

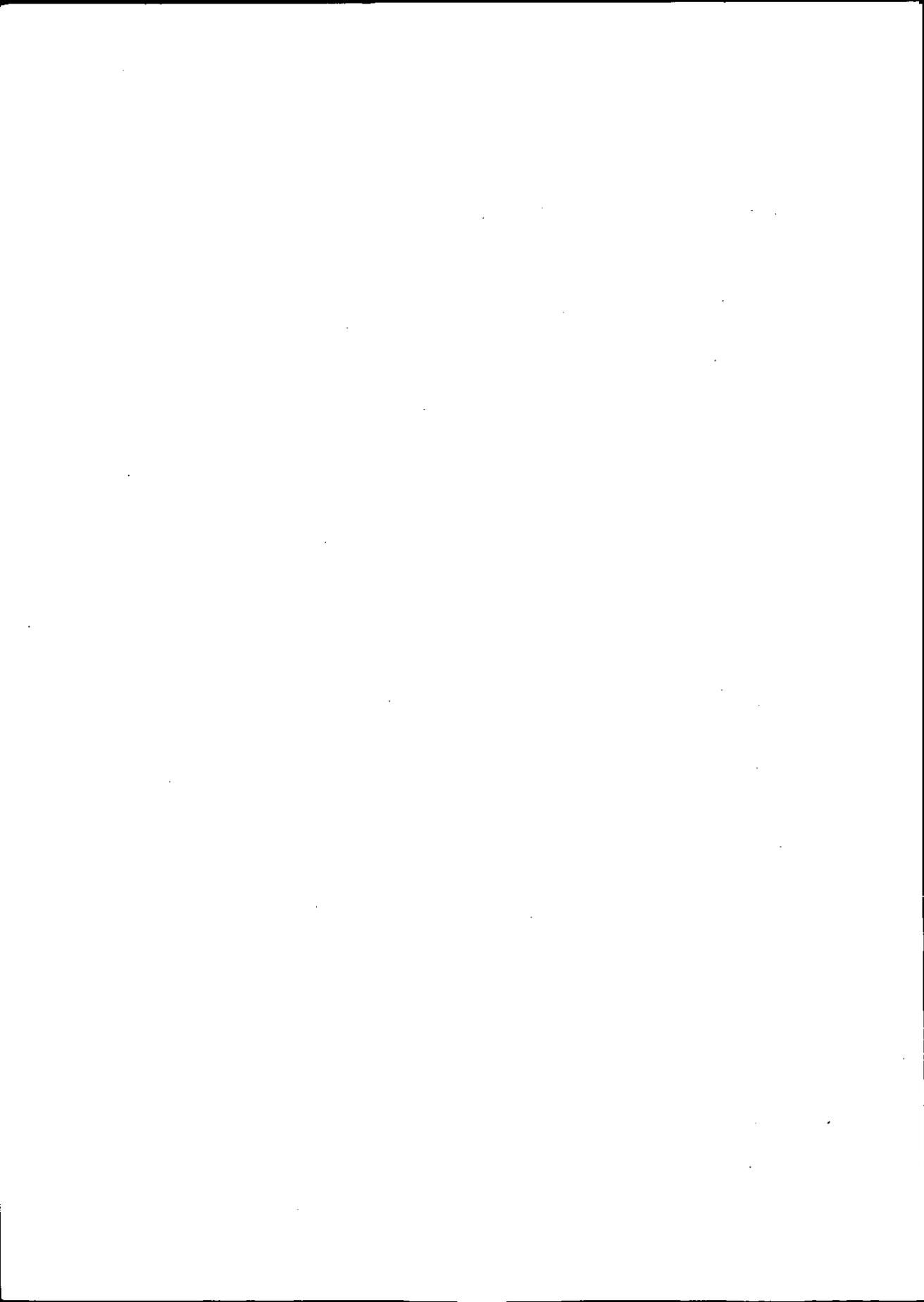
プロジェクト管理の手法 (I)

—渡米システムズ・アナリシス研修団テキスト—

昭和46年6月

財団法人 日本経営情報開発協会





I 序 (Introduction)

いかなる組織においても、データ処理部門の基本的な使命は、マネジメントとその組織体の効率を高め、有効性を発揮する手助けとなるようなシステムと情報処理に関するテクニックを応用することである。

この使命を成功させるには、開発活動が、通常、ある一定期間内に達成できるような特定の目標（例えば、在庫回転率の向上）をもったプロジェクト形式で組織される。不幸なことには、あるプロジェクトの目標を達成しようという試みに際して、データ処理システムの設計者は、他の領域でみられるのとまったく同じような、弱体なマネジメントのやりかたというあやまちを犯しがちである。他の領域でのマネジメントの弱みを補ったり代行したりするためにこそ、データ処理システムが設計されるというのに……。プロジェクトの完成期日がのびのびになったり、予算をオーバーしたり、あるいは製品（すなわちシステム）の品質は一般に期待以下のものであったりする。

同じ企業の別の領域において、このような不幸な事態がおこったら、データ処理に携さわる人々は次のように言うことであろう。“彼等に必要なのはシステムであり、コントロールである。あるいは、例外による管理であり、総合的な経営情報システムである、”と。

データ処理の専門家が経営情報システムをつくれるという信頼を受けるためには、彼等はまず第一にデータ処理プロジェクトのマネジメントに、マネジメントの原則を適用しなければならない。

A アプローチ (Approach)

このコースの目的は、システムとプログラミングのプロジェクトをコントロールするための有用かつ秩序立った方法を提供することである。このことは次のようにしてなされる。

1. プロジェクト・コントロール・システムの各構成要素の定義

2. プロジェクト・コントロール・システム確立のために必要な組織の条件の規定。
3. プロジェクトの組織化と分割の原理に関する検討。
4. 質を確保する為にコントロールする必要のある諸要因の確認。
5. リソースの効果的な使用の為にコントロールしなければならない諸要因の確認。
6. 特定の見積り及びスケジューリングのテクニックに関する記述。
7. プロジェクト・コントロールの実施に関するアイデアの探索。

B 領域 (Scope)

このコースではビジネス・データ処理のアプリケーションのためのシステムとプログラムを開発することに主眼を置き、そうした分野での例を数多く選んである。全く同様な基本原則が、数学、工学および物理学などでのアプリケーションにも成立することは言うまでもない。

データ処理部門の全要員の活動のコントロールに加えて、ユーザー側組織の役割がシステムの開発における重要な関心事と見なされ、これがプロジェクト・コントロール・システムにおける適切なる主題となる。

C コースの構成 (Course Organization)

このコースは (ⅡからⅤまで) 5つのセクションから構成されている。セクションⅠはコースの目的と領域と構成を述べた簡単な序からなっている。

セクションⅡは、プロジェクト・コントロール・システムの要素の定義とそのシステム確立のための組織的な要件の明確化、およびそのシステムが実際に働く環境について述べてある。

セクションⅢでは、プロジェクトの組織化及び分割の原則及び種々のタイプのプロジェクトに要求される典型的なコントロール・ポイントについ

て述べてある。

セクションⅣでは、システムとプログラムにおいて望まれる品質を確保するためにコントロールの必要な諸要因およびそのようなコントロールのなされるチェック・ポイントを確認している。

セクションⅤでは割当てられたりソースの効果的な使用のためにコントロールの必要な諸要因およびそのようなコントロールのなされるチェックポイントを確認している。ここでの議論には、見積りやスケジューリングや予算化等に関するトピックスが含まれている。

セクションⅥはある特定のインストラクション内でプロジェクト・コントロール・システムを導入するさいのアプローチについて述べている。

Ⅱ 定義及び環境 (Definition and Environment)

A 定義 (Definition)

プロジェクト・コントロール・システムとは、マネジメントに次のような能力を与える、標準的な手続きと実務の大成を言う。即ち、

—— ある特定の結果を得るために続けられているタスクの実行状況に対してひんばんに評価をする能力。及び、

—— 進行のおくれや、プロジェクトの優先度の変更などによって必要とされる修正処置をとる能力。

プロジェクトに、なにか思わしくない事態が起っているということを示すだけでは、コントロール・システムとして十分ではない。システムは十分な正確さとすばやさとして、不都合を指摘し、有効なマネジメントを行なえるようにしなければならない。

B 構成要素 (Components)

プロジェクト・コントロール・システムの構成要素とは：

1. プロジェクト・コントロール・マニュアルあるいは、ハンドブックとして記載された手続きの集大成。
2. 報告とスケジューリングに用いられる書式。
3. このコントロール・システムの使用に関してよく訓練されたプロジェクト・リーダー及びコントロール・オフィサー。

