

プロジェクト進捗管理の実戦的手法

(Fitチャートによる進捗管理)

さか い ただ みち
坂 井 忠 通

〈要 旨〉

プロジェクト管理（プロジェクト・マネジメント）においては計画が大切だといわれている。しかし実際のプロジェクト現場では「計画が良いから、悪いから」だけで成否が決まるわけではない。無論、適切な計画は大切だが所詮は「当初の予定」であって、本当は「実施」段階におけるプロジェクト・マネジメントが最重要なのである。日々変化する流動的な現実を正確に把握し、正しい見通しを持って適切かつ迅速に対処していくことによるのみプロジェクトの成否が決まると言っても過言ではない。

ここでは、属人性の高いソフトウェア業界において計画と実施とがなぜ大きく乖離するかの理由と、実施段階における工程管理の重要性を説明する。工程管理のための有効なツールである「ガントチャート」、「PERT図」の特徴を述べた後、さらなる改善のための「Fitチャート」の提言を行なう。

〈キーワード〉

プロジェクトマネジメント 進捗管理 ガントチャート PERT図 見積もり

はじめに

情報処理システムは昭和40年代に入ってから急速に発展してきた。コンピュータの高性能化は高度化する社会の情報処理ニーズに応えるものとして歓迎され、多くの情報システムが企画、開発、推進されてきた。しかしそれらシステム構築のプロジェクト管理はほとんど常に問題含みであった。すべてが予定通り順調に進んだものはむしろ稀であり、多かれ少なかれ問題を起こして来た。稼働予定時期を大幅に遅延したり、品質、性能に致命的な問題が出る場合も少なくなかった。予算を大幅に超過するケースも珍しくはなく、最悪は開発中止に追い込まれたものさえあった。

それから40年経ち現在に至るまで、数多くの経験を積み、各種の手法が提案、試行されてきた。しかし近年のシステムトラブル報道から分かるように、いまだ問題の完全解決には至っていないのが実情である。

以下、本稿ではコンピュータを使った情報システム構築事例を中心に、プロジェクト管理の中で最も中心となる工程管理について工程上の諸問題が発生する背景や原因について触れる。また、その重要性と進捗管理手法について述べ、今までの進捗管理ツール「ガントチャート」と「PERT図」の長所短所を論じ、その改善策としてのツール「Fitチャート」（仮称）の提言とその説明を行う。なお、本稿で述べる考え方や手法は情報システムのプロジェクトだけに限定するものではなく、どのようなプロジェクトにでも適用できるものである。

I. プロジェクトにおける工程管理

プロジェクトとは一時的な目標を一定の期限内に達成するための企画／計画／作業であって、比較的大きな目標を組織として実行する活動を指して言う。（有名な巨大プロジェクトとしては、米国の原爆開発「マンハッタン計画」、月への有人飛行「アポロ計画」などが挙げられる。）一般的には、企業の新商品開発、情報システム開発、工場建設、展示会開催、社内改善運動等であり、社会のあらゆるところでプロジェクトが進められている。

プロジェクト管理（プロジェクト・マネジメント：プロマネと略される）とは、このようなプロジェクトを成功に導くための総合的な管理のことであり、工程（スケジュール）および人員、資金、損益、物的資源などの管理すべてを含む。

1. プロジェクトの管理項目

情報システム構築における管理のポイントは、下記の5項目に尽きる。

- ①機能：目的の業務を遂行する上で、必要な使いやすい機能が備わっていること。
- ②性能： " 、支障の無いように必要な性能が出ること。
- ③品質： " 、安定的に使用できる品質が備わっていること。
- ④納期：必要な時期までに完成して、使用出来ること。
- ⑤コスト：予定した予算の範囲内で完成、完了すること。

プロジェクトでは必ずと言って良いほど、これらの項目に関して問題が発生する。これらはすべて絡んでおり、それぞれが問題の原因であると同時に結果になっている場合が多い。このことがプロジェクト管理をいっそう複雑にし、問題解決を困難にしている。

各項目はそれぞれみな重要であるが、最も中心的なものは、④納期（納入期限）であろう。理由は、諸々の問題点がほとんどの場合ここに集約、反映されるからである。つまりプロジェクト管理では工程管理（進捗管理、スケジュール管理とも表現する）が最も重要であると言える。

情報システムの開発（ソフトウェア製造を含む）には、過去の一般的な製造業の世界と決定的に違う面がある。世間一般の製造、開発（たとえば建物やダム建設など）は材料、部品、工具、機械などの「モノ」によって制約される面が大きい。コンピュータソフトの世界では「作業者がすべて」と言えるくらい、「作業員＝人間」のウエイトが特別に大きい。その理由は、ソフトウェア製造、システム開発のほとんどすべてを大勢の「人間の手作業、知的作業」で行うので、仕事の成果が作業員の個人的能力やセンス、さらには性格や意欲、加えて人間関係、チームワークなどによって大きく左右されるからである。人によっては具体的な「やり方」も相当に異なっている。

