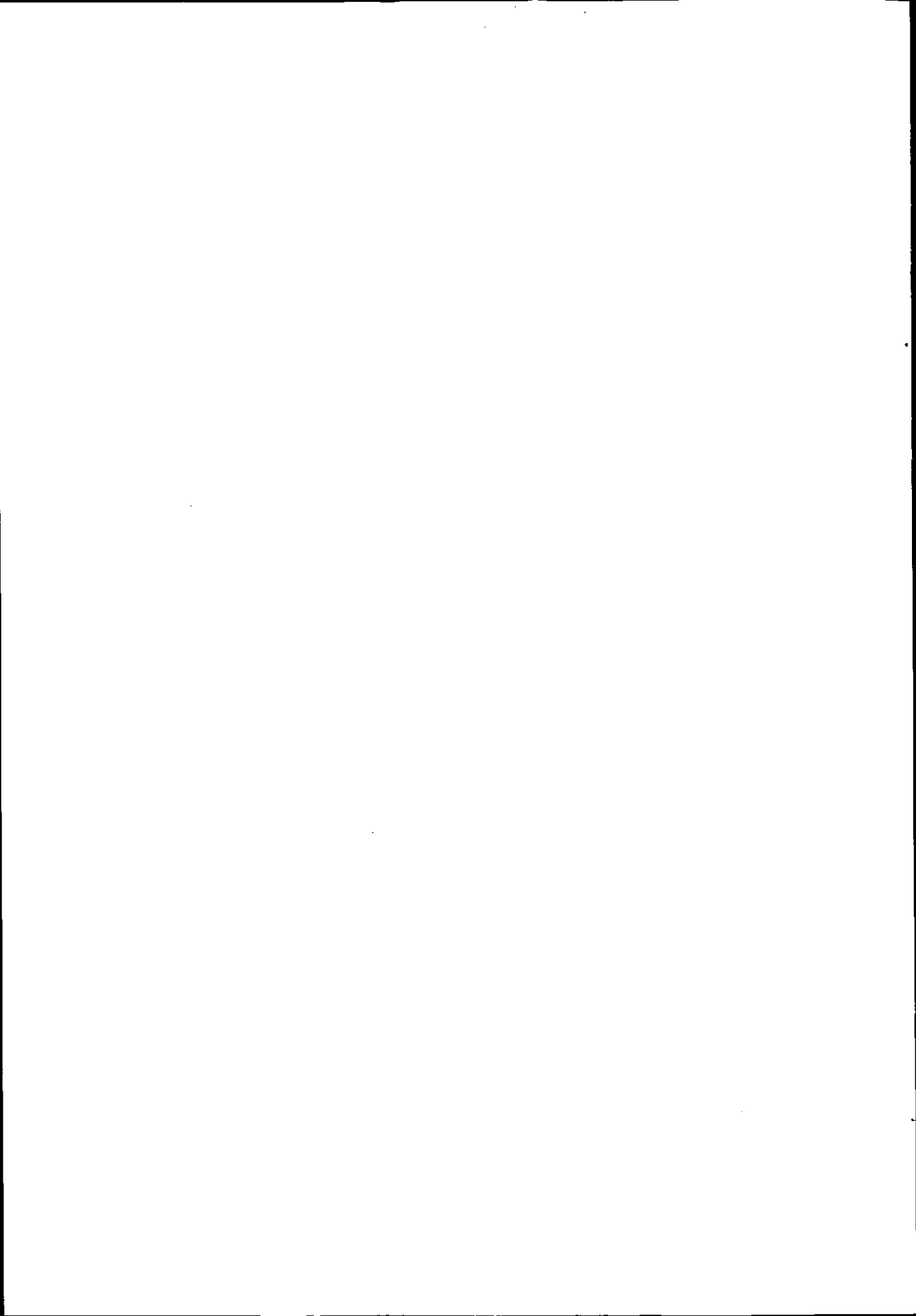
記述項目		記述上のポイント	作業上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
	(3)制約条件	* 時間的, 経済的, 技術的, 組織的, 法的条件としての最終的な制約条件を明確に設定する。	* ユーザ部門と企画・設計部門で十分な意見の統一を図り, 制約条件を決定する。	○	○	△	
3.代替案の作成及び検討	(1)機能・性能	* システム化要求案と代替案の機能, 性能を対応させ, 横断的な比較・検討を行う。 ○ 対象業務の範囲, 機能; 処理効果 ○ 関連部門とのインタ・フェース	* 要求内容に基づいてシステムの理想形態, 必要最小限構成, 現要求の形態等を考慮し, 機能の範囲を設定する。	○	○		
	(2)経済性	* 開発・運用経費を主体に経済性の比較・検討・評価を行う。 ○ 機能と経費 ○ 環境条件を満たすか否かと経費等	* 上記システムの個々の機能, 範囲に応じた開発・運用に必要な経費の見積りを行う。	○	○		費用対効果分析技法等
	(3)効果	* システム化の効果を管理, コスト, 業務・経営局面から比較検討評価を行う。	* 比較・検討に際しては, 各々の代替案の相違点, 長所, 短所を明確にする。				
	(4)技術分析	* 適用しようとする各種技術及びその導入方法等を比較検討評価する。 ○ ハードウェア技術動向 ○ ソフトウェア技術動向			○	○	

記述項目		記述上のポイント	作業上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
	(5)類似システムとの対比	<ul style="list-style-type: none"> <li>* 類似システムの調査結果と代替案を比較検討し、評価する。</li> <li>◦ 相違点, 長所, 短所</li> <li>◦ 機能, 範囲, 経済性, 採用方法等</li> </ul>		○	○		
	(6)体制・工期	<ul style="list-style-type: none"> <li>* 開発完了までの作業工期, 所要人員について比較検討評価を行う。</li> <li>◦ 業務範囲, 機能毎の開発・運用の工数, 体制, 工期</li> </ul>	<ul style="list-style-type: none"> <li>* 費用及び要員, 体制等を考慮し, 代替案の内容の実現程度を充分認識・把握することが重要である。</li> </ul>	○	○		
	(7)開発難易度	<ul style="list-style-type: none"> <li>* 開発作業の難易性を十分予測し, その対策を盛り込み, 比較検討評価する。</li> <li>◦ 作業内容, 作業量, 技術的対応</li> <li>◦ スケジュールの作成</li> </ul>					
4.システム案	(1)システムの目的効果	<ul style="list-style-type: none"> <li>* 最も有利なシステム案を作成し, その目的及び効果を明確に設定する。</li> </ul>	<ul style="list-style-type: none"> <li>* 代替案の検討の結果得られた最適な案がこの案である。</li> </ul>				
	(2)システムの範囲	<ul style="list-style-type: none"> <li>* システム化の業務範囲を明確にする。</li> <li>◦ 対象業務名, 概要</li> <li>◦ 対象業務システム化の範囲</li> <li>◦ 人手作業とのインタ・フェース</li> </ul>	<ul style="list-style-type: none"> <li>* ユーザ部門, 企画設計部門, 運用部門の最終的合意が得られた内容としてとりまとめる。</li> </ul>	○	○	△	

記述項目		記述上のポイント	作業上のポイント	担当部門			特記事項
主項目	詳細項目			ユーザ	企画設計	運用	
	(3)システム概要	<p>* システムの基本的な処理機能（サブシステム）を明確にする。</p> <ul style="list-style-type: none"> <li>◦ 入出力情報</li> <li>◦ 基本機能の構成と処理概要</li> <li>◦ システムの規模，性能</li> </ul>	<p>* 予算・要員等で大幅な変更がない限り，目的，効果，範囲，概要，経費，工期等の内容はシステム化の最終案としてシステム設計に反映できるようにする。</p>				
	(4)経費・工期	<p>* システムの開発・運用に必要な経費，工期，要員等を明らかにする。</p>					



#### 4. ソフトウェア開発計画





## 4. ソフトウェア開発計画

ここでは、前章の要求定義・分析、システム分析等の結果をもとに、開発規模の見積から種々の計画の立案、及びそれらに関連する事柄について述べる。

### (1) 開発計画の種類と内容

システム設計から、プログラムの作成、運用保守・拡張に至るソフトウェア・ライフサイクルの中で作成される16種類の計画について、その目的、盛り込まれるべき事項、又、開発計画立案の実態、開発計画相互の関連、開発計画の変更の要因等について述べる。

### (2) 技法・手法・ツール

開発計画の立案を支援するための技法・手法・ツールについてその概要、特徴、利用方法、利用効果等を述べる。

### (3) 見積方法

開発計画の立案のもとになる見積方法の概要、見積基準、見積の実態等について述べる。

### (4) 開発計画の作成から承認まで

承認の手続の事例、計画作成者と承認者との関係、承認の内容、承認後の権限と義務等について述べる。

### (5) 外注の活用

開発計画段階で外注利用をどの様に組込んでいるかについて、その目的、利用の度合、外注する段階、発注の形態、外注先の評価、外注費の算定方法等の側面から述べる。

### (6) 留意点とチェックリスト

以上述べてきた事柄について、それぞれの留意点を一覧表の形式でまとめている。

#### 4.1 開発計画の種類と内容

ソフトウェアの開発から運用・保守に至るライフサイクルの中で、種々の計画が必要になってくる。

ここでは、アンケート調査の結果及び文献等によって得られた情報をもとに、計画の種類とそれぞれの目的、盛り込まれるべき事項、開発計画相互の関連、ライフサイクル上での位置付け等について述べる。

計画の種類としては下記のことを挙げることができよう。

- (1) ソフトウェア開発における見積、予算、スケジュール等に関する計画
  - ① 工程計画
  - ② 予算計画
  - ③ 要員計画
  - ④ 外注計画
  - ⑤ 要員教育計画
  - ⑥ 設備計画
- (2) 開発作業の実施における管理及び支援等に関する計画
  - ⑦ 開発支援計画
  - ⑧ 工程管理計画
  - ⑨ 費用管理計画
  - ⑩ 品質管理計画
  - ⑪ 検査・検収計画
- (3) 移行、運用、保守等に関する計画
  - ⑫ 移行計画
  - ⑬ 運用計画
  - ⑭ 保守・拡張計画
- (4) その他の計画
  - ⑮ パッケージド・ソフト購入計画

#### 4.1.1 種類と概要

##### (1) 工程計画

前3章の要求分析及びシステム分析等の結果を踏え、開発の対象となるソフトウェア・システムの規模、工程毎の工数、工期等を見積り、納期と工程相互の関係を勘案しつつ実施可能な開発スケジュールとしてまとめることが目的となる。

計画の範囲は、通常、対象となるソフトウェア・システムのシステム設計からプログラムの作成、テストを経て、そのシステムがエンドユーザーによる運用に入る直前までを指すことが多い。

工程計画に盛り込むべき、主な事項としては、下記のものを挙げる事ができよう。

##### ① 全体の規模

対象となるソフトウェア・システムの規模を見積り、総ステップ数、プログラムの本数等で、その概略を把握する。

この際、システムの構成図や、機能図等を添えることは、見積根拠を明らかにする上で、有効である。

##### ② 工程の内容

各工程毎に下記の事項を明記する。

- ① 仕事の範囲、作業内容及び完了の条件等
- ② 必要な工数
- ③ 工期
- ④ 使用を予定している設備、端末等の台数、及びその使用時間等
- ⑤ 作成するドキュメントの種類及びそれに盛り込むべき事項

##### ③ その他

- ① 開発対象システムの名称及び概要
- ② 開発方針
- ③ 計画の立案年月日、納期

⊖ 計画の立案部門，立案者，承認者

(2) 予算計画

(1)の工程計画に基づいて、人件費、設備使用費、材料費、雑費等を見積り、予算を算定する。

人件費には、直接人件費、外注費、アルバイト費等が含まれる。

設備使用費は、工程計画で見積った使用時間に単価を掛けて算出する。

材料費は、工作機械の制御システムの開発のように、ハードウェアの一部を含めて開発する場合に、その材料費を算定する。

雑費には、交通・通信費、パンチ代、磁気テープ、プリンターシート等の消耗品費、コピー代、参考文献やマニュアル等の資料費等々が含まれる。

これらを工程毎に積み上げたのち、全工程を集計したものが直接費であり、これにプロジェクト遂行のための管理費又は部門間接費を加えたものが開発予算となる。

(3) 要員計画

年々、ソフトウェア開発要員の不足が深刻化し、その確保が難しい時代になって来ているが、そのような環境の中で、必要とされる技術、経験を持った要員を確保するとともに、未熟練者の利用範囲をも検討し、それらをソフトウェア開発の各工程に、いかに適切に配置していくかを計画することが要員計画の目的となる。

それは、一つのソフトウェア開発計画内の問題にとどまらず、現在進行中の開発プロジェクトや同じ時期に開発を実施しようとしている他のソフトウェア開発計画等との調整も含むものである。

要員の配置に当っては、単に余力のある人間を割当てるといった方法ではなく、人事管理的要素も強いので、その要員の技術力や経験量に加えて、その性格や要員相互の人間関係にまで配慮しなければならないところにより一層の難しさがある。

又、開発段階だけにとどまらず、運用段階における要員の配置において、開発者から運用担当者への情報の伝達のために開発要員の一部を運用要員として継続させる等、開発・運用相互の関係についての配慮も重要になってくる。

ソフトウェア開発は、個人の能力や適性・やる気といったことに負うところが特に大きい仕事であるため、以上述べて来た諸事項を配慮しつつ要員計画を立てることは、ソフトウェア開発の高度化、効率化に効果を発揮するものである。

要員計画に盛り込むべき事項としては、開発段階における各工程に対し、下記の事項を挙げることができよう。

- ① 担当者の氏名及びその仕事の内容範囲等
- ② 作業期間
- ③ 作業完了の条件

#### (4) 外注計画

外注計画は、外注する目的、外注する仕事の内容や範囲、金額、外注方法及びその他の諸条件を明確にすることが目的で、そこに盛り込むべき事項としては、下記のことを挙げることができよう。

##### ① 外注する目的

外注する目的には種々あり、それを明確にしておく必要がある。その一例を挙げると下記のようなものがある。

- ・社内要員の不足を補充するため。
- ・社内の技術力不足を補うため。
- ・プログラム作成段階の要員を会社の方針として持たないため。
- ・費用の削減を計るため。

##### ② 外注する工程及び仕事の内容

##### ③ 外注することによって得られる成果物

##### ④ 外注方法及び契約形態

- ・一括請負形式
  - ・人員派遣形式
  - ・出来高払い形式
  - ㊤ 外注予定金額
  - ㊦ 発注時点及び納期，作業期間等
  - ㊧ 外注先の選定規準及び必要とされる技術力等
  - ㊨ 進捗状況の把握方法
  - ㊩ 検収方法，検収条件等
- (5) 要員教育計画

要員教育はその目的によって下記の3つの場合が考えられる。

- ① プロジェクトには関係なく，情報処理要員の知識，技術等の向上を目的として行なわれる教育で，新人教育，データベース・システムや新しい言語，又，管理技術等々の講習会への参加，社内研習会や勉強会等がこれに含まれる。

長期的なビジョンに立って計画が立てられるのが通例である。

- ② 1つのソフトウェア・システムの開発に当って，その仕事に必要な知識や技術を修得することを目的とした教育で，（仮に「開発時教育」と呼ぶ）一般に，下記のような事柄がそのテーマになる。

- ・開発するソフトウェア・システムの対象となる業務内容や背景，目的等。
- ・開発用コンピュータのOSやJCL，操作方法等。
- ・使用する言語やデータベース・システム等。

この計画に盛込むべき事項としては，上記の教育テーマ又は内容の他に下記のを挙げることができよう。

- ㊠ 教育者又は社外の場合は教育機関
- ㊡ 教育の対象者

この教育の場合には，対象者は比較的限られており，主にそのソ

ソフトウェア開発の担当者とその管理者，場合によっては，ユーザー部門の人や外注先の担当要員などが含まれる。

又，必ずしも全対象者に一率に全教育内容を施す必要のない場合には，対象者をグループ分けして，教育内容を限定すると，時間の節約を計ることができる。

㊦ 教育方法及び教材，スケジュール等

㊧ 教育成果の判定方法

教育が充分に行なわれて，各担当者が必要な知識や技術を身につけたか否かは，後の開発作業に大きく影響する。このため，試験をする，面接をする，発表させる等，教育成果の判定方法を明らかにしておくが良い。

③ 開発されたソフトウェア・システムが運用段階に入る前に運用担当者や利用者に，そのシステムの使い方や利用の仕方等を修得させることを目的とした教育で，（仮に「運用時教育」と呼ぶ）一般に，下記のような事柄がそのテーマとなる。

- ・そのシステムの概要。
- ・運用計画にもとづいた運用スケジュール
- ・伝票の記入方法やその入力方法等
- ・出力諸表の見方や利用方法等

この教育に盛込むべき事項としては，上記の教育テーマ又は内容の他に下記のことを挙げることができよう。

④ 教育者の選定

㊨ 教育の対象者の検討

この運用時教育の対象者は比較的広範囲に渡り，システム部門の運用担当者をはじめとして，このシステムの入力データを作成する部門の人々，出力諸表を利用する部門の人々で，場合によっては経営者も含まれよう。

又、システムの対象となる業務内容によっては納入業者や得意先の窓口の人々等も考慮する必要がある。

開発時教育と同様に、場合によっては対象者をグループ分けして教育することも考えられる。

㊦ 教育方法及び教材、スケジュール等の決定

(6) 設備計画

設備計画は、コンピュータの使用に先立って、コンピュータ及びその付帯設備の確保、具体的な使用方法や使用予定時間等を決定することが、その目的となる。

又、設備計画は、システムの開発段階に対するものと、運用段階に対するものに分けて考えるべきであろう。

設備計画に盛り込まれるべき事項としては、下記のことを挙げる事ができよう。

① 設備の選定

ソフトウェア開発や運用に必要な設備とその端末機器の数等

② 設備の入手方法

社内設備を使用する場合にはあまり問題にならないが、社外の設備を使用する場合や、新規設備の購入又は借入れる等の場合には、その入手方法、入手先、入手可能時期、保守方法等を明確にする。

③ 設備使用予定時間

総使用時間のみでなく、開発段階においては、1日平均の使用時間、又運用段階においては、日次、月次の使用時間として算出しておくとうよい。

④ 設備の管理体制

社内設備で専門の管理者又は管理部門がある場合には、あまり問題にならないが、そのソフトウェア開発プロジェクトに専用の場合には設備の管理体制を明確にしておく必要がある。



又、他のプロジェクトと共同利用する場合は、予め、調整を行なっておくことも大切である。

#### (7) 開発支援計画

開発支援計画は、ソフトウェア開発の効率化、開発費用の低減、品質の向上等を計るために、各工程において、具体的な開発支援方法を定める。ここで盛り込まれるべき事項としては、下記のことを挙げるべきであろう。

##### ① 支援対象

まず、どのような作業や管理を支援するのか、その対象を明確にする。

たとえば、下記のような事柄が考えられる。

- ① プログラム設計
- ② コーディングの省力化
- ③ デバッグの効率化
- ④ テスト・データの作成及びテストの自動化等
- ⑤ ドキュメント作成の自動化、省力化