プロジェクト・コントロール・マニュアル(しばしば、1つのインストラクションのスタンダード・マニュアルの一部分となる)は次のような各章から構成されるだろう：

プロジェクト組織。

プロジェクトのタイプの定義。

プロジェクト開始の手続き。

プロジェクトのモニタリングの手続き。

これは1つの目次例である。

各章には、推奨される手続きが記載され、今後作り出されるべきコントロール用ドキュメントが規定されていなければならない。システムで用いられる書式には、要員の所要時間報告、ガント・チャート及び標準チェックリストが含まれている。

プロジェクトの管理及び評価にたずさわる責任者は全員、このコントロール・システムの正しい使用方法を修得していなければならない。

C 便 益 (Benefits)

プロジェクト・コントロール・システムを確立することにより、数多くの便益が得られる。その便益とは、

1. マネジメントが、プロジェクトの真の状態を知ることが可能であり、その結果、リソース配分の調整がすばやく効果的に行なわれる。
2. 開発の各段階の終了ごとにスタンダードな要件を定義することにより、そのインストラクションにおいて開発されるシステムやプログラムの質を高める。
3. プロジェクト・リソース追加の要求をうらづける文書が用意されている。
4. 他のプロジェクトに充当可能な未使用の能力を知ることができる。
5. プロジェクト・リーダーに、彼等の職能経験の初期の段階ですぐれたコントロールのやりかたを修得させることにより、経営手腕を伸ばす助けとなる。

D 前提条件 (Prerequisites)

プロジェクト・コントロール・システムを開発し、実施するためには適切な組織状況が確立されていなければならない。最も重要な要素として

システムの開発の使用に対するマネジメントの関与、新しいアプリケーションの実施に関する全体計画、システムおよびプログラミング業務についての方法の標準、およびリソース使用状況のデータ収集に関する統制手続き等が上げられる。

1. 関与（態度表明）（Commitment）

データ処理部門のマネジメント及びその報告を受ける重役会は、プロジェクト・コントロールの改良について組織体の関与を言明しなければならない。この関与は、コントロール・システムの開発に必要な要員の配置に支障がないことを含む。

2. アプリケーション計画（Applications Plan）

アプリケーション導入についての全体的インストラクション計画は次のものを含む：

- a 予想される新しいアプリケーションの一覧表。
- b 導入までの一時的スケジュール。
- c メインテナンス及びその他の継続的努力を必要とする、すでに導入されたアプリケーションの一覧表。
- d ある処理形式から他への転換（例えばバッチ処理からオンライン処理へ）を必要とするアプリケーションの一覧表。
- e 修正処置を必要とするアプリケーションの分析。

この計画は、現在のインストラクションにおける継続的活動の業務量および緊急の場合及び一時的な要求にそなえるリソースの量を予測するのに使用されることもある。

3. 方法の標準（Methods Standards）

方法の標準はシステム開発部門において従われるべき共通のやりかたを用意し、それによってコントロール手続きを共通の基盤に立って開発するのを可能にする。このような標準は、一般には、システムおよびプログラミングのドキュメンテーションの内容、開発活動の行なわれる順

序、及びそれらの開発が実行される流儀等に主眼を置いている。

4. 統制面の状況 (Administrative Environment)

統制面での状況として、その実施計画内で実行されるすべてのジョブの分類及びすべてのプロジェクトにみられる共通のタスクの正式な定義が行なわれていなければならない。正確な記録手続きの整備と、プロジェクトの完了度をあらわす統計数字の作成とは、現在行なわれているプロジェクトのコントロールのためだけでなく、将来のプロジェクトを見積るためのデータを蓄積する基盤ともなり得る。

E プロジェクト・ミックス及びコントロール特性 (Project Mix and Control Characteristics)

ある組織内でのプロジェクトはその目的、その開発時期、およびその使用方法等によって分類されているだろう。よって質、時間、及びリソースのコントロールの統制要因は、プロジェクトのタイプと関連づけられることになる。プロジェクトの各タイプごとに、制約となる要素と融通のきく要素、あるいは最も重要なコントロール問題を引き起こしている要素などの分析例が示される。いったん、組織のプロジェクト・ミックスがわかれば、こうしたチャートあるいは同様の分析法を用いて、プロジェクト・コントロール・システムにおいて強調されねばならない統制要素を確認することが可能となる。

Ⅱ プロジェクト組織 (Project Organization)

A 要約 (Summary)

どんな型のプロジェクトを組織化するにも、そのプロジェクトをはっきりしたフェーズに分けることが望ましい。こうすることによって、各フェーズ内での小さな活動単位を監視する能力がマネジメントに与えられる。同時に、あるフェーズから、次のフェーズへの切換えや移行をするのに必要なコントロールを最少限度に止めることができる。

プロジェクト組織に対する分割的アプローチはマネジメントに次のような機会を与える。

1. プロジェクトの各フェーズすなわち分割に対する詳細なプランニングを要求する。
2. 見積りに対するタイムリーかつ秩序だった訂正を入手する。
3. 各フェーズの終了時に必要なドキュメンテーションの作成を強化する。
4. フェーズごとの査閲の責任を明確化する。

典型的な開発プロジェクトの組織化は下記の7つの主なフェーズに分類される。

1. 開始とプランニング
2. システム・スタディ
3. システムズ・アナリシス
4. システム設計
5. プログラミング
6. テスト
7. 実施・導入

各フェイズの完了時点は、そのプロジェクトのコントロール・ポイントとみることができ、そこで特定のドキュメンテーションの作成とマネジメントの処置とがとられる。プロジェクト組織をもう少し詳しく記述すると

下記のようになる。

1. 開始とプランニング (Intention and Planning)

選 定

承 認

予備調査

プロジェクト・プランの開発

見積り

スケジューリング

予算化

2. システム・スタディー (System Study)

データ収集

インタビュー

現システムのフローチャート化

現システムのドキュメンテーション化

3. システム・アナリシス (System Analysis)

機械的要件の定式化

書式のアナリシス

ファイルのアナリシス

レポートとその用途のアナリシス

4. システム設計 (System Design)

ファイル設計

書式及びレポート設計

プログラム仕様書の作成

事務手続きの作成

システム・マニュアルの作成

5. プログラミング (Programming)

ロジック作成

コーディング

机上チェック

プログラム・テスト

プログラム・ドキュメンテーション

6. テスト (Testing)

システム・テスト

ボリューム・(又はパラレル・)テスト

事務手続きのテスト

7. 実施・導入 (Installation)

移行計画

移行

導入後の監査

このように詳細なプロジェクトの組織化は、より頻繁なコントロールの機会を与え、長期にわたる開発プロジェクトに適したものである。

以上の主要フェーズでの重要な関心事と問題点は下記の通りである。

B プロジェクト開始 (Project Initiation)

プロジェクトの目標達成のさいに直面する多くの問題はそのプロジェクトの発端から内在している不完全性にしばしば原因が帰せられるようである。

プロジェクトの開始にあたっての不適切な準備から生ずる問題は、システムに対する不適切な定義から生ずる問題と互に密接な関連があるけれども、異なった問題である。

不適切なシステムの定義は、通常、データ・ベースの明細が不完全なことによるか、システムに対する例外的な要求に対する考慮が不十分なことから来る結果である。いっぽう、不適切なプロジェクトの定義は、ユーザーとデータ処理組織とが下記の事項についての同意に達していないこと

から来る。

(イ) そのプロジェクトの目的（例えば、コストダウン、サービスの向上等）

(ロ) そのプロジェクトの及ぶ範囲

(ハ) 両グループがそのプロジェクトに投入すべきリソース（経営者の能力も含む）の水準

このように、プロジェクトの選定と承認のプロセスは、効果的なマネジメントとコントロールを行なうための基礎となるのである。最初の段階はどのような種類のプロジェクトが予定されているかを認識することである。そのためには、ユーザーの希望（又は要求又は黙諾）を十分に聞き出して分析し、何をユーザーが望んでいるか、また、その要求に応えるにはどの程度の問題が生ずるかを把握しなければならない。聞き出すべきことではちばん大切なのは、ユーザーの要求に対して現存するデータやプログラムがそのまま使用出来るか否か、あるいはそれらを新たに開発しなければならないか否かである。たとえ現存するデータやプログラムを使用出来たとしても、システムのメンテナンスもしくは増強といった活動がシステム再設計という大きな仕事へと発展していく可能性について注意していなければならない。

もし、新しいシステムの開発が指示された場合に、そのアプリケーションの領域がどこまで深く及ぶべきかが基本的な質問事項である。

1. 現在システムを単純に高速の装置へ移行させるか？
2. 現在の書式とレポートをそのまま継続するか？
3. 手続上の問題は再検討及び変更の対象となるか？
4. マネジメントの方法、慣習、実務等は再検討及び変更の対象となるか？

