ソフトウェア工学 71－13
（１９９０．２．９）

プロジェクト管理シミュレーションモデルの構築・評価

上村松男[1] 山田茂[2] 広中正彦[2] 尾崎俊治[2]

1 日本電気マイコンテクノロジー㈱ 2 広島大学工学部(第二類)

ソフトマネージメント活動には、生産管理活動と生産技術に関する技術管理活動がある。本論文では、ソフトの生産活動で最も重要な工程管理、費用管理および品質管理をマネージメント活動と関係づけて、その管理上の問題点や対策とその課題を指摘した。特に企画・計画段階の重要性については、プロジェクト管理が進むにつれて、ソフト製品化過程に生ずる種々の要因と問題が複雑に絡み合い、時間経過に伴って見えにくくなるためである。このことが、見える管理が要求される所以でもある。このためには、これらの計画管理を体系的、一貫性のある統合的なマネージメント管理システムが必要となる。そこでマネージメント技法である工程管理モデル、費用管理モデル、品質管理モデルを実プロジェクトから検証し、プロジェクト管理におけるマネージメント活動ネットワークチャートを構築した。さらに、プロジェクト管理シミュレーションモデルの有効性を、業務アプリケーションの開発期間の最適化から評価した。

A Construction and Evaluation of Software Project Management Simulation Model

Matsuo Uemura
NEC Microcomputer Technology, Ltd.
13-7, Shibaura 2-chome, Minato-ku, Tokyo 108, Japan

Shigeru Yamada, Masahiko Hironaka, and Shunji Osaki
Department of Industrial and Systems Engineering
Hiroshima University, Higashi-Hiroshima-shi 724, Japan

A Software project management activity consists of two major management factors: Production control and engineering management related to software engineering technologies. In this paper, relating schedule, cost, and quality controls which are very important aspects in the software development process control to the management activities, we discuss theissues on the project management and their decision makings. That is, the schedule control model assisted by a SLAM II (Simulation Language for Alternative Modeling II) tool, the cost control model based on the successive approximation method for an orthogonal array analysis, and the quality control model by using the SOREM (SOftware Reliability Evaluation Method) system incorporated software reliability growth models, are proposed.

1 まえがき

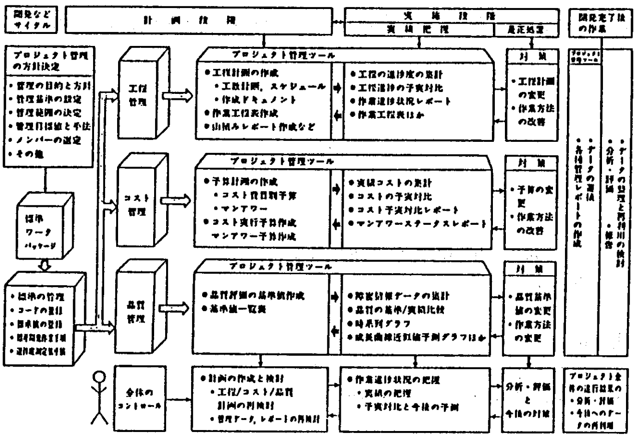
情報システムの高度化・複雑化にともない、プロジェクト管理（企画・計画・事業）には多大な努力が払われている。

ソフトマネージメント活動における生産管理[1]は、予算・費用管理、要員教育・育成管理、文書管理、顧客・ディーラ折衝／管理、費用／工数見積り、受注管理、事業戦略／製品計画、外注折衝／管理、負荷管理、進捗管理、品質管理、人事・勤労・労務管理、組織／制度設定・計画、生産環境・武装化計画、流通・在庫・仕掛管理、運用・保守計画など数多くあるものの管理者の管理事項の一部にすぎない。これらの管理事項以外にも生産技術に関した技術管理事項が山積みしている。これらの計画や管理は、プロジェクト管理が進むにつれてソフト製品化過程に生ずる種々の要因と問題が複雑に絡み合い、時間経過に伴って実態を見えにくくしている。このために、見える管理が重要になっている所以でもあり、これらを体系的、一貫性のある管理が必要である。

本論文では、生産管理のマネージメント技法である工程管理モデル、費用管理モデルおよび品質管理モデルを検証し、マネージメント活動を支援するプロジェクト管理モデルの構築法とシミュレーション実行結果の有効性を評価したので報告する。

2 プロジェクト管理

生産管理で最も重要な要素として、工程管理がある。開発が分散化した形態では、納期を尊守する上で難しい問題である。費用管理については、ソフト活動の特性から非常に計測が困難である。また一方、品質管理については、顧客の要求品質を満たしているかなど最終品質を保証する機能を有しており、これらの管理をいかに経済的に効率よく推進するかがプロジェクト管理のポイントである。（図１参照）



出典）「第2回ECレンポジウムテキスト」, 情報処理振興事業協会(IPA)レグマシステム開発本部, 1987年6月.

図1 プロジェクト管理

生産管理業務では、ソフト関連に限らず企画・計画段階が最も重要である。幸いにソフトにはライフサイクルの概念を持ち、ライフサイクルを意識した計画・立案が前提である。

そこで計画には目標が何かを規定する事業方針計画、なにを開発するかを明確にする戦略計画、さらにはいかに開発するかを具象化する戦術計画があり、それぞれに対応する開発計画、作業計画、要員計画、システム技術計画、変更管理計画書など策定する必要がある。いずれにせよ、目標・指標の設定と管理による生産計画が生産性向上の原点となる。よって管理者は、先の計画がPDCAサイクルを回し、プロジェクトマネージメントの観点と技術管理の両面から計画にそった実行の統制および組織的・継続的に実施するための制度作りが必要である。表１は、現実のソフト生産管理面での問題点を体系化し、計画を重視し管理サイクルを回すことに重点をおいたプロジェクト審査制度を示したものである。

2.1 工程管理モデル

生産管理で最も重要な工程計画は、工程計画の妥当性、工数計画の妥当性、開発製品の新規性・難易度、および開発要員の妥当性などを考慮して行う。一方、工程管理では、工程の定義と成果物（作業、手順）の規定、関連プロジェクトとの適合性、各工程の成果物の確認、開発の各工程が計画通りに完了していることの確認、マイルストーン設定と作業ネットワーク、各工程の進捗度合の把握、工程遅れや仕掛りの検出、工程推進の立案、および関連部門との調整などの作業が必要となる。本節では、工程計画を2.1.1節のライフサイクルモデルの時間軸の上で議論し、工程管理をソフト生産の各工程の生産物に着目し2.1.2節でその対象を明確にする。

	契期	主な内容	主な規定書類
計画審査	発足時	プロジェクト発足時に開発方針、要員計画、予算等についてフィージビリティチェック、評価、査定および承認を目的とする	プロジェクト計画書 システム開発計画書（付属資料）
・開発審査	設計・製造時	設計工程におけるシステム方式、設計品質等設計審査（D．R）を主な目的とする 開発方式面で標準規定の遵守、ノウハウの活用等を促進し併せて計画と実行との差異の分析、アクションを行う。	設計審査関連資料（チェックリスト検討項目表等） 計画審査資料の最新版
総括審査	完了時	完了時点で、予算面、技術面での計画と実績との差異を総括し、実績収集と評価、問題点の抽出と改善等について、反省、まとめを行う	プロジェクト完了報告書 計画・開発審査での総括資料

