

## エンタープライズ系ソフトウェアの生産性に影響を与える要因の分析

古山恒夫

東海大学理学部情報数理学科  
〒259-1292 平塚市北金目 1117

概要：エンタープライズ系ソフトウェアの生産性に影響を与える要因を分析し、次のことを明らかにした。(1) 工数は規模だけでなく開発時のバグ数にも大きな影響を受ける。(2) セキュリティへの要求レベルが高いプロジェクト、スキル・員数ともに不十分なテスト体制のプロジェクト、プロジェクト管理ツール・構成管理ツールを利用しているプロジェクト、ユーザ担当者が受け入れ試験に関与しているプロジェクトは生産性が低い。(3) PM スキルの高いプロジェクトはテストを十分に実施して多くのバグを検出するため生産性が低下している。(4) ユーザの要求仕様への関与が高いプロジェクトは生産性が高い。(5) 作業スペースや騒音環境は単独では影響要因とならないが、他の要因の影響を加速させる効果もっている。

## Analysis of the Factors that Affect Productivity of Enterprise Software

Tsuneo Furuyama  
Tokai University

This paper reports the analysis results of clarifying factors that affect productivity of enterprise software projects as follows. (1) Number of faults detected during test phase is also dominant factor that affect development efforts as well as size. (2) Project where high security level software is required, project where test team consists of fewer members with low skill, project where various development tools are used, and project where customers commit themselves to acceptance test have low productivity. (3) Productivity of the project managed by skillful project manager is low because he/she tries to detect many faults. (4) Project where customers commit themselves to specifications ha high productivity. (5) Although narrow and noisy office does not directly affect productivity, it exaggerates effects of other factors.

### 1. はじめに

ソフトウェア開発では開発するソフトウェアの種別によって生産性のベースラインが異なる。例えば、基本ソフトウェアの生産性は、高い信頼性を確保するために多くのテストを繰り返すことなどにより一般のアプリケーションソフトウェアよりも生産性は低くなる。超大規模なソフトウェアは中規模なソフトウェアより多くの開発要員を必要とし、オーバーヘッドの増大などにより生産性は急激に低下する。開発対象ソフトウェアの種別を限定した場合にも、さまざまな影響要因のためプロジェクトごとに生産性がばらつく。生産性に影響を与える要因とその程度を明らかにすることができれば、たとえそれがモデル式の形で与えられていなくても、プロジェクト計画時の見積りの助けとなる。影響要因がプロジェクトで制御できるものであれば、プロジェクトを組織するとき、あるいはプロジェクト運営上の重要な助けとなる。もし、生産性に影響を与える要因同士の因果関係と影響の程度までが明らかになれば

ば、それらの定量的関係に基づいて生産性予測のモデル式を得ることができる。

ソフトウェアの開発工数に影響を与える要因については COCOMO[1]や COCOMO II [2]がよく知られている。COCOMO では、開発工数を規模のべき乗に調整係数を掛けたモデル式で予測する。規模のべき乗は概ね 1.05 から 1.2 が与えられる。調整係数は工数に影響を与える 15 個の要因で構成され、これらの要因が規模以外の影響要因ということになる。

COCOMO 以外にも開発工数や生産性の影響要因について分析したものがすでにいくつか報告されている[3][4][5][6]。影響要因について信頼度の高い普遍的な結果を得るためには、できる限り信頼度が高く、件数やデータ項目数が多いプロジェクトデータ群にたいして適切な分析方法を用いて分析する必要がある。

IPA (独立行政法人情報処理推進機構) SEC (ソフトウェア・エンジニアリングセンター) では、エンタープライズ系ソフトウェアプロジェクトのデー

タとして、規模・工数・工期などの最も基本的なプロジェクトデータ項目（量的データ）に加えて、プロジェクト特性を示すさまざまなデータ項目（質的データ）を定義し収集している。これらのデータに対して、統計的手法を用いて生産性に影響を与えているデータ項目を抽出し、これまでわれわれが経験的に感じていたこととつき合わせることができれば、プロジェクトの計画作成においてもプロジェクト運営においても大いに貢献できると考えられる。本論文では、重回帰分析や分散分析の手法を用いて生産性に与える要因を分析した結果を報告する。

第2章で分析対象データの概要を、第3章で分析方法を紹介する。第4章で分析結果すなわち生産性または工数と関連の見られる要因の候補を示す。第5章では要因の候補と生産性について考察を加える。

