

プログラム実効処理量に着目した
生産性評価方法の一提案

1G-5

関川弘和 高橋宗雄
(NTT 電気通信研究所)

1. はじめに

ソフトウェア開発において、生産性向上は重要な課題であり、これを的確に管理するためには、生産性を把握する適切な尺度が必要である。従来から一般的に用いられている尺度は、プログラムの実コード行数（出来高）と工数との比である。しかしこの方式は、マクロやサブルーチン等のプログラミング技法を用いると、行数の削減により見かけ上の生産性が下がるという問題があるため、生産性の評価方法として適切とはいえない。

ここでは、機能が同じプログラムが同じ実効処理量（マクロの場合では展開後の行数）になることを前提とし、プログラミング技法の採用による見かけ上の生産性の低下を補正する方法を提案する。

2. 従来の生産性尺度の問題点

(1) 従来の生産性の評価尺度“プログラム出来高/工数”については以下のような問題点がある。

- ① プログラミング技法の適用により見かけ上の生産性が下がる。
- ② 下手な作りをして出来高の大きいプログラムの方が、生産性が高く評価される可能性がある。

(2) ①の解決を目的として、生産性を機能量を用いて“機能量/工数”で表す方法が提案されているが⁽¹⁾、これには機能を機械的に定量化することが困難であるといった問題点がある。

本報告では、これらの問題点を解決するため、プログラミング技法による生産性向上効果を、プログラム出来高情報から求め、これをもとに従来の生産性を補正する評価方法を提案する。

3. 新しい評価尺度の提案

(1) 基本的考え方

- ① プログラミング技法として、マクロ、サブルーチン及びコメント記述の3種について考察する。
- ② 同じ機能のプログラムが同じ実効処理量になると仮定し、この実効処理量に相当するプログラムをすべてマクロ等を用いず手作業で記述した場合の生産性を基準値

とし、これと同じ生産性となるように、生産性の補正係数を導入する。

③ マクロ、サブルーチン、コメント記述による生産性の補正係数をそれぞれ α 、 β 、 γ とし、新しい評価尺度による生産性を、従来の生産性の $(1 + \alpha + \beta + \gamma)$ 倍に補正する。

(2) 補正係数の導出

生産性の補正係数の導出に当たっては、工数の測定は困難であるとの考えから、極力ツール等により簡便に測定可能とするために、工数に関する項目をプログラム出来高に関する項目へ対応付けることとする。これにより、プログラムの解析だけで生産性の補正を行うことが可能となる。

(i) マクロ呼出しを考慮した生産性補正

マクロ利用プログラムのマクロ展開前の出来高をP、マクロ展開後の出来高を M_t 、それぞれの工数を X_p 、 X_m 、補正係数を α とすると(図1参照)、前項②の考え方から次式が成り立つ。

$$\frac{P}{X_p} (1 + \alpha) = \frac{M_t}{X_m} \quad \text{[式1]}$$

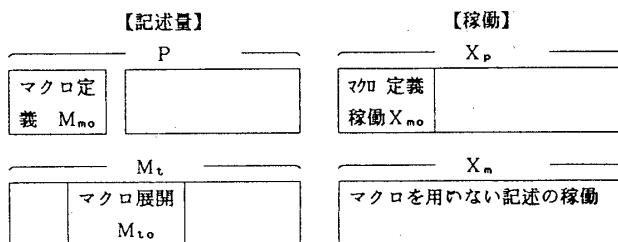


図1 マクロの記述量と稼働の関係

補正係数 α を求めるために、以下の前提を設ける

① [式1]の右辺(マクロを用いない記述の生産性)をマクロ定義を除いた非マクロ部分のプログラム記述に対する生産性に等しいとすると次式となる。

$$\frac{M_t}{X_m} = \frac{P - M_{m0}}{X_p - X_{m0}} \quad \text{[式2]}$$

② 式中の工数に関する項目をプログラム出来高の情報で表わすために、マクロを用いた場合と用いない場合との、単一工数当たりの行数の比 (m_m) を導入すると、 m_m は次式で表すことができる。

$$m_m = \frac{M_{t.o}/M_{m.o}}{(X_{m.o}/M_{m.o}) / \{ (X_p - X_{m.o}) / (P - X_{m.o}) \}} \quad [式3]$$