##### ② 支援方法

上記、支援対象のそれぞれに対し、具体的な支援方法及び使用する支援ツールとその使用方法等を明確にする。

##### ③ 支援体制及びスケジュール

独立した支援部門がある場合には、その部門と話し合い、支援してもらえる項目やその内容、コミュニケーションの方法等を明確にする。

又、独立した支援部門がない場合には、ソフトウェア開発プロジェクトの中に専任をおく等、体制を明確にし、それぞれに時期を検討して、支援のためのスケジュールを作成する。

##### ④ 支援による期待効果

それぞれの支援方法に対し、どのような効果が得られるかを明らか

にする。

#### (8) 工程管理計画

工程管理計画は、工程計画によって設定されたスケジュール通りに開発作業が進むように、進捗状況の実績をタイムリーに把握し、管理していくことが目的である。この計画に盛り込むべき事項としては、下記のことを挙げることができよう。

##### ① 管理項目

ソフトウェア開発の各工程において、具体的に何を管理するか、その項目を明らかにする。

たとえば、プログラム設計においては、設計済みプログラムの本数、プログラムの作成においては、コーディング済みのプログラム本数やデバッグ済みのプログラム本数、検査においてはテストケースの消化数、等である。

##### ② 実績データの収集方法、時点等

上記の管理項目それぞれについて、いつどのような方法で実績データを収集するかを明らかにする。

##### ③ 計画との比較方法

進捗状況が定量的に把握できるように、計画値と実績との比較方法を明らかにする。

#### (9) 費用管理計画

費用管理計画は、開発作業の進行に伴って発生する費用を適確に把握し、費用の消化が工程計画及び予算計画通りに進行するよう管理すると共に、差が生じた場合には、早めに対策を施こせるようにすることが目的である。

費用管理計画に盛り込むべき事項としては下記のことを挙げるできよう。

##### ① 発生費用項目

管理の対象となる費用項目を明確にする。

費用項目としては下記のことを挙げる事ができよう。

① 人件費

人件費としては、要員の直接人件費、アルバイト費、外注費等がある。

② 設備使用費

ソフトウェア開発に必要なコンピュータやその付帯設備等

③ 材料費

計測システム等ハードウェアの一部を含めた開発の場合には、そのハードウェアの作成に必要なICやLSI、プリント板等の材料費。

④ 雑費

雑費としては、交通・通信費、コピー代等

② 費用データの収集方法

上記各費用項目のそれぞれに対して、実績データの収集方法、収集時点等を明確にする。

たとえば、

① 直接人件費については作業日報により、毎朝9時までに前日分を収集する。

② コンピュータ等の使用時間は、ログ情報を毎週末に収計する。

③ 雑費は、出金伝票の処理時点で収集する。

③ 予算計画との対比方法

工程上に週1回、月1回等のチェックポイントを設定し、予算計画と実績との対比時点、及び対比方法を明確にする。

(10) 品質管理計画

品質管理計画は、ソフトウェア開発作業の各工程において作成される種々の成果物に対して、それぞれある一定の品質レベルに保持していく

ための管理方法，時期等を具体的に定めるものである。

品質管理計画として，次のような事項が考えられる。

① 品質管理の対象とする成果物

システム設計書，プログラム設計書，テスト仕様書，プログラム・リスト等，品質管理の対象成果物を明らかにする。

② 品質管理項目

各対象物に対して何を管理すれば良いか，その項目を明確にする。

たとえば，

① 書式

ドキュメント類については，章，項立ての方法，プログラムについては，コーディングの形式やコメント文の付け方等。

② 盛り込むべき事項とその順序

主に，ドキュメント類については，それぞれ最低限盛り込むべき事項，場合によって盛り込むべき事項等に分けて列挙し，かつ書く順番も規定する。

本項は，どのソフトウェア開発に対しても，言えることなので，予め検討して，ドキュメント規約やコーディング規約として，社内基準を定めておくことが望ましい。

③ 品質管理方法，時期，体制等

上記の各項目について，いつ，誰がどのように管理していくかを明確にし，かつ，その結果をそれぞれの成果物にどのように反映させるかを明確にする。

(1) 検査・検収計画

検査・検収計画では，ソフトウェア開発の各工程における成果物，特に検収成果物としてのソフトウェアの品質を，いつ，誰が，どのような方法で確認するか，又，何をもって合格（検収）とするか等を具体的に定める。

検査・検収計画に盛り込むべき事項としては、下記のものを挙げることでできよう。

① 検査項目

何を検査すれば良いか、その項目を明確にする。

② 検査方法及び時期、期間

上記の各項目について、いつどこで、どのような方法で又、どのくらいの期間をかけて検査するかを明らかにする。

特にプログラムについては、単体テスト、結合テスト、総合テスト等、段階を分けて、それぞれについて検査仕様書としてまとめる。

又、テスト支援ツールを利用する場合には、その利用方法等も明らかにする。

③ 検査結果の確認方法の明確化

特に、プログラムの確認については、どのようなデータを入力すれば、どのような出力が得られるかという、入出力の関係及び、処理時間等の性能の目標値等を明確にしておくことが大切である。

又、ドキュメント類については、品質管理計画で検討した品質管理の項目等をもとにして、確認項目を決め、それぞれについて何か書かれていれば良いかを明らかにしたチェックリストを用意し、そのチェックをしながら、ドキュメントを読む等の方法も考えられる。

④ 検査結果の反映方法

検査の結果、満足できない部分があった場合、それを、開発者に対し、どう伝達し、どうするか、又、再検査の方法等も明確にする。

⑤ 検査結果の管理・保管方法

検査結果を検査報告書にまとめると共に検査結果の資料を整理して管理・保管しておくことは、後日の保守、機能変更、拡張等の検査時に役立てることができる。

## 12) 移行計画

旧システム又は、手作業で行なわれている状態より、今回開発されたシステムの運用に移る過渡期には、運用のために準備しておくべき事項や試運転等により発生する諸問題が多い。

移行計画は、それらをどのように解決するか、予め開発部門、運用部門及びユーザー部門合同で検討し、具体的な方法とスケジュールを定めるもので、移行計画に盛り込まれるべき事項としては、下記のことを挙げることができよう。

### ① 手作業又は旧システムと本システムとの並行処理期間

並行処理期間内に確認しなければならない事項を列挙し、それぞれの確認方法を明確にして、期間を推定する。

又、並行処理打切りの条件を明確にする。

### ② 運用部門及びユーザー部門に対する教育計画（先に述べた運用時教育計画）を作成し、それをスケジュールに盛り込む。

### ③ 伝票や特別の帳表のフォーマットを最終確定し、その印刷の手配をする。

### ④ 旧システムが稼働中の場合には、データの継続性が問題となり、データのコンバージョンの方法、時期、期間、確認方法等を明確にする。

### ⑤ ソフトウェアの開発機種と運用機種が異なる場合には、その違いを比較検討し、プログラム・コンバージョンの方法、期間、確認の方法等を明確にする。

### ⑥ その他、並行処理期間（試運転期間）中に発生した問題を解決する方法について、開発部門と運用部門（及びユーザー部門）との間で話し合い、明確にしておく。

## 13) 運用計画

運用計画は、開発されたソフトウェア・システムを運用部門（及びユーザー部門）が、トラブルの発生もなく、スムーズに運用していくための

スケジュールリング，入力データの収集方法やルート及びその管理，データファイルの管理や保全，出力帳表の配布ルート等の運用方法を予め定めておくものである。

運用計画に盛り込むべき事項としては下記のことを挙げる事ができよう。

① データの収集方法

- ① 入力データとなる伝票の書き方等，起票する部門，担当者等を明確にする。
- ② 伝票等の認承者を明確にする。
- ③ 伝票等の収集ルートを明確にする。
- ④ 締め日を明確にする。
- ⑤ これらを利用者マニュアルとして整備する。

② 運用方法

- ① 各プログラムの標準的な使い方，起動方法，JCL等を明確にする。データファイルの更新方法，及び時点，バックアップを取る時点等を明確にする。
- ② バックアップを取った磁気テープやフロッピーディスク等外部記憶媒体の管理方法等，データの管理・保全方法を明確にする。
- ③ これらを運用マニュアルとして整備する。

③ 出力情報の配布方法等

- ① 配布すべき出力帳表を列挙し，それぞれについて誰が利用者かを明らかにする。
- ② それぞれの出力帳表について配布ルートを確立する。
- ③ 配布時期を明確にする。

これらを整理し，日次，月次，年次等のスケジュール表としてまとめる。

#### ④ 保守・拡張計画

ソフトウェアを運用している間に、そのソフトウェアの環境、具体的には対象となっている業務内容が変化してくる。又、運用になれてくるに従い、新しい要求が出てくるのが常であろう。

このように保守・拡張とは、開発当時の環境と、現在の環境との差を明らかにし、そのソフトウェアを現在の環境に適合させるために、機能の変更、追加を行なうことである。

変更、追加が大規模になれば、開発段階と同様に種々の計画を立てる必要が出てこようが、小規模な場合には、それらのうち必要な事項を統合して保守・拡張計画としてまとめることになる。

保守・拡張計画に盛り込むべき事項としては、下記のものを挙げる事ができよう。

- (1) 変更内容及び変更量等
- (2) 変更に必要な期間、人員、設備等
- (3) 検査・検収の方法等
- (4) 現システムから変更・拡張後のシステムへの移行の方法
- (5) 運用形態の変更内容

#### ⑤ パッケージド・ソフト購入計画

パッケージド・ソフト購入計画は、要求分析の結果をもとにして、その要求に一番近いパッケージド・ソフトの選定から、導入、運用に至る間の諸問題を解決していくためのもので、盛り込まれる事項としては下記のものを挙げる事ができよう。

- (1) 要求分析から、必要な機能や性能を洗い出し、その確認方法、期間等を明らかにする。
- (2) 不足する機能の追加方法やそれに必要な期間、費用等を明らかにする。
- (3) 他の既存システムとのインターフェイスの取り方及び、それに必要



な期間，費用等を明らかにする。

(4) 試行期間を明らかにする。

(5) 運用時教育計画を立て，その期間も含めてスケジュールを作成する。

#### 4.1.2 開発計画相互の関連

(1) ソフトウェア・ライフサイクルにおける各計画の位置付け

ソフトウェアの開発から運用，保守に至るライフサイクル上では，先に述べた種々の計画が立てられ，それぞれが実施に移されていくわけであるが，各計画がいつ立てられ，いつ実施に移されていくかは重要な問題であり，そこで，各計画の位置づけの例を示すと図4・1の通りである。

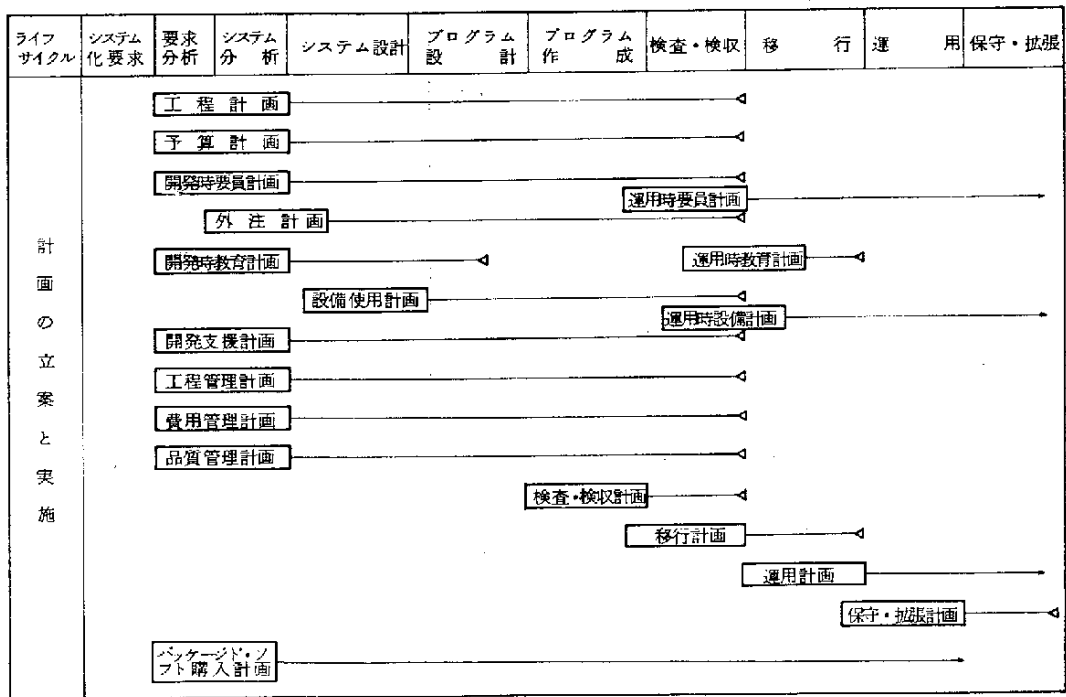


図4・1 ソフトウェア・ライフ・サイクルにおける各計画の位置付け

## (2) 開発計画相互の関連

以上、述べてきた種々の開発計画の分類は、これが唯一というものでもなければ、このようにしなければならないと言うものでもない。企業や組織の習慣、規定等によって、それぞれの計画の呼び名や内容等には、少なからず相違があろう。ここに示したのは、その一例、一方法にすぎないが、ここでは、以上述べてきた各計画の分類をもとにして、それらの相互関係について述べる。

### ① 開発段階に対する計画の相互関連

開発段階に対する各種の計画をその性格から大別すると次の3つになろう。

#### ① 各開発工程における見積、予定、スケジューリング等に関する計画

開発作業の基本になる、対象システムの規模の見積から、各工程に必要な工数の見積り、及び、要員、外注の配置、それらに対する教育、又、使用設備の見積、手配、工程への配分等を行なうもので、下記の諸計画が含まれる。

- ① 工程計画
- ② 予算計画
- ③ 要員計画
- ④ 外注計画
- ⑤ 設備使用計画

ここで中心になるのは工程計画であり、その各工程において必要とされる要員及び設備をより具体的にしていけるのが、要員計画、外注計画及び設備使用計画であり、金の面から見たものが予算計画である。

#### ② 開発作業の実施における管理及び支援等に関する計画

上記で作成された計画に基づいて、開発作業をどう実施していく

かといったことを具体化するための計画で、下記のものが含まれる。

- ① 開発支援計画
  - ② 工程管理計画
  - ③ 費用管理計画
  - ④ 品質管理計画
  - ⑤ 検査・検収計画
- ⑥ 移行、運用、保守・拡張等に関する計画

開発段階から運用段階へ移る移行作業も含めて、開発されたシステムを良好な状態と環境の下で、活用していくために作成されるもので、下記の計画が含まれる。

- ① 移行計画
- ② 運用計画
- ③ 運用段階における要員計画
- ④ 運用時教育計画
- ⑤ 運用時設備計画
- ⑥ 保守・拡張計画

#### 4.1.3 開発計画の変更

計画をキチンと立て、それに基づいて開発作業を実施しても、計画と実績との間にはある程度の差が生じてくることは避けられない問題であろう。

そこで、ここでは、計画の変更になるような原因をアンケート調査から抽出、整理し述べる。

開発計画の変更原因を類別すると、下記のようなになる。

##### (1) ユーザー要求の不備、不足に起因するもの

エンドユーザーがシステム化する対象業務を十分に把握していなかったり、又、分かっていてもユーザーニーズの表現能力に欠けていたり、

あいまいであったり，業務の変化を考えていなかったり，運用時の人手作業量の予測があまかったり等々の理由により，開発途中で，要求変更がたび重なり，計画を変更しなければならなくなったケースである。

この原因は比較的多く，極端な場合には開発されたソフトウェア・システムが使われない事例もあった。

(2) 要求分析，システム分析の不完全さに起因するもの。

ユーザ要求に対する分析力の不足や，導入を急いで十分に分析しないまま，開発作業に入ったため，変更が多発したり，システム規模が大きくなって計画を立て直すといったケースである。

(3) 開発部門とユーザー部門又はユーザー部門間の関係に起因するもの。

ユーザー部門の協力が得られなかった。開発部門主導型の開発になり，ユーザーや運用担当者の意見が反映されなかった。等，開発部門とユーザー部門の関係のまずさから，計画変更に追い込まれたケース。又，複数のユーザー間にまたがるソフトウェア・システムを開発する場合に，ユーザー間のコンセンサスを得ないまま，又は，利害が対立したまま開発に入り，仕様変更が多くなり，計画の立て直しになったケース。

(4) システム設計時の調査不足に起因するもの。

新しいコンピュータであったのに，そのハードウェアやOS，言語等について調査せず，又，データベース・システムをはじめて使うことになったのに十分学習もせずに開発したために，特にプログラム作成段階になって，トラブルが多発し，工数が増大，工期が遅延し，工程計画や要員計画を変更したケース。

(5) 計画時の見積の不備に起因するもの

計画時に工数を過少に見積ったため，又，納期等の関係で無理な工程計画を立てたため，途中で遅れが目立ち，計画を変更したケース。

(6) 実施管理の不備に起因するもの

実施の過程で、その工程が完全に終わらないうちに次の工程へ進んでしまったために、手戻りが多くなり、計画を変更したケース。

(7) 要員不足に起因するもの

途中で必要な要員が確保できなくなったり、作業中の要員がぬけてしまったりして、計画を変更したケース。

(8) 要員の技術不足に起因するもの

社内の要員、外注先のソフトウェア要員、又は協力を求めたメーカーのSE等の技術力が不足していて、工程が大幅に遅れ、計画変更を余儀無くされたケース。

(9) 設計ミスに起因するもの

システム設計等のミスがプログラム作成時に発見され、それが大きな問題となってシステム設計をやり直すように計画を変更しなければならなくなったケース。

#### 4.2 技法・手法・ツール

システム開発要求・分析作業段階に続く作業としては、4.1節にて述べているように各種の開発計画をたてることになる。

これらの開発計画立案作業に当っては、

- (1) 長期計画がなく、目先の計画になっていないか。
- (2) 一部の部門が不必要に先走りしていないか。
- (3) 他のシステム開発計画との調整は、
- (4) 開発体制、受入れ体制の整備計画は、
- (5) 問題点は明確化されたか、
- (6) システム開発作業のガイドラインとなる計画案になっているか、
- (7) その他

などといった点が、的確にされていなければならず、これから紹介する、

技法、手法、そしてツールを大いに活用することの効果は大きいと云える。

#### 4.2.1 技法・手法の種類と概要

技法・手法としては、これから紹介するもの以外にも、独自に開発したり、長年の経験によって最適化した方法を使用しているケースも数多くあると思われる。

全てを紹介するわけにもいかないので代表的なものについて概要を説明する。

##### (1) WBS (Work Breakdown Structure : 作業区分構成)

WBSとは目標を達成するうえで必要となる作業を細分化し、かつ階層構造にして表現したものである。

WBSのねらいは、必要な作業をもれなく洗い出すことである。従って、システム開発計画・立案作業にあたっては、WBSを利用することは大変効果的である。

作業もれのないWBSを作成するために注意したいことを挙げてみる。

- ① どの段階まで詳細化するのかを明確に定義しておく。
  - ② あらかじめ、対象となる全体作業の特性、問題点および制約事項などをは握すること。
  - ③ 記述の仕方は判りやすい表現を心掛け、統一した形式にすること。
- 各種の計画・立案段階だけでなく、他の作業段階でも同様に活用できる一般的な技法・手法である。

