

## ソフトウェア開発において経済性を考慮した チーム人数の求め方†

後 藤 兼 一†† 望 月 正 嘉††† 寺 山 隆†††

ソフトウェア開発では、チームメンバー間で多くのコミュニケーションが通常必要となる。このことはチーム人数が増えると著しい。管理者等は開発にかかる費用と急ぎの度合を勘案し、勤と経験でチーム編成を行っているのが現状である。そこで本稿では経済性を考慮したチーム人数の算出方法を検討する。まず、延作業時間を延主体作業時間と延調整作業時間の和としてとらえ、延主体作業時間と人数との関係を習熟で、延調整作業時間と人数との関係をコミュニケーション量で定式化する。なお、本稿の特色はコミュニケーションに方向性を導入したことにある。次に経済性が評価できるようにするために開発期間が遅れることによる機会損失を金額に変換し、これに開発にかかる費用を加えて、費用・損失関数を定義する。そして費用・損失を最小にするチーム人数の算出方法を示す。最後に実際のソフトウェア開発におけるデータを用い、この方法はチーム編成をする際、管理者等がチーム人数を決定するための一つの目安として有用であることを示す。

### 1. ま え が き

ソフトウェア開発における管理技術の応用研究の一環として、本稿では経済性を考慮したチーム人数の求め方に関する研究を報告する。

ソフトウェア開発においても、他の一般の業務と同様に期間の短縮とか工数の低減は効率化をはかる上で重要な課題である。ところがソフトウェア開発においては、期間の短縮をはかるために、開発の途中で人数を増しても期間を縮めることにはならず、極端な場合はかえって遅れてしまうという特性があることが一般に知られている。その理由として工学的な研究では Brooks<sup>1)</sup>は、人数が増えるとコミュニケーションの量が大幅に増えるからであると説明している。また Mantei<sup>2)</sup>はチーム構造の効果についてまとめている。さらに心理学的な研究として、Shaw<sup>3)</sup>はコミュニケーション構造とグループパフォーマンスについて論じているし、青井<sup>4)</sup>は問題解決能力は集団の大きさ、集団の結合度、リーダーシップの型、競争的雰囲気の高さおよびコミュニケーション構造の型によるとのべている。しかしこれらの研究では、何人のチームにすれば期間あるいは費用を最小にできるかなどの工学的な課題には答えられないのが実状である。実務担当者はチーム人数を客観的に決める方法を求めている。期間

と工数と人数の関係をモデル化し、実験によって検証がされ<sup>5)</sup>、期間の最小化という観点から研究がされてはいる<sup>6)</sup>ものの、費用を最小にするという経済性を考慮した解析は不十分と思われる。また上記研究の前提では自分自身に対するコミュニケーション、すなわち自問自答するような場合が含まれていない。しかしソフトウェア開発では、2.2節でくわしくのべるが、自分自身に対するコミュニケーションが多く発生することにも着目する必要がある。

なお本稿でとりあげる対象業務はソフトウェア開発であるが、ある特定の人の特別な能力に依存するような業務は含まない。また業務には習熟現象があり（なくてもかまわない）、意見調整が多く発生する場合に限定する。本稿ではまず、延作業時間を延主体作業時間と延調整作業時間の和として捉え、延主体作業時間と人数、延調整作業時間と人数との関係を論ずる。次に経済性を評価するための費用・損失関数を用い、費用損失を最小にする人数の求め方を示す。さらにこれらの理論を用い、実際のソフトウェア開発工場におけるデータを分析する。

### 2. 延主体作業時間と延調整作業時間

ソフトウェア開発を分業すると、人と人との間で多くのコミュニケーションが発生するという認識から、延作業時間 ( $H_N$ ) を延主体作業時間 ( $H'_N$ ) と延調整作業時間 ( $H''_N$ ) の和として考えることにする<sup>5), 6)</sup>。したがって

$$H_N = H'_N + H''_N \quad (1)$$

とする。なお Brooks<sup>1)</sup> も延作業時間を個々の作業時

† Method for Computing the Size of a Software Development Team under Considerations of Economy by KEN-ICHI GOTOH (JMA Consultants Inc.), TADAYOSHI MOCHIZUKI and TAKASHI TERAYAMA (Nippon Electric Co., Ltd.).

†† (株)日本能率協会  
††† 日本電気(株)

間とコミュニケーション時間の和として位置づけている。

はじめに主体作業と調整作業を次のように定義しておく。

主体作業：アウトプット製作のために直接必要な作業。すなわち、アウトプットの目的を理解し、関連して必要となる事項を吸収し、アイデアを出し、構想を練り、アウトプットにまとめあげるまでの一連の作業をいう。

調整作業：チームメンバーの意見調整のために必要な作業。すなわち、ある人が他のある人に対して意見を出し、評価を受け、自分の意見を修正するかあるいは相手の考え方を修正させて、意見をまとめあげるまでの一連の作業をいう。

