

## ソフトウェア開発工数見積もりモデルにおける熟練者判断の扱い

角田 雅照<sup>1</sup> 松本 健一<sup>2</sup>

**概要:** 数学的モデルに基づく工数見積もりでは、開発するソフトウェアの規模と、生産性により工数を見積もる。ただし、定性的または定量的に測定されていないならば、生産性に影響する要因を説明変数とすることはできない。そこでそれらの要因を説明変数とする代わりに、PM が定性的、定量的に計測されていない情報に基づき、生産性のレベルを粗い粒度（「高」、「低」など）で推測し、これを説明変数とすることを提案する。ただし、生産性レベルの推測には誤りが含まれる可能性がある。例えば、ある見積もり対象プロジェクトの生産性レベルが、実績値では「低」となるにもかかわらず、見積もり時に「高」と誤推測する可能性がある。そこで、誤推測を前提とした、生産性レベルのモデルへの与え方を提案する。

**キーワード:** 工数予測, プロジェクト管理, 生産性, 見積もり誤差, 誤差マージン

### Introducing Expert Judgment into Software Effort Estimation Models

MASATERU TSUNODA<sup>1</sup> KENICHI MATSUMOTO<sup>2</sup>

**Abstract:** A mathematical model estimates development effort based on software size and productivity. Productivity factors can be used as explanatory variables only when they are measured qualitatively or quantitatively. We propose that a project manager judge productivity level roughly (high or low), and use it as an explanatory variable, instead of using unmeasurable productivity factors as explanatory variables. However, the productivity level includes some errors. For instance, an estimation target project may be judged as high productivity level erroneously, although actual productivity level will be low. So, we propose alternative method considering misjudgment.

**Keywords:** effort prediction, project management, productivity, estimation error, error margin

#### 1. はじめに

工数見積もりはプロジェクトマネジメントの基礎となるものであり、高い見積もり精度を実現するために、これまで数学的モデルに基づく工数見積もり方法が数多く提案されてきた[1]。工数見積もりモデルは単純化すると、開発するソフトウェアの規模と、生産性（開発効率、モデルによって推定される。実績値はプロジェクト完了時に確定する）により工数を見積もっていると見なすことができる。生産性は開発種別や開発言語の種別などにより変化する。

ただし、生産性に影響する全ての要因を、説明変数としてモデルに与えることはできない。モデルに与える（説明変数とする）ためには、その要因が過去のプロジェクトにおいて、定性的または定量的に測定されている必要がある。例えば、非機能要求は生産性に影響すると考えられるが、過去プロジェクトにおいて（順序尺度などで）測定されていないならば、非機能要求に関する情報をモデルに与えるこ

とができない。顧客や開発要員に関する情報など、測定することが容易でないものも多い。

本研究では、これらの（測定されていない）生産性に影響する要因をモデルに与える代わりに、プロジェクトマネージャ（熟練者、以降 PM と表す）が生産性のレベルを粗い粒度（「高」、「低」などの順序尺度）で推測（判断）し、モデルに与えることを提案する。一般に PM は、見積もり対象プロジェクトに関して、定性的、定量的に計測されていない情報を把握している。本研究では、PM はそれらの情報に基づいて、見積もり対象プロジェクトの生産性のレベルを粗い粒度で推測できると仮定する。この生産性レベルは測定されていない要因の情報を利用しているといえる。従って提案方法により、より正確な生産性の推測（すなわちより高い見積もり精度）が期待できる。

ただし、定性的、定量的に計測されていない情報に基づいて生産性レベルを推測するため、その結果には誤りが含まれることが避けられない。すなわち、ある見積もり対象プロジェクトの生産性レベルが（プロジェクト完了時に）「低く」なるにもかかわらず、（見積もり時に）「高い」と誤って推測する可能性がある。そこで、推測に誤りが含ま

†1 近畿大学

Kinki University

†2 奈良先端大学院大学

Nara Institute of Science and Technology

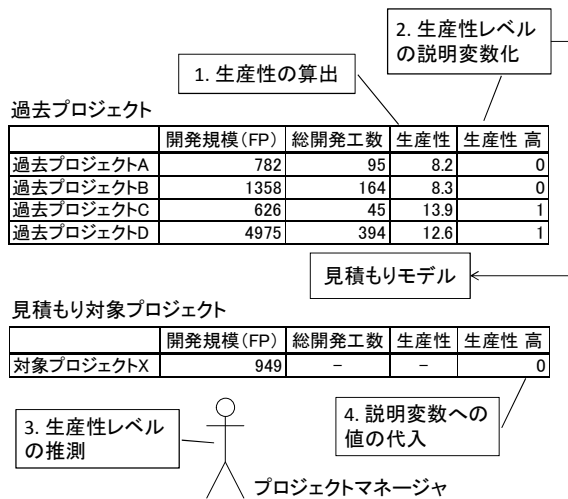


図 1 提案方法による見積もりモデルの構築手順

Figure 1 Procedure of building estimation model based on the proposed method.

