ソフトウェア工学 107－1
（1996．1．18）

役割別工数投入計画のための見積りモデルの提案

内藤 裕幹[†‡]　飯田 元[†]　松本 健一[†]　鳥居 宏次[†]

[†] 奈良先端科学技術大学院大学 情報科学研究科
〒630-01 奈良県生駒市高山町 8916-5

[‡] シャープ株式会社 情報商品開発研究所
〒639-11 奈良県大和郡山市美濃庄町 492

E-mail:{hiroyo-n, iida, matumoto, torii}@is.aist-nara.ac.jp

本稿では，ソフトウェア開発における開発要員の役割（設計者，プログラマ，テスタ）毎に工数投入の見積りが可能な「役割別工数投入計画のための見積りモデル」を提案する．本モデルは，大規模ソフトウェア開発の経験がない管理者が，ソフトウェアの開発における工数見積りを開発要員の役割別にできるようになることを狙いとしている．本研究では，実際のソフトウェア開発過程で収集したデータを使って，提案モデルと収集したデータの適合性の評価，提案モデルを使った見積りと従来モデルを使った見積りの比較による有効性の評価を行ない結果を報告する．

A Model for Estimating the Necessary Human Resources in Parallel Software Development

Hiroyoshi Naitoh[†‡]　Hajimu Iida[†]　Ken-ichi Matsumoto[†]　Koji Torii[†]

[†] Graduate School of Information Science, Nara Institute of Science and Technology
[‡] Information Systems Product Development Laboratories, SHARP Corporation

[†] 8916-5 Takayama, Ikoma, Nara 630-01, Japan

[‡] 492 Minosho Yamatokoriyama, Nara 639-11, Japan

A model is proposed for estimating the necessary human resources in software development, where process steps may overlap, i.e. the next process step may begin before the current step is finished. The aim of this model is to enable managers, who have little or no experience with large-scale software development, to estimate the amount of human resources (designer, implementer, and tester) needed for each process step at various points in time. This model is evaluated using data collected in an industrial environment, and a comparison is made with a conventional model, i.e. the Putnam software cost estimation model.

1 はじめに

機器組み込み型ソフトウェア開発の現場では，ソフトウェアの開発規模が急激に増大してきている．かつて機器組み込み型ソフトウェアは，一人で開発されることが珍しくなかったが，今日では，数十人，数百人で開発する規模にまでなってきている [3][4].

また，機器組み込み型ソフトウェアの分野には，ハードウェアの機能をソフトウェアで代用したり，ソフトウェアの機能をハードウェア化し性能を確保するといったように，ハードウェアとの依存関係が大きいために，ドキュメント化し難いノウハウが数多くある．このため他の部門の経験豊富なソフトウェア技術者や管理者が容易に開発に加われないという状況になっている．

この結果，開発の現場では，大規模ソフトウェア開発の経験がない管理者が，大規模ソフトウェア開発プロジェクトのマネージャーになることが多くなり，Brooks[2] も指摘しているように，開発遅延による問題が発生し易い状況にある．

Brooks は，大規模ソフトウェアが如何にして問題を引き起こすかということを以下の５点のように分析している．

(1) 見積り技術の不備
(2) 人，と，人月が交換できるという仮定
(3) 見積りに対する頑固さの欠如
(4) スケジュール監視の不備
(5) 遅れに対する不用意な人員増強

一方，近年のハードウェアコストの低下により，ソフトウェア組み込み機器が広く一般個人に使用されるようになってきた．このため，プロダクトの納期が市場動向に左右され易く，多くの場合，開発コストよりも納期が優先される．このことから，前工程の作業終了を待たずに次の工程の作業を始めるような開発形態が多く見られる．つまり，図 1に示すように，設計，プログラミング，テストの各開発工程（以下「3 つの開発工程」と呼ぶ）がそれぞれ連携を取りながら並行に開発を進める形態 (以下,「並行開発」と呼ぶ) である．図 1では，3 つの開発工程とそれぞれに投入される工数の状況を示している．