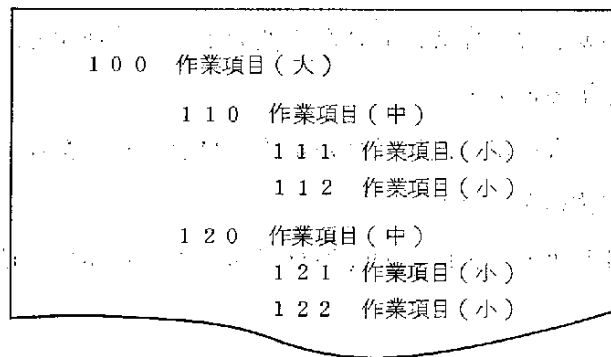
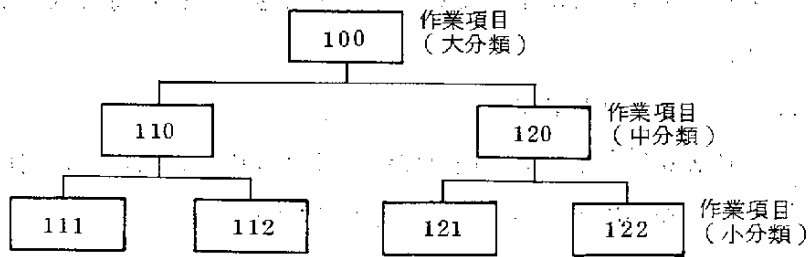


図 4・2 WBS の証述例

(2) PERT (Program Evaluation and Review Technique)

PERT, CPM, PERT/COSTについては、ツールにて述べるのでここでは、その中心となっているアローダイアグラム (矢線図, PERTネットワーク, PERT図などと称している。) について述べる。

アンケート調査でも、PERTの知名度、使用経験数を群を抜いている。その原因がアローダイアグラムにあったと云っても、けっして過言ではない。これによって、正確な工程管理が可能となり、また、作業の手順や相互の関係を明確に表現することができるわけである。

クリティカルパスの識別や、所用工数の把握によって、不測のトラ

ブルや工程遅れが生じても十分対応できる技術が得られたことになった。

### (3) ハンディPERT

PERTを誰でもが簡単に使えるようにしたものが、協同乳業株式会社川寧氏提唱によるハンディPERTである。

この方式は、基本手順を5ステップに分け、最終ステップでネットワークが完成するしくみになっている。

#### ① 作業の分析・単純化

必要となる作業をもれなく洗い出して整理する。

#### ② 作業名のカード化

①で洗い出した作業名の単位でHPカードを作る。

#### ③ 作業の順序づけ

HPカードを作業の性格を考慮しながら作業手順にそって配置する。

#### ④ 作業の関連づけ

各作業間での関係づけを行う。

#### ⑤ 作業フローへの確定

④にて関係づけられた作業を矢線でむすぶとネットワークが出来上がる。

### (4) ガント・チャート

ガント・チャートとは、バー・チャートとも呼ばれ広く用いられている図の描き方である。

作成するのが簡単であり、見やすく、わかりやすい点が特徴である。

プロジェクトをワークパッケージに分割し、その時間的計画を表現したり、プロジェクトの状況を簡単に説明するのに使用できそうである。



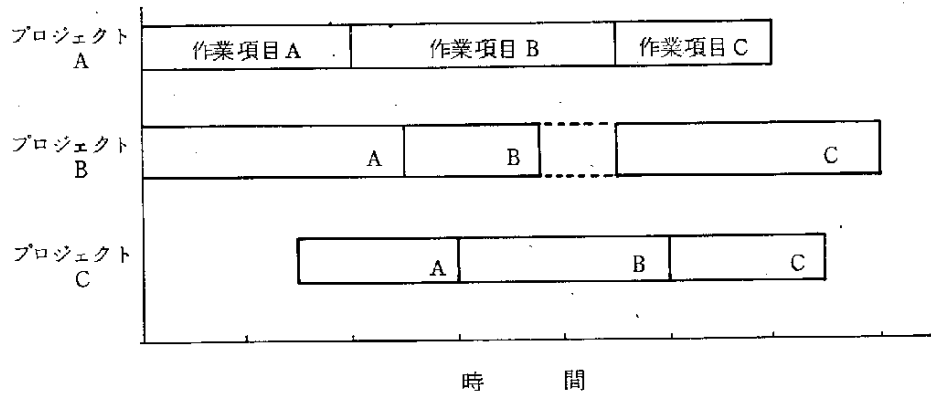


図 4・3 ガント・チャートの記述例

(5) マイルストーン・チャート

ガント・チャートでは表現しにくかった点を補うことができる特徴を持っている。それは、各作業が工程上重要となるポイントや、意志決定すべきタイミングが明記されているところである。

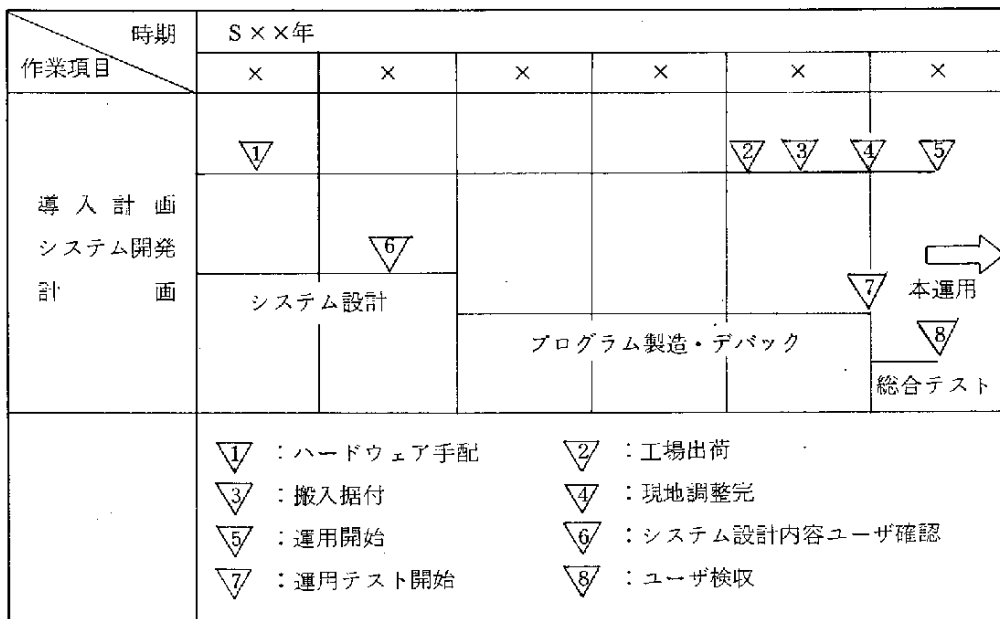


図 4・4 マイルストーン・チャートの記述例

#### (6) トレンド・チャート

予算／実績費用を縦軸に、日程を横軸にとる。そして、計画線を引き、実績値を重ね、順次伸ばしていくことによって、費用の進み具合いと、工程の進み具合いが合わせて管理できる。

### 4.2.2 ツールの種類とその概要

ツールについては、提供者、適用範囲、対象システムの種類・規模などの違いから、数多くのツールが存在している。

利用する立場からでも、使い易いように変更したり、一部分だけ利用したりしていると思われる。

それらのなかから、以下に挙げたツールについて紹介する。

#### (1) PERT

PERTは、1958年にアメリカにおいて開発された、大規模なプロジェクトの計画・管理のためのツールであった。そして、そのプロジェクトとは、ポラリス・ミサイルの開発計画であることから、コストに対する考慮が入っていないため一般の開発計画にそのまま通用するわけにはいかなかった。そこで、資金の最も効率よい活用も考えられるプロジェクト管理方法が考えだされた。これがPERT/COSTと呼ばれているものである。

これに対比させるため、従来のPERTをPERT/TIMEと呼んでいる。

PERTの応用分野は、その開発経緯から、研究・開発のプロジェクト管理が代表的であったが、土木、建築工事の日程管理をはじめとして、現在ではあらゆる方面で利用されている。

#### (2) CPM (Critical Path Method)

米国デュボン社がレミントン・ランド社と共に、1957～8年に開発したものである。

PERTが国家的プロジェクトに向けて開発されたことから、コスト・コントロールにかなり融通性をもたしているのに対して、CPMはコストはコントロールし、スケジュールに融通性を持たすことに重点が置かれている。例えば、最小の費用増加で工期がどれだけ短縮できるかを計算するのに利用できる。

### (3) PERT/COST

1962年頃に、アメリカ国防省とNASAが共同で開発したもので、PERTの時間的要素とともに、費用に関する情報をネットワークに与えて、時間と費用の両面からプロジェクトの開発計画・立案および、その遂行を管理するために利用する。

### (4) RAMP S (Resource Allocation and Multi-Project Scheduling)

人員・資材の配分も伴った日程計画を、しかも1プロジェクトだけでなく競合する複数のプロジェクト間に対しても実施することができる。マルチ・プロジェクト管理用のツールである。

作業の順序関係の表現はPERTのときと同じである。

しかし、各作業での所要時間の把握が、PERTの場合のように一定値ではなく、人員・資材の量によって変われるようになっている。つまり、仕事量(工数)として表現している。

### (5) P P B S (Planning Programming Budgeting System)

アメリカのランド・コーポレーションが最初に開発したといわれており、1961年に米国国防省において、1965年には、連邦政府の各部局でこのシステムを採用した。

また、わが国においても、大蔵省を中心に各省で研究を行った経緯もあった。

このシステムは、目標を設定して、それを達成するための具体的な案をいくつかつくり(Planning)、それらが必要とする資源や効果を

規定し (Programming), もっとも効果的なものを選択するため予算化してみる (Budgeting) 方式になっている。

(6) STEPS (Standardized Technology and Engineering for Programming Support)

STEPSは、日本電気㈱が開発した、体系系化されたシステム開発標準化ツールである。

システム開発および、開発されたシステムの運用・保守を安易にかつ効率よくすることにより、システムライフサイクルを通してトータルコストミニマムを実現するため、

- ① システム開発の作業手順の標準化
- ② 開発作業結果として作成されるワークドキュメントの標準化
- ③ 開発作業で出力されるドキュメントの標準化
- ④ プログラム開発時の仕様作成の標準化
- ⑤ プログラム開発時のコーディン作成の標準化

を目的としている。

(7) HIPACE-SPDS (High Productive Application Creation and Engineering-Standard Procedure to Develop System)

HIPACE-SPDSは日立製作所が開発した、システムの計画、設計・開発作業およびこれらの管理に関する標準手順である。

- ① システム開発の工程に沿った手順
- ② プロジェクト管理手順
- ③ システム管理手順

から構成されている。

(8) SDEM (Software Development Engineering Methodology)

SDEMは、富士通が開発したソフトウェアの開発及びプロジェクト管理のための開発作業標準である。

- ① システム環境（運用，移行，ハードウェア，設備障害対策）
- ② システム開発（入出力，業務処理，ファイル，テスト）
- ③ 管理（技術管理，教育，訓練，プロジェクト管理）

の3つ局面にて実施すべき作業の標準を設定している。

(9) P R I D E (Profitable Information by Design through  
Phased Planning and Control)

P R I D Eは，アメリカのMBA社が開発し，日本システムックスが販売，サポートしている情報システムの方法論である。

- ① システム構造化設計技法
- ② データ管理技法
- ③ プロジェクト管理技法
- ④ ドキュメンテーション技法
- ⑤ コミュニケーション技法

といったプロジェクト管理の技法をとり込んでいる。

(10) P M P (Project Management Program)

P M Pは，協同システム開発㈱の委託を受け，メルコム・オキタックシステムズが開発したソフトウェア開発プロジェクト向けの工程管理支援ツールである。

- ① 基本処理（全体管理，ネットワーク，時間分析，ファイル管理）
- ② 資源割当処理（人員，コンピュータ費用）
- ③ ネットワーク準備処理
- ④ 費用評価処理

の4つの機能を有している。

(11) P C M P (Project Cost Management Program)

P C M Pは，情報処理振興事業協会の委託を受け，日本電子開発株式会社が開発した原価管理を中心とするプロジェクトの計画・管理のツールである。

プロジェクトの原価管理を原価計算機能，見積り支援機能，要員管理支援機能としてとらえ，これをプロジェクト群管理，階層管理，多次元管理の考え方から総合的に管理している。

#### 4.2.3 利用の実態と効果

技法・手法およびツールについて，紹介したが，これらのものが実際のシステム開発計画立案部門において，どれだけ利用されているかをアンケート調査などから明らかにし，内容について分析してみた。

##### (1) 技法・手法

アンケート結果から特徴的なことを挙げてみると，2.4(3)項，図2・7でも述べているように，知名度や使用経験が以外と低いことである。

PERTやガント・チャートは，良く知られており，利用度も高い結果が出ているが，他は予想外に低い知名度と使用度であった。

このことは，技法・手法といったものが，文献などで，名称や概要を知っていたとしても，直ちに活用できたり，試行的に使ってみることに結びつかないとも云えるのではないだろうか。

一方，PERTやガント・チャートが高い値を示していたことは，これらが持っている表記方式が有名かつ秀れているためではないだろうか。計画段階での作業過程や作業の結果を表現するためにだけ利用されているケースも含まれていると思われる。

やはり，使ってみれば，例えそれが部分的，局所的であったとしても，それなりの効果が期待できるのである。

知る機会，使う機会がなく，また必要性も感じないため，技法・手法の知名度や利用度が低いのであれば，計画立案担当者の意識や努力が低いか，もしくは，まだ効果的な技法・手法が現われていないのであろうか。

アンケート調査のなかで、システム開発での失敗の原因をたずねたところ（2.5.2項）、回答で一番多かったのが、要求分析段階での不都合であった。また、作業工程の無理や遅れが原因していたという反省も何件かあった。

要求分析や開発計画段階でのミスや問題の放置は、システム開発に決定的な影響を及ぼすことがある。技法・手法の活用が、これらの失敗を起こさないための助けとなるはずであり、まず試行的にでも使ってみることを推めたい。

## (2) ツール

ツールについてのアンケート調査の内容は、使用経験についてたずねている。

この結果は、4.2(4)項、図2・8にて報告しているが、51%という半数以上のところで何らかのツールを使っているとのことであった。そのなかでも、特に高い使用度を示していたのが、PERTであった。

PERTについては、技法・手法でも同様であったが、高い知名度と使用度が特徴であった。

また、他のツールでは、メーカーが提供しているものが比較的高い使用度を示していた。これは、導入しているコンピュータのメーカーとの関連が強くなってしまふからであろう。

各ユーザ、メーカーとしては、年々のソフトウェア関連費用の増大化に対処すべく、ソフトウェア開発の全工程に渡って、標準化や自動化、また品質の向上のためのツールを開発したり、規定を設けることに大変力を入れてきている。

コンピュータ・システムも、ハードウェアの性能や、提供ソフトウェアの内容を競う段階から、利用者側のソフトウェア開発・保守のための支援ツールや、ノウハウに関しても重要視されてきている。

個々のツールについては、入力や出力仕様に規約があったり、効果についても定量的に出すことが難かしく、自から進んで利用しづらい面がある。

最初から無理してでも具体的効果を出そうとは思わず、一部分（例えば、帳票形式の利用、作業工程の分け方、計画／実績データの収集など）だけでも試行すべきである。

部分的な利用でも、相応の効果があると思われるからであり、さらに一歩進んだ使い方への布石になるからである。

#### 4.3 見積方法

##### 4.3.1 見積方法の概要

ソフトウェア開発計画立案のもとになるものが見積作業である。その見積の確かさによって実施可能な計画になるか否か、ひいては、最終成果物であるソフトウェア・システムの品質にも影響をおよぼすので重要な作業であるが、なかなか正確に見積れないのが現実である。

ここでは、見積方法の概要として、見積の対象となる事項を挙げ、それぞれについてどのような見積方法があるか、及び見積の手順等を述べる。

##### (1) 開発段階における見積方法

###### ① 開発規模

前章で行なわれた要求定義、分析及びシステム分析の結果をもとにして、開発するソフトウェア・システムの開発規模を推定する。

数値化する単位としては次のものがある。

- プログラム・ステップ数
- プログラム本数

開発規模の見積方法としては、下記のもの挙げられよう。

###### ① 機能分割により積上げる方法



この見積方法は、開発すべき、ソフトウェア全体を機能別に分割し、分割した単位ごとに見積るものである。分割する単位は、サブシステム、プログラム、モジュールと詳細にすればする程正確さは増すが、システム設計の一部を行なうことと同じになり、時間がかかる。

又、機能を分割する際、処理フローや関連図等を作成するとより見積りし易くなる。

次に、機能分割されたものに対して、例に示すような処理パターンごとの基準値から見積るのも一つの方法である。

処理パターン		基準値	ボリュームの算出
オンライン	エントリ	20/人日	項目数 × 20ステップ
	会話	40/人日	項目数 × 40ステップ
	照会	10/人日	項目数 × 10ステップ
	⋮		
バッチ	帳票	10/人日	項目数 × 10ステップ
	チェック	20/人日	項目数 × 20ステップ
	転送	15/人日	項目数 × 15ステップ
	集計	15/人日	項目数 × 15ステップ
	⋮		

(出典 (社)ソフトウェア産業振興協会  
58年度技術研修資料「プロジェクト管理」より)

プロジェクト完了時には、この見積りと実績を比較し、処理パターンと基準値の設定を見直す必要がある。

#### ㊤ 既存の類似システムから推測する方法

この方法は、完了したプロジェクトの実績ステップ数などを技術分野、業種、業務内容などにより分類して蓄積しておき、その蓄積データを参考にして推測し、これに安全率を掛けて見積ものである。



れる機能の数等を勘案して、工程毎の開発期間を計算し、それを合計して工期を算出する。

希望納期が決まっている場合には、この時点で、両者の調整を計る。

#### ④ 設備使用時間

① プログラムのデバッグ工程及びテスト工程に対して、生産性の基準を設定する。

単位としては、次のものがある。

ステップ数／人月

プログラム本数／人月

生産性の基準値は、既存の完了プロジェクトの実績値から、開発するソフトウェアの種類、規模毎に標準的な値を求めておき、今回開発するソフトウェア・システムの難しさや、複雑さ等を勘案して設定する。

② 開発規模と上記の生産性の基準より、設備使用時間を算出する。

#### ⑤ 費用

最後に、上記各項の結果から総費用を見積る。

見積る費用項目としては下記のものを挙げる事ができよう。

○ 人件費

直接人件費

外注費

アルバイト費

○ 設備使用費

○ その他直接費

パンチ代（プログラム，テストデータ等）

コピー代（マニュアル，各種仕様書）

印刷費

交通，通信費

：  
。管理費

④ 人件費

上記②で求めた工程毎の開発工数のうち，社内の要員で行なう部分，外注に出す部分及び，アルバイトで済む部分等に分け，社内の要員で行なう部分については，人件費ランク別に工数を分けて，それぞれにランク単価を掛けて集計し，直接人件費を算出する。