つまり作業成果に占める「属人性」が極めて高い点に特徴があり、そのため、

- ・ 作業量の事前把握と要員計画【見積り】、
- ・ 作業途中での進捗把握【工程管理】、
- ・ 作業結果の完成度／精度【機能・性能・品質】の把握、管理

が難しくなっているのである。従ってプロジェクトのリーダーにとっては、「人の管理、指導」あるいは作業員の「人間関係」が非常に重要なのである。一見無機質に思われがちなコンピュータの世界で人間関係がきわめて重要な位置づけにあると言うのは、意外な感じがするかもしれない。

2. 工程管理の重要性と難しさ

プロジェクトの管理項目が多数ある中で最も重要なものは工程管理、つまり作業の進捗管理であることは異論が無いであろう。期限（納期）通りにすべての要求を満たした目的とするシステムを完成して顧客（またはユーザー部門）に引き渡し（あるいは納入）しなければ、たとえ「モノ」が出来たとしてもそのプロジェクトは成功とは言えない。期限を守れない場合、プロジェクトによってはすべてが無意味になったり、社会システムであるならば社会的な大損害、大混乱を生ずることもある。場合によっては遅延による損害賠償として巨額なペナルティを依頼者側から請求されることもある。

納期（期限）は何事にもまして重要な項目と言えるが、現実には多くのプロジェクトが工程遅延を引き起こしている。この結果、品質・性能・損益などが影響を受ける。逆に、それらが原因で工程遅延しているケースも多々ある。これら要素は複雑に絡み合っているので、「工程の進捗状態」は結果であると同時に原因でもある。工程を適切に管理出来ればプロジェクト全体がうまく行く、と言うのはこのような理由による。

<見積りの難しさ>

ところが現実には、この納期がなかなか守れない。その理由は、例えば、

- ・ 予定よりも機材、人材、時間が多く必要になった
- ・ 予定外の事由（作業項目の追加など）や想定外のトラブルが発生した

などである。別な表現をすれば、対象の「作業量、作業負荷」に対する「生産能力、資源の大きさ」とのバランス問題と言える。

*原因としては、過少見積り、または機能追加による過大作業、見通しの甘さ、要員・機材のレベルの低さ、さらには不可抗力、等が挙げられる。

前者を分子に、後者を分母にとって、これを作業の負荷率=A とすれば、基本的には $A \leq 1$ でなければならない。
$$\frac{\text{作業量}}{\text{生産能力}} = \text{負荷率}A$$

Aの値が1.0を超えると、過負荷（作業量が多すぎる、オーバーロード）となり、当然のことながら、納期遅延、性能不良、品質不良、損益悪化などの問題が発生してくる。プロジェクト管理に際しては、このAの値が1.0を超えないようにすべきであるが、現実には、分子の「作業量」の事前見積りが難しく、さらに加えて、分母の「生産能力」（作業成果の属人性）が不確定なため、正確に決められないのが実状である。

* Aの値が1.0以下の場合、過大見積りとみなされる。このケースはあまり無いのだが、認識した段階で速やかに資源節約をするべきである。

<さらなる現実の重石：要員確保の難しさと損益管理>

しかも辛いことに、不確定な見積りに対して、開始時点で負荷率に余裕を見込むことが現実問題として難しい。初めから余裕を見込むべきだ、という「正論」は現場を知らない者の机上の論理と言える。IT企業は厳しい競争に打ち勝って商談を受注するために、仕様の不確定なシステムをギリギリのコスト条件で受注する。だから損益管理上、最初から余裕を見込むことなど許されないからである。さらに、常時人手不足と言われるソフト業界では、必要な時期に必要なスキルを持った要員を必要な数だけ確保することが非常に難しい。「人がすべて」と言われるソフト開発だが、現実には数の確保（頭数合わせ）だけで精一杯で、最悪は要員数不足のまま時間切れで「見切り発車」となるケースも多い。このように、もともと無理な条件（要員の量・質ともに不十分な状態）でスタートするプロジェクトが多いのである。

3. 工程管理の真の問題は何か――計画面と実施面での難問

さて、プロジェクトの工程管理では、まず計画（予定）を決め、次に作業実施段階で進捗を把握して、もし問題があればこれを是正しなければならないわけである。では、それぞれの問題、課題は何だろうか。まとめると<表-1>のとおりである。

※プロジェクトの形態には色々あるが、ここでは最も一般的な形、つまり顧客（利用者企業、ユーザー企業）が開発をベンダー（システム開発会社）に委託する形態を念頭に論ずる。