この開発形態では，設計，プログラミング，テストそれぞれの役割を担った開発担当者が必要である．それぞれの役割に求められるスキルは違うため，見積り時の工数投入計画を作る際は，それぞれの役割

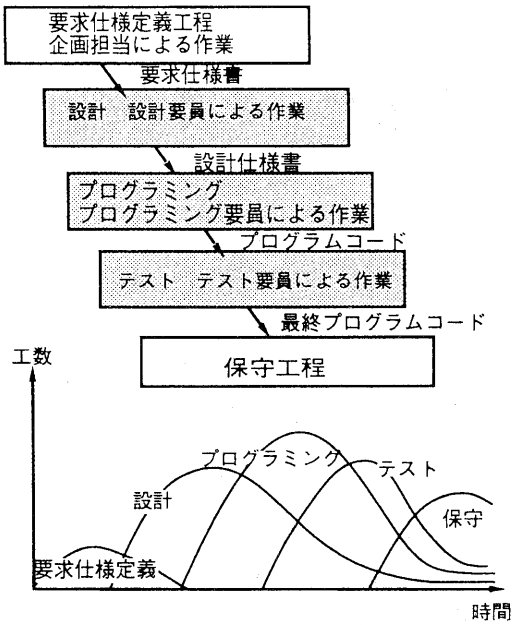

図 1: ソフトウェア開発プロセスと工数投入の現状

別に人数を割り付ける必要がある．勿論，要員が持っているスキルによりコストが変わるため，最適な人数の割り付けが必要である．しかし，大規模ソフトウェア開発の経験がない管理者には，この作業は殆んど不可能である．

開発現場で大規模ソフトウェア開発の経験がない管理者が，プロジェクトのマネージメントをすること，そして，3 つの開発工程が並行に開発を進める形態であることと，役割別に適切な人数を割り付ける必要があるという状況から，並行開発に合った見積りモデル，即ち，開発要員の役割別に工数投入ができる見積りモデルが求められている．このようなモデルがあれば，典型的な開発トラブル，即ち，開発の遅延の発生を防止できると考えられる．

しかし，既存の見積りモデル，例えば，COCOMO モデル [1]，Putnam モデル [5] を用いれば，開発期間の工数投入計画を立てることはできるが，開発要員の役割別に工数投入の見積りをすることはできない．従って，管理者の勘と経験により工数投入を行なうこととなり開発トラブルの発生に継る．

本研究では，並行開発に対応した役割別工数投入計画のための見積りモデルを提案し，その妥当性と有効性について評価を行った結果を示す。

以下，2章で，既存モデルとして，COCOMO モデル，Putnam モデルについて簡単に説明し，3章で，並行開発に対応した見積りモデルを提案し，4章で，その妥当性と有効性について評価を行った結果を報告し，5章でまとめる．

2 既存モデル

2.1 COCOMO モデル

Boehm[1]が提案したCOCOMO(COnstructive COst MOdel) モデルでは，ソフトウェアのソースコードの行数を基に，開発工数 Eと開発期間 Dを以下の式で推定することができる．

$$E = a(L_k)^b \prod_{i=1}^{15} \alpha_i \quad (1)$$

$$D = c(E)^d \quad (2)$$

ここで，各項は，

E	=	開発工数
L_k	=	ソースコードの行数 ($KLOC$)
D	=	開発期間
a, b, c, d	=	モデルの階層と開発モードに対応して決められる係数
α_i	=	ソフトウェア，ハードウェア，開発要員，プロジェクトの各コスト誘因に対する修正係数 ($1 \leq i \leq 15$)

を意味する [1][6].

