技術ノート | TECHNICAL NOTE

品質評価に基づくソフトウェア生産性評価法の検討†

冨士仁*1　大森晃*2　尾越昌子*3　菅野文友*4

A New Indicator of Software Productivity based on Quality Evaluation

*1Hitoshi FUJI　*2Akira OHMORI　*3Shoko OGOSHI　*4Ayatomo KANNO

As an indicator of software producitivity, the ratio of software size to software development effort is often used. The productivity evaluation with this indicator depends on the physical volume of software, does not depend on its quality. Therefore, in some cases, we get a productivity evaluation of which we cannot be convinced immediately. This paper proposes a new productivity indicator that, as elements, explicitly has not only the physical volume of software but also software quality in the broad sense.

Furthermore, we make trial use of the new productivity indicator to real software systems, and we examine its usefulness.

The results of this study can be summarized as follows:

- The values of the new indicator are smaller than the values of the old indicator.
- In some cases, the orders of evaluation of the old indicator and the new indicator are reversed.
- The new indicator is more consistent with developer's actual feelings than the old indicator.

1．はじめに

ソフトウェアの生産性指標としては，従来，主として規模（ステップ数：LOC（Lines of Code））の工数に対する比（以下，従来指標と呼ぶ）が用いられてきた．しかし，従来指標に関しては次のような問題点を指摘できる．

たとえば，同一の仕様に対して，ある一定期間に同一のプログラミング言語を用いて，2人の開発者がそれぞれソフトウェア X と Y を開発したという状況を考えよう．仕様が同じであっても，その仕様を満たすソフトウェアは様々な方法で開発することが可能であり，アルゴリズムの違いやプログラムの作り方などに差異はあるが，たまたまソフトウェア X とソフトウェア Y の双方が同じステップ数で開発されたとする．しかしながら，ソフトウェア X の品質は極めて良いがソフトウェア Y の品質は劣悪であったとしよう．

この場合，従来指標による生産性評価ではソフトウェアの物理的な量に依存し，品質の程度には全く依存しないため，質的に極端な差異が生じていたとしても，ソフトウェア X とソフトウェア Y の生産性は同じであるという不可解な評価になる．本来はソフトウェア X のほうがソフトウェア Y よりも生産性が高いと評価されるべきところであるが，こうした不可

†平成8年5月29日　受付
平成8年11月27日　改訂
*1NTT ソフトウェア研究所
*2東京理科大学　工学部経営工学科
*3(株)日立製作所
*4帝京平成大学　情報研究科
連絡先：〒180 東京都武蔵野市緑町3-9-11（勤務先）

解な評価になるのは，品質の程度が従来指標による生産性評価に明示的に反映されていないからである．

ソフトウェアの生産性指標として，ファンクション・ポイント[1]に基づく指標も近年用いられている．そうした指標は異なるプログラミング言語で開発されたソフトウェアを対象にする場合には，従来指標よりも優れていると言われている．しかし，それは複雑さ要因を考慮してはいるが，ソフトウェアの機能性などを含めた広い意味での品質[2]を扱っているものではないため，従来指標と同様な問題を抱えざるを得ない．

以上のことから従来指標にしてもファンクション・ポイントに基づく指標にしても，前述したような不可解な生産性評価を回避するためには，品質の程度を明示的に判定できるように改善していく必要がある．その場合，双方の指標が質的に異なるため，個別に改善を試みることになる．

ここで，ファンクション・ポイントは計測が難しく習得に時間がかかることや，人によって測定結果がばらつくなどの問題がある．一方，ステップ数はルールさえ定めれば計測を自動化することも可能であり，ファンクション・ポイントに比べて過去の経験が豊富である．さらに，実際の開発現場では依然としてステップ数の計測が主流になっており，そのデータの蓄積も豊富である．こうした現状を踏まえて，本論文では従来指標の改善を試みることにした．

こうした考え方そのものは新しいものではなく，片岡[3]も「生産性評価において品質を考慮すべきである」と指摘している．また，石川ら[4]は開発工数の状況やドキュメント枚数に関する評価といった開発中の間接的な品質評価をとり入れた指標を提案している．