このようなことを考慮することは、そのプロジェクトに何を含ませるべきか、そしてこの対象領域のうちでどこを将来のプロジェクトに残してお

くべきか、といった質問に答える手助けとなる。

0 プロジェクト承認 (Project Authorization)

プロジェクトを選択・承認する経営者 (executive) の権限は、そのプロジェクトに関してデータ処理部門とユーザー部門の両方で必要とする支出額の承認にまで及ぶべきである。また、その経営者はそれら両者をプロジェクト計画に従わせるだけの権限をもつべきである。

経営者はある程度の事前見積りと、その見積りに置いてよい信頼度の表示とを与えられていなければならない。それがあってはじめて、プロジェクト全体をここで承認するか、もっと確実な見積りが作成されてからプロジェクトを段階的に承認するか の決定が行なわれる。プロジェクト承認記述書のアウトラインを例示する。

D プロジェクト・プランニング (Project Planning)

プランニングは、そのプロジェクトの規模にかかわらず必須の、明確な仕事である。プロジェクト・プランニングは下記の事項を含む。

- a 行なうべきタスクを詳細に記述する。
 - b プロジェクトのチェック・ポイントを設定する。
 - c 各々のタスクの実行に必要なスキルを指示する。
 - d 適切な要員を各々のタスクして割り当てる。
 - e 各々のタスクの終了までの時間を見積る。
 - f タスクをスケジュール化する。
 - g プロジェクト予算を作成する。
1. タスクの詳細化のアウトライン (Detailed Task Outline)

プロジェクトの開始段階では、行なわれる主なタスク (書式の設計、インタビューの実施、など) を示した、一般的・予備的なタスクの概要が作成された。ここでは、これからのプロジェクト・プランニングの

助けとなるべく、詳細なタスクの記述書が作成される。例えば、単に“インタビューの実施”といわれていた仕事は、ここで、インタビューのさいに取り上げるべき問題点を述べたインタビュー・ガイドを作成すること、インタビューを行なうこと、見直しをすること、インタビューの要約をつくること、といった部分タスクに分割されるだろう。

こうした詳細なタスクのリストを作成することによって、見積りのためのより精密な基礎が確立し、それぞれ異なったスキルの必要性が浮び上がってくる。

また、詳細なタスクのリストはまた、各々の部分タスクに責任を持つべき組織単位を明らかにすることが出来る。プロジェクト・マネジメントの責任チャートも示される。

2. プロジェクト・チェックポイント (Project Checkpoint)

プロジェクト・チェックポイントは、プロジェクトのあるフェーズもしくはタスクから次に移行するところすべてについて設けるべきである。例えば、システム設計の詳細化からプログラミングに移行する段階ではチェックポイントを必要とする。プログラムからオペレーションに移行する場合も同じことである。

チェックポイントを設ける目的は、その時点までに行なわれたタスクがある一定の水準に達しており、そのインストラクションのスタンダードにのっとっていることを立証することである。

もうひとつのタイプのチェックポイントとして状況変異(contingency)に対するチェックポイントがある。プロジェクト中のいくつかのタスクがクリティカルであることが確認されているかも知れないし、別のプロジェクトでそのタスクについて困難性があったという経験があるかも知れない。

このようなタスクは、プロジェクトの初期に“困難さ”の予測、すなわち状況変異計画を行なうことによって強調されるであろう。こうした

要注意のタスクについてはそのタスクの開始直後、および実行中もある一定の間隔で常に再検討が行なわれるであろう。こうしたタスクのプランニングは再検討の段階であらわになる問題点の予測と、そこで可能な修正処置を予測することを含んでいる。

3. 必要なスキルの指示と要員の割り当て (Designation of Required Skills and Assignment of Personnel)

各々のタスクを実行するのに必要なスキルと経験とは要員割り当ての手引きとなるように明確に記述されねばならない。タスクに必要な典型的なスキルは次記の通りである。

a 調査のスキル (Investigative skills)

データ収集やシステム・テストに必要である。

b コミュニケーションのスキル (Communication skills)

進捗報告、ドキュメンテーションの準備、及びシステム・コンセプトのユーザーへの売込みに必要である。

c 技術面でのスキル (Technical skills)

プログラミングに必要である。

d 分析のスキル (Analytical skills)

システムの要件を定式化するのに必要である。

e 創造的スキル (Creative skills)

システムの詳細設計やプログラミング・ロジックの作成活動に必要である。

各々のタスクが、特定の個人ではなく、特定スキルの組み合わせを要求するとして表現することによって、要員の活用における弾力性が保持されることになる。たとえば、ユーザー部門の要員はある種のデータ収集を行なって、かつシステム・テスト結果の吟味を援助しうるし、デバッグに熟達したプログラマーはみずから幾つかのプログラムを作成しながら、全てのプログラムの机上チェックやデバックを援助すること

ができる。

この様な調整を行なうためには、データ処理部門のマネジメントが、ユーザー部門や運用部門の要員が保有しているスキルについても概要を把握し得ることが必要である。このことは通常非公式に行なわれているが、大きな実施計画の場合は、通常行なわれる個人評価の周期ごとにアップデートされる公式のスキルズ・インベントリーを確立しておくことが望ましい。

スキルズ・インベントリーは、詳細タスクのアウトラインとあいまって現実的なスケジュールを立てるのを助けるタスク対スキルの充当状況 (availability) 分析に用いられる。

もちろん、もっとも重要なスキルはプロジェクト・マネジメントのスキルである。とくに重要な3つの要素は次の通りである。

- a 監督者としての経験と能力
- b そのプロジェクトに使用される機器でのデータ処理経験
- c そのアプリケーションについての知識

プロジェクト・マネージャの候補者を、この各要素について強弱の重み付けをして、次表の如く適性についての評価をすることができよう。

	1	2	3	4	5	6	7	8
監督者経験	W	W	W	S	W	S	S	S
機器経験	W	S	W	W	S	S	W	S
アプリケーション知識	W	W	S	W	S	W	S	S

W : 弱 S : 強

この、大ざっぱな評価方式によってはっきり言えることは、1番(全ての面で弱い)はプロジェクト・リーダーとして不適格であり、8番(全ての面で強い)は理想的であるということだけである。この両極端の間で、マネジメントはある特定のプロジェクトのプロジェクト・リー

ダーにとってどれが最も重量な要因かを定める必要があるし、もっと大切なのは、どのようなマネジメントの支持と追加の訓練とを短期間のうちに提供し得るかを定めることが必要である。このようなマネジメントの支持がないとすれば、プロジェクト・リーダーの弱点がそのままスケジュールを乱す要因となる。

4. タスクの完了時間の見積り (Estimating Task Completion Time)

見積りの1つは与えられたタスクの遂行に必要な時間数の予測値である。とうぜん、よい見積りを行なうためにはタスクを構成するアクティビティを知っておく必要がある。

従って、1つのプロジェクト開発期間中にも、幾つかのレベルでの見積りを作成する必要がある。システム開発アクティビティの予備見積りはシステム・スタディの段階が完了したときにより詳細にわたる見積りになるであろう。プログラミング作業については、プログラミングの詳細記述がはっきりするまではたしかに見積りは不可能である。

プログラミング作業に関する当初の見積りはシステム・デザインが完了した時点で見直されねばならない。間近なものとなり、しかし完全に詳細のわかったタスクに対してだけ見積りの正確さに信を置くというやり方によって、見積りの手続きの経済性を保持するのが、一般的に言って、プロジェクト・リーダーの責任である。見積りのための幾つかの基本ルールは次の通りであろう。

- a タスク遂行のための時間を見積る前に可能な限りタスクの詳細を知ること。
- b 前提となる諸タスクが完了する迄は、そのタスクの詳細見積りを留保すること。
- c 控え目であること。

見積りのための3通りのやりかたがとられる。それは「実績値」、

「直観値」及び「標準値」である。

標準値によるやりかたは、標準の設定に十分な程度の頻度で実施されるタスクに対して推奨される方法である。方法の標準は作業のやりかたを規定するものであり、パフォーマンス標準はタスクの完了に必要な時間を規定するものである。最終的な見積りの値はタスクに割り当てられた個人の能力によって異なる。

実績値によるやりかたは現今のプロジェクトを過去に実施した類似のプロジェクトと比較することを含んでいる。プロジェクト同士の相異点に着目しつつ過去のプロジェクトに費消された実時間を調整し、それを新プロジェクトでの見積りとして使用する。標準がない場合でも、類似のプロジェクトがあり、しかも時間記録が整備されているときには、実績値によるやりかたによって信頼できる見積りを行なうことが出来る。