### 2.1 延主体作業時間と習熟

いま主体作業を複数の同質な作業に分割したとき、1番目より2番目、2番目より3番目、…と作業時間が短くなってゆく習熟現象があらわれる。ここで同質とは同一の質という意見のほか同種の質という意味も含む。Simon<sup>7)</sup>によれば「同じ課題、あるいは同種の課題を繰り返すとき、2度目のほうが1度目よりも、よりよくそれを遂行することができるようなシステム」と定義している。また師岡<sup>8)</sup>によれば「習熟とは同一の機能を果たすための行為の繰返しによる効果があるとき習熟がある」と定義されている。主体作業はアウトプットの目的をいろいろな角度から理解したり、関連して必要となるいろいろな資料を収集・整理し吸収したり、多くのアイデアを出したり、構想を練ったり、最後にアウトプットとしてまとめあげる内容であるから、同種の課題を繰り返しているとみることもできるし、また同一の機能を果たすための行為の繰返しがあるとも考えることができる。したがって主体作業は複数の同質な作業からなりたっているとみることができる。後藤<sup>9)</sup>によれば主体作業に習熟現象があれば、ある作業を  $N$  に同質に分割したときの延主体作業時間は近似的に1人で作業をしたときの主体作業時間の  $N^\alpha$  倍であることが示されている。すなわち

$H_N$ :  $N$  人で作業をしたときの延主体作業時間

$H_1$ : 1人で作業をしたときの主体作業時間

$\alpha$ : 習熟係数 ( $0 < \alpha < 1$ )

のとき、

$$H_N = H_1 N^\alpha \quad (2)$$

となる。式(2)を3章で使う。

### 2.2 延調整作業時間とチーム形態

ソフトウェア開発では会議とか打合せといったようなコミュニケーションの場が多くもたれる。1日中会議とか打合せということもめずらしくない。このコミュニケーションの量が何に比例しているかについて、Brooks<sup>1)</sup>は  $N(N-1)/2$  (ただし  $N$  は人数) に比例するのべている。つまりコミュニケーションの量は

① 人と人との間での

② 最大コミュニケーションルート数に比例する必要がある。ここでコミュニケーションルートとは人と人との間で往来する情報の経路を意味する。さて②の仮定では、 $N$  人が分担する業務の難易度はすべて等しく、しかも  $N$  人の技術レベルもすべて等しく、かつ身分が対等であることが条件となっていると考えられる。しかしソフトウェア開発を含めて一般の業務ではこのような条件になることは少ないと思われる。業務量、業務の不確実性などによる難易度の違い、経歴、経験年数などによる技術レベルの違い、さらに職階、職能などによる身分の違いによって指導性が変わり、その結果コミュニケーションの形態および量も変わるものと思われる。また①の仮定については、前者の人と後者の人は同一人でないことが条件となっていると考えられる。しかしこの条件が成り立つのは、業務が比較的単純でかつ重要性をおびていない場合であって、ソフトウェア開発のような場合は成り立たない場合もあると思われる。なぜならば業務は複雑であり重要性があるから、自分自身に対してもたとえば、“本当にそれでよいのか?”とか“たぶん他の人はこんなことで反対するのではないか”とか“何か抜けはないか”など問いかけ答えることが多くなるからである。したがって本稿では前者の人と後者の人は同一人である場合も含めて検討することにする。なお自分自身に対するコミュニケーションという言葉を用いるのは不自然なので、今後は調整という言葉はコミュニケーションにかわって用いる。その場合、自己調整と他人調整を次のように定義しておく。

自己調整：ある人が自分自身に対して行う調整作業

他人調整：ある人が他のある人に対して行う調整作業

次に指導性の違いを考慮に入れてチーム形態を論ずることにする。自己調整が発生しない場合のチーム形態については先の研究で示されているので<sup>6),7)</sup>、本稿は自己調整が発生する場合について研究する。

まず次のような規則をもうける、すなわち

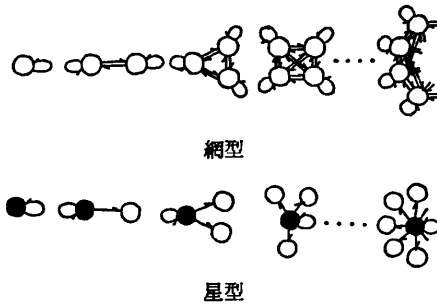


図1 網型および星型の調整

Fig. 1 Adjustment based on net or star type.

- i) チームメンバの任意の2人(a, b)について, aがbよりも指導性を発揮しているか,あるいは同等の場合は, aからaに対する自己調整およびaからbに対する他人調整がおのおの1ルートずつ発生する.
- ii) ただし, a, bの指導性が同等であっても, a, b以外の他のメンバcがa, bに対し指導性を発揮している場合は, a, b間ではi)はおこらない.
- iii) なお, 1メンバ当りの自己調整のルートは1を越えない.
- iv) また, チーム人数が1人(a)しかない場合は, aからaに対する自己調整が1ルート発生する.

ここで上記の規則を用い, チームメンバの指導性が全員同等の場合(網型とよぶことにする)とチームメンバの1人(チームリーダーとよぶことにする)だけが指導性を発揮し他のメンバは同等の場合(星型とよぶことにする)について, おのおののチーム形態を図1に示す. 図1から明らかなように, 網型の場合の調整ルート数は自己調整ルート数が $N$ で他人調整ルート数が $2 \times {}_N C_2$ であるから $N + 2 \cdot N(N-1)/2 = N^2$ となり, また星型の場合は自己調整ルート数が1で他人調整ルート数が $N-1$ であるから $1 + (N-1) = N$ となる.