一方，本論文ではソフトウェアに関する広い意味での品質を従来指標に明示的にとり入れた生産性指標（以下，新指標と呼ぶ）を提案する．さらに，その指標を実際のソフトウェアに対して試用し，新指標の意義について考察する．

2．新しい生産性の指標

2.1 新指標の定義

本論文で提案する新指標は，ソフトウェアの運用段階における品質評価を生産性の評価に明示的に反映させるものである．具体的にはソフトウェアの品質を得点化し（以下，品質得点と呼ぶ），品質得点の満点に対する比率を品質得点率として，この品質得点率と規模（ステップ数）との積を新指標の分子とする．また，新指標の分母にはこれまでと同様に工数を用いる．したがって，品質得点率は(1)式によって，新指標は(2)式によって表すことができる．ここで，新指標における諸要素のうち品質得点率が品質評価を反映する要素である．

品質得点率：品質得点/満点……(1)

新指標：（規模×品質得点率）/工数…(2)

2.2 品質評価

品質得点率は，ソフトウェアの外部特性（プログラムの内部構造は考慮せず，ソフトウェアを利用することによってわかる特性）を用いた品質評価によって求める．その際，JIS X 0129[2]によって示されている品質評価のフレームワークを利用した．これは図・1上部の網掛け部分によって示してある．

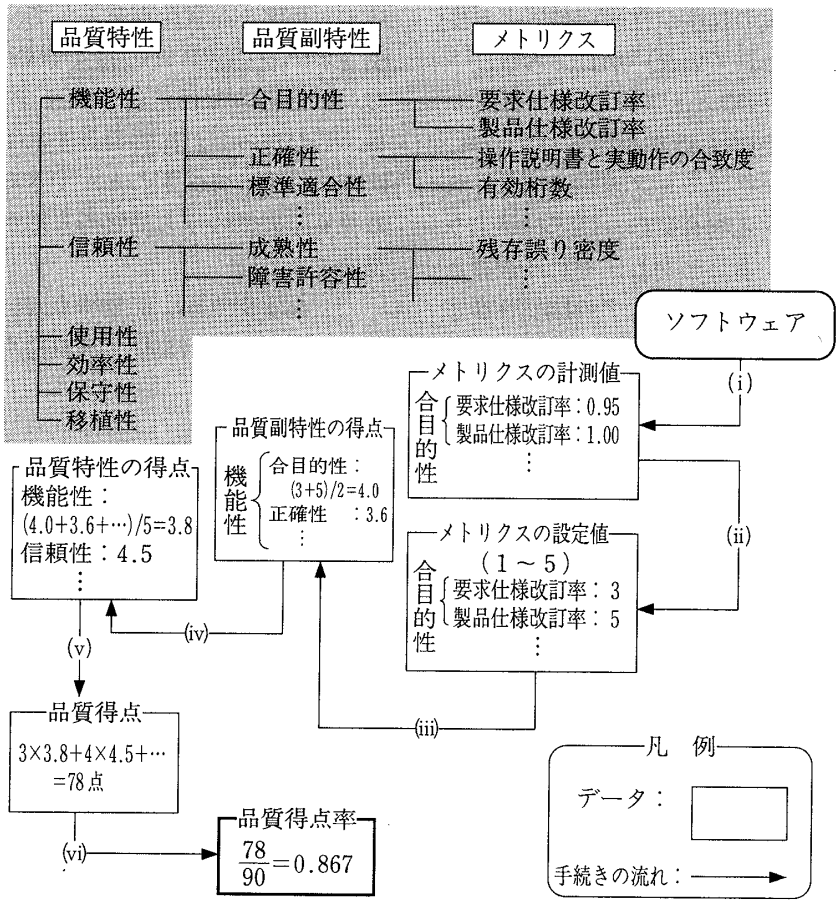
ここでは，まずソフトウェアの品質を6つの品質特性に分解している．そして，それぞれの品質特性をさらに複数の品質副特性に分けている．たとえば，機能性という品質特性は合目的性，正確性などといったように分解される．これら品質特性および品質副特性が外部特性と呼ばれるものである．さらに，その品質副特性には複数のメトリクスが対応している．これらのメトリクスを計測することによって，品質の評価ができるとされている．