直観値によるやりかたでは見積りを行なうのに、実績・標準・経験を結合したある個人の能力に依存する必要がある。直観的に正しく推定し得る人間が居ない時はこのやりかたはいろいろと欠陥がある。

一般的に、経験値によるやりかたはシステム・タスクに適して居り、一方標準値によるやりかたはプログラミング・タスクに有効である。

5. システム開発での見積りのやりかた (An Approach to Estimating Systems Development)

システム開発のプロセスは他の研究開発活動と似ている面が多い。しかしながら、システム部門は通常、そうした伝統的な研究開発活動にくらべてより早く、より精密かつ正確な予算の見積りを必要とされる組織上の立場にある。同時に、見積りを行なうのに十分な時間がない場合が多い。

プロジェクトの初期段階において迅速かつ容易に用いることができ、しかも有用で現実的な見積りを与えてくれる方法を持っているのは有用である。このコースで提出されるアプローチでは、システム開発のため

に必要なリソースに影響する2つの主要な要因について考慮する。

- a システムが開発される業務の広さと複雑さ。
- b アナリストが遂行すべきスタディ、アナリシス、設計といった機能のタイプと複雑さ。

- a システムの広さと複雑さの見積り (Estimating System Breadth and Complexity)

どのビジネス・データ処理システムにも、それを特色づける共通の標準的システム機能がある。(数学的あるいは技術的なアプリケーションについても、同じ様なリストを構成することが出来る)。標準的機能というのは次の様なものである。

- 1.) データ収集
- 2.) データ編集
- 3.) エラーの修正
- 4.) データの順序付け
- 5.) 照 合
- 6.) 計 算
- 7.) データの転送
- 8.) マスター・親データの参照
- 9.) ファイルのアップデート
- 10.) 問合せへの応答
- 11.) データの要約
- 12.) レポートの準備
- 13.) プリント
- 14.) レポート・データの配布

これらの各機能に対してそれぞれいくつかの質問が出てくるがそれへの回答がシステム全体の複雑さの概要を与えてくれることになる。

こうした質問の殆んどは数字で、もしくはイエスかノーかの応答で回答できるものである。かくして、システムの相対的な複雑性を測定しシステム、ファイル、レポートそれぞれの概要 (Profile) を記述するための単純な方法が与えられたことになる。

b) アナリストの機能の複雑性の見積り (Estimating Complexity of Analyst Functions)

通常システムズ、アナリストによって遂行される機能は次の様に類別することが出来る。

1.) スタディ段階 (Study Phase)

- a) 業務統計や文書を集める。
- b) インタビューを行なう。
- c) プロセスを観察する。
- d) ファイルやレポートのサンプルを集める。
- e) 現状のシステムを記述する。

2.) アナリシス段階 (Analysis Phase)

- a) 個々のドキュメントやファイルを分析する。
- b) 情報の用途を関連づける。
- c) 機能面からの要求を明らかにする。
- d) 設計オルタナティブをつくり出す。

3.) デザイン段階 (Design Phase)

- a) 新しいファイルを設計する。
- b) 新しいアウトプットを設計する。
- c) 新しいイン・プットを設計する。
- d) 新しい手続きを設計する。
- e) 新しいプログラム明細を設計する。

さきのステップで行なわれたシステムの複雑性とその叙述はアナリストの機能に関連づけられる。例えば現状システムにおいて50のデ

ータ収集点があることを知った場合、アナリストはおそらく50点全部についての業務統計を集めねばならず、また彼のプロジェクト計画期間中にそれら全部へのインタビューと訓練活動を含めねばならないと想定されるであろう。従ってこの突き合わせの結果は、システム開発プロジェクトのタスクについて詳細計画を行なう上で有効である。

もうひとつ強調すべき点がある。システム開発が進行するにつれて当初考慮した以外のシステムの特徴が遂次明らかになってきて、はじめの見積りを変えねばならなくなることである。従って後続のタスクに対する見積りを改訂出来るようにコントロール・ポイントを設定しておく必要がある。

システムの複雑性を記述する主要な質問項目の一覧表が提示され、アナリストの機能と関連付けられる。

6. プログラミング開発の見積り (Estimating Programming Development)

プログラミング開発の見積りの際の第一の問題は、必要とされる命令の数ないしはコーディングのページ数を確認することである。その大きさは実行される機能の数に応じて変るであろう。

第二の問題は、コーディングの各ユニットを作成しテストするのに必要とされる時間数を求めることである。これはプログラムの複雑さとプログラマーの実質上のスキルと経験によって変るであろう。

プログラムの大きさを見積るための一つの方法は、プログラム仕様書に述べられている各機能を調べ、それをかみくだいて、必要とされる命令群 (instruction sequence) および含まれるデータ・フィールドの近似的な数を示すことである。

命令群は、一般に次のように分類できる。

インプット/アウトプット

計 算

データの転送

論理又は比較

宣言

テーブル操作

編集

それぞれのコンピュータについて、上述の各命令群を実行するのに、ある特定のコーディング行数が標準的に必要とされる、ということが知られている。これらのことは、そのインストラクションに対するリファレンス・テーブルに述べられているであろう。

したがって、たとえば“マスター・インプットからプリントの第13行目を構成せよ”と読めるステートメントは、もしプリントの第13行目に10個のフィールドが置かれ、かつ1回の転送に平均2行のコーディングが必要とされるならば、20の命令が必要であると見積ることができよう。

プログラムの複雑性は、

- プログラムで扱われるべきファイルの数、
- 全プログラム・コーディングに対する論理的コーディングの比率、
- 要求されている計算の長さと複雑性、ならびに、
- オブジェクト時の効率に対する要求、

などによって異なるであろう。

ある所定の大きさと、所定のレベルの複雑性を持つプログラムを完成するために必要とされる時間数について、そのインストラクションでのスタンダードが設定されることが望ましい。このような標準は、ある期間にわたって書かれた各プログラムの大きさ、その複雑さの程度およびプログラムに費される時間についての記録をとることによって、作成されるであろう。標準に関するその他のよりどころとしては、D.H. Brandon らによって出版された資料がある。

標準が作成されていても、プログラム見積りの開始の時点で用いられ個々のプログラマーの経験に基いてある変動するパーセンテージだけ修正されるものである。

7. プロジェクト・スケジューリング (Project Scheduling)

見積りによって、あるジョブを実施するのに必要な実際の時間数が予測される。スケジューリング業務では、要員その他のリソースの利用可能性、および管理的、その他のプロジェクト以外の仕事に失われる時間を考慮に入れる。それからプロジェクト・タスクの各々に対する目標期日を示したタイムテーブルが作られる。

ガントチャート又はPERT/CPMチャートが、スケジューリングを図で示すのに用いられるだろう。PERT/CPMはプロジェクト内のイベントの相互依存性を示すのに特に有用である。若干のデータ処理のユーザーでは、クリティカル・パス上にあるか、関連するプロジェクトからのアクティビティによって、時間が制約されているようなアクティビティ全てに対してスペシャル・レポートの準備を要求できるようにプロセスを改善している。

一般化されたPERT計画が論じられる。

8. プロジェクト予算編成 (Project Budgeting)

予算はドル又は時間の単位によって、プロジェクト計画を数量化したものである。その目的は、プロジェクトに投入されるリソースの使用を監視するために、ある集約的な手段を提供することである。プロジェクト評価に真に有用な道具を提供するためには、予算とタスクの進行とが関連づけられなければならない。標準コストは、プロジェクトに投入されるすべてのレベルの要員について設定されなければならない。時間超過作業に対しては、別個の賃率が設定されるべきである。

E システム・スタディ、設計および導入 (Systems Study, Design and Implementation)

プロジェクト遂行活動は、プロジェクト中に指定された時点における計画的レビューによって一部分管理される。これらのレビュー機能は、品質とリソースの管理に関連づけて、セクションIVおよびセクションVにおいて論じられる。

管理のこうした要素を実行するということは、特定の移行チェックポイント (チェックポイント・レビュー・ガイド参照) において、プロジェクト活動をレビューして、タスクが実施計画の標準とプロジェクトの標準に従って完了したことを確認することを意味している。クリティカルな活動に対しては、タスクの開始直後およびタスクの進行中の予め定められた時点において、レビューがなされる。さらに、プロジェクトに対してリソースの継続的配分を保証し、かつ予期しない厄介ごとを発見するために、一定の時間間隔 (月毎、週毎) にレビューが行なわれる。

大抵のプロジェクトにおいて、ある種の危険が出現するが、それらの主なものとしては、プロジェクト目標の変更、特別な要求、およびプロジェクト・リソースの浪費などがある。