表1 プロジェクト審査制度

2.1.1 ライフサイクルモデル

ソフトは一般に、図２のようなライフサイクルにしたがって開発が進む。すなわち、ソフトを開発することが必要になると、要求仕様を作成する。これをコンピュータの世界の言葉で定義したものが外部仕様である。この外部仕様は機能(what)を中心に表現してある。これをどのように実現するか(how)を記述して内部仕様を作る。次に、実際のプログラム・コードを製造する。この生産物がはじめの要求仕様と一致しているかどうか、また、あらかじめ設けた基準に適合しているかどうかを確認する行為が検査である。図２には保守活動を書かなかったが、ソフトに対する修正もしくは改造要求を新たな要求仕様とみなせば、保守もこのサイクルに含まれる。

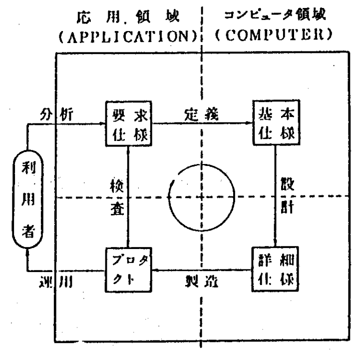


図2 ライフサイクルモデル

まず、要求仕様から外部仕様を定義する段階においては、開発に必要なリソースを見積る必要がある。特に、ソフトの規模、作成期間、要員数（もしくは工数）などが、この時点での重要な見積り対象である。この分野は管理面からの要請も多く、比較的古くから多くのモデルが提案され、発展してきた。設計および製造段階では、プログラムの複雑度が計測の対象となる。複雑度とは、いわばそのプログラムの難しさを表すもので、もし計測ができれば、コーディング時間、エラーの発生頻度、検査や保守のしやすさの予測、問題の生じそうなモジュールの摘出などに使うことができる。しかし、その計測自体がやや面倒であることや、この尺度で測った難しさと人間の考える難しさとが必ずしも一致しないことなどから、現段階にあっては現場への普及はいま一つ進んでいない。

製造および検査段階においては、ソフトの品質がメトリクスの対象である。一般にソフトの品質を表すものとして、①使いやすさ（信頼性、効率、マンマシン・インタフェースなど）、②保守性、③移植性、の三つの特性があげられる。このうち特に信頼性については研究が盛んである。検査工程が独立するにつれて現場への適用も進んでいる。この分野が進んでいる理由の一つは、信頼性がソフトの商品としての価値だけでなく、製造や検査、保守のコストを大きく左右するからである（当然ながら、エラーの生じにくい設計技法やツールに関する研究開発も多い）。このほか、ソフトの「良さ」を総合的に評価する試みが始まりつつある。これは、品質メトリクスとも呼ばれている。この分野は、計測自体の客観性よりも指標的な性格が強く、実用面での普及が先行する分野となろう。

2.1.2 ソフト文書

ソフト文書の作成は、ライフサイクルモデルの中心的な課題であると同時に、製品化にとって重要な要素である。

分析段階とは、システムライフサイクルの最初の段階の現状の調査、検討を実施し、システムに対して要求される要件を明確にするものであるが、この段階では利用者の意向を記載した“要求仕様書”が作成される。開発段階とは、システムライフサイクルの第二段階。“要求仕様書”で規定された要件を満たすシステムを、“定義”、“設計”、“製造”、“検査”の４工程を経て具体化し、システムに関する技術情報を記載したソフト文書、およびプログラムが作成される。運用段階とは、システムライフサイクルの最後の段階。開発したプログラムを利用者環境として設置されたハード上で動作させ、システムのサービスを提供する。システム要件に変化が生じることにより、新たなライフサイクルとして、運用段階から分析段階に戻る。ソフト分析／開発／運用段階における作業工程とソフト文書との位置付けを図３に示す。ソフト製品化段階における文書作成過程をさらに詳述すれば次のようである。

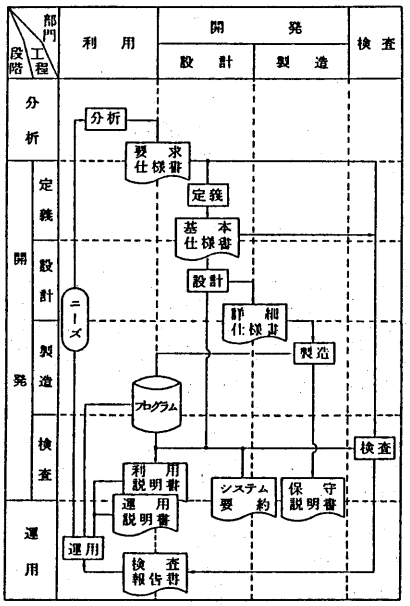


図3 ソフト製品化と文書

定義工程とは、“要求仕様”で規定されたシステム要件を定式化し、システムを構成する人間系、ハード系、ソフト系、三者間の分担、および各々の機能が何であるかを明確にする工程で、“基本仕様書”が作成される。設計工程とは、“基本仕様書”で規定された機能をいかに実現するかの具体方法を検討し、可能な限り詳細（プログラムの作成ができる状況まで）具体化する工程で“詳細仕様書”が作成される。製造工程とは、“詳細仕様書”に基づいてプログラムを作成し、“基本仕様書”、“詳細仕様書”に対応した試験を実施する工程でプログラム、“利用説明書”、“運用説明書”、“保守説明書”、“システム要約”が作成される。検査工程とは、作成したプロダクト（プログラム、“利用説明書”、“運用説明書”、“保守説明書”）が“要求仕様書”を満たしているか確認し、出荷品質を保証する工程で、“検査報告書”が作成される。

２.２　費用管理モデル

ソフト開発や保守において、プロジェクト管理や生産管理などに費やした人件費、工数、マシン費用などを経済的にかつ効率よく計画通り進捗しているかを見積りによって管理するのが費用管理の目的である。具体的には、予算と費用の関係、人員計画と要員配置、見積りと実績の差異分析、標準原価などの費用構造、損益収支（損得）および生産性指標と改善効率などが主な作業である。

ソフトの開発は、開発環境においては方法論、開発技法・手段やツールの発達はあったとはいえ、依然として人間の頭脳労働に負うところが多い。このことがコストや生産性に最も影響する重要な要因である。さらにソフト自身の特性により機能、性能・品質の達成度合いを左右する要因が考えられる。このような背景から見積り手法は古くから研究され、今日数多くのモデルが提案され実用化されている。

本節では、これらの変動要因や相互作用による様々な現象を考慮した、メタモデル方式にもとづくＮＥＣコストモデル[2]で採用している最小二乗法と、新たに提案する直交表による逐次近似法[3]の２つの技法について、モデルの適合精度と生産管理の適正化を比較、検討する。

２.２.１　ソフト生産性要因

生産性とは開発・保守コストと品質要素の相互関係で示される。コストは規模によって導出され、品質は人間要素と環境要素によって得られる。これらの生産性要因は人間の創造的活動をはじめ、人間をとりまく組織、体制などによる要因を計測することは困難である。このため多くの要因間には、計測基準の設定に伴う誤差や対象として考慮しなかった要因との交互作用による影響、およびプロジェクト特性にもとづく測定誤差などにより強い相関がみられる。このような複雑な構造を分析するために、必要に応じて因子分析などにより独立性の評価を行い、事実を反映した説得力のある因子群に要約した。表２はＮＥＣコストモデルの１１の生産性要因を示したものである。