なお網型の調整ルート数が $N^2$ に比例する根拠は後藤の研究からも導くことができる<sup>9)</sup>.

それによると主体作業が発生しない意思決定だけの場合, 延意思決定時間 $H_N$ は延課題理解時間と延意見調整時間の和として

$$H_N = AN + B \frac{N(N-1)}{2} \quad (3)$$

で表せるとのべている. ただし

A: 1人当り課題を理解する時間

B: 1意見調整ルート当り意見を調整する時間

とする.

なお式(3)で示されている延意見調整時間 $BN(N-1)/2$ には本論でいう自己調整による時間は含まれていない. しかし, いま $N$ に比例するとした課題理解時間に, 同じく $N$ に比例する自己調整時間 $BN/2$ が含まれていると解釈すれば, 式(3)は $AN + BN(N-1)/2 = (A-B/2)N + BN^2/2$ と変形できることから網型の調整ルート数が $BN^2/2$ となり $N^2$ に比例するとみなすことができる. また星型の調整ルート数が $N$ に比例する根拠は, 後藤の研究<sup>6)</sup>によれば, チームリーダーが点型の調整(点型の場合は指示ともいう)をチームメンバに個別にすることになるので $N-1$ に比例することになる. したがってこれに自己調整ルートを1加えれば $N$ となる.

さてここで $\lambda$ をチーム形態係数,あるいはたんに形態係数とすれば,  $N$ 人のときの調整ルート数は一般に $N^\lambda$ であらわすことができる. なぜならば,  $N^\lambda$ は $\lambda=2$ で網型の,  $\lambda=1$ で星型のチーム形態での調整ルート数となるからであり, また都合のよいことに調整ルート数が最大になるのが網型であり逆に最小となるのが星型であるから $1 < \lambda < 2$ のときは網型と星型の間の中チーム形態を示すことができるからである.

ここで延調整作業時間は調整ルート数 $N^\lambda$ に比例すると考えれば,

$H_N^*$ :  $N$ 人で作業をしたときの延調整作業時間

$t$ : 1調整ルート当りの調整作業時間

$\lambda$ : チーム形態係数 ( $1 \leq \lambda \leq 2$ )

のとき

$$H_N^* = tN^\lambda \quad (4)$$

となる. 式(4)は3章で使う.

### 3. 経済性を考慮したチーム編成

期間の最小化という観点からのチーム編成の研究<sup>5)</sup>はある. この考え方は手術のごとき人命に関するときとか, 社会的損失が絶大な交通機関の復旧作業のような場合は有効である. しかしソフトウェア開発のような場合は経済性を加味する必要がある.

#### 3.1 延作業時間と作業期間

延作業時間は式(1)に示したとおり延主体作業時間と延調整作業時間の和として考えたから, 式(2)と式(4)の和で示す. したがって延作業時間 $H_N$ は

$$H_N = H_1 N^\alpha + tN^\lambda \quad (5)$$

となる. 作業期間 $T_N$ は, 作業の初めから終わりまで

会員が参加するとすれば,  $T_N = H_N/N$  であるから,

$$T_N = H_1 N^{\alpha-1} + t N^{\lambda-1} \quad (6)$$

となる. なお  $H_1/t$  を作業度とよぶことにし  $q$  ( $0 < q < \infty$ ) で表す. つまり作業度は, 1人で作業をしたときの, 調整作業時間に対する主体作業時間の割合を示している. たとえば業務が複雑であり重要性が高いと調整作業時間が高くなるから, 作業度は低くなる.

### 3.2 費用・損失関係

経済性を加味した上で人数を決めるために, 延作業時間  $H_N$  に比例する費用と, 作業期間  $T_N$  に比例する機会損失の合計を最小にするという考え方を導入する. そこで費用・損失関数  $E_N$  を次のように定義する.

$$E_N = uH_N + vT_N \quad (7)$$

ここで

$u$ : 延作業時間  $H_N$  の単位時間 (たとえば1人・月) 当りの費用

$v$ : 作業期間  $T_N$  の単位期間 (たとえば1カ月) 当りの機会損失

とする. なお  $v/u$  を緊急度とよぶことにし  $p$  ( $0 < p < \infty$ ) で表す. つまり緊急度は, 費用に対する機会損失の割合を示している. たとえば業務に多くの期間がかかることにより社会的損失, 経済的損失が高いと, 緊急度は高くなる.

### 3.3 経済チーム人数

経済的なチーム人数を求めるということは  $E_N$  を最小とする  $N$  を求めることにほかならない. しかし (7) 式を  $dE_N/dN=0$  とおいて整理しても代数的に解は得られない. そこで  $dE_N/dN=0$  とおき整理すると

$$N = p \left( \frac{1 + \frac{1}{q} N^{\lambda-\alpha}}{\alpha + \frac{\lambda}{q} N^{\lambda-\alpha}} - 1 \right) \quad (8)$$

となる. ここで右辺を  $f(N)$  とおけば

$$f'(N) = - \frac{p(\lambda-\alpha)^2 N^{\lambda-\alpha-1}}{q \left( \alpha + \frac{\lambda}{q} N^{\lambda-\alpha} \right)^2} \quad (9)$$