2.3 品質得点率の算出方法

図・1は品質得点率の算出方法も示しており，以下にそれぞれの段階に対する説明を加える．なお，括弧内の説明は図・1との対応を示している．

(i) メトリクスを計測する（「要求仕様改訂率：0.95」，「製品仕様改訂率：1.00」を計測する）．

(ii) 各メトリクスの計測値を5段階で評定する（メトリクスの計測値と評定水準から「要求仕様改訂率：3」，「製品仕様改訂率：5」という評定結果を求める．この評定には原則として5段階の評定水準を用いる．メトリクスごとに計測できる値の範囲を5つの区間に分け，それぞれの区間に1～5の点数を付けた評定水準をあらかじめ決めておく．そして，メトリクスの計測値が評定水準のどの区間に位置するかに応じて，対応する点数をメトリクスに与える．場合によっては，一部のメトリクスに対しては，評定水準を5段階に設定しないほうが適切であることがある．このことについては，3.3節で言及する）．



図・1　品質得点率の算出手順

(iii)　品質副特性ごとに(ii)の平均を求め，それを品質副特性の得点とする（「要求仕様改訂率：3」，「製品仕様改訂率：5」の平均を計算し，品質副特性の得点「合目的性：4.0」を求める）．

(iv)　品質特性ごとに(iii)の平均点を求め，それを品質特性の得点とする（「合目的性：4.0」，「正確性：3.6」などの平均を計算し，品質特性の得点「機能性：3.8」を求める）．

(v)　評価対象のソフトウェアの種類と特徴に応じて各品質特性の重要度を決め，その重要度（W_i）と品質特性の得点によって加重和を求める．この加重和を品質得点とする（「重要度：3」×「機能性の得点：3.8」を計算し，他の品質特性についても同様に計算したものを合計して品質得点を求める）．

$$\text{品質得点} = W_1 \times \text{機能性の得点} + W_2 \times \text{信頼性の得点} + \cdots + W_6 \times \text{移植性の得点} \cdots (3)$$

(vi)　求めた品質得点の満点に対する比率を品質得点率(1)式として求める（「満点：90」と「品質得点：78」の比率を計算し，「品質得点率：0.867」が求まる）．

ここで，重要度と満点の設定について補足する．重要度は絶対的な値を設定することが困難であり，品質特性間で相対的に設定せざるを得ない．そこで，重要度は次のようなルールにそって設定するものとする．まず，重要度は最小値を0，最大値を5とし，0～5の範囲で0.5刻みの値をとる．最小値0は全く考慮に入れる必要がない品質特性に対して用意された値である．0.5刻みの値を持たせる理由は，重要度をできるだけきめ細かく調整できるようにするためである．次に，すべての品質特性の重要度に差がない場合には，平均的な重要度として3を各品質特性に設定する．また，これに伴って重要度の合計を18（一定値）とする．

このように，重要度は6種類の各品質特性の合計が18になるように，最小値0から最大値5までの範囲で0.5刻みで，各品質特性に相対的に割り振られることになる．重要度の合計を18としたことによって，すべての品質特性に関するメトリクスが評定水準の5を満たす場合を考えると，上記(vi)の計算に必要な満点は90となる．

この方法で求めると，品質得点率は最低でも18/90となるが，これはソフトウェアが運用段階に入るとき

に，最低限備えている品質の程度と考えて差し支えないと思われる．

3．ソフトウェアへの試用

3.1 試用対象

新指標を実際のソフトウェアに対して試用し評価を行った．対象にしたソフトウェアは3社合計で11種類である．X社のソフトウェアは通信制御用など，Y社のソフトウェアは放射能の計測用，Z社のソフトウェアはソースコードの測定用のソフトウェアである．これらのソフトウェアの概要を**表・1**に示す．

なお，ソフトウェアBおよびソフトウェアCはそれぞれソフトウェアAおよびソフトウェアBをバージョンアップしたものであり，ソフトウェアJとKは同一ソフトウェアをもとに，別々の機能を有するソフトウェアにバージョンアップしたものである．

品質を測定するメトリクスについては，JIS X 0129[2]および「平成3年度ソフトウェア開発・システムの文書化標準化調査研究報告書」[5]に示されているメトリクスを利用することにした．この試用に際して実際に用いた品質特性，品質副特性およびメトリクスの一覧を**表・2**に示す．