生産性要因	β	T(β)
X_1　開発ｸﾞｰﾙ	0.749	2.041
X_2　開発技法・手法	-0.109	-0.334
X_3　対OS・ﾊｰﾄﾞｲﾝﾀﾌｪｰｽ	-0.459	-1.017
X_4　ﾒﾝﾊﾞｰの経験年数	-2.446	-4.256
X_5　分担形態	0.079	0.340
X_6　仕様変更度	2.227	7.010
X_7　ﾕｰｻﾞｰ特性	0.267	1.247
X_8　製品の汎用性	0.653	0.974
X_9　新規性・難易度	-0.753	-0.728
X_{10}　品質要求	0.892	2.318
X_{11}　ﾁｰﾑの能力	-0.258	-0.502

CONSTANT　5687.010

表2 生産性要因と偏回帰係数

２.２.２　解析法

生産性要因の計測基準で計測した生産管理データによって、回帰式を導出する技法としての最小二乗法と逐次近似法について概説する。さらに、これらの技法にもとづくコストモデル作成手順と解析結果について述べる。

（１）　最小二乗法

業種、業界をもとに開発形態の類似したシステム２０プロジェクトを選定した。目的変数として実工数Ｙ、説明変数としては表２の生産性要因の開発ツールをX_1、開発技法・手法をX_2、…、チームの能力をX_{11}として(1)式に代入し、最小二乗法による重回帰分析を行い、表２で示す偏回帰係数とｔ値を得た。

$$Y=\beta_1 X_1+\cdots+\beta_{11} X_{11}+C+\varepsilon \qquad (1)$$

表２から、ソフトの生産性に影響のある要因について検定・推定を行った結果、分散分析のＦ検定では５％有意で、適合度も重相関係数が99.8％と（見かけ上）良い結果が得た。しかしながら偏回帰係数をみると負の値を示しているものがあり、その要因のみに着目すれば生産性計測基準から矛盾している。このことは要因間において相互作用が働いたためなどである。このように適用上意味のない係数をうのみにして生産性を評価することは危険である。

（２）　逐次近似法

逐次近似法は過去の生産性分析から得られた係数をもとに、先の最小二乗法の偏回帰係数を考慮して３水準を設定し、直交表にしたがって１１の生産性要因に水準値を割りつけて回帰式を評価し、適合度をみながら段階的に水準の幅をせばめていく手法である。以下に、その生産性係数の決定までの手順を示す。

手順１　水準値の設定

生産性の係数は最小二乗法の係数がとりうる範囲を等間隔の３水準に設定する。ここで範囲の決め方は２つの方法がある。例えばの開発ツールの係数は０．７４９であるため水準Ⅰに０、水準Ⅱに０．５、水準Ⅲに１の範囲を等間隔で設定する方法である。もう一つは過去の実績から設定する方法である。ここでは比較、評価を容易にするために最小二乗法の偏回帰係数を考慮した方法を採用した。

手順２　直交表の割りつけ

生産性要因を３水準に割りあてるとしたら、その組合せは膨大となるが、直交表を用いることによって３６通りの実験で全ての要因の組合せが評価できる。そこで最小二乗法の(1)式と同様に生産性要因を代入し、実工数との差

$$Y-(\beta_1 X_1+\cdots+\beta_{11} X_{11})=C+\varepsilon \qquad (2)$$

を求めると２０プロジェクトのＣ推定値C_1、C_2、……、C_{11}が得られる。これらの合計をＴ、平均を$\bar{C}$、変動をＳとすると、Ｔ、Ｓは式(3)、式(4)で

$$T = C_1 + C_2 + \cdots + C_{20} \quad (3)$$

$$S = (C_1 - \bar{C})^2 + \cdots + (C_{20} - \bar{C})^2 = C_1^2 + \cdots + C_2^2 - T^2/20 \quad (4)$$

と表わされる。次にL_{36}直交表によりβ_1、…、β_{11}の各水準に対応させて割りつけ、式(3)および(4)の計算によりT、Sの値を算出する。ここではC₁、C₂、…、C₂₁と$\bar{C}$との差の二乗和であり、Cの推定値の変化の大きさを示している。したがってSが小さいということはCが安定した値であるということであり、回帰式の適合性の良さを示している。係数β_iの各水準を評価するために直交表の割りつけ規則にしたがって同一水準毎のSの合計を表している。

手順3　水準の比較

手順2で得られた結果を利用して、3水準のSの値を比較する。例えばSの値が各水準とも同じであればどの水準を係数としても誤差の大きさは等しくなる。もし水準IIのSが非常に小さく、水準I、水準IIIが大きい場合には係数は水準II近くに存在しているとみなせる。このように各水準のSの変動の大小関係をいくつかのパターンで表し、そのパターン毎に水準間の有意差検定を行う。

手順4　有意差検定

有意差検定には式(5)の分散比Fを用いる。

$$F_0 = (1/20 \mid \overline{S(I,J)} - S(I,J) \mid) / V_e \quad (5)$$

ここでVₑは最小二乗和で得られた誤差分散である。もし、いずれかの要因の水準間に有意差が認められれば、各々のパターンの規則にもとづいて2分法により水準の範囲を決定する。

手順5　生産性係数の決定

有意差検定で、全ての生産性要因の水準間に有意差がないと認められるまで、再度手順1の水準の設定からの処理を繰り返し、Sが安定するまで水準の範囲をせばめていく。これら一連の処理により最終的に得られた3水準の割りあて結果を表3に示す。生産性係数は得られた3水準のうちの水準IIを採用した。

推定	水準 I	水準 II	水準 III
β_1	0.484	0.500	0.516
β_2	0.219	0.234	0.250
β_3	0.172	0.188	0.203
β_4	0.234	0.250	0.266
β_5	0.016	0.031	0.047
β_6	0.000	0.063	0.125
β_7	0.109	0.125	0.141
β_8	0.484	0.500	0.516
β_9	0.047	0.063	0.078
β_{10}	0.047	0.063	0.078
β_{11}	0.047	0.063	0.078

表3 生産性係数

2.2.3　考　察

最小二乗法と逐次近似法の2つの技法により得られたコストモデルの妥当性の比較を行うとともに、各々の生産性要因の分析によって工数削減や生産性指標について考察する。

（1）　妥当性評価

本技法により得られたコストモデルの評価の判断尺度としては一般的な評価としては、寄与率がある。さらに解析に使用したプロジェクトを残しておき、誤差平方和による評価を行った。寄与率は工数の変化を何%各要因が表現しているかを示すものであり、式(6)で表される。

$$\rho = [\, S_e / S_T \,] \times 100 \quad (6)$$

ここで、S_Tは項数の全変動、S_eはコストモデル式の誤差変動である。

この結果、最小二乗法では99%、逐次近似法では89%であり10%の差が認められた。モデルの妥当性としては、最小二乗法の方が若干良い結果が得られている。

一方、誤差平方和は解析に使った20プロジェクト以外の残り10プロジェクトを使って評価した結果、最小二乗法では2.6×10^{10}、逐次近似法では0.9×10^{10}であり最小二乗法の方が2.8倍の差が認められた。この分析結果から、図4に示すように逐次近似法の推定値の方が10プロジェクト中の7プロジェクトが管理実態とよく符合していることが判明した。

以上、これまで行った2つの技法を比較すると、概して逐次近似法による誤差平方和は小さくなる傾向にある。さらに水準の初期設定は、必ず正の値を取りうるので偏回帰係数による負の値は得られない。このことは各要因の工数に及ぼす影響や生産性分析を行う上で好都合である。