## 2. 分析対象データの概要

分析対象データは、2005年度にSECで収集したエンタプライズ系ソフトウェアプロジェクトのデータ1,419件である[7]。そのうち、分析結果の安定している新規開発データ846件のみを分析対象とする。さらに、ファンクションポイント（FP）と工数の両方のデータが揃っているもの534件が今回の分析対象データである。なお、データ定義は参考文献[7]のものを使用している。

### 2. 1 目的変数と説明変数

#### (1) 説明変数が量的データの場合

工数を目的変数とし、規模、工期、開発時のバグ数を説明変数とする。開発時のバグ数（以下では「バグ数」と省略）としては、「回答数の多い」、「検出バグ現象数結合テスト」と「検出バグ現象数総合テスト（ベンダ確認）」の和を用いる。量的データの基本統計量を表1に、各データ項目の相関行列を表2に示す。ただし、後述するように、量的データに対してはすべて対数変換後のデータに対して基本統計量を算出している。本論文では特に断らない限り、量的データはすべて対数変換後の値である。

#### (2) 説明変数が質的データの場合

生産性を目的変数とする。生産性は工数を規模で

表1 分析対象データの基本統計量

	Log(工数 /FP)	Log(工 数)	Log(FP)	Log(工 期)	Log(バ グ数)
平均	0.901	3.570	2.670	0.775	2.220
中央値	0.909	3.527	2.641	0.778	2.217
標準偏差	0.388	0.645	0.496	0.282	0.634
標本数	534	534	534	240	233

(注)単位は次のとおり。工数:人時、工期:月、バグ数:件

割ったものと規模を工数で割ったもののいずれもが使われているが、ここでは生産性=工数（人時）／ファンクションポイント（FP）（を対数変換したもの）とする。値が小さいほど生産性が高いことを意味していることに注意が必要である。

説明変数は、分析結果の普遍性を高めるため、規模と工数の揃ったデータのうちから次の条件を満たすものとする。

- ・データ数（回答数）が50件以上ある。
- ・回答の内容が極端に偏っていないもの、具体的には各レベルに少なくとも10件以上のデータがある。

表2 量的データの相関行列

	規模	工数	工期	バグ数
規模	-	0.799	0.726	0.764
工数	534	-	0.697	0.780
工期	160	239	-	0.566
バグ数	104(3)	233(3)	96(3)	-

(注)右上は相関係数、左下はデータ数、括弧内は値がゼロのデータ数で対数変換できないので分析対象から除外(別掲)

表3に説明変数としてとりあげた26件の質的データ項目を示す。業種（大分類）、アーキテクチャ、主開発言語については、データ白書2006(参考文献[7])にデータ項目ごとの中央値が掲載されており、生産性の順位については明らかであることから直接の分

表3 説明変数(質的データ)

分類	データ項目(*1)
開発プロジェクト全般	新技術利用/役割分担/責任所在/達成目標/優先度/明確度合
	作業スペース/プロジェクト環境/騒音
計画の評価	計画の評価(コスト/工期/品質)
ツール利用	プロジェクト管理ツール/構成管理ツール/設計支援ツール/ドキュメント作成ツール/デバッグ/テストツール
ユーザ要求管理	要求仕様/明確度合
	ユーザ担当者(要求仕様関与/受け入れ試験関与/システム経験)
	要求レベル(信頼性/性能/効率性/セキュリティ)
要員等スキル	PMスキル
	要員スキル(業務分野経験/分析・設計経験/言語・ツール利用経験/開発プラットフォーム使用経験)
	テスト体制

(\*1)「/」が項目の区切りを表す。

析対象からは除外する。

### 3. データの分析方法

#### 3. 1 データの前処理

##### (1) 量的データの対数変換

量的データのヒストグラムを描くといずれも右裾が長く伸びる分布をしていて正規分布とはならない。一方、それを対数変換したものは一般に正規分布の仮説が否定されない。このことから、量的データは生産性も含め、すべて対数変換をしてから分析する。