もし、ユーザーがプロジェクト・チームと密接な関係をもって作業していて、その関係に満足していたとしてもユーザーが彼の学習曲線の向上につれて、システムに関する要求を増すような傾向が強くなってくるであろう。このような要求は、システムに対する変更と追加を求める要求書の形式をとるであろう。また、同様な問題は、プロジェクト・チームが蓄積しつつあるマスター・データの要約や報告書に対する特別の要求書という形でおこってくる。

それぞれの変更あるいは特別の要求ごとに個別に見積りをし、その要求を満足させるためのスケジュールを設定し、特別な要求のおかげで当初のプロジェクトに予想される遅れの説明を、ユーザー部門およびデータ

処理部門のマネジメントに提示することは、プロジェクト・リーダーに課せられた義務である。

プロジェクト・リーダーの下にいるプログラマーとアナリストは、変更と特別の要求とをすべてプロジェクト・リーダーに報告することを要求され、適当な権限なくして、それらを実行することを禁じられていなければならない。

Ⅳ 品質管理 (Quality Control)

品質管理の手続は、次の事を保証するように設計される。

- システム又はプログラムは、明細書に従って準備されたものであり、かつ期待される結果をもたらすこと、
- ドキュメンテーションは、システムの適切なオペレーションを可能にするのに十分であること、
- システム又はプログラムに使用されるテクニックは、メンテナンスと追加開発を可能にするような形式で、ドキュメント化されていること、ならびに、
- システム又はプログラムは、割り当てられたリソースを能率的かつ無駄なく利用していること。

基礎として提供されている21のチェックポイントを用いて、品質管理は次のような時点で実行することができる：

- プロジェクトの選択
- システム・スタディ——第1フェイズ
- スタディの完了
- アナリシスの完了
- データ・ベース設計の完了
- システム設計の完了
- コーディング完了
- システム・テスト計画完了
- システム・テスト——中間時点
- システム・テスト完了
- コンバージョン計画完了
- 導入完了

コントロール・ポイントのアクティビティの要約が提示される。チェック

ポイントまでに行なわれているはずのタスク、そのチェックポイントについての責任、作成されるコントロール用ドキュメント、などを加えて、各チェックポイントの目的が述べられる。

V 時間とリソースの管理

(Time and Resource Control)

時間とリソースの管理は、割り当てられたリソースを用いて指定された時間の枠内で、所期の成果を開発することを保証するものである。このことはプロジェクト開発に対する経過時間の管理のみならず、プロジェクトを遂行するのに使用されている要員と機械の量とタイプを管理することも意味している。

以前に提示した21のチェックポイントのうちでリソース管理の適切なポイントと考えられるのは、次のようなものであろう。

- プロジェクトの承認
- プロジェクト・プランニング
- 要員割当
- 見積り
- スケジューリング
- 予算化
- スタディ完了
- アナリシス完了
- デザイン完了
- コーディング完了
- プログラム・テスト——中間時点#1
- プログラム・テスト——中間時点#2
- システム・テスト計画完了
- システム・テスト計画中間時点
- 導入後の監査

リソース利用に関して、これらチェックポイントで持ち出されるべき質問が提示されている。

Ⅵ プロジェクト・コントロール・システムの導入 (Implementing a Project Control System)

A レビュー (Review)

プロジェクト・コントロール・システムを確立するための前提条件は、セクションⅡに概括的に述べておいた。そこに含まれていたのは、システム確立のためのマネジメント委員会、新しいアプリケーションのインストラクションのための全体的プラン、システムとプログラミングのための方法の標準、およびリソース使用データを収集するための統制的手続き、といった項目であった。

セクションⅢ、Ⅳ、Ⅴではプロジェクト・コントロール・システムを構成するレビューとコントロールのテクニックが論じられた。ここでは、そうしたシステムの導入のためのアプローチに注意を向けよう。

B 事前段階 (Preliminary Step)

プロジェクト・コントロール・システムを導入するについて、くり返し生ずる問題の1つは、データ処理部門の要員の側の協力と了承を得ることである。コントロールシステムは通常、より厳格な規律と、より公式的な報告手順と、より徹底的なドキュメンテーションといった要件を要求する。

このような支持を得るための1つのアプローチは、アナリストやプログラマーがコントロール・システムの開発に貢献する機会を与えることである。これはコントロール・システムを開発し、他の従業員からの提案を求め、任命されたスタディ・グループ内のスタッフの主なメンバーを参加させることによって達成されるかもしれない。

プロジェクト・コントロール・システムを導入するステップは次の通り。

- 1.) インストラクション全体でのプロジェクト・ミックスを検討すること。

- 2.) より強いコントロールを必要とする分野がどこかを知るために、ある期間内でのインストラクション中のプロジェクト遂行の成果を調査すること。
- 3.) プロジェクト・コントロール・マニュアルと必要な書式や手続きを含む、コントロール・システムを設計すること。
- 4.) 過去および現在のプロジェクトを対象とし、このシステムをテストすること。
- 5.) このシステムを使うようにスタッフをオリエンテーションすること。
- 6.) システムを実施すること。
- 7.) システムの有効性を監査すること。

C プロジェクト・ミックスのレビュー (Review Project Mix)

プロジェクト・ミックスをレビューすることはこのシステム上の強調点を確認する機会を与える。本コースのセクションⅡがそのプロセスを述べていた。

D 成果の調査 (Study Performance)

いつまでもコントロールの困難な点を明確にするためにインストラクションの成果記録が調査させるかもしれない。この調査は次のものを含むべきである。

- 実施されているメンテナンスの量とタイプの吟味、
- 代表的なプロジェクトについて、予算と実際との比較、および、
- ユーザー部門およびコンピュータ・オペレーション担当者からの不満とか提案

これらや、それに関連する要因を分析することによりインストラクションでの問題点とコントロールが最も必要とされる分野とが示されるだろう。

E コントロール・システムのデザイン (Design Control System)

プロジェクト・コントロール・マニュアルの目次の例は、このコースのはじめの方で提示された。このマニュアルは簡単であり、かつ要点を得たものであるべきである。その書き方は、記述的ではなく、命令的な文体にすべきである。すなわち、“次の事が行なわれる”という書き方でなく、“貴方が為すべきである”という書き方にすべきである。

プロジェクト・コントロールの書式を実行するための完全な指図も記載しなければならない。

F システムのテスト (Test System)

コントロール・システムをテストする2つの方法を提案する。

第1にコントロール・システムを、完了したプロジェクトに事後的に適用してみるというやり方がある。どんな書式が使用されたことになるか、コントロール・ポイントがどんな効果を持ったはずであろうか、などを知るために、プロジェクトをレビューするのがよいだろう。この分析によりコントロール・システムがたしかに役立つという最初の感想を与えるべきである。

第2のテスト方法は、何か新しい手頃な大きさのプロジェクトを選び、それを新しいプロジェクト・コントロール・システムでコントロールしてみることである。これは全ての書式の実行、指示されたチェックポイントでのレビューの遂行、ドキュメンテーションの要件の固守、などを意味する。コントロール手続きの実用性と適時性は、このようにして、システムが一般的に使用される前により改善される可能性を持たせながら、現実のケースで評価されることになる。

G オリエンテーション (Orientation)

全てのスタッフメンバーは、集合教育と教材の配布により、このシステ

ムへのオリエンテーションを受けなければならない。システムに関する質問が気楽に出るような会合にすべきである。

H 実施 (Installation)

このシステムの実施は、その期間中、マネジメントの特別の注意を必要とする。適切な書式の使用は、要求されるドキュメンテーションの作成と同様に、監視され、励行されねばならない。現に活動しているプロジェクトはすべて、例外の承認が習慣にならないように、直ちに新しいシステムに変換されるべきである。

I 監査 (Audit)

そのシステムが一定期間作用した後、次のことが実行されているか否かを知るための監査を行なうべきである。

——書式や手続きが実際に役立つものであり、利用されているか。

——見積りの手法は現実的な見積り値を与えているか。

——コントロール機能をはたすのに必要な時間が長すぎないか。

プロジェクト・コントロール、システムは、他のいかなるシステムとも同じく、ダイナミックで、当然その全寿命にわたって注意する必要がある。

Checkpoint Review Guide #1

チェックポイント名： プロジェクトの選択 (Project Selection)

目的： プロジェクト目標の定義

ドキュメンテーション

問い合わせもしくは要求書の記録

予備見積りの書式

タスクの定義

1. 目的の記述
2. プロジェクトのタイプの規定
3. 要求書の分析 (フィージビリティ、アナリシス)
4. 予備調査の実施
5. 予備的なマクロ、タスクの概要と見積り
6. 領域とリソース使用のためのガイドラインの設定
7. 必要なスキルの確認
8. インストラクション状況のレビュー
9. 選択すべきオルタナティブの記述