と整理でき,  $\lambda \neq \alpha$  で常に  $f'(N) < 0$  となるゆえ  $f(N)$  は単調減少関数となる. また  $f(0) = p(1-\alpha)/\alpha > 0$  となるから式(8)の解はただ一つだけ存在する. しかもそのとき  $d^2E_N/dN^2 > 0$  であるから最小値となる. したがって Newton-Raphson 法を利用することができ,

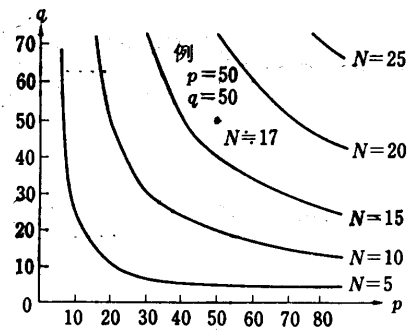


図2  $\alpha=0.5, \lambda=1.5$  の例  
Fig. 2 Example when  $\alpha=0.5, \lambda=1.5$ .

$$N_{i+1} = \frac{f(N_i) - f'(N_i) \cdot N_i}{1 - f'(N_i)}$$

(ただし  $N_0$  は初期値とする) (10)

をもって順次  $N_i$  を求めれば解を数値解析的に得ることができる. たとえば  $\alpha=0.5, \lambda=1.5, p=50, q=50$  のときは  $N_* = 16.6$  と求まる. この方法だと高い精度の解も可能である. しかし実務上は人数を小数点以下をくわしく求めてもあまり意味がない. そこで簡易的に解を求める方法を示す. 式(8)を変形すると

$$q = \frac{p(\lambda-1) + \lambda N}{p(1-\alpha) - \alpha N} N^{\lambda-\alpha} \quad (11)$$

となる. ある  $\alpha, \lambda$  に対して  $N$  をパラメータとすれば点  $(p, q)$  は曲線的な動きをする. たとえば  $\alpha=0.5, \lambda=1.5$  に対して  $N=5, 10, 15, 20, 25$  のときの点  $(p, q)$  の動きは図2となる. 図2において  $p=50, q=50$  のときは  $N_* = 17$  と求めることができる.

## 4. ソフトウェア工場における事例

本稿の理論をもとにソフトウェア開発工場における実態分析をおこない理論の有用性を示す.

### 4.1 実態調査の概要

- ①調査場所: 日本電気株式会社府中事業所
- ②調査対象: 同基本ソフトウェア開発本部
- ③調査業務: OS 言語およびユーティリティの開発プロジェクトを基本設計, 機能設計, 詳細設計, コーディング, デバッグ, インテグレーションの6工程にわけ.
- ④調査範囲: プロジェクトは基本設計からインテグレーションまで全工程にわたって調査する. なお, 終了した部分は実績を, 未完成の部分は予定とする.
- ⑤調査方法: プロジェクトリーダーに対するアンケート

表 1 プロジェクト概要  
Table 1 Outline of sample projects.

プロジェクト																			
項目		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
プログラム	OS	○			○	○	○					○	○		○	○			○
	言語								○	○				○					○
	ユーティリティ							○							○		○	○	○
	その他		○	○								○							
使用言語	HPL	○	○	○	○	○	○		○	○	○	○	○	○	○	○	○	○	○
	COBOL																	○	○
	その他							○						○				○	○
新・改訂	新規			○	○		○	○										○	○
	改訂	○	○			○			○	○	○	○	○	○	○	○			○
難易度	むずかしい				○	○	○	○				○	○					○	○
	普通	○	○	○					○	○			○	○	○	○			○
	やさしい																		
急ぎの度合	特急	○		○		○	○	○		○				○		○			○
	少し急ぐ		○								○	○	○					○	○
	普通				○				○						○				
アウト	ドキュメント枚数	250	500	150	200	700	400	500	360	300	200	1600	500	100	200	300	60	2000	100
	ステップ数 (K)	20	70	7	13	50	25	15	8	5	20	130	15	1	7	20	25	60	30

トおよび補足インタビュー

- ⑥調査月日：昭和57年7月～10月（アンケート）  
昭和58年3月（インタビュー）
- ⑦調査項目：プロジェクト概要として、プログラム種類、使用言語、新規または改訂の別、難易度、急ぎの度合およびアウトプット量(表1)。またプロジェクト内容として月別工程、月別延作業時間、工程別習熟状態および月別チーム形態

4.2 プロジェクト内容調査

- i) 月別工程：月別の工程が基本設計、機能設計、詳細設計、コーディング、デバッグ、インテグレーションのいずれであるか調査した。二つの工程にまたがっている場合は延作業時間の多いほうの工程とした。またプロジェクトによっては工程が省略されているものもある。
- ii) 月別延作業時間：延作業時間  $H_N$  を延主体作業時間  $H'_N$  と延調整作業時間  $H''_N$  に分けて調査した。単位は人・月。なお主体作業と調整作業の定義は2章に示したとおりである。
- iii) 工程別習熟状態：工程別に主体作業を五つ(今回の調査では五つとした)の同質な作業に分割すると仮定し、1番目の主体作業時間  $\tau_1$  を100としたときの2番目、3番目、4番目および5番目の主体作業時