れることを前提とした、生産性レベルのモデルへの与え方を提案する。評価実験では、推測の誤りの程度を変化させ、提案方法の工数見積もり精度への影響を確かめる。

## 2. 提案方法

提案する見積もりモデル構築方法について説明する。提案方法では、PM が生産性のレベルを粗い粒度で推測し、これを説明変数としてモデルに与える。本研究では、生産性レベルの推測に誤りが含まれることを考慮しない方法（単純法）と、考慮した方法（誤り考慮法）、さらに、単純法を誤り考慮法に近づけた方法（3段階単純法）を提案する。誤り考慮法以外は、誤り考慮法と性能を比較することを前提とした見積もり方法である。なお、予備分析を行ったところ、多段階（例えば5段階）で生産性レベルを与えた場合や、直接生産性の推測値を与えた場合、生産性の推測誤差の影響を大きく受けた。そこで生産性レベルは3段階までとした。提案方法によるモデルの構築手順を図1に示す。

本章では提案方法を重回帰分析に適用することを前提として説明するが、提案方法はその他の工数見積もり方法にも適用可能である。重回帰分析は工数見積もりモデルの構築方法として広く用いられている。なお、プロジェクトの特性（開発種別など）を考慮するための方法として、データの層別と説明変数に特性を含める方法があるが、本研究では後者を前提に説明する（あらかじめ層別しておけば、前者にも適用可能である）。

### 2.1 推測誤りを考慮しない方法（単純法）

単純法は、生産性レベルの推測に誤りが含まれることを考慮しない方法である。4つのステップにより見積もりモデルを構築し、対象プロジェクトの工数を見積もる。以下に各ステップの詳細について述べる。

#### ステップ1: 過去プロジェクトにおける生産性の算出

蓄積されているデータ（数学的モデルに基づく見積もりモデルを構築するためには、過去プロジェクトのデータが必要となる）から、過去の各プロジェクトの生産性を計算する。生産性は開発規模÷総開発工数で定義される。その後、過去プロジェクトの生産性の中央値を計算する。

#### ステップ2: 生産性レベルの説明変数化

ステップ1で計算した、過去の各プロジェクトの生産性とその中央値を比較し、生産性が中央値よりも大きい（効率が高い）場合は1、中央値以下（効率が低い）の場合は0とする変数を新たに作成する。この変数とその他の変数（開発規模など）を説明変数とし、見積もりモデルを構築する。

#### ステップ3: 生産性レベルの推測

PM は、見積もり対象プロジェクトに関して把握している情報（非機能要求など）に基づき、生産性が過去プロジェクトの中央値（ステップ1で算出）よりも大きい小さいか（見積もり対象プロジェクトの生産性が、過去のプロジェクトの上位50%に入るかどうか）を推測する。

#### ステップ4: 説明変数への値の代入

ステップ3の推測結果に基づき、ステップ2で作成した説明変数に値を代入する。すなわち、ステップ3で生産性が高い（上位50%に入る）と推測した場合、変数に1を代入し、そうでない場合は0を代入し、モデルを用いて見積もり工数を計算する。

### 2.2 推測誤りを考慮する方法（誤り考慮法）

単純法のステップ3において、推測に誤りがあった場合の影響について説明する。重回帰分析による見積もりモデル構築時には、対数変換を適用することが一般的であり、その場合、ステップ3では以下のような見積もりモデルが構築される（モデルを単純にするために、説明変数は開発規模と生産性レベルのみとした）。

$$y = e^{b+c x_2} x_1^a \quad (1)$$

$y$  は総開発工数、 $x_1$  は開発規模、 $x_2$  は生産性レベル、 $a$ 、 $b$ 、 $c$  ( $c < 0$ ) は定数、 $e$  は自然対数の底である。 $a$  は規模の経済（不経済）を表す定数であり、おおむね1に近い値をとる。 $e^{b+c x_2}$  を  $d$  とすると、式1は以下のように表される。

$$y = d x_1^a \quad (2)$$

$x_2$ （生産性レベル）の値が1（生産性が高い）の場合、( $c < 0$ より) $d$ は小さくなり（その結果 $y$ が小さくなる）、値が0の場合は大きくなる。例えば、前者の場合は $d = 20$ 、後者の場合は $d = 30$ などとなる。生産性レベルを説明変数に含まない、従来の見積もりモデル（式2と同様の形となる）の $d$ は、両者の中間の値となる。ここでは、例えば $d = 25$ であるとする。