責 任

- ユーザー —— 目的および便益の提案と記述。
- D P 部門 —— 要求書の分析、便益の確証、見積りの作成。
- ユーザー —— 記述しなおされた目的とリソース見積りとに対する了承。
- 経営者 —— 予備見積りのレビュー。
—— 要求書の承認又は拒否。

マネジメント・チェックリスト

1. その仕事はデータ処理か？
2. そのプロジェクトは新しいデータベースの作成を必要とするか？もしそうなら、それがどうやって行なわれるか？
3. ユーザーは、たとえ必要であったとしても、インプット・ドキュメントの処理についての彼のやり方を変えることに同意するだろうか？
4. ユーザーは現在のシステムで作成されている報告書の効用に進んで疑問をもつだろうか？
5. もしこれがメンテナンス・プロジェクトであるならば、システムを修正するよりもむしろやり直した方がもっと有効ではなからうか？
6. 何を基礎にして予備見積りはなされたか？
7. 後になって見積らねばならない項目はどれか？
8. プロジェクトの目標は、測定可能な言葉で記述しうるか？

Checkpoint Review Guide #2

チェックポイント名： プロジェクトの承認 (Project Authorization)

目的： ユーザーとデータ処理部門とがプロジェクトを開始するための準備がととのっているかどうかのチェック

ドキュメンテーション

プロジェクトの記述

資金ぐりの記述 (必要な場合だけ)

タスクの定義

1. 優先順位のわりつけ
2. リソースの割り当て
3. この承認によりどれだけのものが認められるようになるか規定すること。
4. リソースの委託
5. 次のレビュー・ポイントの設定

責任

ユーザー —— プロジェクト目標の承認

D P 部門 —— プロジェクトに同意し、当初に必要なリソースを獲保すること。

経営者 —— 費用の承認

マネジメント・チェックリスト

1. このプロジェクトを実行するとした場合には、どんなプロジェクトを

延期したり、中止したりしなくてはならないか？

2. ユーザーは、このプロジェクトにどんな項目が含まれていないかを理解しているか？
3. このプロジェクトを延期することによって、どんな障害がユーザーの組織に生ずるか？
4. このプロジェクトはユーザーの問題にとって、最適な永久的解決を与えるのか、それともこのプロジェクトの後でさらに追加作業をしなければならぬのか？
5. 新しいシステムによって、どんな組織変更が必要とされるか？
6. ユーザーはこのプロジェクトにどのレベルまでのリソースを提供すると予想されるか？
7. このプロジェクトによって得られる便益を、他の方法で達成することは出来ないか？
8. プロジェクト全体を承認する前に、予備調査を行なわねばならないか？

Checkpoint Review Guide #3

チェックポイント名 : プロジェクト計画 (Project Plan)

目的 : 何をしなければならないかを決定する。

ドキュメンテーション

プロジェクト計画書

トラブルの予測

タスクの定義

1. 詳細なタスクのアウトライン
2. 要求されるスキル
3. 組織上の責任
4. 品質管理とリソース管理のためのチェックポイント

責任

D P 部門とユーザーのプロジェクト・チームがプランを用意する。

D P 部門とユーザーのマネジメントが詳細をレビューする。

経営者が要約をレビューする。

マネジメント・チェックリスト

1. タスクの標準的な分割が行なわれているか？
2. レビュー・ポイント及び進捗報告書を準備すべき場所が含まれているか？
3. 各々のタスクに対する組織の責任は明確か？
4. "援助する"、"レビューする"、"相談する"、"実行する"とい

った言葉の意味ははっきりしているか？

5. 設計仕様書のレビューのためのタスクが含まれているか？ テスト結果についてはどうか？

Checkpoint Review Guide # 4

チェックポイント名： 要員の割当て (Personnel Assignment)

目的： 誰を割当てるか？

ドキュメンテーション

プロジェクト要員の (タスクへの) 割り当て

タスクの定義

1. プロジェクト・リーダーの選定
2. D P 要員の選定
3. ユーザー側の要員の選定
4. トレードオフ (やりとり) を行なう。
5. サポートや訓練の必要性の確認

責任

D . P . 部門およびユーザーのマネジメントが選定を行なう。

マネジメント・チェックリスト

1. プロジェクトに割当てられる重要な人物は誰か？
2. 彼らはなぜ重要とみなされるのか？ アプリケーションの知識か？
技術面での経験か？
3. 彼らがプロジェクトの進行中に従事不可能となった場合、どのようにして代理とおきかえるか？
4. プロジェクトに対する影響はどのようなものがあるか？
5. プロジェクト・チームにバックアップ要員は含まれているか？

6. チームに代理者をなじませるのにどれ位の期間がかかるか？
7. このプロジェクトで、ジュニア・レベルの要員はどのように使われているか？
8. ユーザーはプロジェクトに割当てられた要員をどのようにおきかえるか？
9. 重要なデータ処理要員のこのプロジェクトに対する従事可能性は何パーセントの信頼度を持っているか？
10. もし、訓練が必要とされるなら、それはいつ、どのように行なわれるか？

Checkpoint Review Guide #5

チェックポイント名： 見積り (Estimating)

目的： 仕事にはどれだけの時間がかかるか？

ドキュメンテーション

プロジェクト要員の（日数での）割り当て
プロジェクト・タスクの見積りの要約

タスクの定義

1. タスク一覧表のレビュー
2. 詳細の拡張
3. 標準の適用
4. 標準化されていないタスクについての見積り
5. レコードキーピング手続きの確立

責任

プロジェクト・リーダーが“見積り人”の見積りをレビューする。
作業担当者が個別に部分の見積りを行なう。

マネジメント・チェックリスト

1. 詳細なタスク一覧表は構成されているか？
2. タスクの各々に対してスタンダードが用意されているか？ それは見積りのさいに用いられたか？
3. スタンダードのないタスクについて、どのような技法が見積りに用いられたか？
4. もし別個に異なった見積りがなされたとしたら、最終的な見積りではどの程度に妥協がなされるか？
5. この見積りを、すでに完了した類似のプロジェクトをどのように比較するのか？ 時間の記録は比較のために使えるか？

Checkpoint Review Guide #6

チェックポイント名： スケジューリング (Scheduling)

目的： どれだけの時間が必要か？

ドキュメンテーション

PERT ダイアグラム

ガント・チャート

レビュー会議のスケジュール

プロジェクト要員の（期間に対する）割り当て

タスクの定義

1. 見積りをレビューする。
2. 遅れの要因を記述する。
3. クリティカルなタスクを確認する。
4. 最終期限を決める。

責任

プロジェクト・チームが細部を実行する。

マネジメントは遅れと最終期限をレビューする。

経営者は要約をレビューする。

マネジメント・チェックリスト

1. スケジュールには、状況変異に対する遅れの要因を見込んであるか？
2. クリティカル・パス上にあるアクティビティはどれか？
3. テスト時に於てどんな仮定が設けられていたか？
4. ユーザーの参画についてどんな仮定が設けられていたか？
5. 訓練についてはどんな仮定が設けられていたか？

Checkpoint Review Guide #7

チェックポイント名： 予算化 (Budgeting)

目的： プロジェクトの費用はどれくらいになるか？

ドキュメンテーション

タスク別プロジェクト予算

期間別のプロジェクト予算

タスクの定義

1. タスクとその所要時間をリストする。
2. 標準賃率をそれに適用する。

責任

プロジェクト・リーダーが詳細を用意する。

D P 部門およびユーザーのマネジメントがレビューする。

経営者もレビューする。

マネジメント・チェックリスト

1. すべてのリソースについて標準原価が用いられているか？
2. ユーザー側要員の人件費も見込んであるか？
3. すべてのタスクについて費用が洩れなくみているか？
4. 遅れ（たとえば、機械時間の不足）によって起る手待ち時間の費用はみているか？
5. 各々の予算額を裏付ける細部見積りは用意されているか？
6. その見積りで不適當な可能性があるか？
7. 逸脱はどれくらい迅速に測定できるか？

Checkpoint Review Guide #8

チェックポイント名： システム・スタディの第一段階(System Study 1st Phase)

目的： そのプロジェクトに対するユーザーの当初の反応はどうか？

ドキュメンテーション

トラブルのレポート(必要ならば)