##### (2) 質的データの合併

各データ項目のうち、ツールの利用の有無などレベル分けが2つだけのものもあるが、データ項目の中には3レベル以上に分けられているものも少なくない。表3の質的変数では26個中20個が3レベル以上ある。本論文では各データ項目の影響の有無をマクロな意味で明らかにするために、レベル数が3つ以上のものは、近いレベルのものを合併して2つのレベルにまとめる。例えば4レベルある場合は、上位2つと下位2つをそれぞれ合併する。ただし、「テスト体制」のレベルb（スキルは十分、員数は不足）とレベルc（スキルは不足、員数は十分）には順序関係が存在しないので、これらとレベルa（スキル、員数ともに十分）を合併し、レベルd（スキル、員数ともに不足）との差を分析する。

なお、欠損データは補正せず、該当するデータ項目の分析に対してのみ除外する。

#### 3. 2 分析方法

##### 3. 2. 1 量的データの分析方法

3つの説明変数（規模、工期、バグ数）のあらゆる組み合わせに対して、工数を目的変数とする重回帰分析を行い、決定係数（=寄与率）を求める。決定係数はデータ数を考慮した補正後のものとする。

##### 3. 2. 2 質的データの分析方法

###### (1) 一元配置の分散分析

生産性を目的変数として、2レベルの一元配置の分散分析（t検定と同じ）を行うが、まず説明変数ごとに等分散の検定を行い、比較対照の2つのレベルで等分散の仮定が否定されない場合とされる場合に分けて分散分析を行う。分析の結果、P値が5%以下となったデータ項目を影響要因の候補とする。

###### (2) 二元配置の分散分析

要因同士の交互作用の有無を確認するために、表3に示す項目同士を組み合わせで二元配置の分散分析を行う。今回の分析では繰り返し数が異なることから計算は簡便法を用いる。分析の結果、交互作用があると認められた組み合わせに対し、2×2の各レベルの対すべて（6対）に対してt検定を行い、

次の2つのタイプに分類する。

- ・I型：ある特定のレベル同士の組み合わせが他のレベル同士の3つの組み合わせと有意なほど生産性が異なる。相乗効果型とも言える。
- ・II型：Aというデータ項目があるレベルのときにBというデータ項目の2つのレベル間の生産性の差が大きくなる場合である。Bのレベルの影響がAを触媒にして広がるとも言える。

#### 4. 分析結果

##### 4. 1 量的データの分析結果

説明変数の各組み合わせに対する（重）重回帰分析の結果を表4に示す。説明変数の数を増やすに従ってデータ数が減少する。1変数（説明変数が1つ）の場合のデータ数はそれぞれ233, 239, 534、2変数の場合はそれぞれ96, 104, 160、3変数の場合は52であった。

表4 工数に影響を与える量的要因(候補)の重回帰分析

説明変数の数	回帰係数(注)			切片	(重)相関係数	決定係数(補正後)	標本数
	規模	工期	バグ数				
1変数	1.040	-	-	0.794	0.799	0.639	534
	-	1.584	-	2.559	0.697	0.484	239
	-	-	0.814	2.273	0.780	0.607	233
2変数	0.783	0.619	-	1.108	0.870	0.754	160
	0.815	-	0.381	0.780	0.926	0.855	104
3変数	-	0.726	0.645	1.990	0.836	0.692	96
	0.699	0.170 #	0.361	1.013	0.922	0.840	52

(注) # 印はP値が5%を超えるもの。無印はP値が1%未満のもの

###### (1) 1変数の場合

工数と最も相関の高いのが規模で0.799であり、続いてバグ数の0.780である。工数と工期の相関係数は規模やバグ数よりも小さく0.697にとどまっている。工数に対する規模の回帰係数は、1.04で1よりわずかに大きく規模の増加の割合より工数の増加の割合が大きいことを示している。工数に対する工期の回帰係数は1.584、バグ数の回帰係数は0.814で1よりも小さい。バグ数は規模の増加の割合よりも増加の割合が低いことを示している。

###### (2) 2変数の場合

規模とバグ数を説明変数とした場合が最も高い重相関係数0.926を示している。決定係数は0.857であり、工数の86%は規模とバグ数で決定されることがわかる。

###### (3) 3変数の場合

規模、バグ数、工期すべてを説明変数とする重回

帰分析では、工期に関するP値が有意とならなかった。すなわち、工数に工期の影響があるとは言えないことを示している。一般に説明変数を増やすと重相関係数は大きくなる。しかし、今回の分析結果では、3変数による重相関係数は0.922であり、規模とバグ数の2変数を説明変数とする重相関係数0.926よりも低い。これは分析対象であるデータ集合が異なる(2変数から3変数に説明変数を増やした際にデータ集合が縮小した)ために起きた現象である。