以上から α は次式となる。

$$\alpha = \frac{M_{t.o}}{m_m P} - \frac{M_{m.o}}{P} \quad [式4]$$

$M_{m.o}$: マクロ定義記述量

$M_{t.o}$: マクロの展開部分の量

(ii) サブルーチン呼出しを考慮した生産性補正

サブルーチン利用による生産性の補正係数 β も、マクロの場合と考え方は同様であり、次式で表すことができる。

$$\beta = \frac{S_{t.o}}{m_s P} - \frac{S_{m.o}}{P} \quad [式5]$$

$S_{m.o}$: サブルーチン記述量

$S_{t.o}$: 呼出回数を考慮したサブルーチンの延べ記述量

m_s : サブルーチンを用いた場合と用いない場合との、単一工数当たりの行数の比

(iii) コメント記述を考慮した生産性補正

コメントを除いたステップ数を P 、コメント記述量を C とし、全稼働を X_p 、全稼働中のコメント記述稼働を X_c とした時 (図2参照)、従来はコメント記述量は出来高に反映されておらず稼働にはコメント稼働が含まれていたため以下の式が成り立つこととなる。(補正係数 r)

$$\frac{P}{X_p} (1+r) = \frac{P}{X_p - X_c} \quad [式6]$$

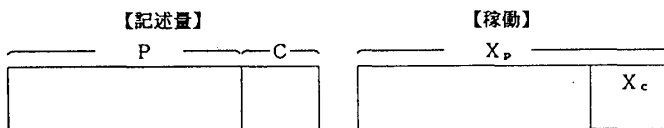


図2 コメントの記述量と稼働の関係

これより、

$$r = \frac{X_c}{X_p - X_c} \quad [式7]$$

以下の前提を設け、補正係数 r を求める。

①コーディング稼働は命令文とコメント行とで同じとする。

②コーディング後は命令文に対してのみ稼働がかかる。
ここで全稼働中のコーディング稼働比を μ とすると

$$X_c = \mu X_p \cdot \frac{C}{P+C} \quad [式8]$$

以上から r は次式となる。

$$r = \frac{\mu \cdot C}{P+C-\mu \cdot C} \quad [式9]$$

以上の結果を整理すると、生産性補正に必要な項目は表1に示すとおりとなる。

表1 生産性補正のための項目

No	項目	略称	備考
1	マクロ定義記述量	$M_{m.o}$	マクロ定義ソース記述量
2	マクロ展開部の量	$M_{t.o}$	マクロ展開部のステップ数
4	サブルーチン毎の呼出回数	$S_{c.t}$	両項目により、サブルーチンの呼出回数を考慮した延べステップ数 ($S_{t.o}$) を得る
5	サブルーチン単位	$S_{m.t}$	
6	サブルーチン記述量	$S_{m.o}$	
7	コメント記述量	C	命令文と同一行のコメントのカウントも必要

4. 計算例

上で求めた式に具体的な値をあてはめ計算した例を示す。

比例定数 m_m を次の仮定のもとに [式3] から求める。

マクロ展開量 ($M_{t.o}$) を定義量 ($M_{m.o}$) の3倍、マクロ化の工数を通常プログラミング工数の2割増しと仮定すると、 m_m の値は以下ようになる。

$$m_m = 3 / 1.2 = 2.5$$

プログラム出来高の各値を表2の例によれば、 $\alpha = 0.02$ 、 $\beta = 0.06$ 、 $r = 0.01$ となり、補正前の生産性 2ks/人年が 2.18ks/人年となる。

表2 生産性補正の計算例

マクロ		サブルーチン		コメント	
P	1 KS	P	1 KS	P	1 KS
$M_{m.o}$	0.1 KS	$S_{m.o}$	0.3 KS	C	0.2 KS
$M_{t.o}$	0.3 KS	$S_{c.t}$	0.9 KS	μ	0.06*
m_m	2.5	m_s	2.5		
α	0.02	β	0.06	r	0.01

*M工程比×M中のコーディング稼働比 = 0.2 × 0.3

補正前の生産性 (P/X_p) = 2 (ks/人年) ($X_p = 0.5$ 人年)

5. おわりに

プログラミング技法としてマクロ、サブルーチン、及びコメントを対象とし、生産性評価を補正する係数の導出を行った。これらは、いずれも出来高に関する基本的な数値の組合せからなっており、実用に供し易いものと考えられる。

比例定数については、その前提条件の妥当性も含めて、今後実データによる検証を行っていく予定である。

参考文献

(1) 秋間 他「ソフトウェア規模の測定尺度：機能量」第32回情報処全大 1H-3, 1986