<表-1>

	工程管理上の問題、課題
《計画段階》	①作業量の見積り評価のミス（特に、過少見積り）
	②生産能力の評価のミス（特に、過大見積り）
《実施段階》	③実施時点における作業量と生産能力のギャップ （内部的な問題と外部的（対顧客）な問題） ④プロジェクト推進、管理ミスや管理の拙劣さ

《計画段階》

①作業量の見積り

まず、作業対象のボリューム（作業量）が作業開始前にはなかなか正確に見積もれない問題がある。そもそも、対象となるシステムの仕様が不明瞭なのである。通例、決まっているのは「〇〇システムを作る」、「××業務をコンピュータ化する」事だけ。本来はシステムの詳細設計が終わらなければ見積もれるはずがないのに、顧客側は最初に見積り（費用総額、期間、その他）を決めるよう要求する。予算管理上から総額を早く知りたい顧客の担当者の立場は理解できるが、まだ設計が終わっていないのだから開発製造するベンダー側からすれば無茶な話である。そのためベンダーはやむを得ず、過去の経験や類似システムから「大まか」に見積もることになる。その際、悲観的な要素は商談競合時点でカットされることが多い。実際に設計がなされていく過程で色々な作業が浮き彫りになってきて、結果として作業ボリュームが増大するのは、こういう事情によるのである。

近年は、さすがにこれらの「ドンブリ見積り」の反省から、より精度の高い「工程毎の分割見積り」が採用されるようになってきた。「未来」を見通すとは言っても、直近の、しかも部分的な工程ならば見通しが立ちやすく、かなり正確な見積りが出来るからである。仮に誤差が出ても部分誤差だから被害が少なく済む。過去のやり方から見れば、工程ごとの分割見積りは格段の進歩と言える。

②ソフト生産能力の評価、判断

生産能力とは、ソフトの場合、簡単に言えば開発要員の「数とスキル（技術力）」である。要員「数」の把握は簡単だが「スキル」の扱いが難しい。その理由は担当者の能力の個人差が大きすぎるからである。筆者の知る限りでも2～3倍の個人差は当たり前で、5～10倍のケースも稀ではない。

さらに、個人差だけがプロジェクトの生産能力の差になるわけではない。組織力、チーム力がこれらをカバーする力を持っているからだ。言いかえれば、プロジェクトマネージャー、プロジェクトリーダーのマネジメント能力の差が作業員の個人差のギャップを埋める力を持っているのである。（マネジメントが悪くて、逆にこのギャップを余計に広げてしまうケースも多くある）

これら「人的要素」をすべて作業開始前に正確に見積もるのは極めて困難である。そもそも、その時点で具体的な作業要員が決まっていないケースがほとんどなのだから当然であろう。従って実際には「仮想的、標準的、平均的な人材」で見積もることになる。つまり、計画はしょせん「絵に書いた予定、予想」であって、厳密な方向付け、運命付けではありえないのである。これをリカバリー（修復）するために、工程の「実施段階」における管理の重要性が浮上してくる。

《実施段階》

③実施段階の「開始時点」における作業量と生産能力のギャップ

プロジェクト開始と同時にプロジェクトマネジャー、リーダーがしなければならないことは、実施責任者として既に作成されている計画の妥当性と開発体制の吟味をすることから始まる。問題は三つある。第一は作業量や作業対象が当初計画から変わっていないか。第二は生産能力、言い換えれば要員／体制の問題である。そして第三にマネジメント能力が挙げられる。

第一の作業量については、増加するケースがほとんどである。開発を依頼する立場のユーザー側としては、決まっている価格の中でシステムに出来るだけたくさんの機能を要求したいのは当然である。さらに設計が進んでいくにつれ、細部の諸問題が顕在化し、これを解決するためにソフトあるいはシステム全体の負荷が増大するのである。この問題は顧客側と開発側との交渉の問題でもある。工程管理を完遂するためには、リーダーに設計作業と顧客との交渉を上手に誘導していく能力が要求される。

第二の生産力については、まず「個人スキル判定」が簡単ではないことが挙げられる。面接や資格（例えば情報処理技術者試験）による評価も絶対とは言えない。一番頼りになるのは「実績」であろう。つまり以前の作業実績、プロジェクトにおける個人評価、チーム評価が決め手となる。しかし現実にはいちいちこれを調べている時間的余裕は無い。

さらに困難さの背景として、恒常的な「要員の確保難」がある。プロジェクト計画に合わせて、必要な時に、適切な人材がすぐに確保／投入出来れば話は簡単だが、現実には全く違う状況にある。優秀な人材は常にビジーであって、仕掛り中のプロジェクトが終わるまではアサイン出来ない。偶然アサイン出来ればそれはラッキーなのである。常時「空いている」人材は、と云えば、どこからも依頼が来ない、どこへも投入出来ない新人や低スキルの人材である。しかし時間は待ってくれないので、通常はスキル等を無視して、苦し紛れに頭数だけ揃えて作業開始（見切り発車）することになる。