タスクの定義

1. 最初のコンタクトの報告書のレビュー
2. ユーザーにどこまで頼れるかの程度を査定する。
3. 領域の拡張を明記する。

責任

プロジェクト・リーダーが要約を行なう。

D.P. 部門、ユーザー、経営者に対し問題点の報告をする。

マネジメント・チェックリスト

1. ユーザー部門のマネージャーはインタビューの対象になってくれるか？
2. ユーザー部門の担当者はインタビューの対象になってくれるか？
3. プロジェクトに対して、どんな種類の抵抗が示されているか？
4. ユーザーのマネジメントが期待しているものは何か？ユーザーの担当者は？
5. 領域のどんな変更が考えられるか？
6. ユーザーは、システムによってどんな問題が解決されると期待しているか？
7. ユーザーが述べている問題で、新システムでも解決されないものはどれか？
8. 現状システムについて、ユーザーはどんな改善点を提示しているか？

Checkpoint Review Guide #9

チェックポイント名： スタディの完了 (Study Completion)

目的： スタディが終了したかどうか？

ドキュメンテーション

進行状況の報告書

見積りの改訂

スケジュールの改訂

予算の改訂

現行システムの記述書

タスクの定義

1. 現行システムの要約を解説する。
2. 発見事項についてユーザーと論議する。
3. 新たな例外事項を明確にする。
4. 領域の拡大を明確にする。

責任

プロジェクト・リーダーが準備する。

D.P 部門およびユーザーのマネジメントがレビューする。

マネジメント・チェックリスト

1. 現行システムは現在十分にドキュメント化されているか？
2. システム・ドキュメンテーションのチェックリストにおける全項目が準備されているか？

3. このドキュメンテーションにユーザーのマネジメントが目を通し了承したか？
4. スタディ期間中に担当者との相談が行なわれたか？ そしてその場合のコメント又は提案は記録あるいは要約されているか？
5. スタディ・チームは現行システムにおける全ての報告書およびファイルの利用状況を十分にレビューしたか？
6. ユーザーはドキュメンテーションに含まれていない追加の例外事項を持っているか？
7. ユーザーはドキュメンテーションをレビューするのにどのくらいの時間を要したか？
8. スタディの段階で発見された追加すべき問題点、ファイル、例外事項にはどんなものがあったか？ そしてこれらはプロジェクトの見積りをどのように変えるか？

Checkpoint Review Guide #10

チェックポイント名： アナリシスの完了 (Analysis Completion)

目的： 要求されている事項を十分に把握したか？

ドキュメンテーション

必要条件の要約 (最終承認)

見積りの改訂

スケジュールの改訂

予算の改訂

タスクの定義

1. デザイン、オルタナティブを論じる。
2. システムの必要条件をレビューする。
3. デザインの見積りを作成する。

責任

プロジェクトリーダーが準備する。

ユーザーのマネジメントが同意する。

D.P 部門のマネジメントが同意する。

経営者がレビューする。

マネジメントのチェックリスト

1. 新しいシステムの機能面での必要条件が詳細に記述されているか？
2. ユーザー、グループのだれがこの必要事項の記述をレビューしたか？
この問題について最も熟知している担当者かマネジメントがレビューに

加わったか？

3. デザイン・オルタナティブとして現在考えられているものにはどんなものがあるか？ 最終的な選択にはどんな要因が影響を与えるか？ どんなトレードオフが可能か？
4. 当初のプロジェクト記述書からみて、プロジェクトの領域にどんな変更がおこっているか？
5. これから先、どんな新規の例外事項および問題点があると予見されるか。

Checkpoint Review Guide #11

チェックポイント名： データ・ベースのデザイン完了 (Data Base Design Completion)

目的： 企画されたデータ・ベースは適切なものかどうか？

ドキュメンテーション

ファイルの明細書

タスクの定義

1. ファイル内容のレビュー
2. 関連するレポートのレビュー
3. ユーザーとの討議
4. ユーザーからの承認をうる。

責任

プロジェクト・リーダーが準備する。

D.P. 部門のマネジメントがレビューする。

ユーザーは詳細にわたってレビューする。

マネジメント・チェックリスト

1. ファイルはユーザーが必要とする全てのデータを含んでいるか？
2. ファイルの中にある全てのデータ要素についてアップデートの手法が開発できるか？
3. データ・ベースをメンテナンスするのに必要なデータの収集にはユーザーが関与してくれるか？
4. 各データ要素が充分詳細に登録されるようになっているか？

Checkpoint Review Guide #12

チェックポイント名： システム設計 (System Design)

目的： このシステム設計は、プロジェクトの今後の継続のため、確固たる基礎になりうるか？

ドキュメンテーション

進行状況の報告

プログラミングの見積りの改訂

プログラミングの予算の改訂

プログラミングの要員の改訂

プログラミングのスケジュールの改訂

タスクの定義

1. ファイル、レポート、処理の明細書のレビュー
2. コントロールのレビュー
3. ドキュメンテーションのレビュー
4. プログラミングの容易性の確認
5. プログラミング見積りの作成

責任

プロジェクト・リーダーが詳細にわたってレビューする。

ユーザー —— 詳細にわたってレビューする。

D.P 部門 —— 詳細にわたってレビューする。

経営者 —— プロジェクトの目的とてらしてレビューする。

マネジメント・チェックリスト

1. プログラミングの明細書は、見積りおよびプログラミングのために充分詳細に作成されているか？
2. コントロールの手続きは規定されているか？
3. 例外事項と、その各々への対処がすべて記述されているか？
4. ドキュメンテーションのチェックリストにある全てのドキュメンテーションがそろっているか？
5. 次善のオルタナティブがあるか？ それにどんな妥協が含まれているか？
6. ユーザーのマネジメントから言われている目的は何か？ ユーザー部門の担当者からは？ その目的は解消されるか？ どうやって？
7. システム作業を行なうのに必要な、別の組織体との協力的行為で、なにか具体的なものはあるか？ そうした協力を期待するのは現実的か？ もし協力がなされないとすればどんな妥協案があるか？

Checkpoint Review Guide #13

チェックポイント名： コーディング (Coding)

目的： コーディングが論理設計に忠実に行なわれているか？

ドキュメンテーション

テスト計画

タスクの定義

1. 机上チェックの際に使用するテストケースをレビューすること。
2. 机上チェックが完了したことを確認すること。
3. テスト・データの準備をレビューすること。

責任

プロジェクト・リーダーはコーディングが行なわれていることを保証すること。

プログラマーのリーダーはそれをレビューすること。

プログラマーは、それに応え、テスト・プランを作成すること。

マネジメント・チェックリスト

1. コーディングの机上チェックが完了しているか？
 - a. インストラクションの使用法は正しいか？
 - b. サブルーチン及びセグメンテーションが正しく用いられているか？
 - c. 標準的なやりかたが守られているか？
 - d. プログラムには適当にコメントが付けられているか？
2. 机上チェックではどのようなテスト・ケースが用いられたか？

3. テスト計画は作成されているか？
4. テスト計画は下記事項を網羅しているか？
 - a メインラインのロジックとジョブの終了までの到達とをテストするための簡単なケースは？
 - b 各ルーチンをテストするためのケースは？
 - c 例外事項の各々をテストするためのケースは？
 - d ジョブの終了条件をテストするためのケースは？
 - e パラメーター・カードとスイッチのセットの状況の結びつきをテストするためのケースは？
 - f 異状なデータの混在やシーケンスのおかしなデータをテストするためのケース（例えば、削除されたマスター・データのあとにマルチプル・トランザクション・データがある場合）は？
5. 言語のきまりとして、標準的なものが用いられたか？
6. 特別なテストが必要な、誤りやすく複雑なルーチンが書かれていないかどうか？
7. テスト・データは作成されているか？

そのテスト・データは、テスト・プランに適合したものかどうか？

Checkpoint Review Guide #14

チェックポイント名： プログラミング (Programming —Third Mac —
hine Session)

目的： プログラマーがマシンと問題とを理解しているか？

ドキュメンテーション

テスト・プランの改訂

見積りの改訂

スケジュールの改訂

タスクの定義

1. 言語の困難度の決定
2. 援助をうけること。

責任

オペレーション部門はテストでの使用状況を定期的報告する。

プログラマーのリーダーは進度をレビューする。

プロジェクト・リーダーは今後のテストを延期したり承認したりする。

マネジメント・チェックリスト

1. コンパイルでのエラー・メッセージが除去されたか？ 除去されな
いで残っているもののうち、どれがプログラムのオペレーションに影響を
与えるか？
2. 最初のテストがプログラム仕様書の解釈の誤りを暴露してくれたか？
言語あるいはマシンの使用法にエラーがあったか？ ロジック・エラー
か？

Checkpoint Review Guide #15

チェックポイント名： プログラミング (Programming (75% of Test Budget))