ソフトウェアの品質評価には本来ならばもっと多くの品質副特性を使い，各品質副特性ごとに複数のメトリクスを用いるべきであるが，開発が終了したソフトウェアからデータを抜粋したため，必ずしも十分とはいえないデータで評価した．しかし，後述の試用結果で示すように従来指標と新指標の生産性評価では差があることが示せた．

この試用結果は，新指標が実務的にも意義があるかどうかを確認することが目的であり，それは達成できたと考える．

表・1 ソフトウェアの概要

会社	ソフトウェア	機能	規模(kステップ)	工数(人月)
X	A	通信制御	21.0	38.09
	B	同上	25.4	8.92
	C	同上	43.8	30.77
	D	データベース用アプリケーション	7.9	20.83
Y	E	ゲートモニター	10.0	10
	F	ランドリモニター	4.4	3.5
	G	エリアモニター	3.0	2
	H	ダスト	6.0	7
	I	サーベイメーター	5.0	4
Z	J	ソースコード測定	8.7	13
	K	同上	5.2	8

3.2 品質特性の重要度

各ソフトウェアごとの品質特性の重要度は，下記のような観点と開発者との意見交換によって決定した．

(1) 通信制御ソフトウェアについては，特に機能性と効率性が重要である

(2) 放射能の計測用ソフトウェアについては，信頼性と使用性が重要である

(3) ソースコードの測定ソフトウェアについては，機能性と使用性が重要である

これらのことを考慮して設定した重要度を**表・3**に示す．なお，重要度は第2章で定義したように6種類の品質特性の合計が18になるように相対的に設定した．

製品A～Dについて，重要度0をつけた移植性は評価対象として考慮しなくてよいことを表す．このため，それ以外の5種類の品質特性についてメトリクスを計測する必要があるが，今回は保守性についてのメトリクスが計測されていなかった．このように品質得点率の算出に使う品質特性のメトリクスが計測されていない場合には，次節に詳述する計算方法を利用した．

したがって，今回の試用対象のソフトウェアで計測されているメトリクスは，表・2に示したようにすべての製品において機能性，信頼性，使用性，効率性に関するものだけあったので，それらに関する重要度しか品質得点率の算出には利用していない．

3.3 その他の方針

新指標によって生産性を評価するためには，前述(2.3節(ii))のようにメトリクスごとに評定水準を定めておく必要がある．今回の試用に利用した評定水準は，あるソフトウェアに対して，あらかじめ下記の一連の手順を数回適用することによって設定した．なお，評定水準を設定するために利用したソフトウェアは，後述する試用対象としたソフトウェアとは別のものである．

(I) 予測に従っていったん評定水準を決める

(II) それをもとに生産性の評価まで行う

(III) この結果が妥当と思われるかどうかについて開発者へインタビューを行う

(IV), (III) の内容を評定水準に反映する

このようにして設定された評定水準の内容を**表・4**に示す．

この評定水準を設定する際に，開発者の意見を反映して，「操作説明書との合致度」など一部のメトリク

表・2　メトリクス一覧

項番	品質特性	品質副特性	メトリクス	算　式	定　義
1	機能性	合目的性	製品仕様改訂率＊1	改訂機能項目数＊3 / 製品機能項目数＊3	製品機能が改訂（追加，変更，削除）された割合，プログラムまたはマニュアルのどちらかで評価する
2		正確性	操作説明書と実動作の合致度＊1	操作説明書と実動作の一致する機能項目数＊3 / 操作説明書の機能項目数＊3	プログラムとマニュアルの機能が一致している割合
3	信頼性	成熟性	残存誤り密度＊2	最終成果物に含まれた障害件数 / 最終成果物の量	ソフトウェア生産における最終成果物（ソースコードや文書）の単位量あたりに含まれる，ユーザに発見可能な障害の件数
4		障害許容性	システム停止発生率＊2	システム停止になった件数 / 発生故障件数	ソフトウェアシステムの稼働中に観測された故障のうち，ソフトウェアシステムのダウンに至ったものの割合
5		回復性	稼働率＊2	稼働状態にあった総時間数 / 観測時間数	ソフトウェアシステムが特定のインストレーションにおいて，一定の観測期間中に稼働状態にあった割合
6	使用性	習得性	マニュアル装備率＊1	マニュアルとして実現されている数＊3 / マニュアルとして備えていなければならない数＊4	対象とする機能に対して，オペレーションマニュアル，文法書，インストールマニュアルなどのリファレンスマニュアルや，オンラインマニュアル，自習書が備わっている割合
7		運用性	メッセージ的確度＊1	原因と処置が明確なメッセージ数 / メッセージの総数	システムに用意されている通知，応答メッセージの中に，原因とそれに対する処置が明確に述べられている割合
8	保守性	解析性	誤り箇所識別率＊2	ユーザーの誤り箇所識別故障件数 / 対象ソフトウェアの故障件数	保守者の故障解析の結果，評価対象のソフトウェアに誤りがあると判明した故障件数に対して，エンドユーザが正しく誤り箇所を識別できた故障件数の割合
9		安定性	誤り混入率＊2	誤り混入件数 / 変更ソースコード件数 または 誤り混入件数 / 訂正した誤り件数	ソフトウェアの改訂によって，新しい誤りが紛れ込む割合（正規化するデータの種類によって，先の二つの算式がある）