更に COCOMO モデルでは，推定した開発工数と開発期間を 3 つの開発工程 (設計，プログラミング，テスト) に配分する比率を示している．各工程の工数の比率は，$32KLOC$規模のソフトウェアの場合で，それぞれ 18%, 54%, 28%である．同じく期間の比率は，34%, 40%, 28%である．工数と期間の配分比率をソフトウェア開発に適用した場合の概念図を図 2に示す．

このモデルでは，図 1に示したように各開発工程が重なっている場合に，期間の配分に基づいて見積りを考えるべきか，各工程への工数の配分に基づいて見積りを考えるべきかがわからない．また，仮に各工程に工数が配分されたとしても，配分された工数を開発の実体に合わせて月々に再配分する手段が提供されていない．

2.2 Putnam モデル

Putnam[5] が提案したモデルでは，ソフトウェア開発における工数と期間の見積り，及び，開発工数の投入計画を作成することができる．

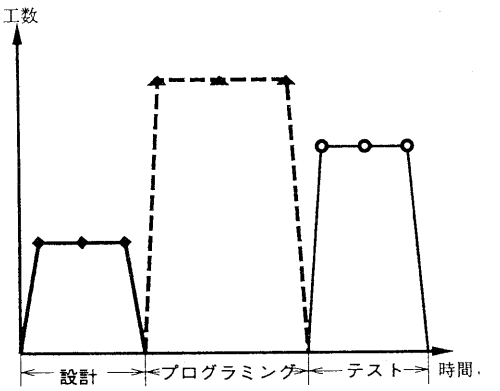

図 2: COCOMO モデル 工数/期間 配分概念図

基本式は，

$$L = C_k K^{\frac{1}{3}} t_d^{\frac{4}{3}} \quad (3)$$

$$Y = K(1 - e^{-\frac{t^2}{2t_d^2}}) \quad (4)$$

$$\frac{dY}{dt} = \frac{K}{t_d^2} t e^{-\frac{t^2}{2t_d^2}} \quad (5)$$

で表される．

ここで，各項は，

L	=	ソースコードの行数 ($SLOC$)
C_k	=	開発環境に依存した補正係数
K	=	開発の工程全体で要求される総開発工数
t_d	=	$\frac{dY}{dt}$が最大となる開発時刻 (開発期間)
Y	=	開発時刻 t までに必要な総開発工数
$\frac{dY}{dt}$	=	開発時刻 t に必要な開発工数

を意味する [5][6].

Putnam モデルが対象とする期間は，要求仕様定義の工程を除いた以降のソフトウェア開発工程全体 (保守工程も含む) を含んでいる．Putnam モデルでは，ソースコードの行数と過去のデータにより推定する補正係数 C_kから，式 (3) を使って総開発工数 Kと開発期間 t_d を決める．総開発工数 Kと開発期間 t_dから，式 (5) により，開発工数の投入見積りができる．ここで，t_dは，ソフトウェア開発の終了時期にほぼ一致すると言われている．図 3に示すように t_dまでが開発期間で以降が保守期間となる．

しかし，Putnam モデルでは，並行開発において開発要員を役割別に投入することが考慮されていないため，役割別の開発要員の工数投入見積りができない．

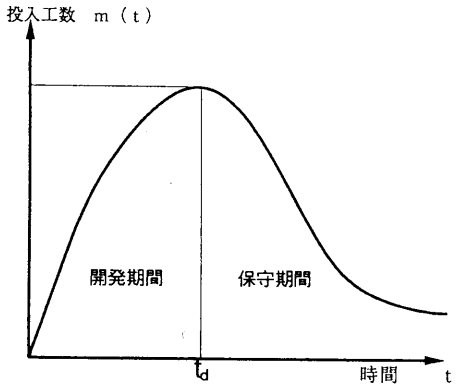

図 3: Putnam モデル 工数投入図

3 提案モデル

ソフトウェア開発は，要求仕様を製品プログラムに変換するプロセスと考えられる．同様に考えると，図1に示すようにソフトウェア開発の3つの工程は，それぞれ要求仕様，設計仕様書，プログラムコードを設計仕様書，プログラムコード，最終プログラムコード（製品プログラム）に変換するプロセスと捉えられる．

これら3つの開発工程を互いに独立した開発者チームで担当することを考えた場合，各工程における投入工数は，Putnam モデルに従うと仮定すると，全体として次のようなモデル化が可能である．