間  $\tau_2, \tau_3, \tau_4$  および  $\tau_5$  を調査した。なおここで同質な作業とは同一または同種のアウトプットを出すための作業をいうことにする。たとえば工程が基本設計の場合、一般にアウトプットはドキュメントとよばれる仕様書である。もしこの仕様書が100ページであれば、最初の20ページを完成するまでの主体作業時間が  $\tau_1$  であり、次の20ページを完成するまでの主体作業時間が  $\tau_2$  ということになる。

iv) 月別チーム形態：自己調整ルート数  $r'$  と他人調整ルート数  $r''$  およびその和である調整ルート数  $r$  を調査した。さらにチーム人数  $N$  を調査した。

4.3 習熟係数  $\alpha$  およびチーム形態係数  $\lambda$  の算出

i) 習熟係数  $\alpha$  の算出：作業に習熟現象がある場合前出の研究<sup>5)</sup>では、作業を  $M$  個の同質な作業に分割するとき、分割された作業を1番目から順に行えば、 $m$  番目の主体作業時間  $\tau$  は

$$\tau = \beta m^{-\alpha} \quad (m=1, 2, \dots, M) \tag{12}$$

で表せるとしている。ここで同質な作業とは同一または同種のアウトプットを出すための作業をいうことにする。ただし、

- $\beta$ : 1番目の同質に分割された作業にかかる主体作業時間
- $\alpha$ : 習熟係数 ( $0 < \alpha < 1$ )

とする。いま、 $m=1, 2, \dots, M$  に対応する主体作業時間  $\tau = \tau_1, \tau_2, \dots, \tau_M$  のデータをもとに式(12)の  $\alpha$  を求めることを考える。式(12)を対数変換すると、

$$\log \tau = \log \beta - \alpha \log m \tag{13}$$

となり  $\log \tau$  は  $\log m$  の一次式となる。したがって最小二乗法を利用することができる。いま  $S = \sum (\log \beta - \alpha \log m - \log \tau_m)^2$  とおけば  $\partial S / \partial \alpha = 0$  と  $\partial S / \partial \log \beta$  を連立させることにより  $\alpha$  は

$$\alpha = \frac{\sum_{m=1}^M \log m \sum_{m=1}^M \log \tau_m - M \sum_{m=1}^M \log m \log \tau_m}{\left( \sum_{m=1}^M \log m \right)^2 - M \sum_{m=1}^M (\log \tau_m)^2} \tag{14}$$

により求めることができる。たとえば  $M=5$  で  $\tau_1=100, \tau_2=80, \tau_3=65, \tau_4=55, \tau_5=50$  のときは  $\alpha=0.439$  となる。

ii) チーム形態係数  $\lambda$  の算出: チーム人数  $N$  でチーム形態係数  $\lambda$  のとき調整ルート数  $r$  は  $r=N^\lambda$  であったから、 $\lambda$  は

$$\lambda = \frac{\log r}{\log N} \tag{15}$$

により求めることができる。たとえば図3の例のときは自己調整ルート数が  $r'=3$  他人調整ルート数が  $r''=7$  から  $r=10$ 、チーム人数  $N=5$  であるから  $\lambda=1.430$  となる。

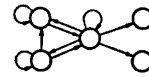


図3 調整ルートの例  
Fig. 3 Example of adjustment route.

さて、図4はプロジェクトを工程別に習熟係数  $\alpha$  およびチーム形態係数  $\lambda$  を求め、プロットしたものである。なおこの図を  $\lambda-\alpha$  図とよぶことにする。小文字のアルファベットはプロジェクトを示す。なお工程によってはプロジェクトが欠落しているが、これはプロジェクトの都合で省略されていたことによるものである。 $\lambda-\alpha$  図は  $\lambda$  の値が高くなるにつれて調整ルート数が増加し、また  $\alpha$  の値が高くなるにつれて主体作業の習熟効果が大きくなることを示している。根拠ははっきりしないが基本設計、機能設計および詳細設計は  $\Delta$  型に、コーディング、デバッグおよびインテグレーションは  $\square$  型に分布していることがうかがえる。

4.4 緊急度  $p$  の設定と作業度  $q$  の算出

i) 緊急度  $p$  の設定:  $p=v/u$  であったから、まず  $u$  すなわち延作業時間  $H_N$  の単位時間当りの費用と、 $v$  すなわち作業期間  $T_N$  の単位期間当りの機会損失を調査すればよい。一般に費用の調査は容易であるが、機会損失の調査は容易ではない。しかし現実的には意思決定者は  $p$  に関して何らかの尺度をもってはいるはずである。