そして、第三の問題がマネジメント能力、つまりプロジェクト管理力である。つまるところ、リーダー（及びサブリーダー）の管理能力（プロマネ・スキル）である。実施段階において日々発生する諸問題の解決を的確にこなし、チームとして持てる人材を適材適所に配置し、適切な部下指導を行って個人の能力を最大限に引き出して組織の力を

発揮することのできる優秀なプロジェクトマネジャー、リーダーがプロジェクト成功のためには絶対に必要なのである。

以上をまとめると、多くのプロジェクトは、開始時点では

- ・やること（対象、ボリューム）があいまい、
- ・作業者のスキル（能力、適性）もあいまい

のまま、見切り発車しているのである。

こういう現実の中でプロジェクトを無事に完成させることは非常に難しい。結局、「プロジェクト実施」段階で作業実体を良く把握して、適時、改善と是正を繰り返しながら（いわば、軌道修正しながら）プロジェクトを推進、完成させるしか方法が無いのである。ここにプロジェクトマネジメントの重要性と日々の工程管理（狭い意味では進捗管理）の重要性が出てくる。

Ⅱ. 工程管理手法（特に進捗管理）の現状と問題

作業には必ず期限があり、管理者はその期限に対して現在の進捗状況はどうかを知る必要がある。すなわち、作業が期限通りに完了するのか、もし前後するならばどのくらいになるのか、あるいは（特に遅れる場合は）その原因は何なのか、その影響はどうなるのか、対策はどうしたら良いのか、等である。

工程管理は先ず「現状がどうなっているか」を正確に知ることから始まる。現状を把握し、そこから「今後の見通し」を判断することが重要である。しかし「現状」を正確に知る（というより「理解・実感」する）ことは意外に難しい。大概是楽観論が勝って、気づくのが遅れ、手遅れになってしまう。そうならないためには、進捗の実態と問題点を「実感できるもの」、「目に見えるもの」、つまりビジュアルにするツールや手法が必要なのである。

1. 基本となるガントチャート

この「ビジュアル」に進捗を示す代表例がガントチャート*〈図-1〉であり、現在も広く使われている。項目を縦軸に、作業予定期間を横軸にとり、作業を「線」で表したもので、これで予定、計画が一目でわかる。今では当たり前の手法だが、直感的に理解できる点は非常に優れている。

* 1910年代にヘンリー・ガント（アメリカ人）によって最初に考案された図表形式。縦軸には作業内容や要員、資源を記入し、横軸には時間軸を記入する。これで各作業の実施時期や資源使用時期を明らかにし、所要期間を横線で視覚的に示せる。それぞれの作業の開始・終了時期の全体像が直感的に把握できるので、管理上、非常に有効な進捗管理方法として広く使われている。

<図-1 ガントチャート例(1)>

	第1週	第2週	第3週	第4週	第5週	第6週	第7週	第8週	第9週
要求定義	△	—————	—————	▼					
概要設計				△	—————	—————	▼		
詳細設計						△	—————	▼	
コーディング							△	—————	▼
テスト								△	—————

2. ガントチャートの実際

では、実際に進捗を管理するうえでガントチャートに問題はないのだろうか。ガントチャートはもともとが「予定」や「計画」を示すものである。工程管理する上では実績を把握し、予定と実績の差（予実差）を把握して、これに対するなんらかの判断と対策（アクション）が必要なのである。この「実績」をチャート上で網線を使ったものが<図-2>と<図-3>である。<図-2>の例では、第3週の終り時点での作業進捗（実績）を網線で示している。<図-3>の例では、さらに第5週が終わった時点の作業進捗*を示している。いずれの場合も、予定に対して現在どのような状態にあるかをチャートから読み取ることが出来る。

*進捗の基準：作業進捗を何で測定するかが問題である。仕様書の枚数、プログラムのステップ数など定量的な「成果物」で見ると、作業完了までの予想日数で見るとある。成果物の量と作業期間とは互いに相関関係にあるが、微妙に異なるものである。また、成果物実績もその「品質や内容のレベル」が絡むため報告者の主観が入り、今後の予想日数は担当者の主観であるため、必ずしも正確なものとは言えない。

しかし単純なガントチャートでは、工程の現状を表現できても、期間の経過とともに作業の進捗状況が変化していく経過、推移をダイナミックに示すことが出来ない。例えば少しずつ遅れが広がっている作業など、実は問題があるのに見落としてしまう欠点がある。〔例えば、プログラム-A（鈴木一郎）の作業は危険な状態にある。など〕