3.1 モデル式

開発時刻 t で必要とされる役割（設計，プログラミング，テスト）別の投入工数 m_{xx} は，

$$m_{de}(t) = \frac{K_{de}}{T_{de}^2} t\, e^{-\frac{t^2}{2T_{de}^2}} \tag{6}$$

$$m_{pg}(t) = \begin{cases} 0 & 0 < t < t_{pg} \\ \frac{K_{pg}}{T_{pg}^2}(t - t_{pg}) e^{-\frac{(t-t_{pg})^2}{2T_{pg}^2}} & t_{pg} \leq t \end{cases} \tag{7}$$

$$m_{ts}(t) = \begin{cases} 0 & 0 < t < t_{ts} \\ \frac{K_{ts}}{T_{ts}^2}(t - t_{ts}) e^{-\frac{(t-t_{ts})^2}{2T_{ts}^2}} & t_{ts} \leq t \end{cases} \tag{8}$$

により与えられる．これらの式をグラフで表すと図4のようになる．

ここで，各項は，

m_{xx}	=	開発時刻 t における投入工数
K_{xx}	=	各開発要員別の総開発工数
T_{xx}	=	m_{xx} が最大となる開発時刻
t_{pg}, t_{ts}	=	工数投入の開始時刻

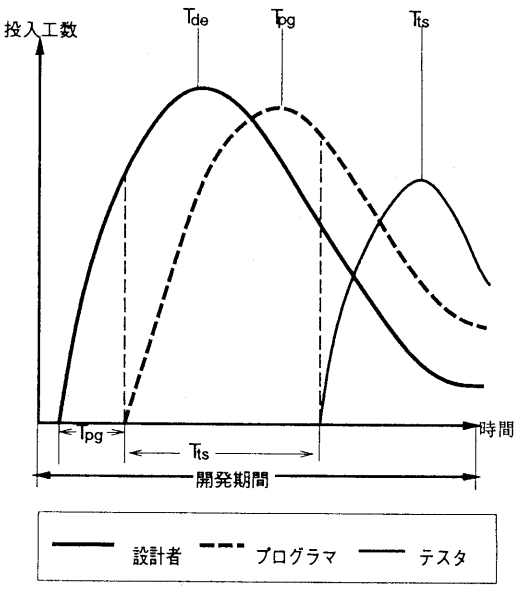

図 4: 提案モデル図

を意味し，'$_{xx}$' は，'$_{de}$'（設計），'$_{pg}$'（プログラミング），'$_{ts}$'（テスト）を意味する．

最大工数投入時刻 T_{xx} は，Putnam モデルでは，開発終了時期を表しているが，このモデルでは，

T_{de}（設計）	：	設計書の第1版完成時点
T_{pg}（プログラミング）	：	結合テスト開始時点（プログラムの第1版完成時点）
T_{ts}（テスト）	：	予定テストケースの第一回終了時点

の時刻を表すものとする．即ち，T_{de}，T_{pg}，T_{ts} までにそれぞれの工程の入力に対する開発を終了し，以降は，各工程の出力の保守が行なわれることを示す．

このモデルでは，各開発要員別の総開発工数 (K_{xx})，最大工数投入時期 (T_{xx})，及び，工数投入の開始時刻 (t_{pg}, t_{ts}) を決めることにより，役割別開発要員の投入工数を求めることができる．

3.2 各パラメータの算出方法

本モデルのパラメータを算出するためには，まず，既存のモデルを用いて総開発工数，及び，開発期間を求め，それを基にモデル式 (6)，(7)，(8) のパラメータを順次算出する．手順を以下に示す．

(1) 事前準備1（総開発工数，開発期間を求める．）
・開発予定のソフトウェアの規模（ソースコードの行数 L_k）を見積もる．L_k を基に総開発工

数E，及び，開発期間Dを求める．ここでは，COCOMOモデルや，Putnamモデルのソフトウェア方程式のような既存のモデルを使う．

・総開発工数Eを役割別の工数に配分し，E_{xx}とする．配分比率は，COCOMOモデルに従う．

・開発工数の補正(K_{de}, K_{pg}, K_{ts}の算出)