今回の実態調査では急ぎの度合を調査してあるので

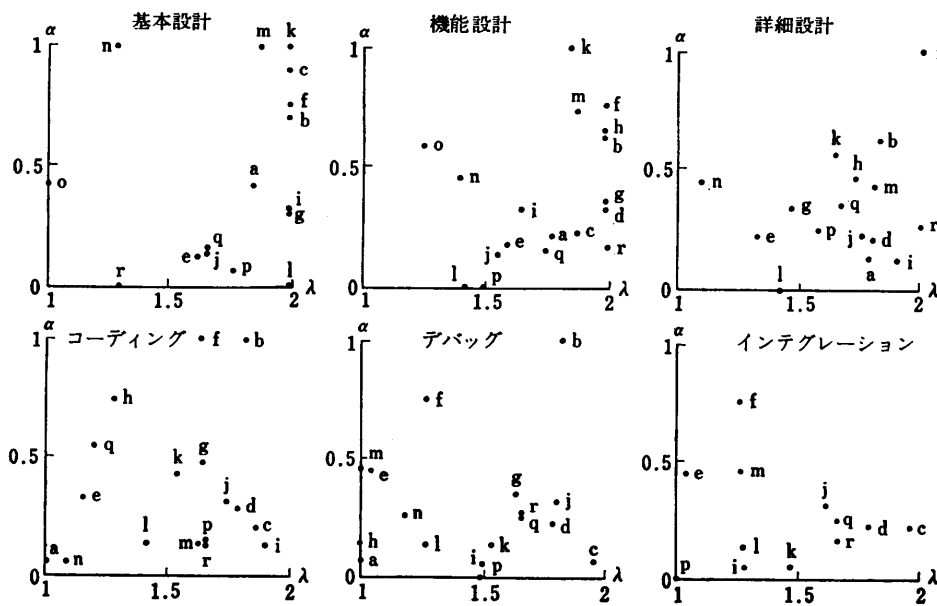


図4 プロジェクトの  $\lambda$  値および  $\alpha$  値 ( $\lambda-\alpha$  図)  
Fig. 4  $\lambda$  and  $\alpha$  of sample projects.

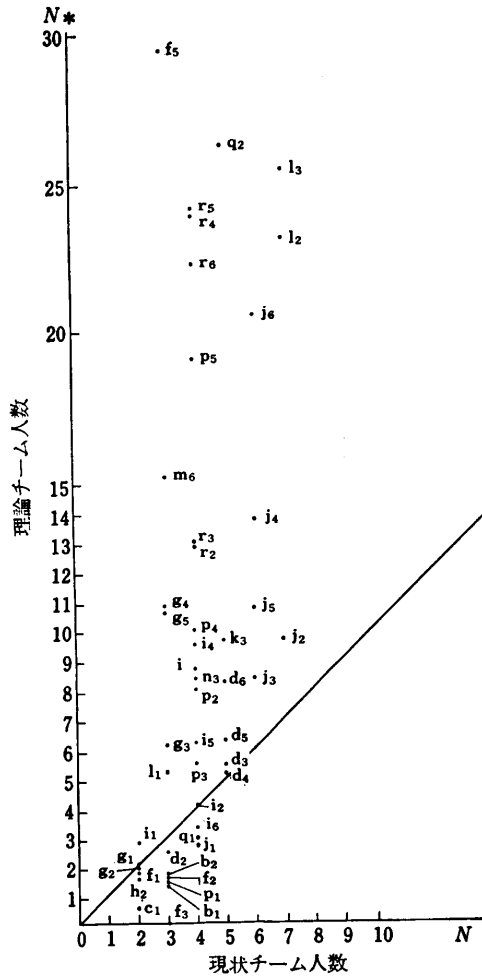


図5 期間最優先の場合

Fig. 5 Time is most important.

(表1), 関係者との協議により特急は  $p=100$  くらいであることが、普通は  $p=1$  くらいであることが合意された。その結果少し急ぐは特急と普通の間をとって  $p=10$  とした。

ii) 作業度  $q$  の算出:  $q=H'_1/t$  であったから、まず  $H'_1$  すなわち1人で作業をしたときの主体作業時間,  $t$  すなわち1調整ルート当りの調整作業時間を先に求める。

〈 $H'_1$  の算出〉式(2)により, チーム人数  $N$  で習熟係数  $\alpha$  のときの延べ主体作業時間  $H'_N$  は  $H'_N = H'_1 N^\alpha$  であったから  $H'_1$  は

$$H'_1 = \frac{H'_N}{N^\alpha} \tag{16}$$

より求めることができる。たとえば  $N=4$ ,  $\alpha=0.439$  で  $H'_N=80$  のときは  $H'_1=43.5$  となる。

〈 $t$  の算出〉式(4)により, チーム人数  $N$  でチーム

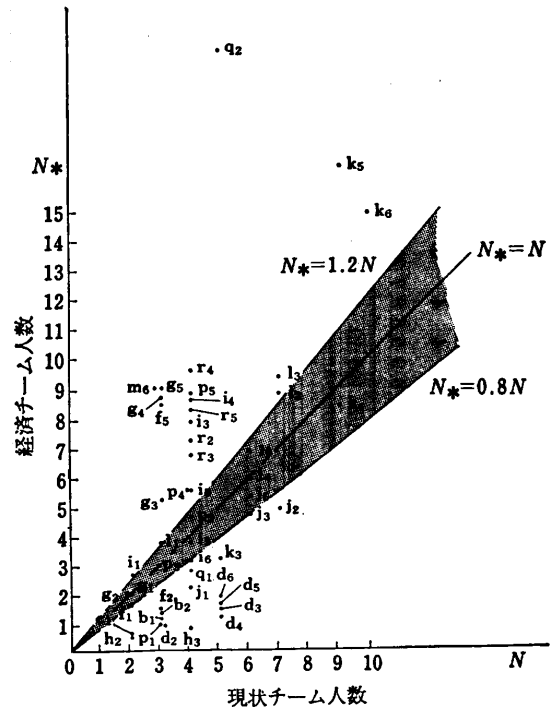


図6 経済性を考慮した場合

Fig. 6 Money is most important.