#### 4. 2 質的データの分析結果

目的変数(生産性)に対するP値が5%以下の説明変数(影響要因の候補)に対して、レベル毎のデータ数、生産性の平均と分散、P値、寄与率を表5に示す。

26個の説明変数のうち、次の9項目の質的要因が生産性に影響要因の候補となった。

- (1) 要求レベルに関するデータ項目のうち、性能・効率性とセキュリティが有意である。性能・効率性は要求レベルが高いほど生産性が高く(危険率5%で有意、以下この節では「危険率」と「有意」を省略)、セキュリティは要求レベルが高いほど生産性が低い(1%)。
- (2) ユーザ担当者に関するデータ項目のうち、要求仕様関与、受け入れ試験関与が有意である。ユーザ担当者が要求仕様に強く関与している場合生産性が高くなる(1%)。受け入れ試験に関与している方が生産性が低い(1%)。
- (3) 開発プロジェクト全般に関するデータ項目のうち、役割分担\_責任所在が有意である。役割分担\_責任所在が明確な方が生産性が高い(5%)。
- (4) 要員等スキルに関するデータ項目のうち、PMスキル、テスト体制が有意である。PMスキルが高いほど生産性が低い(1%)。テスト要員のスキル・

表5 生産性に影響を与える質的要因の候補に関する分散分析の結果と対比するグループ間の生産性の比

分類	データ項目(影響要因の候補)	レベル	データ数	生産性(注1)		P値(上段) 寄与率(下段)	典型的プロジェクト(注2)	
				平均(A)	分散		生産性(B)	生産性の比
要求レベル	要求レベル_性能・効率性	極めて高い+高い	56	0.741	0.133	2.7%	5.51	1.33
		中位+低い	117	0.866	0.111	2.8%	7.34	
	要求レベル_セキュリティ	極めて高い+高い	30	0.928	0.165	0.1%	8.48	1.82
		中位+低い	80	0.669	0.106	10.0%	4.67	
ユーザの関与	ユーザ担当者_要求仕様関与	十分に関与+概ね関与	100	0.79	0.133	0.0%	6.17	1.76
		関与が不十分+未関与	60	1.037	0.146	9.5%	10.88	
	ユーザ担当者_受け入れ試験関与	十分に関与+概ね関与	67	0.821	0.129	0.1%	6.62	1.72
		関与が不十分+全く関与していない	39	0.585	0.109	9.8%	3.84	
開発プロジェクト全般と要員等スキル	役割分担_責任所在	非常に明確	39	0.668	0.142	4.4%	4.66	1.38
		概ね明確+やや不明確	82	0.809	0.119	3.4%	6.45	
	PMスキル	レベル6、7+レベル5(高い)	41	0.984	0.205	0.2%	9.64	1.73
		レベル4+レベル3(低い)	86	0.746	0.129	7.6%	5.57	
	テスト体制	スキル、員数のいずれかが十分	77	0.698	0.097	0.2%	4.98	1.74
		スキル、員数とも不足	25	0.937	0.157	8.8%	8.65	
ツールの利用	プロジェクト管理ツール_利用	有り	42	0.947	0.21	2.3%	8.86	1.49
		無し	85	0.774	0.135	4.1%	5.94	
	構成管理ツール_利用	有り	45	0.984	0.196	0.0%	9.63	2.07
		無し	69	0.668	0.093	15.3%	4.66	

(注1)Log(工数(人時)/FP)

(注2)(B)は(A)を対数逆変換したもので単位は「工数(人時)/FP」である。

員数とも不十分な場合、少なくともいずれかが十分な場合よりも生産性が低い(1%)。

(5) ツール利用に関しては、プロジェクト管理ツール利用(5%)、構成管理ツール利用(1%)が有意である。いずれもツールを利用する方が生産性が低い。

#### 4. 3 二元配置の分散分析結果

##### (1) I型(相乗効果型)

分析結果を表6に示す。6つの組み合わせに対して交互作用が見られた。まず、要求レベル\_セキュリティが高い場合に、次のいずれかの条件が起きると、相乗効果となって少なくとも生産性が2倍以上低下する。①ユーザ担当者受け入れ試験が高い。②ドキュメント作成支援ツールを利用している。③プロジェクト環境・騒音の状況が悪い。また、構成管理ツールを利用している場合に、④役割分担\_責任所在が明確でなかったり、⑤ユーザ担当者のシステム経験が不足していたりすると生産性が2倍以上低下する。さらに、⑥作業スペースに余裕がなく、ユーザ担当者のシステム経験が不足していると生産性が1.7倍低下する。