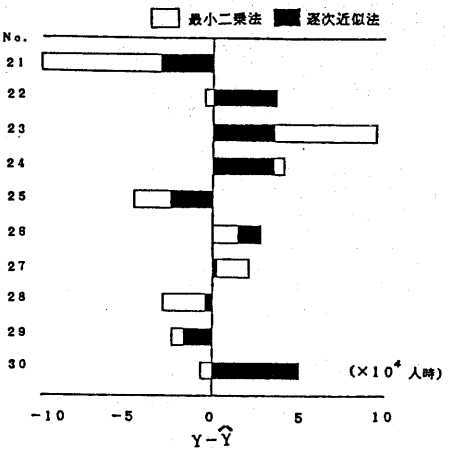


図4 誤差分析

（2）　生産性分析

生産性分析は、各要因が生産性に与える度合いを工数（人時）と生産性（ライン／人時）で計量化し、管理指標を提供するものである。図5の生産性分析は、各要因の特性を過去の実例から重みを加えて工数にかかる影響度を比率尺度によって、先の20プロジェクトを対象に逐次近似法で計算した一例を示したものである。図5で示す技術水準とは全ての要因が平均値をとった時の工数をいう。

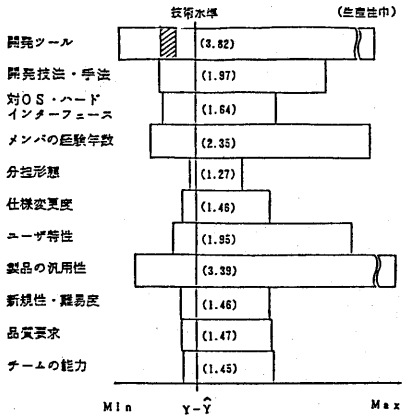


図5 生産性分析

生産性幅はある要因に着目した時、全ての要因を平均とみなしてその要因の生産性が最も高い状態をMinとし、最も低い状態をMaxとしてMax／Minの比で表される。例えば開発ツールの生産性幅は、他の要因を一定にした時、382%変動し得ることを示している。そこで具体例としてプロジェクトのソフト規模が100KLを開発するとした時、開発ツールとして高級言語（COBOL）が40%、システム記述言語（C言語）が10%、そして低級言語（アセンブラ）が50%の配分で実施すると、(7)式から59250人時の投入工数が必要になる。このことから、後者は前者に対して削減工数が5000人時となり（斜線部）、後者の生産性指数としては18.4L／人時が開発環境や努力を考慮した管理指標となる。

$$Y = 0.5X_1 + 0.234X_2 + 0.188X_3 + 0.25X_4 + 0.031X_5 + 0.063X_6 + 0.125X_7 + 0.5X_8 + 0.063X_9 + 0.063X_{10} + 0.063X_{11} + C \quad (7)$$

以上、生産性分析は一般にこれらの要因の組を選定し改善幅の大きい要因をとりあげてプロジェクトの生産性向上目標や改善施策を検討することができる。

2.2.4 費用見積りモデルの課題

ソフト生産性（経営的観点）や生産効率（技術的観点）の評価を進める上で、特に前者の経営的観点で生産性を議論することが多くなってきている。工数見積りの基本となる投入工数に対する成果物の一つである規模（以外にもソフト文書などもある）を表す機能量を測定する指標がない。ファンクションポイントが提案されているものの、重要な考え方であり今後の検討課題である。また、プログラムの論理的な大きさの測定や非手続き型言語の計測、および他言語との生産性比較など確立された方法がないのが実状であり、早急な対策が必要である。

2.3 品質管理モデル

システムは高度化・多様化し高付加価値をもたらすためにシステムインテグレーション指向に進んでいる。ソフト品質（使用性、保守性、移植性）に対する要求は増々厳しくなって来ており、品質目標や保証基準の設定および品質管理手法や制度の確立が急がれている。特にソフト品質において信頼性特性は機能・性能が重要な要素である。

ソフトの信頼性[6]とは「ソフトが要求された機能を規定された条件下で規定された期間、システム故障を起こさない確率」をいう。この確率とはシステムの利用の関数であると同時にソフト内に存在する障害の関数でもある。

一般的にソフト信頼性を評価する方法として、①機能をどれくらい確めたか（原因-結果グラフ）、②内部構造をどの程度網羅したか（論理網羅度）、③エラー発見過程によるエラーの出方はどうか（信頼度評価技法：後述）などのアプローチによる検証が試みられている。

信頼度成長モデルはよく知られているロジスティック曲線やゴンペルツ曲線といった決定論的モデルはあるが、ソフト信頼性データは偶発性を含んだものであるから、確率論的モデルがほとんどである。

ソフトのエラー発見過程を記述するソフト信頼度成長モデルの研究は１９６０年代よりはじまり、今日まで多くの提案がなされている。一般に発見エラー数の成長曲線は、指数分布関数形よりむしろＳ字形を示すことの多いことが経験からわかっている[4][5][6]。

ソフトの信頼性評価方法としては、エラーの定性的予測と信頼度やシステムに潜在する総エラー数の予測は管理者にとってプロジェクト開発計画の際の重要な課題である。

本節では、まずソフト信頼度成長モデル[5][7]に基づきエラー発見過程の推移を把握する。このとき、潜在総エラー数の予測精度などモデルの信憑性は運用面で真に実情を反映した有効なものなのか、という事が実際に使用者から往々にして問題になってくる。そこで、我々は数年来、実用化と導入支援を進めているソフト信頼度評価技法ツールＳＯＲＥＭ[7](Software Reliability Evaluation Method)のソフト信頼度成長モデルの予測性の問題と、運用段階における適用性の問題の２点について議論してきた。それをまとめると以下のようになる。

2.3.1 遅延Ｓ字形ＮＨＰＰモデル

ソフト信頼度成長モデルの予測性を評価するために、遅延Ｓ字形ＮＨＰＰ(Non-Homogeneous Poisson Process:非同次ポアソン過程)モデル[4]を用いた。このモデルは、パラメータの数が少なく、その各々の意味が明確であるという利点がある。さらに、このモデルによって信頼度も定義できる。このモデルでは、時刻ｔまでに発見されたエラーの数M(t)を次のように推定する。

$$M(t)=a\{1-(1+bt)\exp[-bt]\} \qquad (8)$$

ここで、aは潜在総エラー数、bは期待発見率（発見率の上限）を表す。

また、時刻ｔの信頼度R(x|t)は次のように与えられる。

$$R(x|t)=\exp[-\{M(t+x)-M(t)\}] \qquad (9)$$

ここで、式(9)は時刻ｔまでテストが進行したときに、時間区間(t,t+x]でエラーの発見されない確率を表す。今回の分析で適用したソフトが、すべて発見エラー数を１週間ごとに計測していたため、xを7日間とした。

2.3.2 ソフト信頼度評価技法

この分析では、以下に示す２種類の業務ソフトに適用した。

ＣＳ：通信ソフト

このソフトは、デジタル交換機用で、リアルタイム性のきわめて高いシステムである。また、交換システム特有の厳しい運転環境を考慮した、高い信頼性が要求された。システムは、V1という初版から始まって逐次機能が拡張している。V1の全体規模は、50KLである。V2は、V1の機能を拡張したものであり、全体の規模は新規3KL、流用47KL、改造10KLの計60KLである。いずれも Real Time C と呼ばれるＣ言語の一種で記述されている。

ＣＣ：Ｃコンパイラ

これは、リアルタイムＯＳ用Ｃコンパイラである。バージョンは、V1.0から始まって、V1.8までの９つがある。ただし、各々のバージョンアップは、エラーの修正を反映するためのものであった。エラーデータは、V1.0をリリースする以前の検査段階以降、すべてのものが収集されている。ＣＣの全体規模は24KLで、そのうち新規14KL、流用9.4KL、改造0.6KLである。システムはＣ言語で記述されている。