（プロジェクト完了時に）見積もり対象プロジェクトの生産性が低くなる（ $d = 30$ が適している）にも関わらず、（見積もり時に）生産性レベルを高いと誤って推測する（ $d$

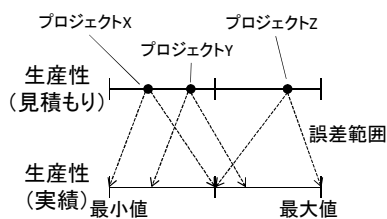


図 2 生産性の推測誤りの範囲

Figure 2 Error range of speculated productivity.

見積もり対象プロジェクト

	開発規模 (FP)	5段階生産性	生産性 高
過去プロジェクトV	2425	1	0
過去プロジェクトW	861	2	-
過去プロジェクトX	904	3	-
過去プロジェクトY	1320	4	-
過去プロジェクトZ	598	5	1

生産性レベルを含まないモデルにより見積もる

図 3 誤り考慮法における生産性レベルの扱い

Figure 3 Productivity level on the error consideration model.

= 20 を当てはめると、見積もり精度が大きく低下する。この場合、従来のモデル ( $d = 25$ ) を用いたほうが、精度の低下が小さくなる。

本研究では、生産性レベルの推測の誤りは多くの場合、一定の範囲内に収まると仮定する。例えば見積もり対象プロジェクトの生産性が、過去のプロジェクトの下位 20%以下になると推測した場合、実際の生産性はほとんどの場合、下位 50%以下に収まると仮定する。逆に生産性が下位 30%~50%程度になると推測した場合、実際の生産性が下位 50%以上（上位 50%以内）になることは十分起こりうると仮定する（図 2）。

提案する誤り考慮法はこの仮定に基づいており、従来のモデルよりも見積もり精度が低下することを避けることを狙いとする。具体的には、生産性が下位 20%以下、もしくは上位 20%以上になると推測した場合、生産性レベルを説明変数に含んだモデルを利用し、それ以外の場合には従来の（生産性レベルを含まない）モデルを用いる。直感的には、生産性を 5 段階（数字が小さいほど生産性が低い）で評価し、1 の場合は生産性レベルが低い、5 の場合は生産性レベルが高い、それ以外の場合は生産性レベルが不明確であると判断するイメージである（図 3）。

これにより、前者のケースでは実際の生産性はほぼ下位 50%以下、もしくは上位 50%以内に収まるため、生産性レベルを含むモデルを用いても、少なくとも（従来法と比較して）精度の低下は起こらない。それ以外の場合は従来法を用いるため、同様に少なくとも精度が低下することはない。以降に誤り考慮法の詳細な手順を説明する。

**ステップ 1: 過去プロジェクトにおける生産性の算出**

蓄積されているデータを用いて、過去の各プロジェクトの生産性を計算する。その後、これらの生産性の中央値と、

20 パーセンタイル値、80 パーセンタイル値を計算する。

**ステップ 2: 生産性レベルの説明変数化**

単純法のステップ 2 と同様にモデルを構築する。すなわち、各プロジェクトの生産性と中央値を比較し、中央値より大きい場合は 1、それ以外では 0 とする変数を作成してモデルを構築する。また、生産性レベルを含まない見積もりモデル（従来法に基づくモデル）を構築する。

**ステップ 3: 生産性レベルの推測**

PM は、見積もり対象プロジェクトに関して把握している情報に基づき、生産性が過去プロジェクトの 20 パーセンタイル値（ステップ 1 で算出）よりも小さいか（過去のプロジェクトの下位 20%に入るか）、80 パーセンタイル値よりも大きい（過去の上位 20%に入るか）を推測する。

**ステップ 4: 説明変数への値の代入**

ステップ 3 で生産性が高い、または低いと推測した場合、生産性レベルを説明変数に含むモデル（ステップ 2 で作成）を用いて、工数を見積もる（生産性レベルが高いと推測した場合は変数に 1 を代入し、低いと推測した場合は 0 を代入する）。生産性レベルが不明確であると判断した場合、生産性レベルを含まないモデル（従来法によるモデル。ステップ 2 で作成）を用いて見積もる。