##### (2) II型(触媒型)

分析結果を表7に示す。6つの組み合わせに対して交互作用が見られた。⑦ユーザの要求仕様への関与があって要求仕様が明確である場合は、ユーザの要求仕様への関与がない場合に比べて2.33倍生産性が向上する。また、プロジェクト環境・騒音の状況が悪くなると、次に示すデータ項目ではレベルによ

る生産性の差が増大する。⑧役割分担が非常に明確である場合はそうでない場合より3.75倍生産性が高い。⑨テスト体制でスキル・要員ともに不十分である場合はそうでない場合に比べ3.81倍生産性が低下する。⑩ユーザ担当者のシステム経験がある方がない場合に比べて2.91倍生産性が向上する。⑪ユーザ担当者が受け入れ試験に関与すると3.62倍生産性が低下する。

さらに、⑫作業スペースが狭いと要求レベル\_セキュリティの強さが生産性に2.79倍の差をもたらす。

## 5. 考察

### 5. 1 量的データの分析結果について

#### 5. 1. 1 重回帰分析の結果の検討

4. 1の分析結果から量的変数を用いた工数の予測式としては次の式が最も適切であることがわかる。

$$\log E = 0.815 \log S + 0.381 \log D + 0.780 \quad (1)$$

ただし、 $E$ は工数(人時)、 $S$ は規模(FP)、 $D$ はバグ数を表す。

式(1)の両辺それぞれから $\log S$ を引いて式を書き直すと次の式に変形される。

$$\log(E/S) = 0.196 \log S + 0.381 \log(D/S) + 0.780 \quad (2)$$

式(2)は、生産性(規模あたりの工数)は規模とバグ密度の一次式で表され、生産性は開発時バグ密度の3乗根と規模の5乗根に比例して低下することを意味している。生産性が規模の5乗根に比例して低下することは、工数が規模の1.2乗に比例して増加することと同じことである。1.2乗は基本COCOMO

表6 二元配置の分散分析の結果(I型:相乗効果型)

データ項目	相乗効果を生み出す組み合わせ				左記以外のデータ					P値	生産性の比(*2)
	レベル	データ項目	レベル	データ数	生産性		データ数	生産性			
					平均	分散		平均	分散		
要求レベル_セキュリティ	極めて高い+高い	ユーザ担当者_受け入れ試験関与	関与が不十分+全く関与していない	19	1.094	0.132	87	0.655	0.096	0.0%	2.75
		ドキュメント作成ツール利用	有り	17	1.034	0.158	89	0.675	0.103	0.0%	2.29
		プロジェクト環境_騒音	時としてかなり騒音があり、電話も作業をときどき中断する	17	1.197	0.112	89	0.667	0.095	0.0%	3.39
構成管理ツール利用	有り	役割分担_責任所在	やや不明確+不明確	26	0.987	0.111	79	0.657	0.109	0.0%	2.14
		ユーザ担当者_システム経験	経験が不十分+未経験	16	0.989	0.153	83	0.659	0.104	0.0%	2.13
作業スペース	(*1)	ユーザ担当者_システム経験	経験が不十分+未経験	19	0.903	0.19	85	0.669	0.121	0.0%	1.72

(\*1) やや狭くオープンスペース、思考の集中は持続しにくい環境+明らかに狭くオープンスペース、資料や計算機の設置場所もない。(\*2)典型的プロジェクト(対数逆変換後)での比

表 7 二元配置の分散分析の結果 (II 型:触媒型)