<図-2 ガントチャート例(2)>

ガントチャート 第3週の終り時点



作業項目	担当者	第1週	第2週	第3週	第4週	第5週	第6週	第7週	第8週
プログラム-A	鈴木一郎	△	■	△					
プログラム-B	伊藤竜太	△	■	■	■	△			
プログラム-C	佐藤次郎			△	■	■	■	■	△
プログラム-D	鈴木一郎				△	■	■	■	△
プログラム-E	田中雅夫		△	■	■	△			

<図-3 ガントチャート例(3)>

ガントチャート 第5週の終り時点



作業項目	担当者	第1週	第2週	第3週	第4週	第5週	第6週	第7週	第8週
プログラム-A	鈴木一郎	△	■	△					
プログラム-B	伊藤竜太	△	■	■	■	■	△		
プログラム-C	佐藤次郎			△	■	■	■	■	△
プログラム-D	鈴木一郎				△	■	■	■	△
プログラム-E	田中雅夫		△	■	■	△			

3. デジタル表示の工程表

ガントチャートの直感的な表示方法は分かり易い半面、どうしてもアナログ的な曖昧さが残る。つまり、遅れが出ているのは分かるが危機的状況か否かが、ピンと来ない欠点がある。例えて言えば「黄信号なのか？、赤信号なのか？」の判断であって、これによって対策アクションの早さ、強さが違って来る。プロジェクトの状況が本当は危機的であるならば、これを早く実感して強力な手を素早く打つことが工程管理ではもっとも大切なことなのである。この直観力はある程度リーダー個人の人格特性（センス）に依存するものではあるが、誰でもが客観的に判断、判定出来るツールや方法があれば望ましい

のは言うまでもない。


このアナログ的な「あいまいさ」を解決するために、〈図-4〉の様なデジタル表示する方法も過去に試みられた。つまり達成率の指標が予定の「80%以下は要注意、60%以下は危険」というように数値で明確に判断できるような工夫である。これは定量的になるため判断が簡単明瞭になり、また、グラフ化しての分析などに役立つ。だが現場担当者からは「感覚的に分かりにくい」ので不評であった。さらに数値を意図的に「少しいじれば」評価が大きく変わり、白か黒の判断を「デジタルに間違ってしまう」危険性も大きい。そうすると数値の細かな精度、妥当性が新たに問題となってくる。実際は「灰色」である現場実態を「白か黒か」に断定することは想像以上に難しいのである。

しかも数値化は、同時に「数字の一人歩き」の危険性を含んでいる。灰色を見て、本当は黒なのに一旦「白」と表記されると誰も注意を払わなくなり、潜在する致命的な問題点が見落とされ、手遅れになる危険性が大きい。そのため現場では、これにアナログ的な線表を併記することも行われた。デジタル工程表は一見正確に見えるが、直感性（見やすさ）と信憑性（現場実態の正確な反映）に問題が出る危険性があるという難点があった。

〈図-4 デジタルな工程表の例〉（数値をプログラムのステップ数で進捗を表している例）

*ここでは、プログラム-Aが現在〈第5週末時点〉2週間遅れの未完成。
Eは予定通り完成。但し、予定よりステップ数が680と+80増加している。
(%表示はその時点までの予定に対する達成度を示す。)

デジタル工程表 例



作業項目 担当		予定・実績	第1週	第2週	第3週	第4週	第5週	第6週	第7週	第8週
プログラム-A	実績		50 (100%)	150 (75%)	15 (50%)	210 (70%)	250 (83%)			
鈴木一郎	予定	300	50	200	300	-	-	-	-	
プログラム-B	実績		70 (70%)	140 (70%)	220 (73%)	350 (88%)	430 (86%)			
伊藤竜太	予定	600	100	200	300	400	500	600		
プログラム-C	実績		-	-	50 (50%)	240 (96%)	380 (109%)			
佐藤次郎	予定	700	-	-	100	250	350	500	600	700
プログラム-D	実績		-	-	-	60 (60%)	150 (83%)			
鈴木一郎	予定	500	-	-	-	100	180	250	400	500
プログラム-E	実績	680	-	150 (75%)	300 (86%)	500 (100%)	680 (113%)	-	-	-
田中雅夫	予定	600	-	200	350	500	600	-	-	-

この例で分かる通り、デジタル表示の工程表は「直感性」に劣るため、かえってプロジェクトの進捗を把握し難くする欠点がある。また、成果物で測定する場合でも、プログラムなどは作成過程で見積りより膨らんだり小さくなったりするケースがあるため、基準となる数値が変動することにより係数値の意味／価値が不明朗になる面も否めない。この例のようにプログラムのステップ数で管理しているやり方は、「製造量の管理」には適しているが作業の時間的な「進捗」管理には、やや不適當と言わざるを得ない。