外注に出す部分は，外注に出す工数に標準外注費単価を掛けて外注費を算出する。

アルバイトの部分も，工数にアルバイト費の標準単価を掛けてアルバイト費を算出する。

これらを全て集計して人件費を算出する。

⑤ 設備使用費

設備使用費は，上記④で求めた設備使用時間に単価を掛けて算出する。

使用する設備が数種類ある場合には，それぞれに費用を算出したのち，集計する。

⑥ パンチ代

パンチ代のうち，プログラムの分は，上記①で求めた開発規模に一定の比率を掛けて数量を算出し，それに単価を掛けてパンチ代を計算する。

一定の比率とは，単体デバッグ用のダミーメイン・プログラム等，使い捨てプログラムや，JCL等を見込んだものである。

ただし，開発規模をプログラム本数で見積っている場合には，平均ステップ数を想定した上で，ステップ数に変換する。

テストデータの分は，単体デバック用のテストデータと総合テ

スト用のデータに分け、単体デバック用のデータ数は、過去の実績から例えば、20ステップに1データという基準値から算出する。

総合テスト用のデータは、その対象になる業務内容に依存するので、業務内容から見積る必要がある。

最後にこれらを集計して総パンチ代を算出する。

㊦ コピー代

作成するドキュメント類のコピー費用を見積るのが目的で、これももともとなるのは開発規模である。何ステップに1ページ書くかという基準を、各ドキュメント毎に過去の実績値から算出しておき、各ドキュメントについて、開発規模とこの基準から作成ページ数を計算して、それに配布必要部数及び単価を掛けてコピー費用を算出する。

㊧ 印刷費

ユーザーズマニュアル等を印刷する場合には、上記で算出したページ数を業者に提供して見積書を取っておくと良い。

㊨ 交通・通信費

ユーザーや、開発用コンピュータが遠方にある場合には、大雑把にでも、往復する回数や通信回数を見積って費用を算出しておくと良い。

又、その他の費用といっしょにして、過去の実績から、例えば人件費に一定の比率を掛けて算出することも簡便な方法である。

㊩ 管理費

上記を全て合計して直接費を求め、それに、過去の実績から求めた管理費比率をかけて、管理費を算出する。

以上を全て集計して総費用を算出する。

## (2) 運用段階における見積方法

運用段階に対する見積には下記のことを挙げることでよい。

### ① 入力データの作成費

日々の入力データ量を推定し、その量から、人件費及び、カードパンチをする場合にはパンチ代を計算する。

対象になる業務内容により、時間が経過するにつれて、データ量が増加するような場合には、その増加率を想定し、半年後、1年後、2年後等の費用を算出しておくことも重要である。

### ② 出力資料の作成費

日次、月次、年次それぞれについて出力資料を列挙し、その出力ページ数を推定して、それぞれの配布先数を掛け、コピー費用を算出する。

報告書として、製本する場合などは、その費用も見込んでおく。

これも、1年後、2年後の費用を計算しておくが良い。

### ③ 設備使用費及び運用担当者の人件費

運用計画より日次、月次、年次の処理量を推定して、それぞれのコンピュータ時間を計算し、単価を掛けて設備使用費を算出する。

次に処理量から、運用担当者の必要作業時間又は日数を推定し、人件費単価を掛けて運用担当者の人件費を算出する。

### ④ その他消耗品費

入力になる伝票やプリンターシート等の費用を算出する。

印刷が必要な場合には、業者からの見積書を取って、一枚単価に直したのち、消費量を掛けて費用を算出する。

以上を合計したものが運用段階における費用になるが、期間の単位を明確にしておくことが大切で、月間の費用、年間の費用を求め、かつ、数年先までの費用を算出しておくが良い。

#### 4.3.2 見積基準

見積における基準値には多々あるが、ほとんど過去の実績値に依存することが多いので、完了した開発プロジェクトの実績値をたんねんに蓄積することが大切である。

以下、主な見積基準値の入手方法、条件等について述べる。

##### (1) 開発規模を機能分割により見積る場合の基準値

要求分析，システム分析の結果を元にサブシステム，プログラム，モジュール等に分割して，それぞれを見積る場合の基準値を求めるには，まず，過去の完了プロジェクトを事務処理，科学技術計算，プロセス制御関係等に分類する。

- ① 事務処理については，次にオンライン処理とバッチ処理とに分け，それぞれについて，下記の例のような構成要素に分解して，ステップ数（定義文，実行文，コメント文），ファイルの数，扱っているデータ項目数等，相関関係のありそうなデータを抽出する。

事務処理における構成要素の例

- データ エントリー
- データ チェック
- 照会
- 帳表出力
- ファイル更新

次に，上記の構成要素毎に多くの事例のデータを収集し，その平均値や分散，相関関係等を蓄積する。

個々のデータには，処理の概要等をつけておけば，類似のデータを見つけ易くなる。

- ② 技術計算及び，プロセス制御関係も手順は同様であるが，事務処理のようにパターン化が容易ではないので，構成要素の分類や抽出するデータの種類等は工夫が必要であろう。

## (2) 開発工数算出のための生産性基準値

これも、過去の完了プロジェクトの実績値から求める。

まず、事務処理、科学技術計算、プロセス制御関係等に大別し、次に、総開発ステップ数を工程毎の実工数で割って生産性を算出し、総ステップ数、対象業務の概要、工期、開発言語・コンピューター、担当者の氏名、経験年数等を付して、蓄積すると良い。

蓄積された結果から、各工程の平均値、分散、開発規模との相関、工程間の相関等も付随した情報毎に算出して見積りに利用する。

## (3) ドキュメントのページ数

前記と同様、事務処理、科学技術計算、プロセス制御関係等に大別し、各ドキュメント毎に、開発総ステップ数をドキュメントのページ数で割って算出し、種々の付随情報を付けて蓄積する。

特に、それぞれのドキュメントの質を評価してランク付けしておく  
と良い。

### 4.3.3 見積の実態

アンケート調査結果等から、見積の実状について、いくつかの側面より述べる。

#### (1) 工数等の見積基準値を持っているユーザー

2.4(5)の表2.9で示したように、開発計画の立案時に工数等の見積基準を持っているか否かを問うたところ、持っている66件(57.9%)、持っていない48件(42.1%)ということであった。

この結果を見る限り、見積基準値を整備しているユーザーの方が多  
いものの、今だに見積基準値もないまま、勘にたよっているというユ  
ーザーが40%以上もあることがわかる。



(2) 複数の見積基準値を持っているユーザー

2.4(5)の表2.10で示したように開発するソフトウェアの種類によって見積基準値を変えているか、否かについては変えている44件(66.7%)、変えていない22件(33.3%)ということであった。

この結果を見る限り、 $\frac{2}{3}$ は変えており、比較的多くのユーザーが見積基準値の整備を行なっているように思われる。しかし、回答者全体(114件)の比率を見ると $\frac{44}{114} \times 100 = 38\%$ とまだまだ少数派であることがわかる。

(3) 見積の際、最も重要視している事項

工数等の見積の際最も重要視している事項については図4.5に示す結果となっている。

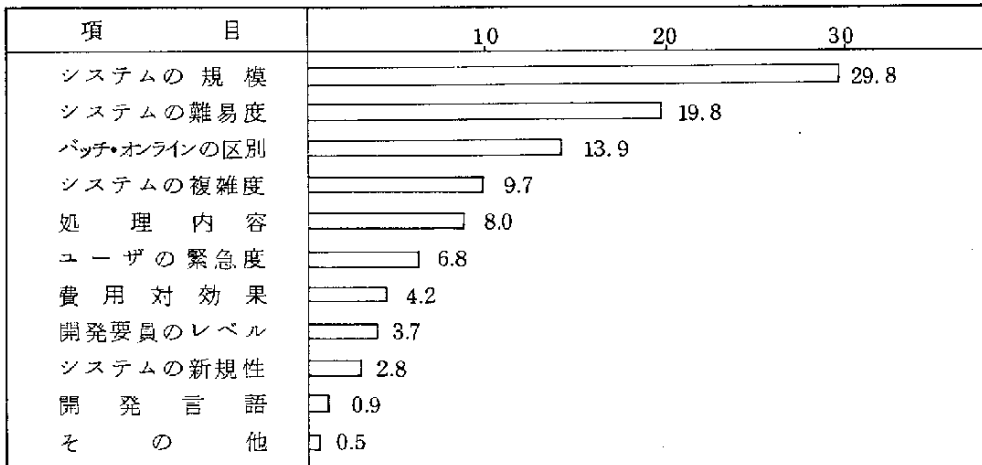


図4・5 見積の際最も重要視している事項

この結果を見る限りにおいては、システムの規模と答えたものが一番多い。システムの規模は、見積の基本になるところであり、この誤差が、後々の開発計画の作成や実施に大きく影響してくることを考えると、妥当な結果と考えられる。

又、次にシステムの難易度、バッチ/オンラインの別、システムの複雑度、処理内容と続いているが、見積時点での不確実性を如実に物語るものであろうことがうかがわれて興味深い。

一方、開発言語などは、一番問題のない部類に属し、今では開発経験を積み、色々な問題が解決された状態にあることがうかがわれる。

(4) 工程別、工数、及び期間配分

ここでは、アンケート結果に基づいてソフトウェアを、事務処理と科学技術計算およびプロセス制御関係に分け、それぞれについて、各工程毎の工数比および期間比の平均値をグラフ化したものである。

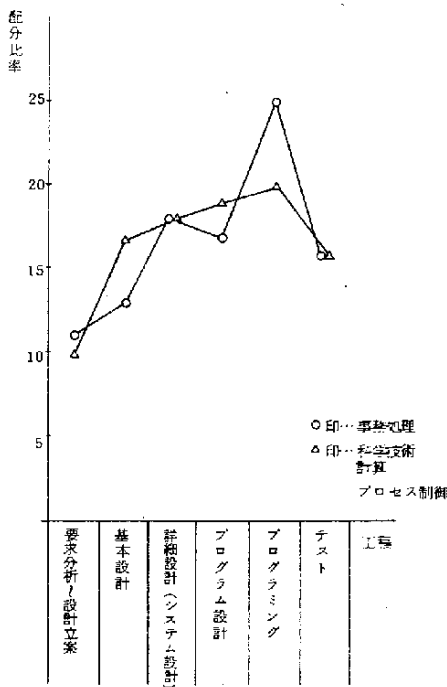


図4・6 工数に関する配分

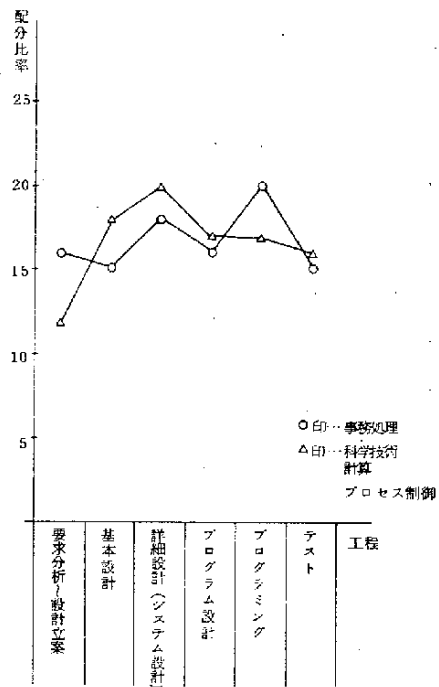


図4・7 期間に関する配分

これらの図を見る限りでは、事務処理は、開発工程の後半に工数も期間もかかっていることがうかがえ、一方、科学技術計算およびプロセス制御関係は、前半の設計段階に重点がおかれていることがわかる。

#### 4.4 開発計画の作成から承認まで

開発計画を立案し、計画書としてまとめて発表する迄の手順、その内容を審議し承認する機関、そして方針が決った後に計画を実施に移すための体制といったことが、企業や官公庁などで具体的にどのように行われているかについて述べる。

実際に、開発計画を作成し承認されるまでの作業体制は、チーム編成して大がかりに行う場合から個人レベルで全ての計画を立てるケースまで、対象システムや組織によって異なり、最適イメージを提示することが大変難かしい。

ここでは、よりわかりやすくするため具体的な内容によって説明する。

##### 4.4.1 手続き

システムの開発作業における初期段階の計画立案から、計画案が承認され実行段階に入るまでの手順について述べる。

##### (1) 初期検討段階（企画グループ等）

初期段階の開発計画は、企画グループ（担当グループのものが行うこともある。）から、下記項目を整理し、まとめた結果を提案として出される。

- ① ニーズの確認
- ② ユーザ窓口の確認
- ③ システム化要求の概要
- ④ 必要とする各種資源
- ⑤ メリット

⑥ おおまかな線表（時期）

⑦ 担当グループの選定

この内容をコンピュータ部門内の審議会にはかる。その結果に基づきコンピュータ部門の長により承認されると、選定された担当グループでは基本構想の構築に入る。

(2) 基本検討段階（担当グループ等）

基本構想または、基本計画概要の作成を行う。

内容的には、可能な限り前段階としての検討項目の詰めを行う。特に下記項目については重点的に行う。

① 必要とする各種資源

② メリット

③ 時期

この結果を再び審議会で検討する。そして、審議に基づきコンピュータ部門の長は、この計画をプロジェクトとして実施することの決定をこの段階で下す。

(3) 基本計画段階（担当グループ等）

あらかじめ定められている様式にそった基本計画書を作成する。

その主な項目を挙げる。

① システム開発に当たっての経緯

② システム化の対象

③ 他システムとの関連

④ システム概要（処理の概要，システム構成等）

⑤ 運用（運用手順，リソース，運用費用等）

⑥ 開発スケジュールと費用計算

出来上がった基本計画書に対する審議を行い、システムの内容と必要リソースの最終決定が下されると実行段階に入る。

(4) 実行段階（担当グループ，コンピュータ室等）

基本計画が承認されたことにより、プロジェクトの編成を、担当グループとコンピュータ室とで行い、システム設計から本番運用へと本格的な開発作業に入っていく。

(5) まとめ

本事例の場合では、コンピュータ部門で各作業段階のメンバを構成しており、ユーザ部門が参加することはあまりないようである。理由は、経験・知識の点でギャップがあること。また、連絡を密にすればメンバでなくても不便はないとのことである。ただし、複数のユーザ部門にまたがる場合にはやはり、調整等に時間がかかっているようである。

計画の承認もコンピュータ部門の各セクションが集まった部会での審議で評価され、コンピュータ部門の長が行っており、この組織が体制・スキルとも十分力を持っているからと思われる。

特に、最近の傾向として、新規開発計画は少なく、メンテナンス（機能追加、改造等）的な計画が多くなっている点は注目したい。

また、基本計画段階で計算された数値（各種の数量と費用）とその実績値を決められた様式で記録していることは大変重要なことである。

計画値と実績値の対比分析と蓄積により、各種作業段階の評価や、別の開発計画への活用がはかれるからである。

#### 4.4.2 計画作成者と承認者の関係

コンピュータ部門内に企画から分析設計・製造そして運用までできる部門が存在している場合は強力な主導性を発揮することができ、また開発作成もスムーズに編成されているようである。

一般的には、プロジェクトが混成チームで編成されることや、承認者が不明確であったりすることが多くある。しかも、顧客やエンドユーザといった利用者部門との関係でも、システムの内容とは直接関係がなさ

そうにみえる作業や機能を要求してきたときに、適確に対応できる業務知識や豊富な経験が不足しがちである。

好ましい作業環境作りといったことがらについても、計画作成段階で考慮しておきたい点である。

アンケート調査では、開発計画を承認する人は誰であるかを問い、その開発要求がどこから出されたのかを対応させて調査してみた。結果は、2.4(2)項、図2-6にて述べている。

開発計画立案の承認者としては、コンピュータ部門の長が一番多く、次いでトップ、ユーザ部門の長と続いているが、コンピュータ部門の長による承認のほうがユーザ部門の長による承認よりも多いということは、強力なコンピュータ部門を持っているところが多いか、もしくは、開発作業の中心は、やはり、コンピュータ部門が責任をもって行うべきであるとの考えが強く存在しているためと思われる。

システム化の要求はどこの部門から出されることが多いかを調査した結果が、2.2(1)項、表2-3にある。結果としては、ユーザ部門からの要求が93.9%を占め圧倒的であった。その要求に対する開発計画のほうの承認者がコンピュータ部門の長であった場合、結論に対して、ユーザ部門からのクレームが出ることも考えられるわけで、開発計画立案作業の責任者、担当者は、高い業務知識と、方針を持ち、説得力のある計画を作る必要がある。

#### 4.5 外注の活用

最近のOAブームに代表されるように、コンピュータの目覚ましい普及は、一方でコンピュータ部門の強化の必要性が高まり、コンピュータ要員の不足に拍車をかけている。

しかし、限られた予算や要員の高齢化、さらにはスキルの不足などが影響して、実際にはこの要求に十分応じきれてない状況である。

そのような状態を補なうための方策として、外部の専門業者を利用する場  
合がある。

外部からの助けや外部への委託によって処理されている部分は、年々増加  
してきており、今後も質的な変動はあったとしても増えつづけることは確か  
であろう。

このような状況から効率の良い外注の利用法について考えてみた。

#### 4.5.1 外注利用の目的・判断基準

外注を必要とする理由としては、アンケート調査結果からみると、工  
程や工数の不足や要員不足、またスキル・経験の不足によるところが多  
いようである。

また、外注する作業工程が、あらかじめ決まっているところも割り合  
い多くみうけられている。特に労働力集約型の作業や、作業自体が独立  
していて、行いやすい部分に多いようである。

今回のアンケートでは、ライフサイクルの各項目毎や、作業内容別で  
はなく、全体としての外注を利用する場合の理由を調査している。

仮に、細かく調査したとしても、同じ理由が挙げられたであろうが、  
傾向については差が出たのではないだろうか。

例えば、スキル不足のため外注を利用するケースは、システム分析、  
設計段階での理由として多いと思われる。

また、外注する工程をあらかじめ決めている場合としては、プログラ  
ミング・オペレーション・データエントリ作業段階で多くなる。

要員不足やピーク時の対策として、外注を利用するケースは、作業段  
階や内容によっても差はなく、全般的にみても高い値になろう。

外注利用においては、あらかじめ計画段階で決まっている場合と、期  
日に遅れるなどの問題が予測され、間際になって外注利用が決まるケー  
スがある。特に、後者の場合は、外注先の選択が十分でなく、また発注

条件も無理な内容が多くなる傾向にあるため、そのような事態にならぬように十分な外注利用計画をたてる必要がある。

外注計画をたてて、余裕のあった発注を行うことは、発注元と発注先の双方にとって、高いメリットがある。

- ① 適任者を割り当てられる。
- ② 質の良い仕事が期待できる。
- ③ 融通性がより高くなる。

また、あらかじめ決まった工程や作業内容を外注している場合でも、以下のようなメリットがある。

- ① 事情の判っている、気心の知れた会社、担当者を確認しやすい。
- ② 外注管理のためのノウハウが得られ、より効率的な発注が行える。
- ③ 発注先の経営安定化に寄与する。
- ④ 不測の事態が発生しても、対応がしやすい。
- ⑤ 外注先のスキルが向上する。

#### 4.5.2 外注を利用する段階

システム開発段階から、計算業務自体を委託するまで、様々な作業が委託されているが、それらは下記のように分けることができる。

- ① システム分析・設計段階
- ② プログラム製造段階
- ③ オペレーション
- ④ 入力データ作成（データエントリ）
- ⑤ コンピュータ処理（委託計算）