データ項目(触媒)	レベル	データ項目	レベル	生産性		データ数	P 値	生産性の比(*2)
				平均	分散			
要求仕様_明確度合	非常に明確+かなり明確	ユーザ担当者_要求仕様関与	十分に関与+概ね関与	0.745	0.123	53	0.0%	2.33
			関与が不十分+未関与	1.112	0.189	32		
プロジェクト環境_騒音	時としてかなり騒音があり、電話も作業をときどき中断する	役割分担_責任所在	非常に明確	0.369	0.088	10	0.0%	3.75
			概ね明確+やや不明確	0.944	0.153	21		
		テスト体制	スキル、員数のいずれかが十分	0.525	0.041	14	0.0%	3.81
			スキル、員数ともに不足	1.106	0.123	14		
		ユーザ担当者_受け入れ試験関与	十分に関与+概ね関与	0.992	0.145	18	0.0%	3.62
			関与が不十分+全く関与していない	0.434	0.105	13		
ユーザ担当者_システム経験	十分に経験+概ね経験	0.489	0.112	13	0.3%	2.91		
	経験が不十分+未経験	0.953	0.183	18				
作業スペース	(*1)	要求レベル_セキュリティ	極めて高い+高い	0.983	0.193	16	0.1%	2.79
			中位+低い	0.537	0.119	22		

(\*1) やや狭くオープンスペース、思考の集中は持続しにくい環境+明らかに狭くオープンスペース、資料や計算機の設置場所もない。(\*2) 典型的プロジェクト(対数逆変換後)の比

における Embedded の場合と同じ値である[1]。

さらに式(2)は次のような式にも変形できる。

$$\log(E/S^{1.196}) = 0.381 \log(D/S) + 0.780 \quad (3)$$

式(3)は、規模の重み付き生産性はバグ密度の3乗根に比例して低下することを意味している。

ところで、式(1)を対数逆変換して生のデータを変数とする式に変換すると次の式が得られる。

$$E = 6.03 S^{0.815} D^{0.381} \quad (4)$$

説明変数・目的変数ともに対数の世界での回帰分析では、対数変換前の変数のスケーリングファクタは回帰分析の切片を変化させるだけで、回帰係数には影響を与えない。また、両対数の世界で回帰分析を行う線形モデルは、元の変数の世界では式(4)で表されるような各変数のべき乗の積のモデルと等しい。各変数の間にそれぞれ高い相関がある場合は、各変数のべき乗の総和、すなわち回帰係数の総和はほぼ一定になる。表 3 から容易に計算できるように、ソフトウェアの生産性の場合、この値は 1.3 前後である。したがって、相関の高い説明変数を増やすことは各回帰係数の値が小さくすることになり、必然的に有意ではない変数が増える。今回の分析では 2 変数で十分である理由はこの理由による。

### 5. 1. 2 バグ数を説明変数に加えた効果

バグ数を説明変数に加えることにより重相関係数が 0.926、決定係数が 86%にも達する。この結果の普遍性を確かめるために、業種(大分類)、アーキテクチャ、主開発言語のうちデータ数が 10 以上の分類

項目について重相関係数と決定係数を調べた。その結果を表 8 に示す。すべてのデータ項目におけるすべての分類項目にわたって重相関係数が 0.92 以上、決定係数が 0.82 以上となった。特に、金融保険業では、重相関係数が 0.968、決定係数が 0.933 にも達し、規模とバグ数で工数の 93%が説明できることを

表 8 説明変数にバグ数を加えた効果

分類	層別	データ数	説明変数が規模		説明変数が規模とバグ数	
			相関係数	決定係数(補正後)	重相関係数	決定係数(補正後)
業種	製造業	15	0.884	0.764	0.922	0.825
	卸売・小売業	11	0.943	0.876	0.943	0.861
	金融・保険業	41	0.939	0.880	0.968	0.933
	サービス業	12	0.953	0.900	0.957	0.898
アーキテクチャ	2階層クライアント/サーバ	15	0.827	0.660	0.930	0.843
	3階層クライアント/サーバ	33	0.892	0.789	0.929	0.853
	イントラネット/インターネット	49	0.926	0.855	0.930	0.859
主開発言語	COBOL	20	0.870	0.743	0.943	0.876
	VB	16	0.869	0.738	0.931	0.847
	Java	15	0.922	0.839	0.952	0.890

示している。これらのことから、工数は規模とバグ数で85%程度決定されるということはエンタプライズ系ソフトウェア開発では普遍的な事項と考えてよい。

一般にバグ数は計画段階ではなかなか把握できない。工数に対するバグ数の影響がこれだけ大きいことは計画段階では工数予測に限度があることを示している。また、よく知られた事実、「テスト工程でのバグ数を減らすこと、すなわち作業のやり直しを少なくすること」が工数の増大を抑える、そのためには上流工程で設計レビュー等によりバグを減らしておくことが重要である、ということを裏付けている。