Ⅲ. ガントチャートの限界と PERT 図

1. ガントチャートの限界

すでに述べたように、ガントチャートは直観的に工程の進捗状態を理解、判断するうえで非常に優れたものである。しかしいくつかの限界を持っている。それは、ある時点での「静的な進捗状況」は分かるが、それが過去にどのような経過をたどってそうになっているのか、問題の本質は何なのか、従って今後どうなると予想されるか、などは上手く表示できない点である。

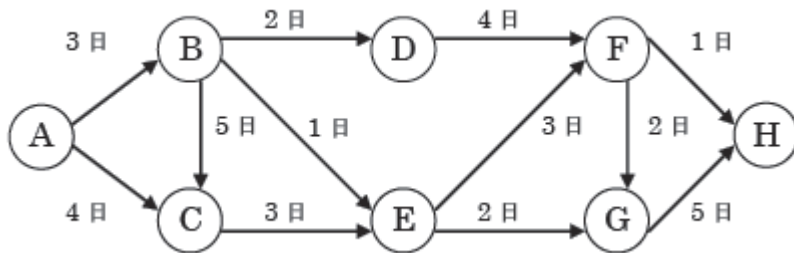
つまり現時点の静的記録としての意味はあるが、工程の進捗状況をダイナミックに現わすことができないのである。過去の経緯を含めての現在が分からなければ、プロジェクトの実態や工程推進のための問題点出や解決方策を導き出すヒントを得るためには不十分なのである。

2. PERT 図の登場

ガントチャートの限界を打破する手法としては PERT 図<図-5>があげられる。大規模で複雑なプロジェクトにおける作業の工程計画・管理手法のひとつで、プロジェクト全体を構成する各作業の相互依存関係をネットワーク図で表し、各作業の日程計画を作成するとともに作業全体の所要時間を算出し、さらにクリティカルパス*を認識することで工程のボトルネックを明らかにし、所要時間の短縮（あるいは遅延防止）を図る手法である。

*クリティカルパス：複数の作業工程が関連している時、もっとも時間を要する工程（ネットワーク図上でのルート）部分を言い、この工程が遅延すれば全体工程が遅れる危険性のある工程（ルート）を指す。

<図-5 PERT 図の一例>



*線は作業工程を表し、数字はその工程の所要日数である。開始点④～終了点⑧は工程の区切り時点を示す。

PERT : program evaluation and review technique

PERTは、対象とするプロジェクトに必要なタスク（作業）を相互に関連付けて分析する手法である。各タスクの完了までに必要な時間を分析し、プロジェクト全体を完了させるのに必要な最小時間を特定することが出来る。

PERTは1958年に米海軍の原子力潜水艦搭載用ミサイル「ポラリス」の開発に際し開発されたという。PERTの特徴はネットワーク図で、個々の作業を矢線で表し、それらの前後関係を結び合わせたアローダイヤグラムによってプロジェクト作業全体を表現し、工程を管理する。PERTによってクリティカルパスの概念が誕生した。

ガントチャートでは、多数の作業が複雑に入り組んだ工程を計画・管理しようとする、作業相互の依存関係が分からなくなるうえ、管理の重点ポイントがどこにあるかをつかむことが難しい。また、ある特定の作業の進捗が遅れた場合に、それが全体にどのような影響を及ぼすかについてすぐさま判断することが困難で、実施段階に入ってから計画変更や状況変化に対する柔軟性に欠ける。PERT図はこの問題を解決出来るものである。しかし一般に紹介されているアローチャート（矢線図）のままでは非常に使いにくい。現場が使えるためにはもっと進捗の実感が湧く、直観的で分かりやすいものでなければならない。

そこで以下に簡単なサンプル問題「ライスカレーが食べられるまで」を付した。条件に従って作成した解答例である。○付数字は作業の結節（始点、終点）を示す。横線は作業を時間軸で表したもの。縦線（実践、点線）はそれぞれの順序性、因果関係を示している。この表によって作業日程とその前後関係などが直観的に分かる。（ここでは、簡単な作業なので進捗管理はしていない。）例えば、コンロの数が増えたり、作業員数が変化した場合、PERT図で示せば容易に答が出せる。