形態係数  $\lambda$  のときの延べ調整作業時間  $H'_N$  は  $H'_N = tN^\lambda$  であったから  $t$  は

$$t = \frac{H'_N}{N^\lambda} \tag{17}$$

より求めることができる。たとえば  $N=4$ ,  $\lambda=1.430$  で  $H'_N=120$  のときは  $t=16.5$  となる。

以上の方法によりプロジェクトごとに工程別に  $H'_1$  および  $t$  を求め、作業度  $q$  を算出する。たとえば  $H'_1=43.5$ ,  $t=16.5$  のときは  $q=2.64$  となる。

#### 4.5 経済チーム人数の算出

経済チーム人数を求めるためには4.3節で算出した習熟係数  $\alpha$  およびチーム形態係数  $\lambda$ , 4.4節で設定または算出した緊急度  $p$  および作業度  $q$  の値を式(8)に代入し, 式(10)によって順次  $N_i$  を求めれば経済チーム人数  $N_*$  が得られる。

たとえば  $N=4$  で  $\alpha=0.439$ ,  $\lambda=1.430$ ,  $q=2.64$  が算出され  $p=100$  のときは  $N_*=3.0$  となる。つまりチーム人数が3.0人のとき, もっとも経済的であるということである。

さて, 図5,6は実態調査のプロジェクトを工程別に現状チーム人数と理論チーム人数との関係をプロットしたものである。図5は  $p=\infty$  すなわち期間最優先, 図6は緊急度に応じ経済性を考慮したもので, 小文字





のアルファベットはプロジェクトを示し、サフィクスは工程で、1は基本設計、2は機能設計、3は詳細設計、4はコーディング、5はデバッグ、6はインテグレーションを示す。なお図5,6で a, c, e, n, o のプロジェクトが欠落している。これは実態調査はしたのであるが、いくつかのチームが集まった形になっていたためデータからはずしたことによる。また工程が欠落しているプロジェクトがあるが、これは工程が省略されていることによる。

#### 4.6 現状チーム人数と理論チーム人数の差に対する考察

図6において、もし実態が理論どおりであるならば直線  $N_* = N$  のまわりにばらつくはずである。しかし現状チーム人数と理論チーム人数の間に差が生じている。

相対誤差を20%とし  $0.8 \leq N_*/N \leq 1.2$  を満たすプロジェクトは実態と理論とが一致したデータと考え、まず i)  $1.2 < N_*/N$  を満たすプロジェクトについて、リーダーに  $N_*$  を示し増員しなかった理由を表2に、ii)  $N_*/N < 0.8$  を満たすプロジェクトについても同様にリーダーに  $N_*$  を示し減員しなかった理由を表3にまとめた。理由を大別すると工数面、設備面、予算面、工程面、管理面およびその他となった。まず増員しなかった理由(表2)では、6-1の“現状人員で十分と考えるから”が1件あるほかは増員したほうがよいことを認めている。その理由として多いのは1-1の“スキルの高い人も低い人もいないから”および1-2の“スキルの高い人がいない、低い人では教育に時間がかかる”である。また3-1の“予算がないから”というのもその次に多い。ここで予算とはソフトウェア外注費のことである。次に減員しなかった理由(表3)では、6-1の“現状人員で十分と考えるから”と6-2の“逆に人を増やしたいほどだから”とで3件ある外は減員したほうがよいことを認めている。その理由として多いのは1-2の“スキルがそろわないから”と1-3の“フルタイムで時間のとれない人がいるから”および2-2の“後工程で参加する必要があるから”である。

以上のように各プロジェクトリーダーはかずかずの理由により、経済性を加味した最適チーム編成にすることの困難性を認めている。したがって経済性を重視するならば、管理者等は減員したほうがよいチームから増員したほうがよいチームに人をダイナミックに移動させることを積極的におこない、より経済的なチーム

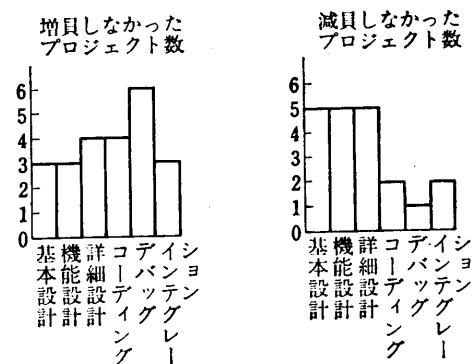


図7 増員または減員しなかったプロジェクト数  
Fig. 7 Number of project not to increase or decrease.