（１） 信頼度評価尺度によるモデルの選択方法

ソフト信頼度成長モデルをプロジェクトに適用する際、まずどのモデルを使用するかを決定しなければならない。その方法の一つとして、図６で示すようにモデル固有のエラー発見過程の傾向を知ることでモデルを選択することができる。次に、どのモデルが最も適合するのかを評価する必要がある。適合性の評価方法としては、カイ二乗検定法やコルモゴロフ・スミルノフ検定法などの統計的な検定や、偏差二乗和や推定精度などの絶対的な指標が有効である。

以上、これら一連の適合性評価や信頼度評価尺度（ＳＯＲＥＭの管理指標［7］）の比較を反復し、最終的に最も適合性のよいモデルを選択することが出来るため、モデルの推定精度の向上が期待できる。

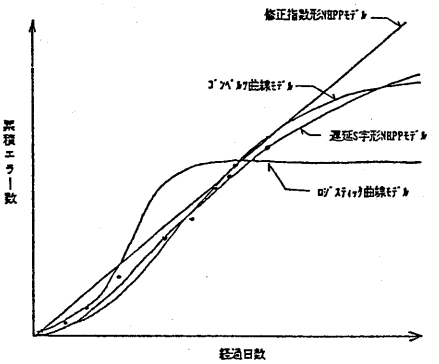


図6 モデルとエラー発見過程

（２） 検査段階におけるモデルの安定

モデルの安定とは、出荷まで正しい予測をするモデルが得られ、それ以後の発見エラー数を入力してもaの値がほとんど変わらなくなった状態（リリース時点の95%信頼区間からaの値がはずれなくなった状態）をいうことにする。つまり、どこまで検査を進めればモデルが安定するのか、ＣＳのV1、V2を用いて検討した。

検査の初期段階 t_k まではaの推定値が激しく変化し、それ以後は、ほぼ同じ値をとることがわかった（リリース時点の95%信頼区間に、aの値が入り続ける）。V2でもこのような傾向が見られた。この t_k がどの時点なのかを検討することで、モデルの安定する時点がわかる。

まず、検査開始から、リリース時点までを100とする。このとき安定する時刻 t_k は、V1が40.5時点、V2が57.4時点であった。この結果からは、安定する時点の規則性が見えない。これは、V1とV2のリリース時の信頼度が、異なるためだと考えられる。(V1は、83.7%、V2は60.0%だった。)

そこで次に、V1、V2とも80.0%でリリースしたと仮定して t_k を計測する。検査開始から、信頼度80.0%の時点までを100とする。このとき、安定する時点 t_k は、V1が41.8時点、V2が49.1時点であった。ここでは、安定する時点が検査段階の中程であることがわかる。また、時刻 t_k の信頼度は、V1が0.0007%、V2が2.97%であり、かなり信頼度の低い時点で安定していることもわかった。

以上のことから、検査工程が40～50%進んだ時点で（または信頼度が、数%の時点）で、モデルが安定し、検査要員の配置、テストのしかたなどを考え直すことができると考えられる。しかし、これはＣＳのみから得られたものなので、今後のデータの積み上げが必要であることは言うまでもない。

（３） リリース後におけるモデルの安定

ここでリリースとは、社内向け限定リリースのことをいう。ＣＣは、リリース後のエラー発見過程が得られている。

図７は、ＣＣの発見エラー数実績である。全体的に、おおむねＳ字のカーブをなしている。しかし、V1.0、V1.5、V1.7の各リリース時に、小さいＳ字の切れ目が見られる。原因は、以下のようなことが挙げられる。V1.0リリース時は、このソフトが初めてリリースされた時であったからである。V1.5の時は、このソフトがVSLIチップに組み込まれ、大量の新しいユーザがついたからである。またV1.7の時は、あるバグが修正され、このことによって、それまで動かなかった部分が動くようになったからである。このような利用環境の変化により、突然エラー発見数が増えることがわかっている。いいかえれば、バージョンアップは、エラーの修正を反映しており、この図はリリースとエラーの増え方の関係をよく表している。モデルが安定しているかどうかは、図８を見るとわかる。V1.2リリース以前は、潜在エラー数の推定値が、激しく変化している。そ

の後、V1.5リリース前までは、潜在総エラー数の推定値が安定した。しかし、その後V1.8まで潜在総エラー数の推定値が増え続けている。モデルの安定に関しても、利用環境の変化に対する影響が大きい。前述のV1.5とV1.7での特徴をよく表している。ただし、V1.0の影響はわからない。これは、V1.0以前の検査段階データを細かくとれば、わかるであろう。

以上のように、潜在総エラー数の推定値はリリースする度に大きくなる傾向がある。リリースすると、テストの十分されていなかった部分がユーザに使われたり、エラーを修正したために使えなかった部分が使えるようになったりする。そのために潜在総エラー数の推定値が、増加するのである。つまり、潜在総エラー数の推定値は、テストの網羅度とも関係する値だと言うことができる。

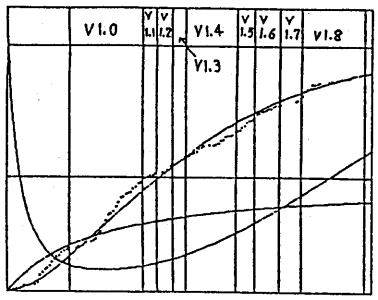


図7 ＣＣの発見エラー数とバージョン

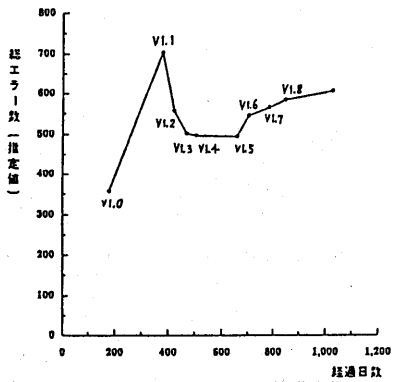


図8 ＣＣの潜在総エラー数推定値

（4） 信頼度を考慮した出荷時期の決定

ソフト出荷問題について、議論するとき考慮しなければならない点は、潜在するソフトエラー数により評価される信頼度と、エラーを発見して修正・除去するために費やされた時間、および修正コストである。ＳＯＲＥＭの最適化は、目標となる信頼度水準を設定し、達成される信頼度と出荷時期の関係から、プロジェクト管理者が、総合的に判断して決定する方式をとっている。つまり、目標信頼度を満足しながら、時間やコストを最小化することが理想である。その理由としては、信頼度は実施される検査技術や、ツールあるいは検査要員の習熟度など、検査時間に依存した複雑な要因の交互作用がプロジェクトによって考えられるからである。以下に、遅延Ｓ字形ＮＨＰＰモデルを使った最適化の適用について議論する。

分析には、図９で示すＣコンパイラの発見エラーデータ（プロット点）を用いて、推定最適化を行った。このデータを使って、生成したモデルM1(t)から推定してみると、潜在総エラー数は約 607個、現時点での残存エラー数は約67個であった。また、期待発見率は約0.0036、信頼度は信頼度区間を 3日として推定すると56.22%が得られた。このとき出荷可能な目標信頼度水準を80%とした時、最適リリース日RL1まで残り340日であることがわかる。