### 5. 2 分散分析一元配置の分析結果について

4. 2で述べた9つの要因候補について、次の観点から考察を加える。

- ① 対象となるグループの生産性が生産性の全体平均 (0.901 (表 1)) より高いか
- ② 比較対象グループ同士で典型的なプロジェクトの生産性の比率はどの程度か
- ③ 要因候補は筆者の感覚と合致するか

典型的プロジェクトの生産性の比率を表 5 に合わせて示している。ここで典型的プロジェクトとは、対数変換後のデータの平均値を逆変換したものである。比較対照グループ同士の生産性の比率は 1.3~2.1 倍の間にある。すなわち、これらの要因単独の影響により、1.3~2.1 倍生産性が異なると言える。

#### (1) 要求レベルの影響要因

「セキュリティの高いプロジェクトは生産性が平均以下に低下する」ということは、セキュリティの高いプロジェクトは高度な機能や複雑な構成を持ったソフトウェアの代名詞と考えると納得のいく結果である。

性能・効率性への要求レベルが低いと生産性が低下するという事は、セキュリティへの要求レベルの高さと矛盾する。矛盾の意味は与えられたデータだけでは説明できない。しかし、生産性そのものは全体平均より高く、現実にはそれほど大きな問題ではないと考えられる。

#### (2) ユーザ担当者の関与

「ユーザ担当者の要求仕様への関与がないプロジェクトは生産性が平均以下に低下する」ということは筆者の感覚と一致する。また、プロジェクト運営にとって重要な指針となる。

受け入れ試験に関与すると生産性が低下するが、生産性そのものは平均より高いので問題は少ない。むしろ、品質 (信頼性) の向上に役立っているのではないかと推測される。

#### (3) プロジェクト全般と PM のスキル等

役割分担\_責任所在が非常に明確なプロジェクトはそうでないプロジェクトに比べて生産性が高いが、そうでないプロジェクトでも生産性の平均は全体平均より高いことからこの結果のインパクトは大きくない。

「PM スキルの高いプロジェクトの方が低いものより典型的なプロジェクトと比較して 1.73 倍生産性が低い」という結果は筆者の感覚、おそらく一般的な常識にも反する。生産性そのものも平均より低い。典型的なプロジェクトと比較した場合、PM スキルの高いプロジェクトは低いものより平均的に 2.6 倍以上大規模であるが、このことを考慮しても意外な結果である。この点を明らかにするために、PM スキルレベルに対するバグ数とテストケース数を調べた結果を表 9 に示す。ここでテストケース数とは「テストケース数結合テスト」と「テストケース数総合テスト (ベンダ確認)」の和 (対数変換したもの) である。表 9 から明らかなのは PM スキルが高いプロジェクトはそうでないものに比較して、テスト工程で統計的に有意 (5%) などほど多くのバグを検出していることがわかる。検出バグ数は典型的なプロジェクトと比較して 2.3 倍の違いがある。さらにテストケース数でも同様の差がみられる。このことから、「スキルレベルの高い PM は多くのバグを検出するためにテストを十分に行う。そのために生産性が低下する」ということがわかる。

「スキル・要員とも十分なテスト体制のプロジェクトはそうでないプロジェクトに比べて生産性が平均以下に低下する」ということは筆者の感覚に合致する。

表 9 PM スキルレベルに対するバグ数とテストケース数

データ項目	バグ数		テストケース数	
	高い	低い	高い	低い
PM スキル				
平均	2.625	2.264	3.769	3.419
分散	0.258	0.245	0.235	0.314
標本数	22	19	24	19
P 値	2.7%		3.4%	

#### (4) ツールの利用

「プロジェクト管理ツール、構成管理ツール、設計支援ツールのいずれを使用しているも生産性は平均以下に低下する」ということがわかる。しかし、例えば構成管理ツールは品質 (信頼性) の向上に寄与している可能性も高く、単純に否定的に捉えることはできない。

### 5. 3 分散分析二元配置の分析結果について

二元配置の分散分析の結果から得られた 12 の組

み合わせに含まれる、①要求レベル\_セキュリティ、②ユーザ担当者\_要求仕様関与、③ユーザ担当者\_受け入れ試験関与、④構成管理ツールの利用、⑤役割分担\_責任所在、⑥テスト体制は、単独で生産性への影響要因ともなっている。⑦ユーザ担当者\_システム経験は、一元配置の分散分析の結果ではP値が9.3%であり、二元配置で影響要因の候補となることに違和感はない。しかし、⑧要求仕様\_明確度合、⑨作業スペース、⑩プロジェクト環境\_騒音、⑪ドキュメント作成ツール利用は、一元配置の分散分析では全く影響要因の候補となりえないものであった。