提案したモデルの各開発要員別の総開発工数(K_{xx})は，時刻tが0から∞までの累積工数を表すが，総開発工数Eを配分して求めた工数E_{xx}は，時刻tが0から開発期間Dまでの累積工数を表している．即ち，

$$E_{xx} = \int_0^D m_{xx}(t)\,dt \tag{9}$$

となる．式(9)より，K_{de}, K_{pg}, K_{ts}を算出する．ここで，$\frac{E_{xx}}{K_{xx}} = \alpha_{xx}(0 \leq \alpha_{xx} \leq 1)$を補正係数として導入する．各工程の補正係数$\alpha_{xx}$は，適用する組織の過去のソフトウェア開発データの解析により定める．

(2) モデル式(6)のパラメータ算出．

・T_{de}算出．

時刻tが0からDまでの累積工数を$\alpha_{de}K_{de}$と置くことによって，次式を得る．

$$\int_0^D m_{de}(t)\,dt = \alpha_{de}K_{de} \tag{10}$$

上式より，式(6)は，

$$T_{de} = \frac{D}{\sqrt{-2\ln(1-\alpha_{de})}} \tag{11}$$

とすることができる．

(3) モデル式(7)のパラメータ算出

・t_{pg}算出．

t_{pg}は，プログラマの工数投入の開始時刻を示す定数である．ここで，設計書の第1版の進捗率をβ_{de}%とすれば，

$$\int_0^{t_{pg}} m_{de}(t)\,dt = (1-e^{-\frac{1}{2}})\beta_{de}K_{de}$$

かつ，

$$\gamma_{de} = (1-e^{-\frac{1}{2}})\beta_{de}$$

と置くことによって，式(6)は，

$$t_{pg} = T_{de}\sqrt{-2\ln(1-\gamma_{de})}$$

とすることができる．即ち，開始時刻は，前工程の開発の進捗を表す進捗率β_{de}を定めることにより求めることが可能である．従って，適用する組織の過去のソフトウェア開発データの解析により定めることが可能であると考えられる．

・T_{pg}算出．

プログラミングの開発期間D'は，t_{pg}だけ短くなっているので，$D' = D - t_{pg}$である．総プログラミング工数(K_{pg})と開発期間D'から，モデル式(6)のパラメータT_{de}の求め方と同じようにT_{pg}を算出する．式(11)と同様に，

$$T_{pg} = \frac{D'}{\sqrt{-2\ln(1-\alpha_{pg})}} \tag{12}$$

を導く．

(4) モデル式(8)のパラメータ算出

・t_{ts}算出．

t_{ts}は，テスタの工数投入の開始時刻を示す定数である．ここで，プログラムの第1版の進捗率をβ_{pg}とすれば，

$$\int_0^{t_{ts}} m_{pg}(t)\,dt = (1-e^{-\frac{1}{2}})\beta_{pg}K_{de}$$

かつ，

$$\gamma_{pg} = (1-e^{-\frac{1}{2}})\beta_{pg}$$

と置くことによって，t_{ts}は，t_{pg}と同様に算出することができる．即ち，

$$t_{ts} = T_{pg}\sqrt{-2\ln(1-\gamma_{pg})}$$

となる．

・T_{ts}算出．

テスト期間D''は，更にt_{ts}だけ短くなっているので，$D'' = D' - t_{ts}$である．以下T_{pg}同様にT_{ts}を算出する．即ち，

$$T_{ts} = \frac{D''}{\sqrt{-2\ln(1-\alpha_{ts})}}$$

となる．

4 モデルの評価

4.1 概要

本章では，提案したモデルについて，ソフトウェア開発現場から収集したデータに基づいて適合性を検証する．続いて，同じデータを使って，提案モデルを用いた見積りと従来法による見積りを比較し，有効性を確認する．