そこで、このモデルをもとに”目標信頼度を満たすべき出荷時期を約100日早める”ことを条件として、最適化を行う。方法としては、潜在総エラー数を一定にして、期待発見率を操作する。このような操作を繰り返した結果、期待発見率を約0.004として生成したモデルM2(t)において、現時点での残存エラー数が約50個、信頼度が62%となった。この結果、最終的に最適リリース日RL2を求めると、現時点からあと250日という値が得られた。このことから、検査がほぼ中間にさしかかっている時、出荷時期の最適化は、検査要員の強化で、信頼度を満たした出荷時期を、予測することが出来ることがわかったので、次に進捗管理が比較的容易な期待値を操作する方法を評価している。

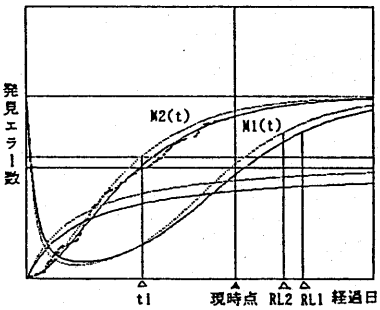


図9 遅延Ｓ字形ＮＨＰＰモデルによる
Ｃコンパイラの最適化

以上、４つの技法について要約したが、これらのどの方法をとったとしても、最終的な決定は、プロジェクト管理者の経験や、凡例によるところが実際には多い。このため妥当性の向上には、多くのプロジェクト実績や、評価経験を積み上げるための努力が今後とも重要と考える。

2.3.3 ソフト信頼度成長モデルの課題

ソフト信頼度成長モデルの今後の課題として、３つの対象と評価尺度例を示す。①ソフト工学技術の評価として、設計審査の結果からシステムテストのパスの選び方、環境設定など新技術に対する良し悪しの判断尺度が必要となる。②生産・開発活動の評価としては、ソフト信頼性はテストの量に比例して増加する。またテスト費用は、エラー発生や改善度合いに非常に深い関係を持っていることから、プロジェクト活動データから観測される客観的な信頼度尺度を設定する。③新規機能や設計仕様変更の評価では、一般にソフトの信頼性は設計変更が加わるにつれて低下する。そこで、開発・運用上の特性を監視し、新しく加えた特性や設計変更を制御できる尺度が要求されているなどへの適応型ソフト信頼度成長モデルの研究・開発が急がれている。

3 プロジェクト管理シミュレーションモデル

プロジェクト管理におけるスケジューリングと負荷計画は、納期遅れ、設備稼働率、工程負荷平準化、在庫適正化など重要な問題である。例えば、この基本となる工程の終了の判断となると、期日になった、予定工数に達した、出荷品質を満たしたなど、その基準は様々である。また、ソフトは個別受注生産方式であるため業務分析や機能分析、いわゆる科学的マネージメントシステムＷＢＳ(Work Breakdown Structuer)技法による活動の詳細化など非定型作業による計測をはじめ、定式化が難しい。さらには、製品仕様は顧客仕様の傾向が強くなり（付加価値）、しかも製品のライフサイクルが極端に短縮される。このような傾向は、ソフト事業においてますます強まるであろう。このようなことから、新製品の導入や先の諸問題に素早く対応できるシミュレーションシステムの構築が望まれている。

一方、シミュレーション技術の発達によりＳＬＡＭⅡ[9](Simulation Language for Alternative Modeling)に代表されるような優れた汎用シミュレーション言語が開発され、効率的に工程の定量的評価を行うことが可能となっている。本システムの特徴としては、1979年に開発された離散型／連続型共用のシミュレーション用言語でパソコン版も提供されている。こういった汎用性を有することから米国を中心に、生産ライン、在庫管理、スケジューリング、輸送網、通信網、コンピュータシステム、軍事など、極めて多岐にわたるシステムの設計・分析・運用に適用されている。

3.1 モデルに基づくプロジェクト管理技術

一般的なプロジェクト管理技術としては、工程管理ではネットワーク技法、費用管理ではＷＢＳ手法、品質管理では統計手法が過去の論文から多く用いられている。

シミュレーションモデルの具体的な活用／用途としては、①プロジェクト管理として、検査段階における投入工数、進捗、出荷時期、品質状況、テストケース／項目の妥当性、検査環境の整備など、②体制／制度あるいは組織診断としては、分散開発形態、受入検査、外注品質、外注工数、発注期間、外注スキル、適正な契約とソフト購買、契約形態の改善など、③システム維持／運用および営業支援活動として、リリース後の品質（障害処理）、新規（次世代）システムへのアプローチ、システムの寿命予測、セールスポイントの決定などが考えらる。本節では、プ

ロジェクト管理の進捗に関する評価を試みたので概説する。なお、汎用シミュレーション言語としてＳＬＡＭⅡを用いた。

３.２　シミュレーション

プロジェクト形態には、①個別受注生産プロジェクト、②既存システム保全プログラム、③研究開発プロジェクトなどがある。

プロジェクトとは、明確な達成目標をもち要求機能品質を実現するための作業体制・組織である。このことから、プロジェクトは技術・製品化ノウハウの蓄積や知的財産権（特許・著作権・商標）などを確保する上で一般的に体系化されることが多い。

マネージメント活動をサブプロジェクト単位に労力を配分すると表４となり、プロジェクト構成ネットワークに割り当てられる。このPERTネットワークフローにおいて、各アクティビティは対応するプロジェクトの作業を表す。ここで一つのアクティビティの作業を行う要員の集合をプロジェクト単位と呼ぶ。各プロジェクト単位への要員数の最適配分は、線形計画法を応用することにより決定できる。

次にソフト生産管理は次の手順で行う。①要求仕様定義をもとにシステム分析を行う。②労力投入配分と最小プロジェクト単位まで掘り下げたスケジュールを立てる。③PERTネットワークで示した開発計画が計画通りに進行するかどうかを診断するシミュレーションを行う。④③で得た情報を②にフィードバックする。

記号	内容	期間（三角分布）		
		最短	最尤	最長
A	要求仕様定義	0.8	1.0	1.2
B	開発計画	0.5	0.5	0.7
C	基本設計	0.4	0.5	0.6
D	機能設計	0.7	0.8	1.3
E	詳細設計	1.3	1.5	1.7
F	コーディング/単体テスト	2.0	2.3	3.0
G	結合テスト	0.8	1.0	1.5
H	既存プログラムの流用	2.5	3.5	3.8
I	結合テスト	1.6	2.5	2.8
J	DB設計	2.5	3.0	3.3
K	DB構築	2.5	2.8	3.3
L	統合テスト	0.8	1.0	1.3
M	機能設計	0.7	1.0	1.1
N	詳細設計	1.3	1.4	1.6
O	コーディング/単体テスト	2.2	2.5	3.0
P	基本設計(通信系)	0.7	1.0	1.2
Q	機能設計	0.8	0.9	1.2
R	詳細設計	1.3	1.5	1.7
S	コーディング/単体テスト	1.8	2.0	2.4
T	結合テスト	1.8	2.1	2.5
U	開発計画	0.5	0.5	0.7
V	テストケースの設計	1.3	1.4	1.6
W	評価パッケージの開発	1.3	1.5	1.6
X	評価プログラムのテスト	2.2	2.4	3.0
Y	結合テスト	1.5	1.9	2.5
Z	製品テスト	1.0	1.2	1.5

表4 費用見積りモデルに基づく期間配分

３.２.１　ソフトマネージメント活動モデル

シミュレーションの遂行にあたり、次のようにモデル化を行う。

(1) 各アクティビティの相互関係はPERT図で表現する。
(2) 複数のアクティビティが一つのノードに集まる場合は、すべてが揃わなければ次工程に移行できないものとする。
(3) 各アクティビティの完了期間は、過去の経験と分析結果から推測した分布に従う。