注）＊1：開発開始から終了時点までで計測，＊2：一通りの不具合が発生すると思われる半年以上の稼働後に測定，＊3：マニュアルに記載されている最小機能項目単位に計測，＊4：機能仕様書からマニュアルの最小機能項目を抽出して計測

表・3　品質特性の重要度

ソフトウェア	品質特性					
	機能性	信頼性	使用性	効率性	保守性	移植性
A, *B*, *C*, *D*	4.5	4.5	2.5	4	2.5	0
E, *F*, *G*, *H*, *I*	3	4	4	3	3	1
J, *K*	4	2	4	3	2	3

注）網掛けされた品質特性に関するメトリクスは計測されていないので，品質得点率の算出には利用していない．

スについては3段階の水準にするとともに評定水準を厳しく設定した．これらのメトリクスは良好な値をとるのが当然であると思われるものである．

品質得点率の計算に用いる満点90は，すべての品質特性についてのメトリクスが計測されていることを前提として設定されているが，今回扱ったデータではその前提が満たされないことがあった．メトリクスが計測されていないと評価ができないため，ある品質特

表・4　評定水準の範囲

項番	メトリクス ＼ 評定水準	5	4	3	2	1
1	製品仕様改訂率	0～0.05～	0.1～0.15～	0.2	～1	
2	操作説明書との合致度	－	－	1	～0.99～ 0.8	～0
3	残存誤り密度	0～0.02～0.05～	0.1～	0.2	～1	
4	ダウン発生率	0～0.05～0.01～	0.15～	0.2	～1	
5	稼働率	1～0.99～0.95～	0.9～	0.85～0		
6	マニュアル装備操率	－	－	1	～0.99～ 0.8	～0
7	メッセージ的確度	1～ 0.8～	0.6～	0.4～	0.2	～0
8	誤り箇所識別率	1～ 0.8～	0.6～	0.4～	0.2	～0
9	誤り混入率	0～ 0.02～0.05～	0.1～0.25～1			

性のメトリクスが計測されていないと，その特性に関する得点が0になってしまう．この場合，品質得点率の分母の計算と分子の計算に用いている品質特性の数が同じではなくなり，メトリクスが計測されている品質特性すべての得点が最高点をとったとしても，満点90には及ばないという不整合が起こる．たとえば，表・3のソフトウェアA～Dについて効率性に関するメトリクスが計測されていないとすると，重要度4を設定しているにも関わらず得点が0になるため，他の品質特性の得点がすべて5だとしても品質得点は70になる．

そこで，このような場合に対する対処策として，今回の試用評価においては品質得点率の計算方法を変更した．具体的には通常の満点90からメトリクスが計測されていない品質特性の相当分（その特性の重要度と評定の最高点である5との積）を減じるという方法をとった．

また，ある品質特性の重要度を0に設定した場合，その特性に関するメトリクスに対する評価結果は品質得点にまったく反映されなくなる．しかし，この場合は合計18の重要度が残りの品質特性に割り振られているため，残りの品質特性の得点が最高点の5になれば，品質得点は通常の満点と同じ90が得られる．したがって，重要度を0に設定した品質特性があったとしても，上記のような計算方法の変更を行う必要はない．