**2.3 3 段階単純法**

誤り考慮法は、生産性の低い場合、高い場合、それ以外の場合の 3 種類に対応した見積もりを行っている。それに対し単純法は、生産性の低い場合と高い場合の 2 種類に対応した見積もりを行っている。3 段階単純法は、単純法を誤り考慮法に近づけた方法である。以下に 3 段階単純法の詳細な手順を述べる。

**ステップ 1: 過去プロジェクトにおける生産性の算出**

誤り考慮法のステップ 1 と同様に、過去の各プロジェクトの生産性を計算するとともに、それらの中央値、20 パーセンタイル値、80 パーセンタイル値を計算する。

**ステップ 2: 生産性レベルの説明変数化**

各プロジェクトの生産性を 20 パーセンタイル以下、80 パーセンタイル以上、それ以外の 3 段階の順序尺度とする。次に、これをダミー変数化する（目的変数との関係が線形とは限らないため）。具体的には、各プロジェクトの生産性と 20 パーセンタイル値を比較し、20 パーセンタイル値より小さい場合は 1、それ以外では 0 とする変数を作成する（低生産性を表す）。さらに、各プロジェクトの生産性と 80 パーセンタイル値を比較し、80 パーセンタイル値より大きい場合は 1、それ以外では 0 とする変数を作成する（高生産性を表す）。これら 2 つのダミー変数を説明変数に含んだモデルを構築する。

**ステップ 3: 生産性レベルの推測**

PM は見積もり対象プロジェクトに関して把握している情報に基づき、生産性が過去プロジェクトの 20 パーセンタイル値（ステップ 1 で算出）よりも小さいか、80 パーセン

タイトル値より大きいか、もしくはそれら以外かを推測する。

#### ステップ 4: 説明変数への値の代入

ステップ 3 において生産性が 20 パーセント値よりも小さいと推測した場合、低生産性を表す変数に 1、高生産性を表す変数に 0 を代入して工数を見積もる。80 パーセント値よりも大きいと推測した場合、それぞれに 0 と 1 を代入して見積もる。それら以外の場合は両方に 0 を代入して見積もる。

なお、提案方法の閾値 (20, 80 パーセント値) は、箱ひげ図の箱が 25, 75 パーセント値を表していることを参考に決定した。ただし、25, 75 パーセント値だとデータ全体の 50% に該当し、対象が少し広すぎるため、提案方法の閾値はそれより小さめに設定した。

### 3. 評価実験

#### 3.1 データセット

ISBSG データは、International Software Benchmark Standard Group (ISBSG) が収集した 20 カ国のソフトウェア開発企業のデータである[5]。本研究ではリリース 9 のデータを用い、Lokan ら[9]の条件(データの品質が A または B, FP が IFPUG 法で計測されているなど)に従ってデータを抽出した。さらに欠損値(変数に値が記録されていないこと)が含まれているプロジェクトを除外した、593 件を実験に用いた。また、説明変数についても Lokan らと同じ 4 つの変数(未調整 FP, 開発種別, 言語種別, プラットフォーム)を用いた。開発種別, 言語種別, プラットフォームはカテゴリ変数のため、ダミー変数化して用いた。

Kitchenham データは、Kitchenham が 2002 年に文献[8]の中で公開した、あるソフトウェア開発企業のデータである。データセットに含まれる 145 件のプロジェクトのうち、欠損値が含まれるプロジェクトを除いた 135 件を用いた。説明変数は、工数見積もりに用いるには不適切な変数(プロジェクト管理者による工数の見積り値など)と開発期間を除いた 2 つの変数(調整済み FP, 開発種別)である。開発種別はカテゴリ変数のため、ダミー変数化して用いた。

Desharnais データ[2]は、Desharnais によって収集されたカナダのソフトウェア開発企業における 80 年代のデータセットである[4]。データセットに含まれる 81 件のプロジェクトのうち、欠損値が含まれるプロジェクトを除いた 77 件を用いた。説明変数は、開発年度と調整済み FP (ファンクションポイント), 開発期間を除いた 5 つの変数(開発チームの経験年数, PM の経験年数, 未調整 FP, 調整係数, 開発言語)である。このうち開発言語はカテゴリ変数のため、ダミー変数化して用いた。