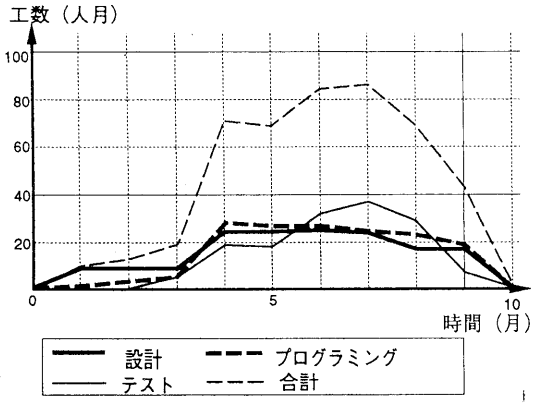

図 5: ソフトウェア開発プロジェクトの実測データ

4.2 収集データ

本評価で用いたデータ全7件の内の1件を図5に示す．図中の月数は，開発開始月を1とした相対月を表す．このデータは，機器組み込み型ソフトウェア開発に要した工数を役割別の開発要員毎に月別に集計したデータである．開発期間は，一部要求仕様設計も含んだ設計工程から出荷前のテスト工程終了までである．

4.3 分析

4.3.1 提案モデルと実データの適合性

提案モデルのパラメータ（K_{xx},T_{xx}）は，最小2乗法により推定した．Putnamモデルにおいては要求仕様の工程は，モデルの範囲外であるのと同様，本モデルでも各工程において要求仕様工程（開発内容を把握するための工程）と考えられる工数は，推定の対象外とする．パラメータ推定の結果を表1に示す．表中の de，pg，ts は，それぞれ設計者，プログラマ，テスタを表す．

最小2乗法によるパラメータ推定の効果を判定するために分散分析を行なう．

モデル式(6)は，

$$\ln\{\frac{m_{de}(t)}{t}\} = \ln(K_{de}) - 2\ln(T_{de}) - \frac{t^2}{2T_{de}^2}$$

のように書き換えることができ，更に，

$$X = t^2, Y = \ln\{\frac{m_{de}(t)}{t}\}$$

と変数変換することによって，

$$Y = \overline{Y} - \frac{1}{2T_{de}^2}(X - \overline{X})$$

表 1: 推定パラメータ／検定結果

事例	役割 (xx)	推定パラメータ t_{xx}	T_{xx}	K_{xx}	ϕ	F値	α
1	de	1	4.9	188.4	6	27.81	.5
	pg	2	4.5	193.0	5	8.70	5
	ts	4	2.2	123.0	3	33.18	5
2	de	0	5.0	79.4	6	58.93	.5
	pg	2	2.3	65.5	3	22.99	5
	ts	5	1.6	25.3	1	27.14	20
3	de	0	3.3	25.9	6	78.69	.5
	pg	1	2.9	18.4	5	30.40	.5
	ts	4	1.5	22.6	2	34.06	5
4	de	0	7.1	73.6	13	46.07	.5
	pg	3	4.4	181.6	10	63.21	.5
	ts	11	1.4	37.9	2	24.43	5
5	de	0	5.9	19.5	12	43.61	.5
	pg	3	4.6	14.3	9	61.33	.5
	ts	9	2.6	7.3	3	40.57	1
6	de	1	3.7	57.4	9	149.7	.5
	pg	3	3.1	71.6	7	72.82	.5
	ts	8	1.5	30.0	2	6.03	20
7	de	0	3.4	28.9	6	47.36	.5
	pg	1	3.4	18.6	5	22.34	1
	ts	4	1.7	13.7	2	33.22	5

となり，XとYは直線関係にあることがわかる．$\overline{X}$ および$\overline{Y}$は，収集したデータの平均値である．

ここで，事例1について，回帰直線によって説明されるYの変動部分を表す回帰による平方和（要因R）と回帰直線によって説明できないYの変動部分を表す残差平方和（要因e），及び，分散比Fを求め，表2に示す．

結果は，有意水準5%で有意と判定でき，回帰に意味が有ったと考えられる．他の事例については，F値のみを表1に示す．

設計工程については，全事例有意水準0.5%で有意と判定でき，プログラミング工程については，4件が有意水準0.5%，1件が有意水準1%，2件が有意水準5%で有意と判定でき，テスト工程については，1件が有意水準1%，4件が有意水準5%で有意と判定できる．