これらの作業段階で、どのように外注利用が行われるかを調べてみた。

##### (1) システム分析・設計段階

この段階の委託の形態には、コンピュータ処理迄を前提としたシステム分析設計作業を委託する場合と、コンピュータ利用にとらわれず、



事務改善や事務の合理化を目指したコンサルティングを委託する場合がある。

前者は、要員不足とスキル不足から利用するが多い。

また、後者では、外部に委託すること自体が客観性や独立性を持って作業に当れるため、効果を上げることにつながる一面も持っている。

システム分析・設計段階を外部委託する場合、出す側にとって注意しなければならないことは、外注者のスキルが高くなるため、コストが高くつくこと、またコミュニケーション不足や、要求変更、仕様変更が発生したときの対応を事前に検討し、後から融通性がなかったなどの不満が生じないようにする。そのためにも、発注元に担当者が窓口を設定し、共同作業体制やきめ細かな外注管理体制を敷くことが重要である。

特に、仕様変更はある程度避けられないので、あらかじめそれを予測した弾力性のある設計要求を出すべきである。

要員不足、スキル不足が委託理由の上位を占めているように、システム分析、設計能力をもった人材の育成は時間がかかり、制約も多いのが一般的である。今後も、この段階の委託は増えていくと思われる。

一方、機密保護や、融通性に乏しいのは困るとして、外注していないところも多く、他の作業段階と比較すると、外注利用率は最も低い。

## (2) プログラム製造段階

プログラムの作成は、外注することに決めているケースも割り合い多くみうけられる。

外注を利用する理由としては、要員不足や作業期間が厳しく期限に間に合わない場合などが多い。

## (3) オペレーション

コンピュータ能力の向上や、システム開発が増えていることにより、システム分析・設計やプログラム作成だけでなく、コンピュータや端

末機のオペレータ要員についても不足や労働条件の悪化が進んできている。

一般に、要員の増加は、企業にとって厳しく制限するのが常であり、特にオペレーションのような労働力集約型の職種は、その要員を外部に委託する傾向が強い。

委託形態としては、計算処理自体も委託する場合と、要員を派遣してもらい自前のコンピュータのオペレーションを委託する場合がある。

理由としては、労務管理によるところが最も多く、次いで要員不足である。

最近では、コンピュータ室の運用管理全体を外部委託してしまうケースも出ている。

ここで注意しなければならない点は、労働条件の悪さをそのまま外注へ移しても、結局は定着率の低さや、スキル不足、モラルの低下、コスト高につながり解決したことにはならないことである。

システム機能の不足、運用形態、機能改善の余地などについても合わせて検討すべきである。

#### (4) データエントリ

オペレーション委託と同様に、要員不足や労務管理対策、また仕事が集出して発生したときなどに利用している。

データ入力は全て外部に委託するケースや要員を派遣してもらい継続して利用している場合が多くみられる。

#### (5) 計算業務委託

外部の計算センタへ処理を委託するケースは、コンピュータの普及によって少なくなった一面と、逆に、システムの増大と要員不足やピーク時対策、さらには障害対策などの面で積極的に外部を利用して行く傾向とが現われてきている。この違いはシステムの規模にも相関があり、減少傾向は小規模なユーザによくあらわれ、外部委託傾向にあ

るのは、大規模なユーザに多くみられる。

#### 4.5.3 発注の形態

外注利用の形態は、大きく2つに分けることができる。

##### (1) 業務委託（一括発注）

外注会社が作業管理までも行う。作業としては、独立した内容のものが多い。

##### (2) 要員派遣

要員を発注先へ派遣し、作業内容や管理を発注先が行う形態をいう。

最近では、業務委託の比率が高まってきており、行政当局からも、その方向へ進むように指導してきている。

特に、要員派遣については、労務管理の面や、就業条件の不明、定着率の低下などを招く恐れがあるため、労働者の保護と雇用の安定の面で、現在、労働省で調査し、中央職業安定審議会に諮り、人材派遣事業法として制度化をはかるべく作業を進めている。

#### 4.5.4 外注先の評価

外注先の評価は大変重要なことである。

選定作業や結果に対する正しい評価が行われなかったとき、その影響は発注元に対しても思いもかけないほどの影響を及ぼすことがある。

外注先を選ぶ基準としては、スキルや作業方針、経営者や担当者の考え方や人間性など、常識的なことがらが多い。

外注利用するうえで、特に注意を払いたいことがらを挙げてみる

- (1) 品質に対する考え方や具体的な対策をもっているか。
- (2) 納品後の保守については事前に検討と確認をしておく。
- (3) 一括発注であっても、進捗報告や作業途中のレビュー、チェック項

目の確認などに手を抜かないこと。

(4) 外注先とのコミュニケーションを密にする。

(5) 外注先の作業環境や作業条件にも注意を払う。

別な見方をすると、優秀な外注先を見つけたならば、出来る限り継続した好ましい関係を保つことにある。

#### 4.5.5 ソフトウェア・プロダクトの利用

外注の活用と似たところをもっているものとして、ソフトウェア・プロダクトの利用がある。

最近では、ユーザに限らず、コンピュータと係り合いのある事業全体として云えることに、ソフトウェア費用の急激な増加傾向があり、要員の不足に拍車がかかっている状況がある。

その対策の一つとして、ソフトウェアの有効利用、汎用化が叫ばれている。

しかし、ソフトウェア流通に関しては、最近の調査報告書などを読んでも、欧米に比べ、日本は大きく立ち遅れていると述べられている。

今でも、品質の良いもの、適応性の高いもの、拡張性や融通性に富んだものといったソフトウェア・プロダクトに対しては、強い需要が見込まれており、ニーズに合ったソフトウェアプロダクトの提供が望まれている。

また、昭和57年度から、プログラムの割賦販売契約、及び購入資金借入保証契約並びに、リース契約による取引に対して機械類信用保証法が適用されるようになり、プログラムの割賦販売、リース等による取引に伴うリスクを保険する制度が確立されたことも大きく寄与してくると思われる。

ソフトウェア・プロダクトを利用するつもりのない利用者が、導入しない理由として挙げていることは、大体以下の理由が多い。

- (1) ニーズに合ったものがない。
- (2) 拡張性、適応性に乏しい。
- (3) 価格的に合わない。
- (4) ソフトウェア・プロダクトの情報がない。
- (5) 自前の要員や、所有のソフトウェアで十分対応可能である。

#### 4.6 開発計画の留意点とチェックリスト

本章では、4.1節での開発計画の説明、4.2節での技法・手法・ツールの紹介と効果的な使い方、4.3節では見積り方法についての説明、そして4.4節での体制、4.5節では外注の活用について、といった開発計画段階に関連した内容の説明を行ってきた。

これらのなかから、注意点や留意点をまとめてチェックリストを作成した。このチェックリストが開発計画をたてる作業において参考になればと思う。

		留 意 点	チェック
見 積 及 び 計 画 立 案	工程計画	システム構成図、機能一覧表等はできているか。	
		各工程の仕事の範囲、作業内容、作成するドキュメント種類と内容等は明確か。	
		見積の基準値は最適なものか。	
	予算計画	予算計画には人件費、設備使用費、材料費、その他雑費等の直接費の他に、管理費又は部門間接費を加えたか。	
		標準単価は適切か。	
	要員計画	他のプロジェクトとの関係を考慮したか。	
		要員配置に当っては人間関係をも配慮したか。	
		開発段階と運用段階の関係も考慮したか。	
	外注計画	外注する目的は明確か。	
		外注方法は明確か。	
外注先の能力の把握はできているか。			
要員教育計画	教育の目的は明確か。 目的によって3つの場合に分けることができる。 ⅰ) 一般的な教育 ⅱ) 開発的教育 ⅲ) 運用的教育		

		留 意 点	チェック
見 積 及 び 計 画 立 案	設備計画	見積の基準は適切か。	
		開発時設備計画と運用時設備計画とを分けて考えたか。	
		設備の使用予定時間は、総使用時間のみでなく、一日一人平均の時間も算出して、実施可能か否かを判定したか。	
		設備の管理体制は明確か。	
	開発支援計画	支援対象は明確か。	
		支援方法は明確か。	
		支援体制は明確か。	
		支援による期待効果を明確にしたか。	
	工程管理計画	管理項目は明確か。	
		実績データの収集方法、時点を明確にしたか。	
		工程計画との対比方法は明確か。	
	費用管理計画	発生費用項目は明確か。	
		発生費用の収集方法及び収集時点を明確にしたか。	
		予算計画との対比方法を明確にしたか。	
	品質管理計画	品質管理の対象、及び項目は明確か。	
		管理方法、時期を明確にしたか。	
		管理体制を明確にしたか。	
	検査・検収計画	検査、検収の項目は明確か。	
		検査方法、時期、期間を明確にしたか。	
		検査結果の確認方法を明確にしたか。	
		検査結果の反映方法を明確にしたか。	
		検査完了の条件は明確か。	
	移行計画	検査結果の管理、保管方法を明確にしたか。	
		手作業、又は旧システムと本システムとの並行処理期間及びその期間に確認すべき事項を明確にしたか。	
運用時教育計画は立てたか。			
伝票や帳表等の印刷手配をしたか。			
データの継続性は保たれているか。			
プログラムのコンパージョンは必要か。そのスケジュールは明確か。			
運用計画	並行処理期間中に発生する問題の解決のための体制は明確か。		
	データの収集方法、時点を明確にしたか。		
	運用方法を明確にしたか。		
	出力帳表の配布方法、時点を明確にしたか。 日次、月次の運用費用及び数年先までの運用費用を見積ったか。		

		留 意 点	チェック
見積及び計画立案	保守, 拡張計画	変更内容及び変更量は明確か。	
		変更に必要な期間, 人員, 設備は明確か。	
		検査, 検収の方法は明確か。	
		現システムから変更, 拡張後のシステムへの移行の方法は明確か。	
		運用形態の変更点は明確か。	
	パッケージド・ソフト 購入計画	要求分析の結果と候補に上っているパッケージド・ソフトとの機能, 性能の差は明確になっているか。	
		不足する機能の追加方法, 期間, 費用等は明確になっているか。	
		既存のシステムとのインターフェイスは明確になっているか。	
		試行期間は明確か。	
		運用時教育計画を立てたか。	
技法・手法・ツール	品質向上を図るには, 道具を使うことである。		
	知る努力, 使う努力をしているか。		
	部分的, 補助的にでも使えないか。		
	適用範囲と活用部分とにずれはないか。		
	使用後の評価とより効果的な利用法を検討する作業を計画しているか。		
開発計画の作成から承認まで	計画書は要領よく, 判りやすくまとまっているか。		
	過去のデータを参考にしているか。		
	開発要求部門の要求仕様は把握できているか。		
	開発要求部門と計画承認部門との調整は十分か。また問題が発生してもすみやかに調整できる体制か。		
	関連する業務に精通しているか。		
外注の活用	外注利用	要員不足が慢性化していないか。	
		品質に対して十分認識している外注先であるか。	
		契約後に仕様変更が生じた場合の対応は。	
		標準化や汎用性について具体的な指示を出せるか。	
		外注先の労働条件や作業環境に注意を払っているか。	
		レビューのための工数・日程は組み込んでいるか。	
		外注先とのコミュニケーションは不足していないか。	
		十分な発注・契約期間があるか。	
		外注先要員の移動がはげしくないか。	
	ソフトウェア・プロダクトの利用	ニーズに合っているか。	
拡張性・適応性に優れているか。			
物件の正確で詳しい情報が入手できているか。			

		留 意 点	チェック
外 注 の 活 用	ソフトウェア・プロダ クトの利用	自社内に似たソフトウェアが存在していないか。	
		費用面での妥当性の検討は十分か。	
		保守や改造要求への対応はどうか。	
		自社のコンピュータシステムとの互換性はあるか。	



## 5. ソフトウェア開発計画立案の実際例



## 5. ソフトウェア開発計画立案の実際例

本章では、ソフトウェア開発計画立案の具体的な例として、3社の事例を紹介することにする。

### 5.1 A社の事例

#### (1) A社のコンピュータ部門の概要

A社（資本金184億円，従業員1万1千名）のメインコンピュータはIBM 3081-K，人員構成は，管理者4人，システムエンジニア25人，プログラマ6人，オペレータ4人の計39人となっている。また，昭和58年度のソフトウェア開発予算は約5億2千万円である。

#### (2) システム開発の高度化・効率化に対する考え方

ここでは，A社がシステム開発の高度化，効率化のために採用しているマネジメント体系やライフサイクルの考え方等を示す。

##### ① コンピュータ部門の役割の変化と要員育成方針

A社では，コンピュータ部門を単なるプログラム開発あるいはコンピュータのオペレーション部門から脱皮させ，コンピュータ問題に限らず，あらゆる業務の改善を役割とするような集団に変化させようとしており，次のようなことを課題として取組んでいる。

- ㉑ 組織を横断する業務の効率化
- ㉒ トップの意志決定支援システムの開発
- ㉓ 教育コンサルタント機関としての役割強化
- ㉔ システム・プログラムの大衆化（ユーザ自身によるシステム・プログラム開発の支援）
- ㉕ システム開発のスピードアップ
- ㉖ 人材育成のスピードアップ

これらの課題から解することは，システムの開発はユーザ部門自身が行

うようにし、コンピュータ部門はそのサポートにまわるとのことである。これは、拡大される業務に対処するための一つの方法として注目される。ただし、製品としてのプログラムの品質を確保し、単工数当りの生産性を上げるためには、ユーザ部門に対するかなりの教育が必要となろう。

また、A社では、開発業務の効率化のために、次のような方法を用いている。

- ② ソフトウェア・パッケージの導入と活用
- ③ 外注の利用
- ④ コンピュータ部門のレベルアップ

②のソフトウェア・パッケージとしては、PRIDE、ADABAS、MARK IV、等を導入して活用している他、問題解決のための手法としてF-SPANあるいは、運用のためのBELシステムやA-AUTOを自主開発し、これらは、外部に対して販売もされている。(図5-1)

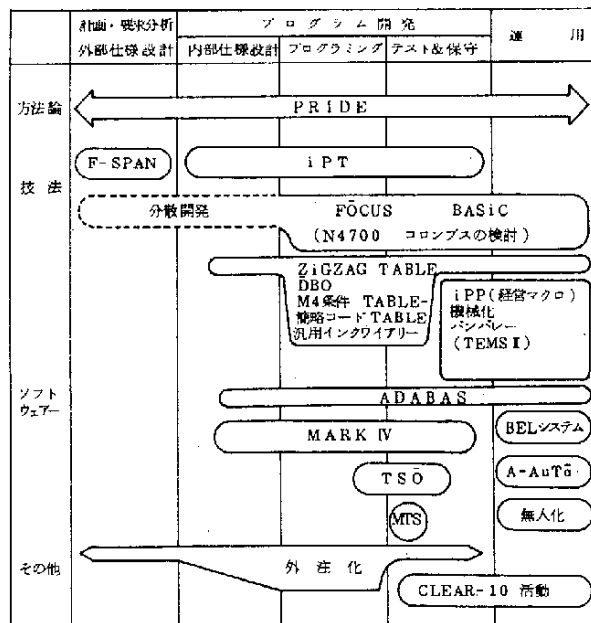


図5・1 A社で利用しているツールや手法等

また、外注も積極的に利用しているが、利用する場合の方針として、概要設計までは必ず自社内で行うようにし、要求仕様が不明確なまま発注することはしないようにしている。

A社が、システム開発の高度化、効率化の有力な手段として考えていることは、㉔のコンピュータ部門のレベルアップである。A社では、コンピュータ部門における要員育成を次のような方針に基づいて行っている。

- ㉔ 全員が高い目標にチャレンジし、生き生きと働ける職場風土の形成
- ㉕ 職場活性化対策の継続的实施を行うなかでの企業目標と個人成長目標のリンク
- ㉖ 単なるコンピュータ化ではなく、業務全体の改善の志向
- ㉗ 問題解決型人間（業務改善者）の養成
- ㉘ 変化に即応できる柔軟な体質作り
- ㉙ スキルフルな人材の養成（どんな問題にも即座に対応でき、かつ他部門でも通用する人間の育成）
- ㉚ 他社に先んじる高い技術力の保持
- ㉛ 技術専門集団の設置と技術力向上対策
- ㉜ 計画的な人材育成
- ㉝ 層別の必要スキルの提示と自己分析
- ㉞ キャリアパターンの設定と人材点検の実施
- ㉟ システム・プログラム教育体系の整備と実施