#### 3.2 評価指標

工数見積もり精度の評価指標として、BRE (Balanced Relative Error)[10]の平均値を用いた。値が小さいほど工数見積もりの精度が高いことを示す。提案方法の見積もり精

度を評価するために、従来法(生産性レベルを使わないモデル)との評価指標の差分を求めた。差分の値が正の場合、従来法よりも精度が改善したことを示し、値が負の場合、精度が悪化したことを示す。

#### 3.3 実験手順

実験は 5-fold cross validation に基づいて行った(後述の手順参照)。試行回数が少ないと統計的に見積もり精度の差を確かめることが難しくなるため、5-fold cross validation を 4 回繰り返した(合計 20 回評価される)。モデル構築時には AIC に基づく変数選択を行った。

推測の誤りを生じさせるため、テストデータの実測値に  $n\%$  の誤差を発生させ、その生産性に基づいて生産性レベルを決定した。生産性の実測値を  $x$ , 推測値を  $\hat{x}$  とするとき、誤差  $n$  は以下の式により計算される。

$$n = \begin{cases} \frac{\hat{x} - x}{x} \times 100, \hat{x} - x \geq 0 \\ \frac{\hat{x} - x}{\hat{x}} \times 100, \hat{x} - x < 0 \end{cases} \quad (3)$$

直感的には、実際の実測値と推測した生産性レベルとの間に  $n\%$  の誤差があるイメージである。一般に計測誤差は正規分布に従うと仮定するため、実験においても生産性の誤差は正規分布に従うと仮定した。平均は 0 とし、標準偏差を変化させて(初期値 0, 増分 0.1, 最大値 2)正規分布に従う  $n$  をランダムに発生させ、式 3 から誤差を含む生産性を逆算した。

なお、生産性レベルの推測時には、具体的な推測値を決定する必要はなく、2 章で述べた程度の推測でよい。上記は 2.2 節で述べた、生産性レベルの推測の誤りが一定の範囲内に収まることをシミュレートするための方法である。

実験は各見積もり方法に対し、各データセットを用いて以下の手順で行った。

1. データセットをランダムに 5 等分する。
2. 5 つのうち 1 つをテストデータ、残りをラーニングデータとし、後者を用いてモデルを構築する。
3. テストデータにおいて、誤差を含んだ生産性を計算し、それに基づき生産性レベルを決定する。
4. 構築したモデルを用いて、テストデータのプロジェクトの工数を見積もり、評価指標を計算する。
5. 誤差の標準偏差が 2 を超えていない場合、標準偏差を 0.1 増やし、手順 2~4 を繰り返す。
6. テストデータを変更し、手順 2~5 を 5 回繰り返す。
7. 手順 1~5 を 4 回繰り返す。

### 4. 実験結果及び考察

#### 4.1 従来法との比較

従来法と提案方法との見積もり精度の差分と、生産性の誤差との関係を図 4 に示す。横軸の生産性誤差は、生産性

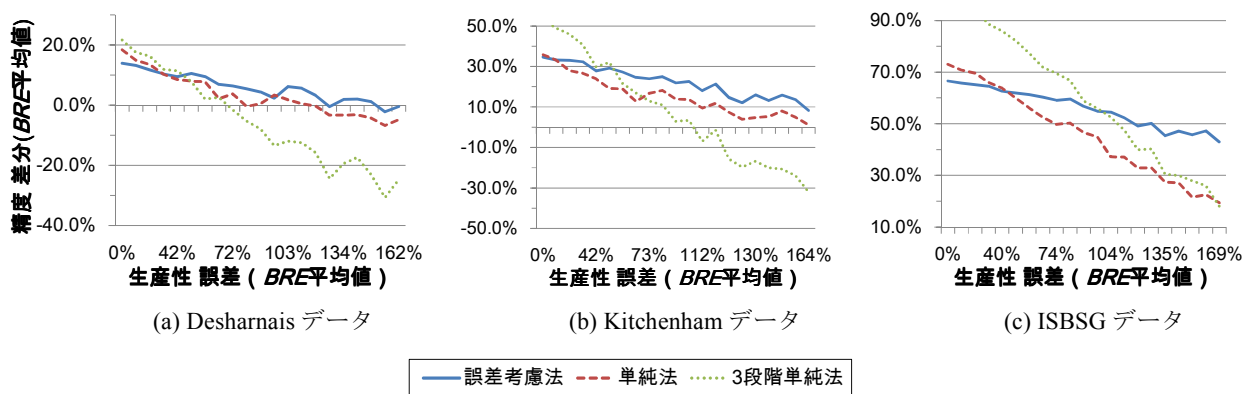


図 4 見積もり精度の差分と生産性の誤差との関係

Figure 4 Relationships between difference of estimation accuracy and error of productivity.