4．新指標の試用結果

4.1　X社のソフトウェアへの試用結果

X社によって開発時に記録されていたメトリクスとその計測値を**表・5**に示す．これらの測定値に表・4の評定水準を適用し，各メトリクスの値を評定した．その結果から求めた新指標と従来指標との比較を**図・2**に示す．工数の単位は人月であり規模（ステップ数）の単位はキロ・ステップである．

図・2においては，まず第1に，すべてのソフトウェアにおける生産性の評価が，新指標のほうが低くなっていることがわかる．そして，この減少率〔1－新指標の値/従来指標の値〕×100は，最低35％から最高60％にもなった．

次に，生産性の順位をみると，従来指標では$D<A<C<B$という順であったのに対して，新指標では$A<D<C<B$となり，ソフトウェアAとDの順位が逆転していることがわかる．この結果について開発者にイタンビューしたところ，実際の感覚に近いことが確認できた．

4.2　Y社のソフトウェアへの試用結果

Y社によって計測されていたメトリクスとその値を**表・6**に，4.1節と同様の手順でまとめた比較を**図・3**に示す．

図・3においてもすべてのソフトウェアについて，従来指標に比べて新指標のほうが生産性が低くなっている．この場合の減少率は18％～25％であり，あまりバラツキはみられなかった．

4.3　Z社のソフトウェアへの試用結果

Z社によって計測されていたメトリクスとその値を**表・7**に，4.1節と同様の手順でまとめた比較を**図・4**に示す．

ここでは，2つのソフトウェアしか比較していないが，減少率は20％～30％程度であり，生産性の評価順位が逆転している．この逆転の結果も開発者の実感と合致した．

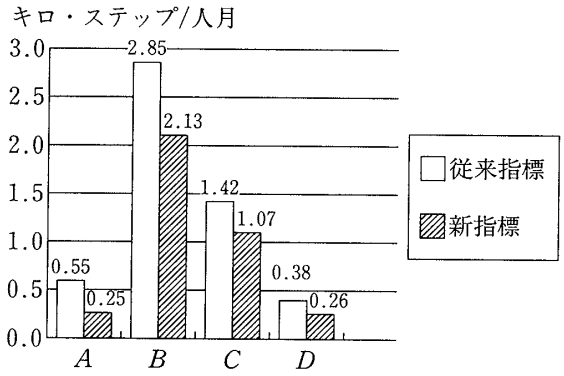
4.4　考　察

1)　試用に用いたメトリクス

メトリクスとしてはこれまでもいろいろなものが提案され使われてきたが，研究レベルにおいても実務レ

表・5　X社のソフトウェアの計測メトリクス

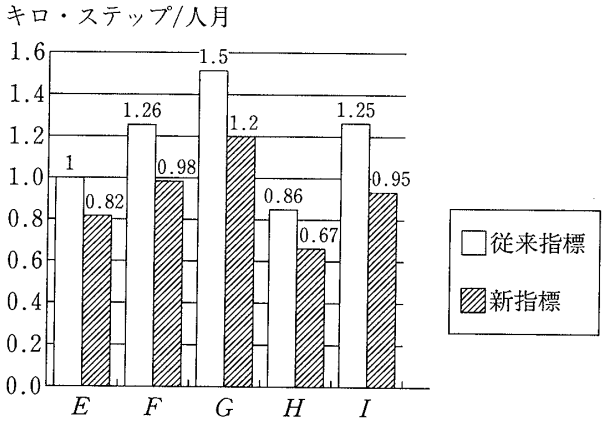
項番	メトリックス	ソフトウェア			
		A	B	C	D
1	操作説明書との合致度	0.98	0.99	0.99	0.99
2	残存誤り密度	0.14	0.08	0.02	0.25
3	ダウン発生率	0.25	0.00	0.00	0.00
4	稼働率	0.88	1.00	1.00	1.00
5	マニュアル装備率	1.00	1.00	1.00	1.00
6	メッセージ的確度	0.31	0.41	0.36	1.00
7	誤り混入率	0.14	0.16	0.05	0.25