二元配置での分析結果 12 件を生産性の比でみると、1 件（作業スペースとユーザ担当者\_システム経験）を除いてすべて 2 倍を超えている。中には 3 倍を超えているものもある。この結果から、二元配置の分析によって詳細かつ影響の大きい要因の組み合わせを発見することができたと言える。

具体的な組み合わせとして興味深いものは次のものである。

- (1) 要求仕様が明確な場合、ユーザ担当者の関与の有無が生産性に大きな差を生み出す。ユーザ担当者が関与して明確な要求仕様を作成した場合は生産性が向上するが、逆の場合は却って生産性が低下する。これはユーザの意見を聞かずに要求仕様を固めることが問題であることを示している。
- (2) プロジェクト環境\_騒音に関係する要因の組み合わせ6つのうち5つはII型であって、騒音状況が悪そうに思えても却って生産性が向上している。これは開発活動が活発に行われていて騒々しいことを表している可能性がある。
- (3) 作業スペースが関係する組み合わせがI型とII型にわかれているが、II型に分類されたものも実はI型に近いものである。作業スペースの狭さは、生産性へのプラス要因はなく、他の要因との組み合わせによって生産性を低下させるだけに働くものと考えられる。
- (4) セキュリティの高いソフトウェア（一般に複雑な構成を持ったソフトウェア）を開発する場合は、ユーザが受け入れ試験に関与することやドキュメント作成ツールを利用することはやむを得ないにしても、騒音や作業スペースなどの開発環境は整えてことが重要であると言える。

## 6. おわりに

エンタプライズ系ソフトウェア開発におけるプロジェクトの生産性に影響を与える要因を分析し、次

のことを明らかにした。

- (1) 工数は規模だけでなく開発時のバグ数にも大きな影響を受ける。
- (2) セキュリティに対する要求レベルが高いプロジェクトやスキル・員数とも不十分なテスト体制をもつプロジェクト、プロジェクト管理ツールや構成管理ツールを利用しているプロジェクトは生産性が低くなる。
- (3) PM スキルの高いプロジェクトも生産性が低下するが、これはPMが多くのバグをテスト工程で検出しているのが一因である。
- (4) ユーザ担当者の要求仕様への関与が高いプロジェクトは生産性が高くなる。
- (5) 作業スペースや騒音環境は単独では影響要因とならないが、他の要因の影響を加速させる効果をもっている。
- (6) ふたつの要因による生産性への影響は、要因単独の場合よりも大きい。

謝辞 本研究は、経済産業省の支援により、東海大学とIPA SECが共同で実施したものである。SEC 鶴保所長はじめ定量データ分析部会委員各位及びSEC 研究員の方々のご協力に深く感謝致します。

参考文献

- [1] Boehm, B. W.: Software Engineering Economics, p. 767, Prentice-Hall, Inc. (1981).
- [2] B. W. Boehm et al.: Software Cost Estimation with COCOMO II, p. 502, Prentice Hall PTR, 2000.
- [3] Maxwell, K. D., Van Wassenhove, L. and Dutta, S.: Software Development Productivity of European Space, Military, and Industrial Applications, IEEE Trans. Software, Vol. 22, No. 10, pp. 706-718 (1996).
- [4] Chulani, S., Boehm, B. W., and Steece, B.: Bayesian Analysis of Empirical Software Engineering Cost Models, IEEE Trans. Software Eng, Vol. 25, No. 4, pp. 573-583 (1999).
- [5] Maxwell, K. D. and Forselius, P.: Benchmarking software Development Productivity, IEEE Software, Vol. 17, No. 1, pp. 80-88 (2000).
- [6] Tsunoda, M, Monden, A., Yadohisa, H., Kikuchi, N. and Matsumoto, K.: Productivity Analysis of Japanese Enterprise Software Development Projects, Proceedings of the MSR'06 workshop, pp. 14-17, May (2006).
- [7] 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター著作監修: ソフトウェア開発データ白書 2006、日経 BP 社、2006.