《サンプル問題 「ライスカレーが食べられるまで」》

〈条件〉

*作業人数、機材などは充分ある。ただし、コンロは1個しかない。

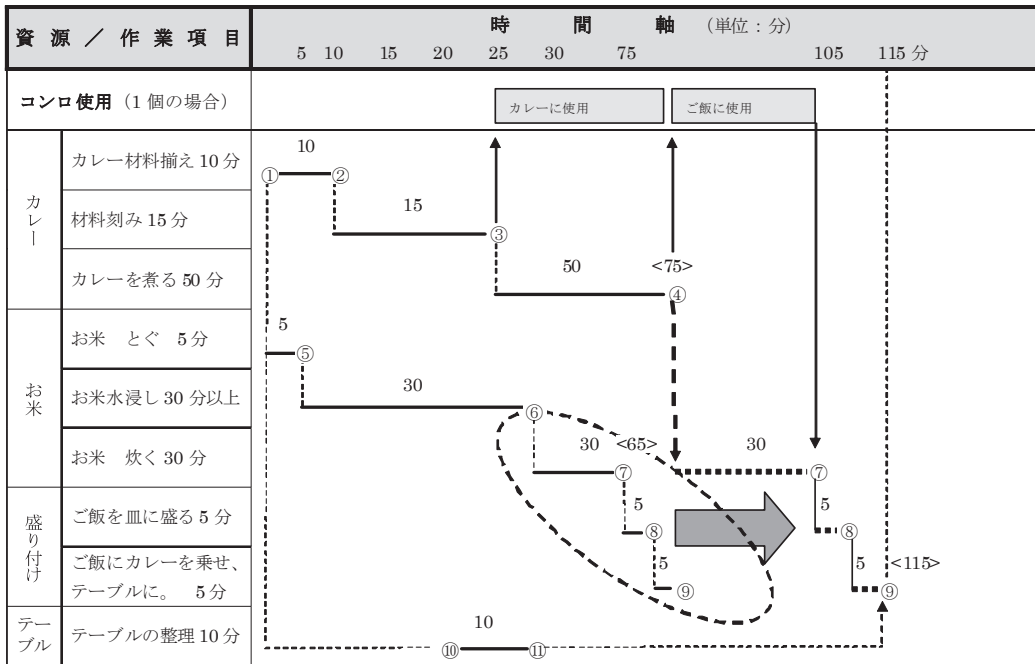
- ・カレー料理の材料を台所にそろえるのに、10分掛かる。
- ・カレーの材料を刻んだりして鍋に入れるところまで15分掛かる。
- ・コンロにナベを架けて、カレーを煮るのに50分掛かる。
- ・お米をとぐのに5分掛かる。
- ・米を水に浸し終わってから、ご飯を炊くのは30分掛かる
- ・米はといだ後、水に浸し、30分以上待つてから炊く。
- ・ご飯をお皿に盛るのに5分
- ・カレーをご飯の上に盛って、整理されたテーブルに乗せるのに5分。
- ・テーブルを整理するのに10分

①問題

- a. 上記の条件で、スタートしてから、「何分後」に食事が出来ますか？
- b. コンロが2個使えたらどうなりますか？

※①の回答——〈図-6〉に示す通り、aでは115分、bでは75分となる。
 コンロが2個だと → (40分) だけbが早くなる。

〈図-6 「ライスカレーが食べられるまで」〉



IV. 実戦的な改善策：Fit チャートの活用

では、我々が実際にプロジェクトを管理／推進していくうえで、どのような管理ツールを使用すればよいのだろうか。プロジェクトの現場では、(a) ガントチャートの分かりやすさ（直感性）、簡便さと、(b) PERT 図の持つ全体的相互関連が分かるもの、との両方を備えたチャートがあればよい。さらに (c) 作業の進捗が時間の経過に合わせてダイナミックにとらえられるものが望ましい。このような合理的ツールはすでに一部では使用されてきたが、これまで明確に説明したものが無かった。本稿ではこれを「Fit チャート」（仮称）と名付け、以下に説明を行う。

1. Fit チャートの作り方

Fit チャートとは、実際に進捗を見る際に便利に使える図表で、基本的には既存のガントチャートと PERT 図を併せたもの、と言える。ポイントは以下のとおりである。

(1) 計画作成時点のポイント

- ・基本的にはガントチャートとして、作業工程あるいは使用資源毎の「時間軸」で線が引かれる。時間の量的判断は成果物のほか、プロジェクトリーダーの「見通し、予測」で行う。担当者がすでに決まっている場合には要員特性（スキルなど）を加味して作成する。
- ・各作業は相互の順序性（前後関係）、依存関係を最大限に明示し、作業相互の関連と因果関係を結線で明記する。

(2) 実施時点の記入ポイント

- ・すべての作業の工程進捗を定期的にチェックし、そのたびに「その作業があと何日あれば完了するか」の見通しを決め工程表の作業線上にマークを記し、チェック時点とそれらのマークを縦方向にすべて接続する。この時できる「ジグザグ」線を「稲妻チャート」とも呼ぶこともある。
- ・進捗はすべて時間（期間）で把握し、「完成までの所用時間がどのくらいか」で行なう。つまりあと何日（あるいは何週間）で作業が完成するか、を予測して線表に現すのである。この「あと何日（あるいは何週間）」というのが実は重要なのである。日数の根拠として成果物を使う事はもちろん望ましいことである。（例えば、設計段階ならば、システム仕様書のページ数を、あるいはプログラムテスト段階ならば、テスト仕様書のテスト消化項目数、など）但し、重要なのは「あと何日で終わるか」と残日数を明記することである。そこには作業者個々人の人格特性や予想されるトラブル（職場の人間関係、健康問題、家族事情なども含め）が総合的に加味されているからである。