本モデルの特徴は、完了期間の確率分布を導入した点にある。通常のクリティカルパスを用いる手法では前述のソフト開発における問題点である不確定要因による作業の遅れや、その影響を考慮できない。

３.２.２　PERTネットワークフローの構築

モジュールの流れはＳＬＡＭネットワークチャートで表現する。これは、PERTネットワークモデルの各アクティビティに、その作業の完了期間の分布を付記したものである。各ノードでは多様な情報を保持でき、結果の出力時に利用できる。これをソース（始点）からシンク（終点）まで100回程度繰り返しシミュレートすることで、進捗度のばらつきの様子を推測する。

各プロジェクトの構成として、A-B-C-D-E-F-G-L-Zは業務アプリケーション、A-B-C-H-I-L-Zは既在プログラムの流用、A-B-C-J-K-L-Zはデータベース、A-B-C-M-N-O-T-Zは通信関係業務、A-B-P-Q-R-S-T-Zは通信系、A-U-V-W-X-Y-Zはテスト関係のパスを表している。（図１０参照）このＳＬＡＭネットワークチャートをプログラム化すると、シミュレーションプログラムが生成される。

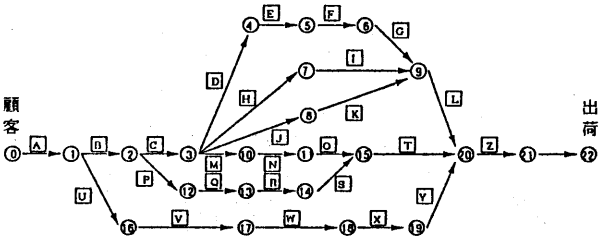


図10 マネージメント活動ネットワークチャート

```
GEN,NEC,SOFTWEAR PERT NETWORK,2/11/1990,
LIM,,1,700;                    100,,NO,,NO,YES/100;
NETWORK;
        CREATE;
        ACT,TRIAG(0.8,1.0,1.2),,N1;
N1      COLCT,FIRST,NODE 1;
        ACT,TRIAG(0.5,0.5,0.7),,N2;
        ACT,TRIAG(0.5,0.5,0.7),,N16;
N2      COLCT,FIRST,NODE 2;
        ACT,TRIAG(0.4,0.5,0.6),,N3;
        ACT,TRIAG(0.7,1.0,1.2),,N12;
N3      COLCT,FIRST,NODE 3;
        ACT,TRIAG(0.7,0.8,1.3),,N4;
        ACT,TRIAG(2.5,3.5,3.8),,N7;
        ACT,TRIAG(2.5,3.0,3.3),,N8;
        ACT,TRIAG(0.7,1.0,1.1),,N10;
N4      COLCT,FIRST,NODE 4;
        ACT,TRIAG(1.3,1.5,1.7),,N5;
N5      COLCT,FIRST,NODE 5;
        ACT,TRIAG(2.0,2.3,3.0),,N6;
N6      COLCT,FIRST,NODE 6;
        ACT,TRIAG(0.8,1.0,1.5),,N9;
N7      COLCT,FIRST,NODE 7;
        ACT,TRIAG(2.5,3.5,3.8),,N9;
N8      COLCT,FIRST,NODE 8;
        ACT,TRIAG(2.5,2.8,3.3),,N9;
N9      ACCUM,3,3;
        COLCT,FIRST,NODE 9;
        ACT,TRIAG(0.8,1.0,1.3),,N20;
N10     COLCT,FIRST,NODE 10;
        ACT,TRIAG(1.3,1.4,1.6),,N11;
N11     COLCT,FIRST,NODE 11;
        ACT,TRIAG(2.2,2.5,3.0),,N15;
N12     COLCT,FIRST,NODE 12;
        ACT,TRIAG(0.8,0.9,1.2),,N13;
N13     COLCT,FIRST,NODE 13;
        ACT,TRIAG(1.3,1.5,1.7),,N14;
N14     COLCT,FIRST,NODE 14;
        ACT,TRIAG(1.8,2.0,2.4),,N15;
N15     ACCUM,2,2;
        COLCT,FIRST,NODE 15;
        ACT,TRIAG(1.8,2.1,2.5),,N20;
N16     COLCT,FIRST,NODE 16;
        ACT,TRIAG(1.3,1.4,1.6),,N17;
N17     COLCT,FIRST,NODE 17;
        ACT,TRIAG(1.3,1.5,1.6),,N18;
N18     COLCT,FIRST,NODE 18;
        ACT,TRIAG(2.2,2.4,3.0),,N19;
N19     COLCT,FIRST,NODE 19;
        ACT,TRIAG(1.5,1.9,2.5),,N20;
N20     ACCUM,3,3;
        COLCT,FIRST,NODE 20;
        ACT,TRIAG(1.0,1.2,1.5),,N21;
N21     COLCT,FIRST,PROJECT COMPLETION,20,
        TERM;                          10.0,0.1;
        ENDNETWORK;
INIT,,,NO;
FIN;
```

図11 シミュレーションプログラム

将来的には、このアクティビティにマネージメント活動のＷＢＳを作成する必要がある。ＷＢＳは、プロジェクト全体からそれを構成するシステムレベルへ、さらにサブシステムレベルへと順次トップダウンで展開されるものであり、容易にしかも漏れなく全作業を把握・識別することが可能である。ＷＢＳを最末端までブレイクダウンすれば、それぞれのワーク（作業）は単一のリソースに一致するところまでブレイクダウンされることにより、リソースの見積りが可能になる。また、これを上位にサマライズすれば、各コントロールに必要なリソースを配分できる。そのためには、まず①生産物を規格化、共通化することによりノウハウの蓄積を可能にする。②作業手順や作業指示方法を統一することにより、作業効率や管理レベルが向上させる。③開発対象、環境、組織やプロジェクトごとに適した開発手法を採用することにより、個人差による品質のばらつきを最小にとどめるなどの整備が必要となる。

３.３　シミュレーションの結果

シミュレーション評価では、完了時間の分布を過去のプロジェクトから採り入れた実際的な分布を費用見積りモデルから導入した（表４参照）。これをＳＬＡＭⅡでシミュレートした結果を図１２に示す。出力結果

として各ノードの到着時間経緯の平均、標準偏差、変動係数、最小値、最大値を算出した。またヒストグラムを用いてプロジェクトの完了時間の回数の累積値を出力した。これは100回のシミュレーション結果なので、このデータは中心極限定理を適用することができ、平均値の99.7%の信頼区間は、

信頼区間＝平均±３（標準偏差／$\sqrt{\text{シミュレーション回数}}$）　　(10)
＝11.0±0.1(ヶ月)