図・2　X社のソフトウェアの生産性

表・6　Y社のソフトウェアの計測メトリクス

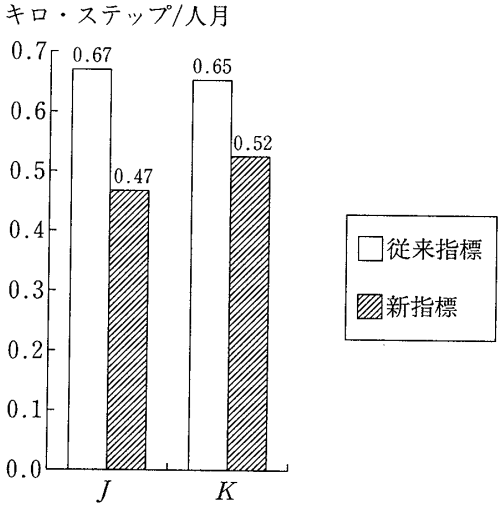
項番	メトリクス	ソフトウェア				
		E	F	G	H	I
1	製品仕様改訂率	0.07	0.16	0.60	0.05	0.14
2	操作説明書との合致度	1.00	1.00	1.00	1.00	1.00
3	稼働率	0.998	1.000	0.999	0.995	0.993
4	マニュアル装備率	1.00	1.00	1.00	1.00	1.00
5	誤り箇所識別率	1.00	1.00	1.00	0.50	0.67



図・3　Y社のソフトウェアの生産性

表・7　Y社のソフトウェアの計測メトリックス

項番	メトリクス	ソフトウェア	
		J	K
1	製品仕様改訂率	0.21	0.16
2	操作説明書との合致度	0.98	0.98
3	ダウン発生率	0.13	0.00
4	メッセージ的確度	1.00	0.98
5	誤り箇所識別率	0.87	0.95



図・4　Z社のソフトウェアの生産性

ベルにおいても，これだけのメトリクスを計測すればよいといった基準を得るまでには至っておらず，未だに各企業がデータを蓄積しながら自社の開発に適したメトリクスを模索している．

本論文では，JIS X 0129[2]および「平成3年度ソフトウェア開発・システムの文書化標準化調査研究報告書」[5]に示されている品質を測定するメトリクス（**表・2**参照）を利用した．これらの中では今回利用されていないメトリクスも示されているが，どのメトリクスを使えばよいかという点は言及されておらず，各企業で必要なものを測定すればよいと書かれている．このため，特にメトリクスの選択についての吟味はしていないが，今回用いたメトリクスが各社で測定され開発管理に利用されているものであることを考慮すれば，意味のないメトリクスを用いた比較とは思われない．

また，試用結果の比較は各ソフトウェアについて会社ごとに行った．このため比較対象となるソフトウェアは，お互いにメトリクスの種類と測定値の有無は同じであり，同一条件のもとで比較したことになる．したがって，今回のように比較するだけならば測定されたメトリクスが少なかったことは問題にはならないと考える．

さらに，表中のメトリクスは開発終了時点で測定可

能なもの（表・2項番1，2など）と，ある程度の運用期間が経過した後にしか測定できないもの（表・2項番3，4など）とに大別できる．前者は特に問題はないが，後者についてはメトリクスの計測時期を決める必要がある．しかし，一般的には，運用開始後にソフトウェアの品質の測定が行われるため，今回はこれに準じた時期に測定したデータを用いた．これについても一般解を求めることはできないため，各企業で行われているようにソフトウェアの種類や企業の方針に応じて決めたものを使うことで問題はないと思われる．

品質評価に用いるメトリクスによって新指標の値が変わることが考えられるが，どのようなメトリクスを使用すべきなのかについて，本論文では検討していない．この点に関する検討は今後の課題として残されていると考える．

2) 試用結果

今回検討した新指標を使った生産性評価の全体的傾向は，従来指標より少なくとも2割程度は評価が低くなったことが特徴的であった．これは，従来指標では明示的に考慮されていなかった品質評価の影響であり，物理的な量だけではなく品質評価も考慮に入れたために得られた結果である．

新指標の評価結果に対しては，あくまでも会社ごとにソフトウェア間での順位でしか捉えていないが，従来指標と順位が逆転することがあった．そして，逆転した順位のほうが実感に近いという感想が開発者から得られた．これは，数字に表す際には従来指標が使われている現状であっても，開発者が経験的に判断している生産性の順位に近いのは新指標であったということである．