編成にすることが必要であろう。

また図7は表2および表3をもとに、各工程ごとに増員しなかった、あるいは減員しなかったプロジェクト数を示したものである。図7によれば、管理者等は可能ならば増員したかったのはデバッグ工程であり、また可能ならば減員してスキルの高い人にしたかったのは基本設計、機能設計および詳細設計の3工程であることを念頭に入れた上でチーム編成をすることが必要であろう。

以上が示すように本稿の提案はソフトウェア開発において経済性を加味したチーム編成を考える上で有用であると思われる。

## 5. むすび

本稿の特色はコミュニケーションに方向性を導入し、かつ自己調整という考えをもちいたことにある。この仮定により網型から星型まで連続的にその形態を入で示すことが可能となった。本稿はこの考え方をもとにソフトウェア開発工場におけるデータを分析した。その結果提案による方法はソフトウェア開発においてチーム編成をする際、管理者等がチーム人数を決定するための計画の一つの目安として有用であることが明らかになった。

しかし、まえがきにもものべたとおり特定の人の特別の能力に依存するような場合には限界があると思われる。さらにチーム人数も10人くらいまでで、それ以上は入が変わったり、サブチームが発生するなどすることが考えられる。このような場合には別の研究を待つほかはない。なお本稿の成果を整理すると

① 1人で作業をしたときの主体作業時間  $H_1$  と習熟係数  $\alpha$ , 1調整ルート当りの調整作業時間  $t$  と

チーム形態係数  $\lambda$  および緊急度  $p$  がわかれば式 (10) によって経済チーム人数  $N_*$  を求めることができる。

② 式(11)により数値解析的ではなく簡易的に経済チーム人数  $N_*$  を求める図(図2)を作成しておくことができる。

③ 多くのプロジェクトを  $\lambda$ - $\alpha$  図にプロットすることにより、事前に  $\lambda$  および  $\alpha$  を推定できるようになる。また逆に、あるプロジェクトを  $\lambda$ - $\alpha$  図に位置づけることにより管理者等はチーム形態および人の編成について事前に手がうてるようになる。

④  $H'_i$  および  $t$  が未知の場合は、現状チーム人数  $N$  および  $N$  人で作業をしたときの延主体作業時間  $H'_N$  と延調整作業時間  $H'_N$ , および  $\alpha$  と  $\lambda$  が既知であれば、式(16)と式(17)から  $H'_i$  および  $t$  を求め①の方法で  $N_*$  を求めることができる。

⑤ 実態調査から経済チーム人数  $N_*$  を求め、現状チーム人数  $N$  との差から増員または減員しなかった理由および工程を分析した。その結果、本稿による方法は経済的なチーム編成を考える上で有用であることが示された。

なお本稿の仮定では Brooks の考え方がベースとなり、延作業時間を延主体作業時間と延調整作業時間の和として考え、式(2)および式(4)によりチーム人数の関数として表した。しかしこの仮定の検証は2章でも示したように、小学児童を被験者とした実験結果を土台としている。

実際のソフトウェア開発での検証を土台としなかった理由は種々の特性をもったソフトウェア開発プロジェクトからサンプルを選び出し、さらに質的に同等な人を数多く選び出し、同じプロジェクトを1人、2人、3人、…のチームに、実験されていることを意識せず、数カ月にわたって作業をすることは現実的には困難であることによる。小学児童を被験者とした実験では「色紙による貼絵図案化作業」が選定されている。筆者は小学児童によるこの作業は成人によるソフトウェア開発作業とチーム編成効果を考える上で類似する点が多いとした。しかし将来可能であれば上記のような実験がされることを切望する。

また本研究には直接関係ないが、図4に関して本文中にもものべてあるが、基本設計、機能設計および詳細

設計は  $\Delta$  型に、コーディング、デバッグおよびインテグレーションは  $\nabla$  型に分布している根拠についての解析は今後の研究課題であろう。

謝辞 最後に本研究を進めるにあたり研究の場を提供していただくと同時に種々のアドバイスをいただいた日本電気ソフトウェア(株)登家正夫常務取締役役に心よりお礼申し上げます。またご指導いただいた慶応義塾大学の浦昭二教授および千住鎮雄教授に心よりお礼申し上げます。さらに研究開発プロジェクト管理という立場からアドバイスいただいた(株)日本能率協会高遠秋良常務取締役役にも誌面をかりて心よりお礼申し上げます。

### 参 考 文 献

- 1) Brooks, F. P.: The Mythical Man-Month, pp. 13-26, Addison-Wesley, Reading (1975) (山内正彌訳: ソフトウェア開発の神話, pp. 15-24, 企画センター, 東京 (1981)).
- 2) Mantei, M.: The Effects of Programming Team Structures on Programming Tasks, *Comm. ACM.*, Vol. 24, No. 3, pp. 106-113 (1981).
- 3) Shaw, M. E.: Communication Networks, in Berkowitz, L. (ed.), *Advances in Experimental Social Psychology*, Vol. 1, pp. 111-147, Academic Press, New York (1964).
- 4) 青井和夫: 小集団の構造と機能, 集団・組織・リーダーシップ, p. 127, 培風館, 東京 (1962).
- 5) 後藤兼一: 協同作業において作業時間を最小にするための人数の判定方法, 日本経営工学会誌, Vol. 31, No. 1, pp. 28-34 (1980).
- 6) 後藤兼一: チームリーダーの有無と作業時間の関係, 日本経営工学会誌, Vol. 32, No. 5, pp. 348-355 (1981).
- 7) Simon, H. A. (富田 勝訳): 機械はなぜ学習すべきか, *bit*, Vol. 14, No. 9, pp. 6-12 (1983).
- 8) 師岡孝次: 習熟性工学, pp. 8-9, 建帛社, 東京 (1969).
- 9) 後藤兼一: チームの人数および意見の不一致度と習熟を考慮した意思決定時間モデルについて, 日本行動計量学会誌, Vol. 8, No. 1, pp. 41-51 (1981).

(昭和58年5月9日受付)

(昭和58年9月13日採録)