となる。ここで、入力データの三角分布の最尤値に対してクリティカルパスを結ぶと、約10ヶ月程度でプロジェクトは完了する。マネージメント活動PERTを設計する上で、ノード(WBS)のコスト（重み）を決めるためには、他のパスとの日程調整が必要不可欠であることはいうまでもない。なぜなら、平行作業の最も遅れた作業が進捗度を決定することが大きな理由の一つである。表４の三角分布に従えば平均的に約11ヶ月で完了することになる。この結果から完成期間の確率は、10.5ヶ月では5%、11.0ヶ月では58%、11.5ヶ月では95%、11.9ヶ月で100%と推定できる。より信頼（納得できるかどうかは管理者の判断）できる値を得るためにはシミュレーション回数を増やすことが必要である。例えば、スケジュールの遅れに対して①工数を増やす（要員の再投入、残業／休出）、②納期を遅伸する（段階的リリース）、③線表上でやりくりする（並列）、④工数を削減（機能を削減、やり方の改善）などシミュレーションの段階での評価が、構築の際考慮すべき重要な点である。

```
                    S L A M   I I   S U M M A R Y   R E P O R T

 SIMULATION PROJECT SOFTWEAR PERT NETWOR        BY NEC

 DATE  2/11/1990                               RUN NUMBER  100 OF  100

 CURRENT TIME     .1039E+02
 STATISTICAL ARRAYS CLEARED AT TIME    .0000E+00


              **STATISTICS FOR VARIABLES BASED ON OBSERVATION**

                    MEAN      STANDARD  COEFF. OF  MINIMUM    MAXIMUM   NO.OF
                    VALUE     DEVIATION VARIATION   VALUE      VALUE     OBS

 NODE 1           .996E+00  .824E-01  .827E-01  .825E+00  .115E+01   100
 NODE 2           .256E+01  .903E-01  .579E-01  .139E+01  .178E+01   100
 NODE 3           .203E+01  .100E+00  .489E-01  .183E+01  .227E+01   100
 NODE 4           .300E+01  .185E+00  .618E-01  .258E+01  .339E+01   100
 NODE 5           .449E+01  .199E+00  .444E-01  .409E+01  .492E+01   100
 NODE 6           .691E+01  .287E+00  .415E-01  .628E+01  .784E+01   100
 NODE 7           .534E+01  .281E+00  .526E-01  .444E+01  .585E+01   100
 NODE 8           .500E+01  .188E+00  .375E-01  .447E+01  .536E+01   100
 NODE 9           .863E+01  .355E+00  .412E-01  .777E+01  .939E+01   100
 NODE 10          .298E+01  .126E+00  .423E-01  .268E+01  .324E+01   100
 NODE 11          .441E+01  .148E+00  .332E-01  .412E+01  .472E+01   100
 NODE 12          .255E+01  .147E+00  .576E-01  .227E+01  .286E+01   100
 NODE 13          .351E+01  .163E+00  .465E-01  .317E+01  .386E+01   100
 NODE 14          .499E+01  .191E+00  .383E-01  .459E+01  .544E+01   100
 NODE 15          .715E+01  .235E+00  .329E-01  .658E+01  .764E+01   100
 NODE 16          .156E+01  .978E-01  .626E-01  .136E+01  .178E+01   100
 NODE 17          .299E+01  .129E+00  .431E-01  .274E+01  .333E+01   100
 NODE 18          .445E+01  .157E+00  .353E-01  .409E+01  .489E+01   100
 NODE 19          .697E+01  .247E+00  .354E-01  .648E+01  .759E+01   100
 NODE 20          .974E+01  .311E+00  .320E-01  .914E+01  .106E+02   100
 PROJECT COMPLETI .110E+02  .333E+00  .304E-01  .103E+02  .118E+02   100
1                        **HISTOGRAM NUMBER21**
                           PROJECT COMPLETI

OBS  RELA   UPPER
FREQ FREQ   CELL LIM 0        20        40        60        80        100
                     +    +    +    +    +    +    +    +    +    +    +
  0  .000  .100E+02 +                                                   +
  0  .000  .101E+02 +                                                   +
  0  .000  .102E+02 +                                                   +
  0  .000  .103E+02 +                                                   +
  2  .020  .104E+02 +*                                                  +
  3  .030  .105E+02 +*C                                                 +
  8  .080  .106E+02 +**** C                                            +
  7  .070  .107E+02 +****     C                                        +
 13  .130  .108E+02 +*******        C                                  +
 12  .120  .109E+02 +*******              C                            +
 13  .130  .110E+02 +*******                    C                      +
 10  .100  .111E+02 +******                           C               +
  8  .080  .112E+02 +*****                                C           +
  6  .060  .113E+02 +****                                    C        +
  7  .070  .114E+02 +*****                                      C      +
  2  .020  .115E+02 +*                                           C     +
  4  .040  .116E+02 +***                                           C   +
  1  .010  .117E+02 +                                               C  +
  3  .030  .118E+02 +*                                              C  +
  1  .010  .119E+02 +                                               C  +
  0  .000  .120E+02 +                                               C  +
  0  .000    INF   +                                               C  +
---               +    +    +    +    +    +    +    +    +    +    +
100               0        20        40        60        80        100


              **STATISTICS FOR VARIABLES BASED ON OBSERVATION**

                    MEAN      STANDARD  COEFF. OF  MINIMUM    MAXIMUM   NO.OF
                    VALUE     DEVIATION VARIATION   VALUE      VALUE     OBS

 PROJECT COMPLETI .110E+02  .333E+00  .304E-01  .103E+02  .118E+02   100
```

図１２ シミュレーション実行結果

４　むすび

ソフト生産管理で特に計画の重要性を示した。またシミュレーションモデルでは、費用見積りモデルで得られた期間から、マネージメント活動ネットワークチャートを構築して、実際の業務アプリケーションの最適化を試みた。しかし実際のプロジェクト管理の場面では、納期遅れ、予算の大幅な超過、品質不良・性能不足など重要性を認識しながらも抜本的対策がないというのが実状である。したがって、良い計画とそれに基づく実行の統制やプロジェクト失敗の原因と要因、または成功例を積み上げて分析・評価を地道に重ねる基本姿勢が結局は大切なのである。このような背景から、シミュレーションプロジェクトを遂行するためには、今後かなり多くのプロセスを踏む必要があり、かつ多大な時間を費やさなければならない。シミュレーションプロセスとしては、マネージメント活動データの計測、モデリング、シミュレーション実行代替案の作成、出力結果の分析、妥当性評価および経済性、さらには結果のプレゼンテーションなど、結果をいかにうまく表現し説得力をもたせ、評価・決断に結びつけていくかがプロジェクトへの適用の鍵である。

最後に、日頃ご指導を頂いている日本電気C＆Cソフト開発グループの藤野支配人に謝意を表します。また本論をまとめるにあたり、データ解析に協力いただいたNMITの阿部君に感謝致します。

参考文献

[1] 藤野喜一,花田収悦共編：”ソフトウェア生産技術”,(社)電子情報通信学会,東京(1985).
[2] 寺本雅則,上村松男,松尾谷徹：”ソフトウェア・メトリクスにより生産性と品質を向上”,日経エレクトロニクス No.344(1984).
[3] 田口良一,横山巽子：”ビジネスデータの分析”,丸善,東京(1975).
[4] 酒巻恒一：”ソフトウェアの信頼性－品質管理のためのソフトウェア信頼性予測”,電子情報通信学会技術研究報告 R81-8(1981).
[5] 山田茂：”ソフトウェア信頼性評価技術”,HBJ出版局,東京(1989).
[6] 山田茂,尾崎俊治：”ソフトウェアの信頼度成長モデルとその比較”,電子通信学会論文誌,Vol J65-D No.7(1982).
[7] 上村松男,寺本雅則共訳：”ソフトウェアの信頼性(M.L.Shooman著)”,マグロウヒル出版社,東京(1989).
[8] 上村松男,樋口潔,阿部勝徳,藤野喜一：”ソフトウェア信頼度成長モデルの検証(遅延Ｓ字形ＮＨＰＰモデルの基本ソフトウェアへの適用)”,電子情報通信学会技術研究報告 R88-1 6(1988).
[9] 森戸晋：”ＳＬＡＭⅡによるシステムシミュレーション”,構造計画研究所,東京(1982).