目的： 実質的なテスト完了日はいつか？

ドキュメンテーション

テスト・プランの改訂

見積りの改訂

スケジュールの改訂

タスクの定義

1. テスト・プランの進度をレビューすること。
2. テスト・プランが忠実に守られているかどうかをレビューすること。

責任

オペレーション部門は75%完了を報告する。

プログラマーは準備を行なう。

プログラマーのリーダーがレビューする。

プロジェクト・リーダーは上記の事項が行なわれていることを確認すること。

マネジメント・チェックリスト

1. テスト・プランは守られたか？
2. テスト・プランのうちどれだけが行なわれたか？ (結果が出ているもの)

3. 更にテスト・ケースを追加する必要があるか？
4. テストの最中に、予期しない状況が発生したか？
5. プログラム仕様書の変更が必要か？
6. プログラムがまだループしているか？ あるいは、思いがけない状態で終りになっていないか？

Checkpoint Review Guide #16

チェックポイント名： システム・テスト・プラン (System Test Plan)

目的： システムは徹底的にテストされるか？

ドキュメンテーション

システム・テスト・プラン

機械時間の見積り

見積りの改訂

スケジュールの改訂

タスクの定義

1. プログラム・テストの完了を確認すること。
2. 予備的なオペレーションの指図書をレビューすること。
3. システム・テスト・プランのレビューを行なうこと。
4. システム・テスト・データのレビューを行なうこと。

責 任

プロジェクト・リーダーは計画を準備する。

システムズ・アナリストは計画を準備する。

ユーザーはテスト・データ及びその結果を準備する。

オペレーション部門はドキュメンテーションをレビューする。

マネジメント・チェックリスト

1. システム・テスト・プランには、エラーのテスト及びエラーの修正手続きが盛り込まれているか？

2. プログラムはすべて、十分にテストされたか？
3. テスト・データはすぐ使えるか？ どのように作られたのか？
4. なまのデータが用いられている場合は、例外やエラーが含まれるように追加もしくは変更が行なわれているか？

Checkpoint Review Guide #17

チェックポイント名： システム・テスト (System Test (50% of Budget) (✓B 25% and 75%))

目的： 現実にテストの完了するのはいつか？

ドキュメンテーション

システム・テスト・プランの改訂

見積りの改訂

スケジュールの改訂

タスクの定義

1. 必要とされるプログラム変更の数をレビューすること
2. ジョブの実行時間をレビューすること
3. プログラム・インターフェイスとエラーをレビューすること
4. 結果をレビューすること

責任

プロジェクト・リーダーはインターフェイスに関する諸問題をレビューする。

ユーザーは、結果をレビューする。

アナリストはインターフェイスをレビューする。

マネジメント・チェックリスト

1. システム・テスト・プランのうちどの部分が行なわれたのか？
2. プログラム変更が必要になった箇所はどのくらいか？

3. 上記の変更 に何かあるパターンがあるか？
4. 原因は何だったのか？
5. ファイルの仕様書に変更が必要とされたか？ 理由は？ その回数は？
6. 各プログラムからのアウトプットは後続のプログラムにうまくインターフェイスしているか？

Checkpoint Review Guide #18

チェックポイント名： システム・テストの完了 (System Test Completion)

目的： システムはテストされた環境条件の中で稼働したか？

ドキュメンテーション

進捗レポート

機械時間見積りの改訂

スケジュールの改訂

タスクの定義

1. 結果をレビューすること。
2. なされた変更をレビューすること。
3. 必要となったドキュメンテーションの変更をレビューすること。
4. ユーザーの承認をうること。

責任

オペレーション部門

プロジェクト・リーダー

ユーザーとD.P.部門のマネジャー

経営者要約のレビュー

マネジメント・チェックリスト

1. システム・テストの間に個々のプログラムに何か変更がなされたか？
2. それらは、なぜ必要であったか？

3. システム・テストの結果は、前もって予定されていた結果に一致したか？ もし相違があるなら、どうやって解消するか？
4. ユーザーはシステム・テストの結果を承認したか？ それに最終承認を与えたか？
5. システム・テスト・プランは完全に実行されたか？ もし、実行されなかったとしたら、それはなぜか？
6. なまのデータがテストで使われたか？ もしそうなら、そこに例外事項を含むように変えられたか？ もしそうでなかったとしたら、それはなぜか？
7. ドキュメンテーションやユーザーへの指図書は変更されなければならぬか？

Checkpoint Review Guide #19

チェックポイント名： ボリューム・テスト計画 (Volume Test Plan)

目的： ボリューム・テストに対する組織の準備は充分か。

ドキュメンテーション

進行についてのメモ (中断の言いわけ)

タスクの定義

1. ユーザーとD.P 部門の準備状態をレビューすること
2. 計画をレビューすること
3. データの準備
4. インプット手続きの準備

責 任

プロジェクト・リーダーは計画を準備する

ユーザーとD.P 部門は計画をレビューする

マネジメント・チェックリスト

1. どんなデータがボリューム・テストに使われるか？
2. データの精度、誤差、完全性について何がわかっているか？
3. オペレーションの指図書はテストのために適当であるか？
4. テストを吟味したり、立証したりするのにどんなスタッフが必要か？
それらの人々は従事可能か？
5. 結果を前もって予定しておくことは可能か？ テストはいかにして立証するか？
6. どんな異常や予期せざることがテストのときにおこったか？ それらはシステムの弱点を示しているか？

Checkpoint Review Guide #20

チェックポイント名： コンバージョン前のチェック (Preconversion Check)

目的： コンバージョンに対する組織の準備はできているか？

ドキュメンテーション

進捗レポート

進行についてのメモ

コンバージョン・スケジュールの改訂

タスクの定義

1. ボリューム・テスト結果をレビューすること
2. 事務員、D P 部門の準備態勢をレビューすること
3. ドキュメンテーションをレビューすること
4. うまく行かない場合の措置 (フォール・バック手続き) を設定すること
5. 最終的なアクティビティを確認すること

責任

プロジェクト・リーダーは計画を用意する

D P とユーザー部門のマネジメントはレビューを詳細に行なう

経営者はレビューを詳細に行なう

マネジメント・チェックリスト

1. ユーザーはコンバージョンにそなえているか？

2. コンバージョン問題を扱うのに要員の追加が可能か？
3. コンバージョン・プログラムはテストされているか？
4. コンバージョンの途中での変更をコントロールする準備ができてい
るか？
5. コンバージョン期間中は、インクアイアリーをどう処理するか？
6. もしコンバージョンがうまくいかなかったら、どんなフォールバック
手続きが使われることになっているか？
7. コンバージョン期間中のコントロールをだれが行なうか？ どんな方
法で？
8. コンバージョンの作業においては何が最終的なタスクであるか？
9. コンバージョンの終了をあらわすシグナルは何か？ どうやって？
10. ユーザーの指導書は作成されているか？
11. ユーザーには口頭説明が与えられているか？

Checkpoint Review Guide #21

チェックポイント名： 導入後の監査 (Post-Implementation Audit)

目的： 他のプロジェクトにも適用できるような教訓として何を学んだか？

ドキュメンテーション

マネジメント・レポート

プロジェクトの見積り値と実際値についてのレポート

タスクの定義

1. 現実化された便益をレビューすること
2. ユーザーとDP部門のオペレーションをレビューすること
3. 必要とされた改良点を確認すること
4. プロジェクトのコスト、予算およびスケジュールをレビューすること

責任

DP部門とユーザーのマネジメントレポートを準備する。

経営者 —— 要約をレビューする。

マネジメント・チェックリスト

1. 予期された便益はなんであったか？
2. それらは達成されたか？
3. もしそれらが達成されていないならば、それらを達成するのに、さらにどんな努力が必要となるか？
4. もしコストを削減することが目的であったならば、本当に実現したか？
？ どんなふうに？

5. 新しいシステムは急を要する仕事や緊急事態に対して使用されるか？
もし、そうでなければ、どんなシステムが使われているか？
6. 個々のトランザクションごとの処理時間は減少したか？
7. ユーザーはさらにどんなリポートを要求しているか？ それはなぜか？
それはいかんにして用意されるか？
8. ユーザーはこのシステムにどんな苦情をもっているか？
9. どんな改良が可能か？
10. プロジェクト予算は超過したか？ それはなぜか？ 新しいプロジェクトにおいて、それを防ぐには何ができるか？

請求 番号	経 46-27	登録 番号			
著者名	日本経営情報開発協会				
書名	プロジェクト管理の手法(1) システム・プロジェクト管理				
所属	帯出者氏名	貸出日	返却 予定日	返却日	

禁無断転載・使用

昭和46年7月

財団法人 日本経営情報開発協会

東京都千代田区霞が関3-2-5

霞が関ビル

TEL (581) 6401 (代表)