全体では，有意水準5%で有意と判定でき，回帰に意味が有ったと考えられる．

尚，事例2, 事例6のテスト工程の検定結果が他

表 2: 事例 1 の分散分析表

役割	要因	平方和	自由度ϕ	平均平方	F値
de	R	1.5289	1	1.5289	27.81
	e	0.3298	6	0.05497	
	計	1.8587	7		
pg	R	1.1332	1	1.1332	8.70
	e	0.6515	5	0.1303	
	計	1.7847	6		
ts	R	4.0088	1	4.0088	33.18
	e	0.3624	3	0.04345	
	計	4.3712	4		

と大きく異なっているが，調査の結果，収集されたデータ数が少ないための誤差が原因と考えられる．

4.3.2 提案モデルによる見積りの有効性

提案モデルと従来モデルを用いて工数投入計画を立てその結果から有効性を評価する．ただし，提案モデルは，役割別の工数投入計画を求めるモデルであるため，比較にあたっては，各役割別の工数の合計で比較する．

評価方法 モデルにより求めた工数投入データと実際の開発時に投入された工数データとの誤差の平方和の比較により評価する．

見積り方法 提案モデルは，ソフトウェアの規模，総開発工数，開発期間が既に決まっていることを前提としているため，今回の評価でも，総開発工数と開発期間は与えられているものとする．

提案モデルでの見積り手順を以下に示す．

(1) 総開発工数を COCOMO モデルにより，設計，プログラミング，テストに配分する．配分率については，過去の開発事例データより得られた値 (ケース 1) と COCOMO モデルの推奨値 (ケース 2) とを使用する．配分後の開発工数を補正し，K_{de}, K_{pg}, K_{ts}を求める．ここで，補正係数α_{xx}は，過去の開発事例データにより求めた値の平均値 (0.9) を使用する．

(2) モデル式 (6) より，t_{pg}を算出．ここで，進捗率β_{de}は，過去の開発事例データにより求めた値の平均値 (0.3) を使用する．

(3) モデル式 (7) より，t_{ts}を算出．ここで，進捗率β_{pg}は，過去の開発事例データにより求めた値の平均値 (1.4) を使用する．

以上により，投入工数を見積もる．

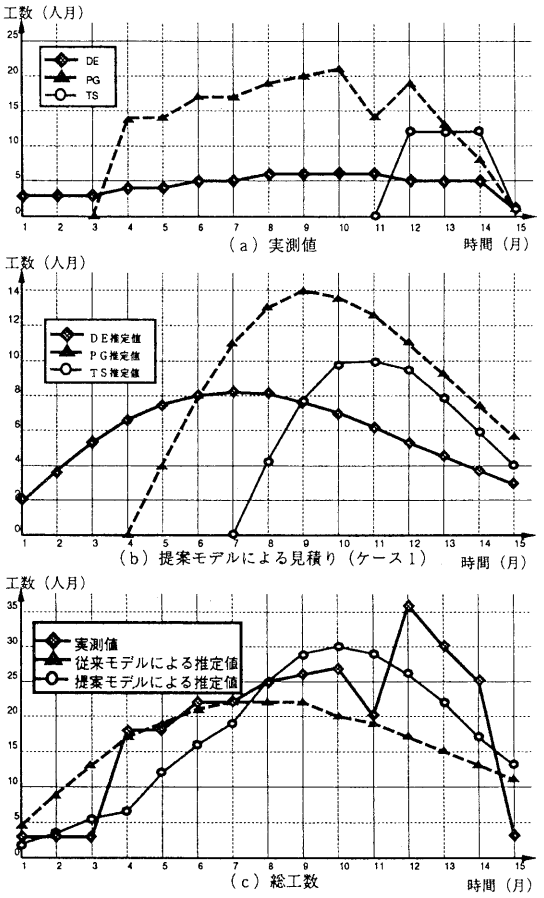

図 6: モデル適用図

従来手法での見積りとして，Putnam モデルの式 (5) により，投入工数を見積もる．ここで，総開発工数と開発期間は，既知として適用する．

収集したデータの一事例に関して，提案モデルと従来方法で行なった見積り結果を図 6に示す．図 6 の (a) は，各工程別観測値，(b) は，提案モデルによる工程別の見積り，(c) は，(a)，(b) それぞれの各工数の合計値と Putnam モデルによる推定値を表す．