要員育成の全体系を図5・2に示す。

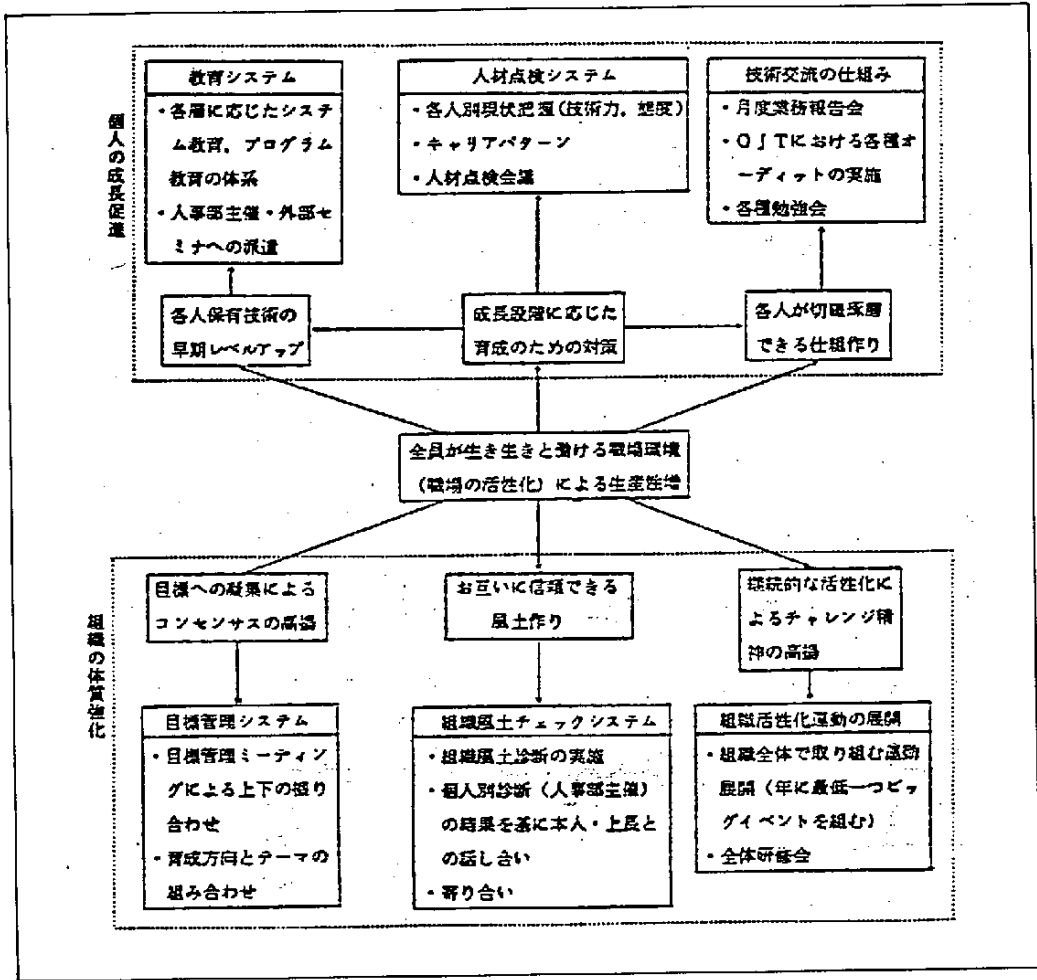


図5・2 A社における要員教育の全体系

② システム開発の効率化に対する取り組み方

ここでは、システム開発の高度化、効率化のために、A社が採用している次の3つのことについて紹介する。

- ① システム・マネジメント体系
- ② ライフサイクル
- ③ システム開発組織

③ システム・マネジメント体系

A社では、システム・マネジメント体系として、図5・3に示すようにPRIDEを中心として考えている。

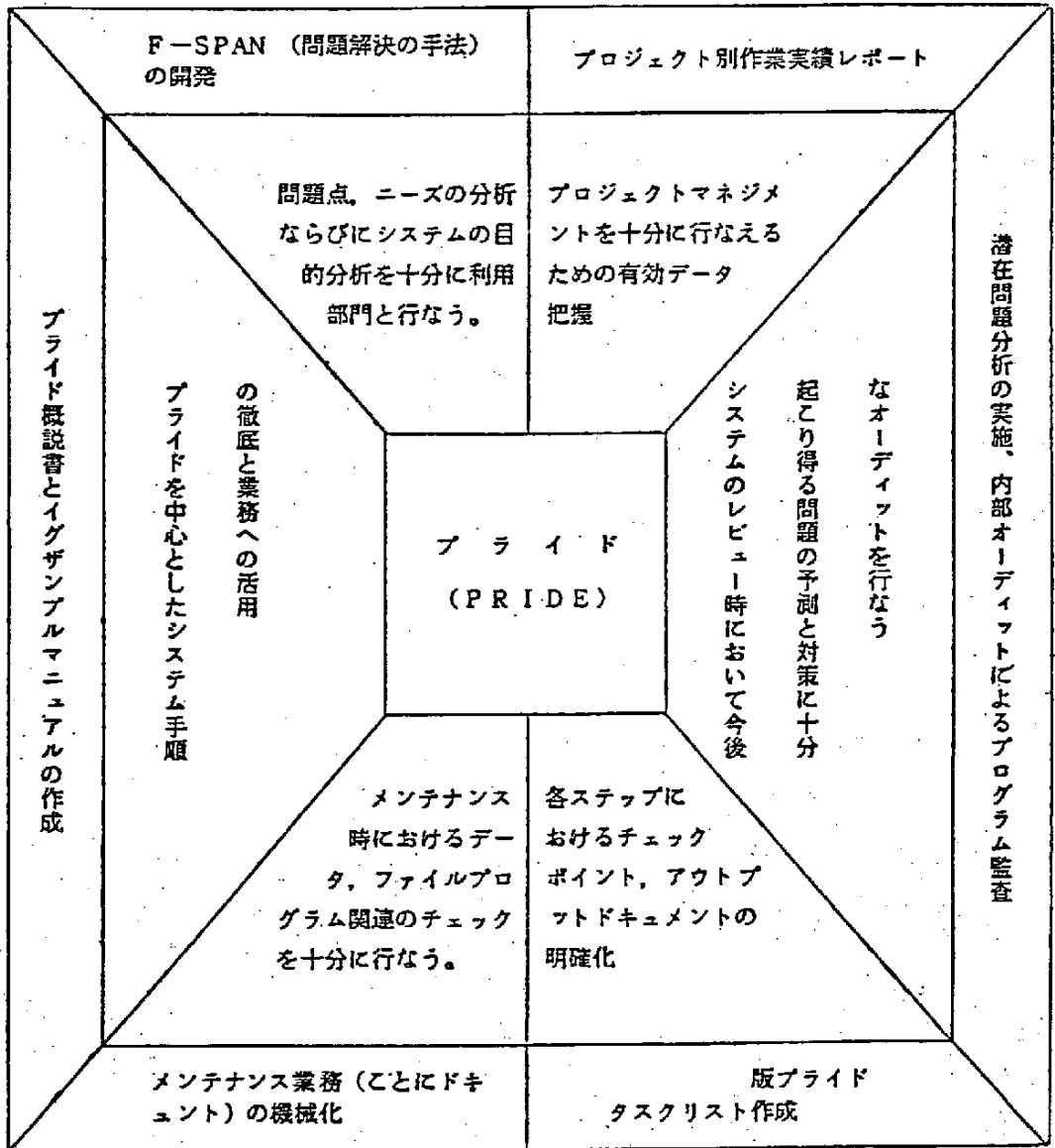


図5・3 A社におけるシステム・マネジメント体系

ここで特徴的なものは、PRIDEの活用、問題解決手法のためのF-SPANの開発、および潜在問題分析の実施である。PRIDEとは、システム設計手法を標準化したものであり、A社ではA社独自のタスク・リストを作成し、設計手順と各手順におけるチェックポイントを明確化している。F-SPANは、A社が独自に開発した問題解決の手法で、問題点・ニーズ分析、目的分析、機能設計の3部からなっており、新しくシステム開発を行う段階で、利用部門との設計構想を固めるうえで有効なものである。これについては(3)ソフトウェア開発計画立案の方法～でその概要を紹介する。また潜在問題分析とは、システム開発のチェックポイントで必ず次以降に起りうる問題要因、影響部分、予防策、緊急策という形で事前に分析し、予防対策を打つ手法である。

#### ④ ライフサイクル

A社では、図5・4に示すようなライフサイクルの考え方を採っている。各フェーズにおける作業概要は次のとおりである。

##### ㉑ 予備調査

問題についての背景や目的、問題領域の範囲、関連部門の明確化、費用の概算見積り等を行う。

##### ㉒ 現状ニーズ調査

対象とする業務に関するニーズ調査を行う。

##### ㉓ システム概案作成

インプット、アウトプットを中心にして概念図を作成し、合わせてシステム評価見積書を作成する。

##### ㉔ サブシステム設計

人間に係わる部分の詳細な検討と、コンピュータに係る部分のラフな検討を行う。

##### ㉕ システム詳細設計

コンピュータを中心として、人間-コンピュータ系の1つ1つのプロシジャ-を設計する。



① プログラム設計

プログラムの構造を設計し、プログラム仕様書を作成する。

② コーディング・単体テスト

プログラムをコーディングし単体テストを実施する。

③ システムテスト

プログラムの統合テストを行う

また、各フェーズ毎のレビュー・ポイント、レビュー参画者、最終検閲者を表 5.1 に示す。

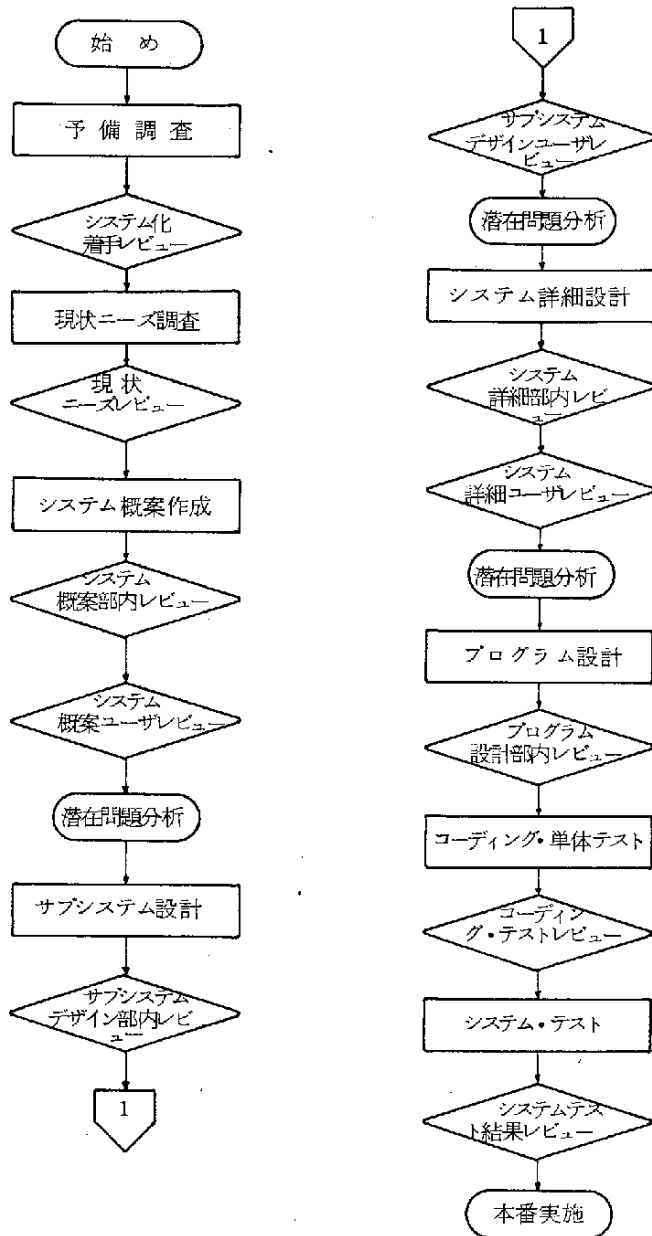


図5・4 A社におけるライフサイクル

⑤ システム開発組織

A社の開発組織は、ユーザの役割分担の明確化とユーザが積極的に参画できる仕組となっている。メンバと役割を表5・2に示す。

この中で注目されることは、品質管理の観点から、プログラム技術グループを設置し、各フェーズの部分検討の際に参画させ、技術面からの監査とアドバイスを行っていることであろう。ソフトウェアの品質管理は、実際に開発に携わる者以外の人が行うべきであり、その意味では良い方法といえる。

(3) ソフトウェア開発計画立案の方法

A社におけるシステム開発計画立案の方法の特長はA社が自主開発した手法F-SPANの活用である。そこで、ここでは、F-SPANの概要を紹介することにする。

F-SPANは、F(A社の名前) Systematic Problem Analysis method by Needs の略であり、問題の当事者が衆知を集めて、コンセンサス・ベースで問題を整理・解決する体系的手法である。F-SPANの実施手順を図示すると、図5・5のようになる。またF-SPANによる問題分析から概要設計までの進め方の特徴は次のとおりである。

- ① 問題点分析シートを用いて問題点を整理する。
- ② 1泊2日の合宿で、提起された問題の因果関係を分析し、重点問題を発見する。
- ③ 重点問題を起点として、問題の解決目的をボトムアップ、トップダウン手法を用いて追求し、解決策としての目的を明確にする。
- ④ 2泊3日程度の時間をかけ、関係者全員が問題解決策の具体的イメージを共通に理解できるレベルまで、解決目的の機能をブレイクダウンし、システムの概要設計を行う。

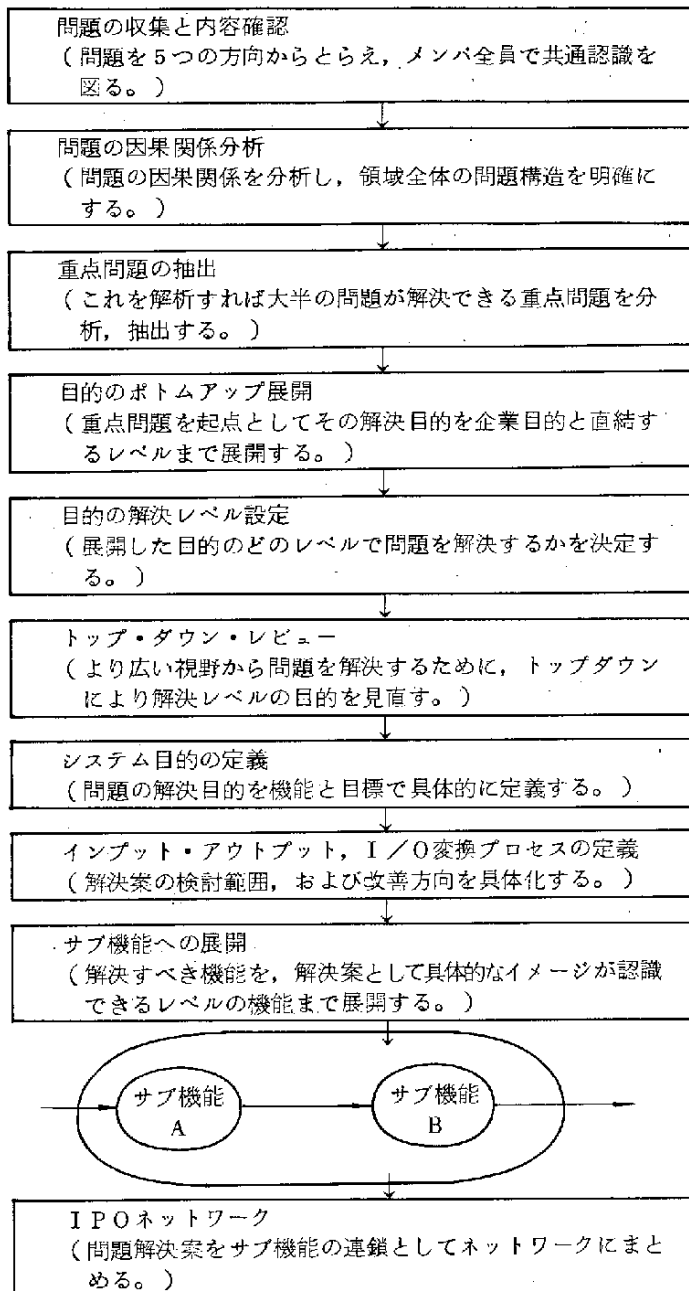


図5・5 FSPANの実施手順

表5・1 各レビューのポイント、参画者、最終検定者

レビューの種類	レビューポイント	レビュー参画者	最終検定者
システム化着手 レビュー	インタビュー結果に基づきコンピュータ利用の有効性を判断し、他のプロジェクトの優先順位等を勘案し、システム化の実行性を検討する。	主任、グループリーダー ユーザー担当者	ユーザーマネージャー 電算部長
現状ニーズレビュー	現状ニーズが正しく把握されているかどうか、又ユーザー全員が共通認識にたてるかどうかの検定を行なう。	主任、グループリーダー システム設計者	ユーザーマネージャー 担当者
システム概要 部内レビュー	実行可能性についての部内レビュー	主任、グループリーダー システム担当者 運営グループマネージャー プログラムオーディター	電算部長
システム概要 ユーザーレビュー	ユーザーニーズが満足されたシステムか。 システム概要の内容、効果、コスト、スケジュールの検定を行う。 このレビューでスケジュールが確定する。	主任、グループリーダー システム担当者 ユーザー担当者	ユーザーマネージャー 電算部長
サブシステムデザイン 部内レビュー	人間系の運用性、実行可能性の現状システムとの構造点の検討を行う。	主任、グループリーダー システム担当者 運営グループマネージャー プログラムオーディター	電算部長又は主任
サブシステム ユーザーレビュー	サブシステムのロジック検討と人間系のつめ、効果、費用、スケジュールの見直し	主任、グループリーダー システム設計者	ユーザーマネージャー 電算部長
システム詳細 部内レビュー	コンピュータ系の実行可能性もれのチェック	グループリーダー システム担当者 運営グループマネージャー	主任
システム詳細 ユーザーレビュー	最終的なシステム内容および効果、コスト、スケジュールの確認	主任、グループリーダー システム担当者	ユーザーマネージャー ユーザー担当者
プログラム設計 部内レビュー	データベース・プログラム構造等最終的な技術レビュー	グループリーダー システム担当者 プログラムアナリスト	プログラムオーディター 運営グループマネージャー
コーディング・テスト レビュー	プログラムのダブルチェックとテスト結果のチェック	プログラマー	プログラムアナリスト
システムテスト 結果レビュー	システムテストデータに基づく最終結果のチェック	プログラマー システム担当者	主任 ユーザー担当者

表5・2 A社における開発組織

組 織		役 割
電 算 部 門	部 長	プロジェクトサブマネージャーとしてシステム開発の承認と各フェーズ終了時点のレビューに参画し、承認を行なう。
	主 任	プロジェクトリーダーを補佐すると共に、必要に応じ各アクティビターの検討に参画し、各フェーズ終了時点の潜在問題分析の実施と内部検討会において承認を行なう。 スケジュール管理の総合調整も行なう。
	グループリーダー	実践部隊のプロジェクトリーダーとしてシステム設計者の補佐を行なうと同時に各アクティビターに参画し、システム監査も待せて行なう。 プロジェクト単位のスケジュール管理も行なう。
	システム設計者	メインシステム設計者としてPRIDEの手順にそった作業を実施する。
	プログラム設計者	プログラム設計者としてのプログラム仕様書作成を行なう。 A社においてはシステム設計者とプログラム設計者が同一人の場合が多い。
	プログラマー	プログラム仕様書に基づいてコーディングテストを実行する。
	プログラム技術 グループ	プログラム設計監査人として各フェーズの終了時の内部検討会に参画し、技術面よりの監査とアドバイスを行なう。
ユ ー ザ ー 部 門	運営グループ マネージャー	各フェーズの終了時の内部検討会に参画し、DP処理運用面よりの監査とアドバイスを行なう。
	ユーザーマネージャー (部長又は課長)	プロジェクトマネージャーとしてシステム推進に参画し、各フェーズ終了時点およびそれ以外に規定されているアクティビターの終了時点に参画し承認を行なう。
	ユーザー担当者	共同のシステム設計面として、システム検討に参画すると共に人間系の設計では中心的役割をになう事もある。又ユーザー運用マニュアルの作成を行なう。

## 5.2 B社の事例

### (1) B社のコンピュータ部門の概要

B社(資本金1,298億円,従業員1万3千人)のメインコンピュータはIBM3081,及び3033であり,システムセンタと呼ばれるコンピュータ部門の人員構成は,図5・6のようになっている。また,昭和58年度のソフトウェア開発予算は,約2億5千万円である。会社の規模からすると比較的小さいが,これは,主要なシステムの開発を既に終了し,現在はメンテナンスが中心となっているためである。