- ・一度記入したジグザグ線は消さずに残し、進捗チェックが次回、次々回と時間軸で進むにつれ累積記録していく。すると、工程が進むにつれて徐々にジグザグ線の縞模様が出来上がるようになる。これはちょうど木材の「年輪」のようなもので、それまでの経緯がダイナミックに表現されたものになる。

2. Fit チャートの使い方

ジグザグの縞模様が入った Fit チャートは作業工程の実態を良く現わしてくれる。縞模様は過去のチェック時点ごとの進捗ポイントを線で結合したもので、その時点の進捗だけでなく、経過もダイナミックな推移として示してくれる。つまり進捗が順調だったか否かを如実に現わしてくれる。具体的には以下のようになる。

①<予定通り>

縦線がチェック時点と同じ（垂直になる）作業は予定通り順調に進捗しており、完成時期を守れる見通しである。理想的な進捗を示している。

②<遅延、又は前進>

縦線が「<」になって左側に凹んでいる作業部分は、予定よりも遅れていることを示す。逆に右側に飛び出す「>」の部分は作業が予定より早めに進んでおり、作業完了が「前倒し」になっていることが分かる。

③<進捗ペースは変わらず>

さらに、縦線から出来る縞模様が平行に揃っている場合は、その期間は「同じペース」で進んできたことを示す。例えば「遅れがまったく改善されていない」場合などもすぐに分かる。

④<遅延の拡大>

縞模様の「間隔が狭く」なっている場合、そこは作業の「進捗速度の低下」がある。時には「作業の遅延がさらに拡大」していることを示すことになる。その該当作業については「実状を念入りに重点チェック」し、必要に応じて「抜本的な支援／強化」をしなければならないことが分かる。

⑤<前進の拡大、または遅延の縮小>

逆に、縞模様の間隔が広がってきた場合は、作業がそれ以前より早めに進んでおり、従って作業の完了時期が早まる可能性を示す。この場合も原因分析*を行って単なる楽観論なのか否かの見極めを行い、本当に順調ならばその余力を早めに他の遅延している作業の応援に回すとか、「作業予定や要員配置を早めに見直す」ことにもつなげる。

*原因としては・作業量の見積りが過大だったのか、・状況が当初と異なってきた（ex. 機能が一部不要になった、etc.）のか、・あるいは作業担当者が予想以上に優秀だったのか、などである。最悪のケースでは、進捗報告が間違っている（ミスや虚偽）場合もあるので、

単純に喜んだり油断したり、管理を甘くするようなことは慎まなければならない。（順調という報告は鵜呑みにせず、報告が本当に正しいか否かの吟味が常に必要である）

3. Fit チャートのサンプル

Fit チャートの例として〈図-7〉に情報システム開発の架空事例の全体工程表を示した。〈図-7 「プロジェクトZ ソフト開発工程表」〉にあるとおり、この図例はソフト開発部署の使う工程表で、全体工程との関連を見ながら、ソフト開発の工程を詳細に現わしている。3月下旬までの毎月1回、進捗実績を記録してある。

たとえば、当初は業務設計が遅れ気味で、ソフトの共通部分が先行して作成されていた。しかし業務の詳細設計の遅れから、入力処理や編集処理に1カ月以上の遅れが目立ったが、2月下旬には回復の見通しが立ってきている。この間、具体的には要員のスキルや健康問題、あるいは仕様のミス、誤解などがあつたであろうが、都度対策を打って工程の遅れを取り戻してきたことが示されている。

V. おわりに

これまでに述べてきたFitチャートは、今まで使われたチャートの実戦的な改善版である。その特徴は、縞模様（年輪模様）から作業進捗がダイナミックに実感できることで、なにか問題がある場合は縞模様の間隔の変化から読み取って、早めに問題を捕まえることが可能になる点である。微妙な変化から大きな潜在的問題を早めに引き出すことのできるメリットは大きい。プロジェクト管理では小さな変化を見逃さないことが大きな問題への早期対処につながるからである。その結果として、当初に計画された工程を守り、プロジェクトを成功させることが出来るのである。

本論が読者のご参考になれば幸いである。

〈参考文献〉

- ・WEBサイト：「Itpro」
- ・WEBサイト：フリー百科事典『ウィキペディア（Wikipedia）』
- ・WEBサイト：IT@上マネジメント用語辞典
- ・WEBサイト：Weblio
- ・「ガント式工場管理法―作業、賃金及利潤」 麓三郎 訳／（日本評論社出版部）