両者の見積り結果と収集したデータの誤差の平方和を表3 に示す．ケース 1 は，過去の開発事例データによる配分率を使った場合，ケース 2 は，COCOMO モデルの推奨配分率を使った場合の誤差である．

比較の結果は，従来モデルに比べて提案モデルの方が誤差が少ない．この結果から 3 つの開発工程に

表 3: モデルによる誤差の平方和

事例	提案モデル		従来モデル
	ケース 1	ケース 2	
1	1547	1291	5470
2	198	269	360
3	43	60	69
4	657	813	1028
5	22	27	11
6	191	306	273
7	34	40	90

分割して見積もることによって，見積り精度が上がったと言える．

事例 5，6 では，提案モデルの見積りの方が誤差が大きい．事例 5 は，たまたま，従来モデルが実際の開発事例に適合したのではないかと考えられる．今後，事例を増やして更に検討が必要である．事例 6 は，ケース 1 での誤差が少ないことから，総工数の配分に問題があると考えられる．

5 むすび

本研究では，設計者，プログラマ，テスタという役割別に開発要員を投入するための見積りモデルを提案し，提案モデル式の各パラメータの求め方を示し，その妥当性と有効性について評価を行なった．妥当性については，最小 2 乗法によるパラメータ推定，及び，その有効性を判定するために分散分析を行ない結果が有意であることを確認した．有効性については，提案モデルと従来モデルを用いて工数投入計画を立て，その結果と別途ソフトウェア開発過程から収集したデータとの誤差の平方和を比較して，提案モデルの方が誤差が少ないこと，即ち，見積り精度が上がったことを示した．

提案モデルを利用することにより，役割別の工数投入見積りが可能となり，開発初期の工数投入見積り誤りによる開発遅延が少なくなることが期待できる．

提案モデルでは，工数投入の開始時刻（t_{pg}, t_{ts}）を，適用する組織の過去のデータの解析により定めることとしているが，工数投入の開始時刻を変えることは，その工程の開発期間を変えることであり，式 (3) からも解るように開発時期と総工数に影響するため，実際の見積りの局面では，総工数と開発時期を決める段階で，開発の条件（例えば，納期を優先するとか，コストを優先するとか）を考慮し，試行錯誤することが予想される．従って，工数投入の開始時刻に関して開発の条件を考慮した指標を考案する必要がある．

謝辞 第一著者に対して本研究の機会を与えて頂きましたシャープ (株) 情報商品開発研究所坂田安男所長および斗谷充宏主任研究員に，ならびに，資料提供等調査に御協力頂いた事業部の方々，協力会社の方々に深く感謝します．

参考文献

[1] B.W.Boehm: "Software Engineering Economics", IEEE Trans. on Software Engineering,VOL.SE-10,NO.1, pp. 4–21, January 1984.

[2] F.P. Brooks,Jr.: "The Mythical Man-Month", Datamation, December 1974, pp. 44–52.

[3] 小泉，渡辺，林，杉村："機器組み込みソフトウェアにおけるテスト効率に関する考察", ソフトウェアシンポジウム'95,pp.181–187(1995).

[4] 仲島，板嶋，門脇，坂口，朝見："ソフトウェアプロセス成熟度の改善の取り組み事例", ソフトウェアシンポジウム'95,pp.124–130(1995).

[5] L.H.Putnam: "A General Empirical Solution to the Macro Software Sizing and Estimating Problem", IEEE Trans. on Software Engineering,VOL.SE-4,NO.4, pp. 345–361, July 1978.

[6] 山田，高橋："ソフトウェアマネジメントモデル入門", 共立出版株式会社 (1993).