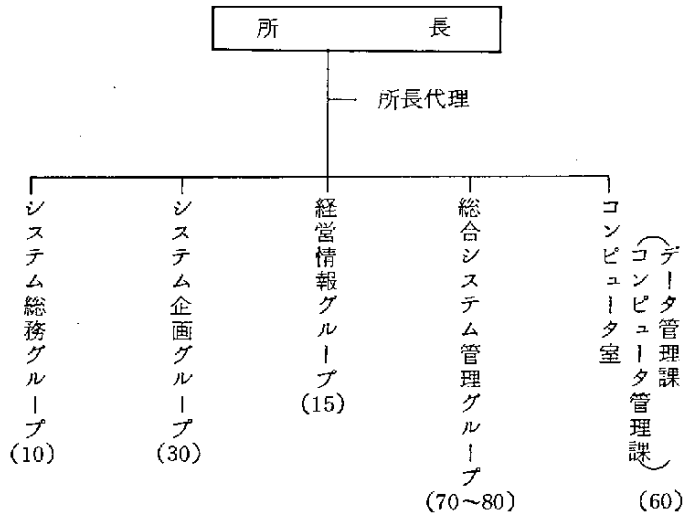


図 5・6 B社のコンピュータ部門の人員構成

(2) システム計画立案の手順

B社におけるシステム計画立案の手順を示すと、図 5・7 のようになる。それぞれの段階における作業内容と、担当者は次のようになる。

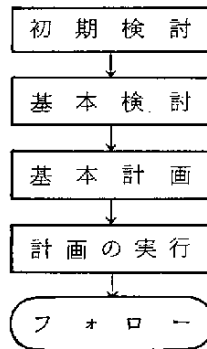


図 5・7 B社におけるシステム計画立案の手順

① 初期検討段階（企画グループ）

この段階では、ニーズの確認，原局窓口の確認，システム化の概要（対象・範囲），リソースやメリット，及び完了時期の推定そして，担

当グループの選定を行う。

この段階を終了すると、その結果に対する部会審議が行われ、それに基づいて、所長はプロジェクトとして正式に承認する。

② 基本検討段階（担当グループ）

開発に必要なリソース、メリット、また開発スケジュールについて可能な限り検討し、基本計画概要を作成する。

この段階を終了すると、検討結果に対する部会審議が行われ、それに基づいて所長は、プロジェクトの実施を正式に決定する。

③ 基本計画段階（担当グループ）

②の検討結果に基づいて、計画の実施に必要となる基本計画を作成する。この段階で、②の内容に大幅な変更がある場合には企画グループと協議のうえ、部会の審議を受ける。

この段階を終了すると、基本計画に対する部会審議が行われ、システムの内容、及び必要なリソースを最終的に決定する。

④ 実行段階（担当グループ、コンピュータ室）

詳細設計から運用開始までを行う。

⑤ フォロー

運用1年後を目安に、計画実績を中心にフォローし、部会に報告する。

(3) 基本計画書の内容とスケジュール例

B社では、各種ドキュメントの内容に対する標準項目が定められている。ここではそのうち基本計画書の内容を示すとともに、予算計画のためのフォーマット及び開発スケジュールの一例を示す。

① 基本計画書に記述する標準項目

① はじめに

必要に応じシステム開発に当っての経緯等を記述する。

② システム化の対象

対象とする業務、現状の分析、システム化のニーズ、システム化の



範囲，効果を記述する。

㉔ 他システムとの関連

他のシステムと関連がある場合にはここにまとめて記述する。

㉕ システム処理の概要と運用時の費用

システムの構成，運用時のスケジュール，運用時の必要リソース，費用を記述する。

㉖ 開発作業スケジュールと費用

システム開発，または拡充，改善のスケジュール，規模，所要リソース，所要費用を記述する。

㉗ フォーマット例

B社における予算計画のためのフォーマット例を図5・8，図5・9に，また，開発スケジュールの例を図5・10に示す。

(4) B社の外注利用法

B社は外注の利用について独特の方法を採っている。それは次のようなものである。

① 外注先にはプログラム作成以降の業務のみ発注する。

② 費用は出来高払い，すなわち最終ステップ数に応じて支払う。

なかでも②の費用算定方法は，最近余り行われていない。それは次の理由による。

① ステップ数の見積りを正確に行うことが難しい。

② プログラムの品質に問題が生じる場合が多い。

B社では，過去の実績を数量的に把握すること，及び厳密なコーディング基準を設定することで，比較的正しい見積りの実現，そして，品質の確保に成功している。

システム化費用（開発・メンテ）

作成（改訂） 年 月 日

		計 画 値		実 績 値		備 考	
		数 量	金 額	数 量	金 額		
マシン費用	社員人件費	人・月		人・月			
	C P U 関 係	時間		時間			
	3 8 0 0 関 係	枚		枚			
	計						
委託作業費	プログラミング	オンライン	ライン		ライン		
		バッチ・アセンブラー	ライン		ライン		
		P L / 1	ライン		ライン		
	メンテ	プログラム・メンテ	時間		時間		
		設計・その他	時間		時間		
	小 計						
	せん孔料	プログラム	件		件		
		移行データ等	件		件		
	小 計						
	計						
消耗品費	デバック用紙	枚		枚			
	オーバーレイ	組		組			
	磁気テープ	本・月		本・月			
	MSS カートリッジ	カートリッジ・月		カートリッジ・月			
計							
合 計							

（システム設計用紙 様式 K-1）

（ 頁）

図 5 ・ 8 B 社における予算計画書のフォーマット(1)

システム化費用（本番）〔月額〕

作成（改訂） 年 月 日

		計 画 値		実 績 値		備 考	
		数 量	金 額	数 量	金 額		
社 員 人件費	プログラム関係	人		人			
	データ管理関係	人		人			
	計	人		人			
マ シ ン 費 用	C P U 関 係	時間		時間			
	3 8 0 0 関 係	枚		枚			
	その他プリンター関係	枚		枚			
	M S S 関 係	カートリッジ		カートリッジ			
	DASD 関 係	3 3 5 0	パック		パック		
	端 末 機 関 係	件		件			
	O C R 関 係	時間		時間			
	裂 断 機 関 係	時間		時間			
計							
せ ん 孔 料		件		件			
消 耗 品 費	磁 気 テ ー プ	本		本			
	3 8 0 0 用 紙	枚		枚			
	事 前 印 刷 帳 表	枚		枚			
	O C R ドキュメント	枚		枚			
	M S S カートリッジ	カートリッジ		カートリッジ			
	計						
合 計							

（システム設計用紙 様式 K-2）

（ 頁 ）

図 5・9 B社における予算計画書のフォーマット(2)

83年 1月 7日  
82年12月24日

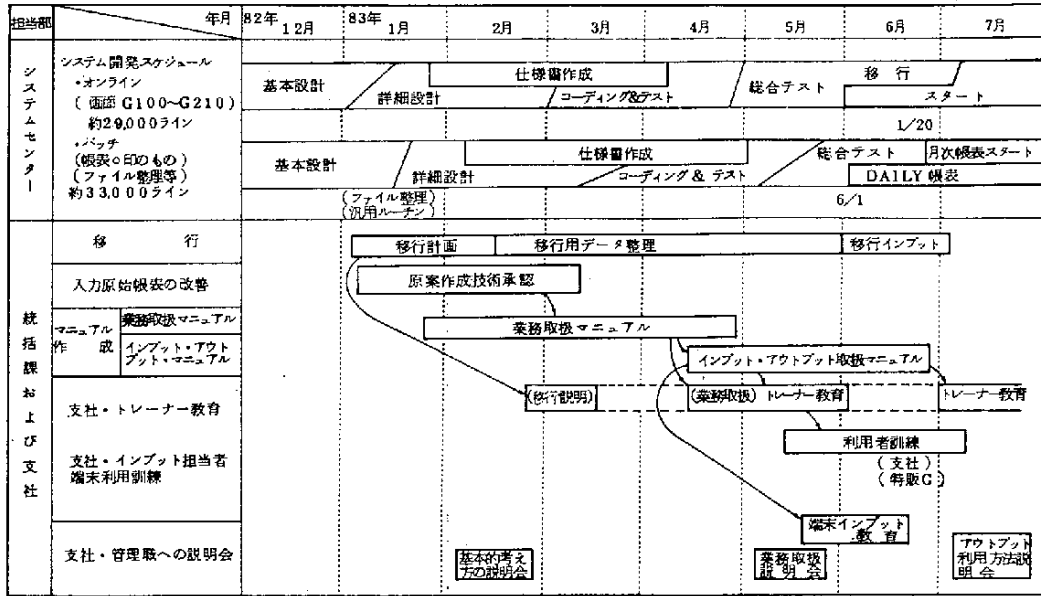


図5・10 B社における開発スケジュールの例

### 5.3 C社の事例

#### (1) C社のコンピュータ部門の概要

C社(資本金1551億円、従業員3万5千人)のメインコンピュータはIBM3081組織編成は、図5・11のようになっている。また、部員数は約180名である。

#### (2) 期間活動計画の策定

C社では、基本的に年度単位の業務を行っており、年度始めに期間全体の業務計画、つまり期間活動計画を策定している。そして上期の終りに予算の見直しを行い、改めて下期の計画を策定している。そこでまずC社における期間活動計画の種類と内容を示す。

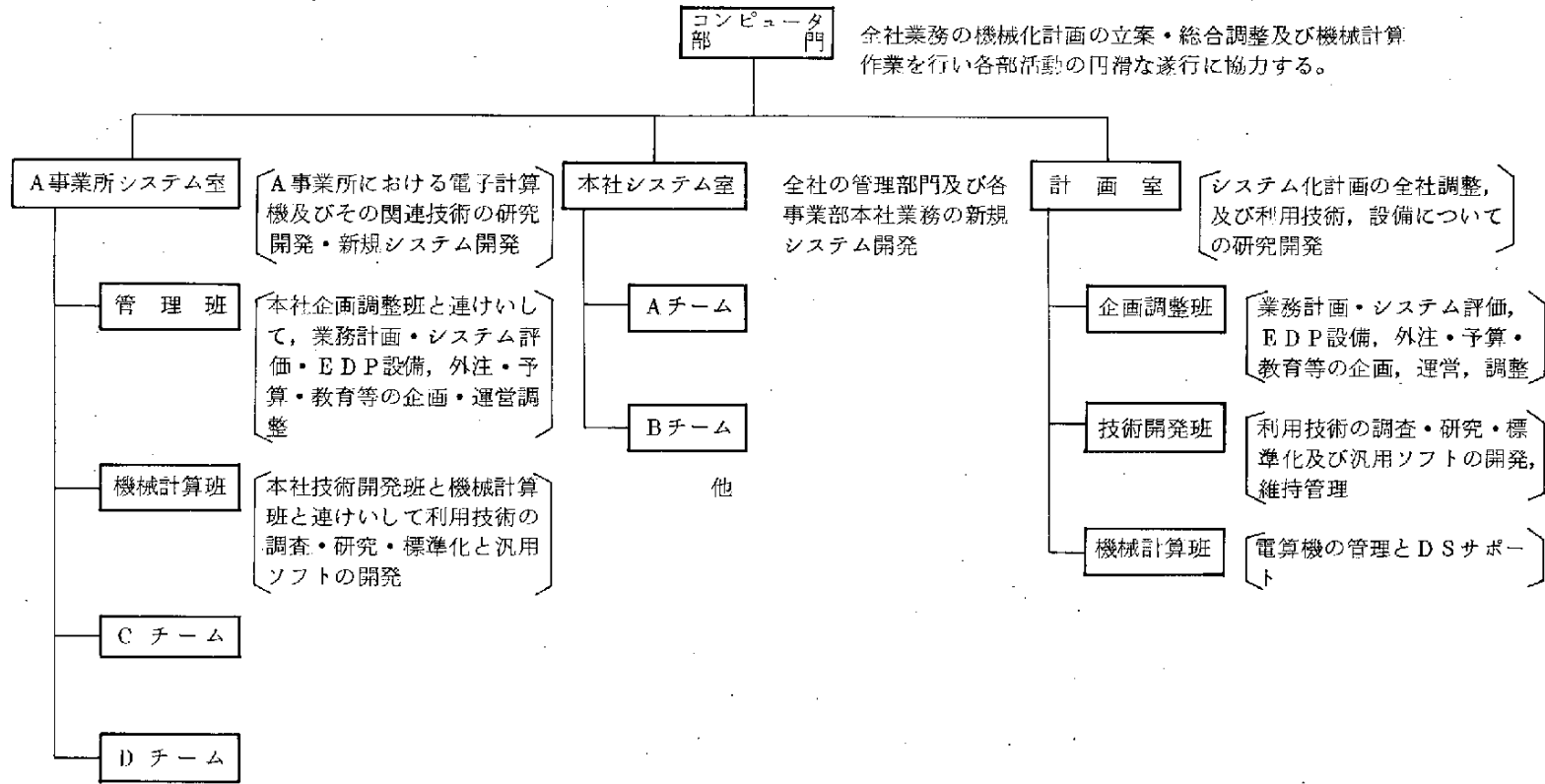


図 5・11 C社の組織編成

① 業務計画

システム開発、大型メンテナンスのスケジュールをシステム開発の各段階毎に線表で記入する。業務計画のフォーマット例を図5・12に示す。

② 設備計画兼操業計画

業務単位(コンピュータ処理単位)にコンピュータ関係設備の使用量を要求するため、システムの稼動時期に合わせて各設備の使用計画・オペレーションのスケジュール、伝送のスケジュールを数量又は線表で記入する。設備計画兼操業計画のフォーマット例を図5・13に示す。

③ ソフトウェア外注計画

C社では、プログラミング作業は外注しており、システムの大きさに応じて外注費用を見積る。

④ 本社計算機稼動計画

既に稼動しているシステムは、前期実績あるいは年度実績を参考にして月別稼動時間を見積る。新規に稼動するシステムについては、プログラムの構造、ボリューム、及びデータ量等が似ているシステムを過去のデータから探し出し、それを参考にして月別稼動時間を見積る。

⑤ 要員計画

現状の要員数に、今後システム開発等に要する要員数を見積り、それらを加えたものを計画とする。要員計画のフォーマット例を図5・14に示す。

次に、期間活動計画策定の手続であるが、C社では、業務計画を策定する前に経理・人事部と接衝し、予算、要員、設備、外注各計画立案の前提条件を設定してから業務計画立案作業を始めることにしている。従って当然、業務計画をまとめる過程では、優先業務の選定、スケジュールの調整といった内容の業務が伴ってくる。これらの作業はすべて、企画調整を担当するスタッフの仕事となっている。しかし、スタッフは直接業務計画立









案段階での業務間の調整にはタッチしていない。C社における期間活動計画の策定フローを図5・15に示す。

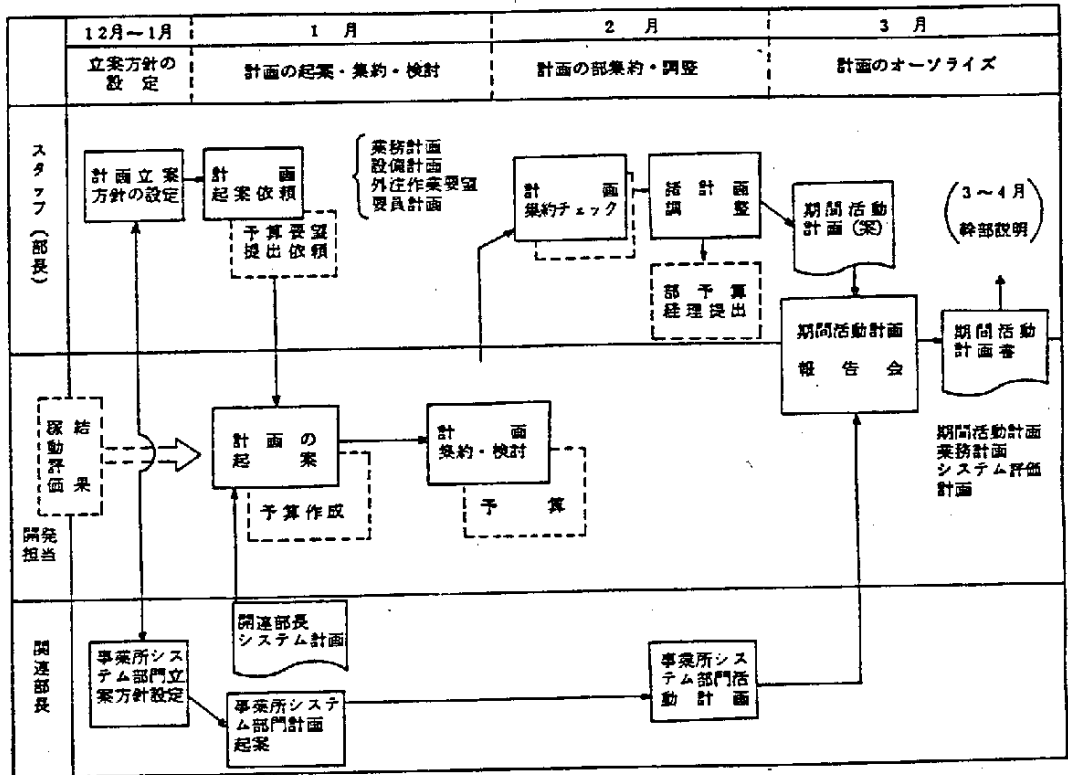


図5・15 C社における期間活動計画の策定フロー

### (3) 開発費用の計算方法

C社では、構想立案段階で費用の概算を見積り、基本設計段階で検定している。なお、変更ある場合は、詳細設計段階で見直しを行っている。C社での費用項目とその内容は次のとおりである。

#### ① 社内労務費

労務費単価/人・月 × 開発人数 × 期間(月)

#### ② ソフトウェア外注費

- ② 業務委託の場合
  - 1ステップ当りの単価×全プログラムのステップ数
- ③ 外部要員の場合
  - 外注労務費単価／人・月×人数×期間（月）
- ③ 開発用ソフトウェア費
  - 開発負荷の軽減を主目的としたソフトウェア費（ソフトウェアパッケージ等の購入費など）
- ④ 電子計算機処理費用
  - ① CPU費用＝CPU単価／分×開発，テストのCPU時間／月×期間（月）
  - ② その他の費用＝CPO（Concurrent Peripheral Operation）単価／分×開発・テストのCPO時間／月×期間（月）
  - ③ オンライン費用（開発・テスト期間中に発生する費用）
  - ④ レンタルの場合：機器レンタル料／月×発生期間
  - ⑤ 機器買取の場合：本稼動時迄に生ずる減価償却費用及び保守料
- ⑤ システム化要望部署が費用管理する機器及び作業用材料費用
  - ① 機器費用
    - ② レンタルの場合：機器レンタル／月×発生期間（月）
    - ③ 買取契約の場合：本稼動時迄に発生する減価償却費用及び保守料
    - ④ 作業用材料費用：作業用材料単価×使用料
- ⑥ 端末機の導入諸経費用
- ⑦ 端末側の設備工事費用
- (4) C社における品質管理

C社では、システム開発の各段階において検定者による検定を行い、品質の維持管理を行っている。また、検定は、職制活動に基づいて行っており、第1次検定、関連セクション検定、最終検定の3段階に分けて行っている。

- 第1次検定：原則として、システム化要望部署、情報システム部のシステム化担当室長が行い、検定内容によってはチームの主査が代行する。
- 関連セクション検定：情報システム部計画室が必要と思われる場合に行う。
- 最終検定：原則としてシステム化要望部署、情報システム部双方の部長が行い、検定内容によってはシステム化要望部署、情報システム部のシステム担当室長が代行する。

なお、計画立案段階の検定の考え方は次のとおりである。

① システム化検討着手検定

システム化の要望内容について、問題の整理状況、問題の解決方向としての、コンピュータ利用の有効性を判断し、更に期間活動計画等を考慮して検討する。

② 構想立案検定

システムの必要性と目的の妥当性、優先順位を検定し、更にシステム目的を表現するために各種案の中から選択されたシステム素案の最適性及び開発順位の合理性に関し、重点的に検定する。併せて今後のシステム開発を効率的に進めていくためにシステム開発推進体制についても検定する。