5．おわりに

本論文では，ソフトウェアの属性として物理的な量だけを対象にしたソフトウェア生産性指標（従来指標）の欠点を鑑みて，物理的な量だけでなく品質の程度もとり入れた新しい生産性指標を提案した．新指標では品質の程度を品質得点率として具体化しており，計測されたメトリクスからそれを算出する方法も示した．さらに，実際のソフトウェアに対して新指標を試用し，その結果について検討した．

3社の11ソフトウェアに対して新指標を用いたところ，生産性の評価順位は場合によって従来指標とは異なる結果となった．こうした結果について面談によって開発者に吟味してもらったところ，開発者からは一様に新指標のほうが従来指標に比べて実感に近い評価であるという意見が得られた．

第1章で例示したように，2つのソフトウェアのステップ数（厳密に言えば，1単位工数あたりのステップ数）は同じであるが，品質には差異がある場合には従来指標による生産性評価は不可解なものとなる．しかし，同じステップ数で同じ品質のソフトウェアを比較する際には，従来指標でも特に問題は起こらない．

ここで，ステップ数が2つのソフトウェアで異なり，さらに品質も異なる場合を考えてみる．このような場合に対して，新指標と従来指標を比較した結果を示したのが第4章であり，新指標による改善がみられたことは既に述べたとおりである．次に，ステップ数が同じで品質が異なる場合，すなわち，第1章で例示したような場合も同様に生産性評価は開発者の実感に近い結果になると予想される．

一方，同一の仕様を完全に満たしている2つのソフトウェアが，品質は同じでステップ数が異なるという状況では従来指標と新指標とでは差異は生じるが，いずれの指標によって評価してもステップ数の大きいほうが生産性がよいことになる．しかしながら，従来指標では品質の程度が明示的に反映されていないために，品質に関するそうした状況を的確に把握した上での生産性評価になるという保証はない．

本論文では，ファンクション・ポイントによる生産性評価は対象にしなかった．しかし，異なるプログラミング言語で開発されたソフトウェアの生産性の比較も必要であるため，ファクション・ポイントによる生産性評価に品質の程度を加味していくことは重要である．その際，本論文で提案した新指標が参考になると考える．

ソフトウェア・メトリクスの研究は，まだまだ不十分な点が多く必ずしも検証が十分にできない状況下にある．たとえば，本論文ではJIS X 0129[2]や「平成3年度ソフトウェア開発・システムの文書化標準化調査研究報告書」[5]に示されているすべてのメトリクスを検討することが理想的なアプローチであったと思われるが，現時点では計測が困難なメトリクスもあり，そのようなアプローチはとれなかった．こうした状況のもとではあるが，本論文では新指標に意義があることを示した．

新指標を実務に適用しようとする場合には，各企業に適したメトリクスを探し出していくといった努力が必要になる．この点を考慮すると新指標を適用しなが

ら自社に適したメトリクスを模索していくことが，メトリクスの研究に寄与することにもなるであろう．さらに，メトリクスの計測といった品質管理の要素はオーバヘッドになるため，納期に追われて十分に行われていないという現状もある．このため，新指標を導入することによってメトリクス計測のモチベーションを高めれば，継続的な品質管理活動を推進することにもつながるであろう．

謝　辞

本研究の当初，東京理科大学在学中の小林大輔氏に様々な点でご協力いただいたことを感謝いたします．

参考文献

[1] Jones, C.(1991)：Applied Software Measurement-Assuring Productivity and Quality, McGraw-Hill.（鶴保征城・富野壽監訳（1993)：「ソフトウェア開発の定量化手法，構造計画研究所).

[2] JIS X 0129（1994)：“ソフトウェアの製品評価―品質特性及びその利用要領”，日本規格協会.

[3] 片岡雅憲（1985)：“ソフトウェアメトリックスの現状と課題”，「情報処理」，**26**，pp.42-53.

[4] 石川晴美・田上典子（1992)：“ソフトウェア生産性指標の試行評価”，第12回ソフトウェア生産における品質管理シンポジウム報文集，pp.89-96.

[5] 情報化技術標準化研究センタ（1992)：“平成3年度ソフトウェア開発・システムの文書化標準化調査研究報告書”，日本規格協会.