- ① 業務委託の場合
  - 1ステップ当りの単価×全プログラムのステップ数
- ② 外部要員の場合
  - 外注労務費単価/人・月×人数×期間(月)
- ③ 開発用ソフトウェア費
  - 開発負荷の軽減を主目的としたソフトウェア費(ソフトウェアパッケージ等の購入費など)
- ④ 電子計算機処理費用
  - ① CPU費用=CPU単価/分×開発、テストのCPU時間/月×期間(月)
  - ② その他の費用=CPO(Concurrent Peripheral Operation)単価/分×開発・テストのCPO時間/月×期間(月)
  - ③ オンライン費用(開発・テスト期間中に発生する費用)
  - ④ レンタルの場合: 機器レンタル料/月×発生期間
  - ⑤ 機器買取の場合: 本稼動時迄に生ずる減価償却費用及び保守料
- ⑤ システム化要望部署が費用管理する機器及び作業用材料費用
  - ① 機器費用
    - ① レンタルの場合: 機器レンタル/月×発生期間(月)
    - ② 買取契約の場合: 本稼動時迄に発生する減価償却費用及び保守料
    - ③ 作業用材料費用: 作業用材料単価×使用料
- ⑥ 端末機の導入諸経費用
- ⑦ 端末側の設備工事費用
- (4) C社における品質管理

C社では、システム開発の各段階において検定者による検定を行い、品質の維持管理を行っている。また、検定は、職制活動に基づいて行っており、第1次検定、関連セクション検定、最終検定の3段階に分けて行っている。

- 第1次検定：原則として、システム化要望部署、情報システム部のシステム化担当室長が行い、検定内容によってはチームの主査が代行する。
- 関連セッション検定：情報システム部計画室が必要と思われる場合に行う。
- 最終検定：原則としてシステム化要望部署、情報システム部双方の部長が行い、検定内容によってはシステム化要望部署、情報システム部のシステム担当室長が代行する。

なお、計画立案段階の検定の考え方は次のとおりである。

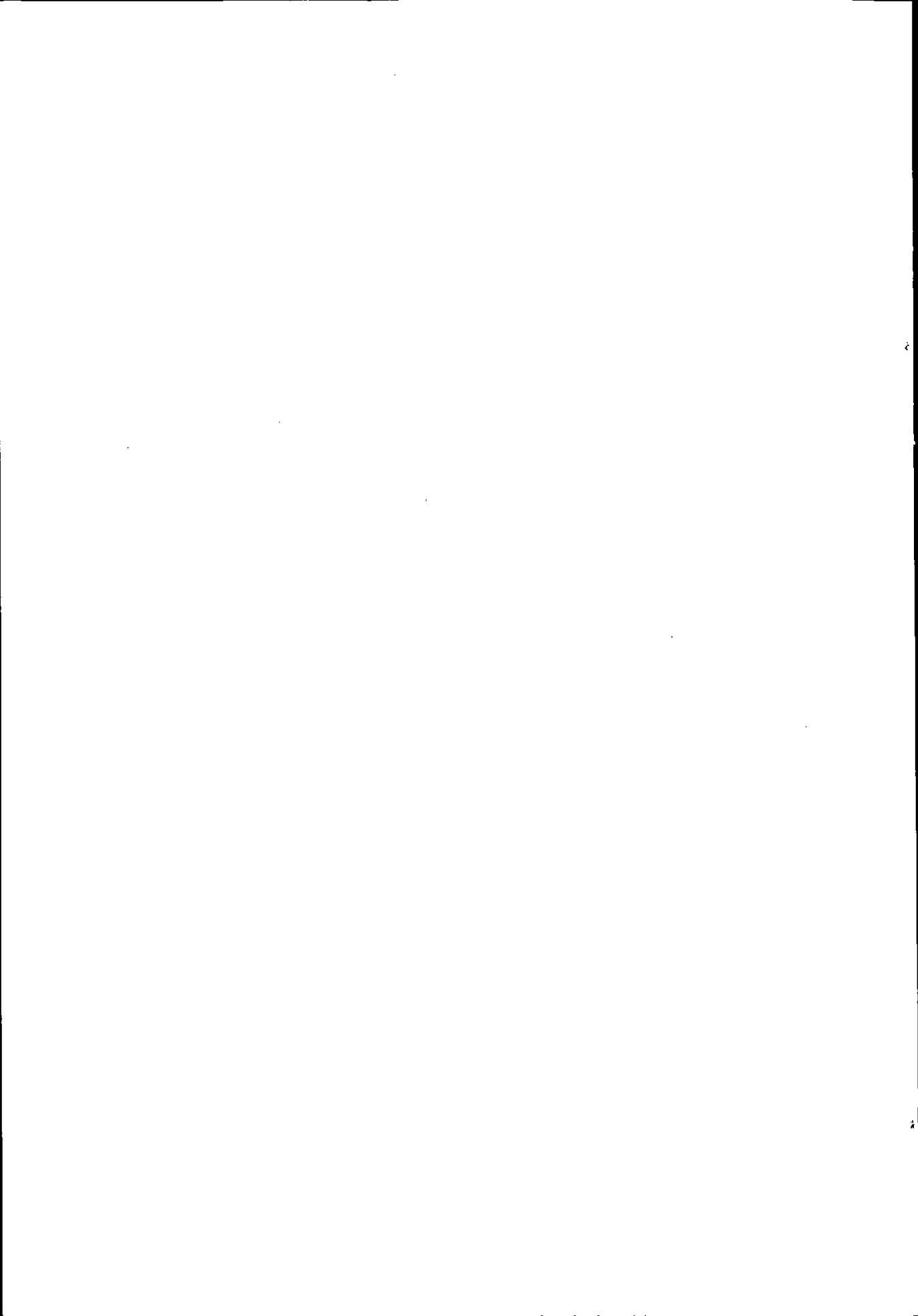
① システム化検討着手検定

システム化の要望内容について、問題の整理状況、問題の解決方向としての、コンピュータ利用の有効性を判断し、更に期間活動計画等を考慮して検討する。

② 構想立案検定

システムの必要性と目的の妥当性、優先順位を検定し、更にシステム目的を表現するために各種案の中から選択されたシステム素案の最適性及び開発順位の合理性に関し、重点的に検定する。併せて今後のシステム開発を効率的に進めていくためにシステム開発推進体制についても検定する。

## 6. ソフトウェア開発計画の今後の課題と将来動向





## 6. ソフトウェア開発計画の今後の課題と将来動向

民間企業におけるソフトウェア開発計画の現状については第2章で詳述したが、本章では、それを基にしてソフトウェア開発計画の今後の課題とその将来動向について述べることにする。

### 6.1 ソフトウェア開発計画の今後の課題

第2章の分析結果及びヒアリング調査結果から、ソフトウェア開発計画の今後の課題としては、大きく次の6つに整理することができる。

- (1) 総合的かつ計画的なソフトウェア開発
- (2) マルチ・プロジェクト・マネージメントの推進
- (3) 要求仕様の明確化
- (4) 外注する場合の基準や手続きの制定
- (5) 各種計画の確実な立案
- (6) 品質に対する意識の向上

以下、それぞれについて述べる。

#### 6.1.1 総合的かつ計画的なシステム開発

ソフトウェア開発計画の最初のステップは、開発するソフトウェアの目的や概要を明らかにすることであるが、それには企業全体としてのシステム化に対する確固たる方針が必要である。場当りの開発は、システム間の整合性を無くしたり、ライフサイクルが短くなったりして結果として無駄な投資になることも多い。今後、INSやニューメディアの発展とともに、本社と支社や工場等を結ぶ広域的な情報システム、OA化、さらにLANなどによる分散システム化がますます進むと思われるが、これらのシステムが有機的に結びついて全体としての効果をより高いものにするためには、長期的な展望のもとに企業の経営方針を加味

した総合的かつ計画的なソフトウェア開発が必要である。

#### 6.1.2 マルチ・プロジェクト・マネージメントの推進

多くの場合、コンピュータ部門を中心として複数のソフトウェア開発が行われるが、どのような企業であっても無限の資源（予算、要員、計算機、等）に恵まれているわけではないので、それぞれのソフトウェア開発プロジェクト間には複雑な関係が生じる。このような場合、マルチ・プロジェクト・マネージメントの考え方を取り入れて、限られた資源で最大の効果が得られるように各プロジェクト間の調整を図る必要がある。

マルチ・プロジェクト・マネージメントとは、次のような目的を持つものである。

- (1) 企業全体としての目標を達成するために全てのソフトウェア開発を完了させる。
- (2) 限られた資源の配分について、適切な意思決定ができるように、長期と短期の2つの立場からみた優先順位を決める。
- (3) 各ソフトウェア開発の作業と定常的な業務を調整し、各要員が支障なく仕事ができるようにする。

この目的を達成するためには、マルチ・プロジェクト・マネージメント担当セクションを作り、次のような業務を行えばよい。

- ① ソフトウェア開発計画の調整
- ② プロジェクトの監視と管理
- ③ プロジェクトの評価
- ④ 要員教育、標準化の推進、他

#### 6.1.3 要求仕様の明確化

前2項は、企業全体としての今後の課題であったが、個々のソフトウェア開発に関しても多くの課題がある。その1つで特に重要と思われる

ものは要求仕様の明確化である。ソフトウェアの開発は、要求仕様に基づいて行われるものであるから、必要かつ十分な要求仕様を定義することは、ソフトウェア開発を成功させる第一歩であるといえる。この点に関して、アンケート調査によると、現在次のような問題が生じている。

- ① システム化に関する検討が不十分
- ② ユーザの意向が反映されない。
- ③ 将来動向の見極めが不十分
- ④ 技法等の活用がなされていない。

要求仕様の明確化に関する具体的な方法については、第3章で詳述したが、要求定義に関する今後の課題としてまとめると次のようになる。

(1) 費用対効果を中心としたシステム化に関する十分な検討

ある業務をシステム化しようとする場合、最も重要なことは、システム化による効果の予測、つまり費用対効果の分析である。表6-1は、効果測定項目体系を示したものであるが、このように種々の観点からシステムの効果について十分な検討を行うことが必要である。

表6・1 効果測定項目体系

区分 影響局面		測定項目*	効果	マイナス効果
			(電算化前に比べて、以下の事例があれば効果である。)	(電算化前に比べて、以下の事例があればマイナス効果である。)
管理局面	業務処理	迅速性	迅速化	遅延化
		正確性	正確化	不正確化
		簡易性	簡素化・標準化	煩雑化
		弾力性	弾力化	硬直化
	データの利用	適時性	向上	低下
		容易性	容易化・高度化	不便(読みにくいなど)
		利用範囲	拡大	縮小
	データの管理	容易性	管理の容易・適切	管理の困難・不適切
		安全性	安全性(秘密保護など)向上	安全性低下
	人事・組織	モラル	勤務環境改善 モラル向上	勤務環境悪化 モラル低下
連携		整備、連携の円滑化	連携の組語など	
コスト局面	-	-	電算化前の事務処理方式を続けた場合の経費と、現在の方式での経費比較	
			経費減少	経費増大

\*この項目の範囲は、「コンピュータ技術へ直接的な影響を及ぼした効果」とする。

出典：「電子計算機利用の効率化に関するガイドライン(II)」行政管理局

## (2) ユーザ部門との十分なコミュニケーション

ソフトウェアの開発はユーザの意向を十分反映させた形で進めなければならないが、アンケートによると、コンピュータ部門が自己の都合のよいように仕様を決めてしまうことが多いようである。要求仕様は、原則的にはユーザ部門が独自に作るべきであるが、コンピュータ部門がそれに関与する場合は、ユーザとの十分なコミュニケーションを図り、ユーザの意向を正しく反映するようにしなければならない。

またこの場合、将来の技術動向や企業戦略といったことも十分検討し、ライフサイクルの長いシステム作りを目指すべきであろう。

## (3) 技法等の積極的な活用

要求仕様をより信頼性の高いものにするためには、第3章で紹介したような技法やツールを積極的に活用していくことも今後の課題といえる。

### 6.1.4 外注する場合の基準や手続きの制定

アンケート調査によれば、何らかの形で外注を利用している企業は77.2%に上っているが、今後、各企業内で開発されるシステムはますます増えることが予想されるので、外注する場合の基準や手続き等の制定は、非常に重要な今後の課題といえよう。

外注に関する基準や手続きとしては次のようなものがある。

- (1) 外注する業務や作業内容の設定
- (2) 外注先の選定基準
- (3) 外注費用の見積り（算定）基準
- (4) 標準契約条項および瑕疵担保責任の分担方式
- (5) 外注会社の善管義務（品質保証基準）や機密保護に関する取り決め事項の制定
- (6) 社内の監督体制、検査体制に関する基準

(7) 品質評価基準

(8) 外注先評価基準

これらの基準や手続き等の中で今後、特に重要になると思われるものは、外注先に対する品質保証の要求である。これは、次に示すようなものであるが、外注する業務が高度化していくことが予想されるので、今後十分に検討しておく必要がある。

- 品質保証に係わる各種計画の提出
- 各開発段階の作業内容と管理要領の提出
- テストの方法や管理要領の提出
- エラーやバグに関する処理要領の提出
- ドキュメントの内容と管理要領の提出
- 品質管理活動要領の提出、他

#### 6.1.5 各種計画の確実な立案

ソフトウェア開発計画の段階において立案される計画については第4章で詳述したが、アンケート調査の結果では、それらの計画が確実に作られているとは言えないようである。工程計画、予算、要員計画、移行計画といった非常に重要な計画であっても、実施率が70%程度であり、少なくともこれらの計画を確実に作ることが今後の課題といえよう。

また、計画立案の際利用できる技法についても、ほとんど利用されていないのが現状であり、やはり今後の課題といえる。特に必要な作業を構造的に把握するためのWBS、それぞれの作業の担当者、チェッカ、承認者等を明確にするTRMについては、ぜひとも利用すべき技法である。

さらに、開発する規模や必要となる費用の見積り方法についても、各企業においてデータを収集し、可能な限り科学的に行うことを考えるべきであろう。

### 6.1.6 品質に対する意識の向上

最近、新規開発は余り無く、既に開発したプログラムの保守作業がほとんどであるような企業も見られるようになった。このような傾向は、今後ますます高まると考えられるが、この保守を容易にするためには、プログラムが保守性を中心として高い品質を有していることが必要である。つまり、正確性や信頼性といったソフトウェアが本来的に必要である品質特性の他に、他人がみても理解し易いプログラムであることが重要となるのである。このような保守性を含めた品質を考えると、品質を高めることは生産性を下げるとする人もいるがこれは誤りであり、高い品質を持つプログラムを作ることが結果的に高い生産性を得ることができると思われるべきである。

## 6.2 ソフトウェア開発計画の将来動向

ソフトウェア開発計画立案の将来を考えてみると図6.1のようになろう。開発の基本計画、つまりシステム化に関する検討は、その時の企業を取り巻く環境、そして企業の戦略によって決まるので、機械化はむずかしいが、要求定義から各種計画を立案することはコンピュータをフルに利用することになろう。このようなシステムは米国のソフトウェア会社で一部実用化されているが、将来的には、我が国においても、パソコン等を利用した計画立案が一般化することが予想される。

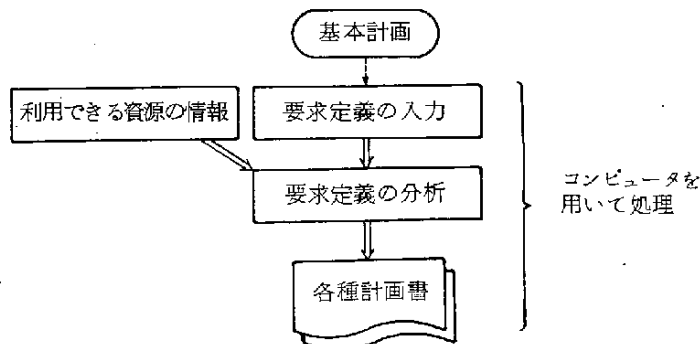


図6・1 将来のソフトウェア開発計画の立案

参 考 文 献

- 1) 「多機能めざす都市銀行の第三次オンライン・システム」  
日経コンピュータ 1983. 12. 12
- 2) 「新世代オフィス・システムの研究」  
日経コンピュータ 1983. 12. 26
- 3) 「情報処理サービス業における管理運用技術の標準化に関する調査研究  
報告書」 (社)日本情報センター協会 (昭和55年)
- 4) 「オンライン需要調査報告書」  
(財)日本情報処理開発協会 (昭和57年)
- 5) 「ユーザの20~40%は、思うように使っていない」 日経コンピ  
ュータ 1983. 11. 14
- 6) 「バックログ問題解決の道を探る」 日経コンピュータ 1982.  
12. 27
- 7) 「ソフトウェア・パッケージ導入の決め手は何か」 日経コンピュータ  
1982. 8. 23
- 8) 高橋延匡 「ソフトウェア生産技術の動向と位置づけ」 電子通信学会  
誌 1983年4月号
- 9) 高村真司 「ソフトウェアの分業について」 電子通信学会誌  
1983年4月号
- 10) 「効率的なシステム開発・運用の促進」 FACOMジャーナル  
1983年-3月号
- 11) 宮本勲・東谷秀夫 「ソフトウェア・エンジニアリングへの道」 共立  
出版 bit Vol. 9. №4
- 12) 第6回ソフトウェア工学国際会議講習会資料」 (社)情報処理学会  
1982年9月
- 13) 「情報処理サービス業の標準化に関する報告書」 (財)日本情報センター

協会（昭和56年5月）

- 14) 前川良博 「EDPシステム設計ハンドブック」 日刊工業新聞社
- 15) 通商産業省機械情報産業局 「ソフトウェア・プロダクト年鑑—プログラム調査簿—」 (株)ソフトウェア産業振興協会 ソフトウェア流通促進センター 1982年版
- 16) 占部都美 「経営意思決定のためのコンピュータ活用ハンドブック」 (株)中央経済社
- 17) 関根智明 「PERT・CPM入門」 (株)日本科学技術連盟
- 18) 「情報システムのユーザズ・ガイド(II)—プロジェクト管理—」 (株)日本情報処理開発協会
- 19) 「要員派遣サービス(FMサービス)利用の現状と動向に関する調査研究およびソフトウェア開発の現状とソフトウェア・プロダクトの活用に関する調査研究報告書」 (株)日本データ・プロセッシング協会
- 20) 南修優 「EDPシステム設計入門」 オーム社 昭和57年12月10日
- 21) 水野滋監修 「新QC七つ道具」 日科技連 昭和59年1月21日
- 22) 「FACOMシステム分析と設計」 UASOOLIN\*1
- 23) 大塚純一 「フロチャート入門」 日本能率協会 昭和51年9月30日
- 24) 国井利泰 「ソフトウェア・ツール」 共立出版 昭和57年2月15日
- 25) 「事務管理ハンドブック」 産業能率短期大学
- 26) 各省庁電子計算機利用効率化共同研究会 「ソフトウェア開発の効率化研究結果報告書(昭和55年度)」 行政管理庁 昭和56年3月
- 27) 「56年度技術研修資料プロジェクト」 (株)ソフトウェア産業振興協会
- 28) 「電子計算機利用の効率化に関するガイドライン(I)」 行政管理庁



—— 禁無断転載 ——

昭和 59 年 3 月 発行

発行所 財団法人 日本情報処理開発協会  
東京都港区芝公園3丁目5番8号  
機械振興会館内  
Tel (434)8211 (代表)

印刷所 株式会社 タケミ印刷  
東京都千代田区神田司町2-16