の BRE 平均値を算出したものである。

単純法と 3 段階単純法では、Desharnais データと Kitchenham データにおいて生産性の誤差が大きい場合、BRE 平均値の差分が 0 を下回った（従来法よりも精度が悪化した）。これに対し、誤り考慮法の場合、実験において変化した誤差の範囲（誤差の標準偏差が 2 以下）では、どのデータでも BRE 平均値の差分が 0 を下回ることがほとんどなく、下回った場合もごくわずかな程度であった。

実験結果より、誤り考慮法が 3 つの提案方法の中で最も生産性の誤差に対しロバストであり、生産性の誤差が大きい場合でも、従来法よりも精度が大きく低下する可能性は低いといえる。生産性レベルの推測の誤りの程度（生産性の誤差）が不明確な場合、3 つの提案方法と従来法では、誤り考慮法が最も適切（多くの場合において、最も高い精度を期待できる）といえる。

#### 4.2 提案方法間の比較

単純法と誤り考慮法を比較すると、Desharnais データでは誤差が 22%以内の場合、Kitchenham データでは誤差が 0%の場合、ISBSG データでは誤差が 33%以内の場合、前者が若干改善の程度が大きかった。それ以外の場合では後者のほうが改善の程度が大きかった。

3 段階単純法と誤り考慮法を比較すると、どのデータセットにおいても、少なくとも生産性の誤差が 37%以内の場合には前者の改善の程度が後者を上回っていた。ただし、生産性の誤差が 91%を上回る場合、どのデータセットにおいても後者の改善の程度が上回っていた。よって、もし生産性レベルの推測誤りの程度（生産性の誤差）が把握できるならば、誤りが小さい場合は 3 段階単純法を、大きい場合は誤り考慮法を用いるとよいといえる。

#### 4.3 考察

提案方法において生産性レベルを推測するには、生産性の具体的な推測値を決定する必要はない。ただし、生産性レベルの推測誤差と熟練者（人手）による工数見積もりの誤差は同程度となる可能性がある。すなわち 3.3 節において、生産性レベルの推測誤差は生産性の誤差に従うと仮定

したが、例えば生産性の BRE 平均値が 100%の場合、熟練者による工数見積もりの誤差も同様に BRE 平均値が 100%となる可能性がある（生産性の分子（開発規模）は定数と見なせるため、見積もり工数の BRE が  $n\%$  の時、見積もり工数を生産性に換算して BRE を求めると  $n\%$  となる）。

この仮定に従うと、生産性の BRE 平均値が 100%で、提案方法による工数見積もりの BRE 平均値が 120%の場合、熟練者による工数見積もりの精度（BRE 平均値が 100%）のほうが高くなるため、提案方法を用いるべきではない。そこで、従来法に加えて生産性の推測誤差（すなわち熟練者による工数見積もりの誤差）を提案方法の比較対象とした。具体的には、従来法と生産性の BRE 平均値を比較し、値が小さい方をベースラインとし、ベースラインと提案法との差分を比較した。

結果を図 5 に示す。差分が負の場合、提案方法はベースラインよりも精度が悪化したことを示す。差分が 0 より大きい（提案方法がベースラインよりも精度が高い）時、ほとんどの場合において誤り考慮法の精度が最も高かった。図には示していないが、差分が 0 より小さい場合、熟練者による工数見積もりのほうが従来法よりも精度が高かった。

よって、生産性レベルの推測誤差と、熟練者による工数見積もりの誤差が同程度であると仮定すると、熟練者による工数見積もりの精度が低い場合は誤り考慮法を適用し、高い場合は熟練者による工数見積もりの結果を採用すべきであるといえる。さらにこの結果より、もし生産性レベルの推測が可能であるという仮定が正しくないとしても、熟練者による工数見積もりの精度が低い場合、熟練者による工数見積もりの結果に基づき生産性レベルを決定すると、誤り考慮法により見積もり精度を改善できるといえる。

工数見積もりのプロセスが未成熟な組織では、熟練者による見積もり誤差が大きくなることは十分ありうると考える。特に Desharnais データでは熟練者による見積もりの BRE 平均値が 42%以上の場合、提案方法が有効に働いており、現実的にありえない誤差ではないと考える。

なお、熟練者による工数見積もりを実施していれば、過

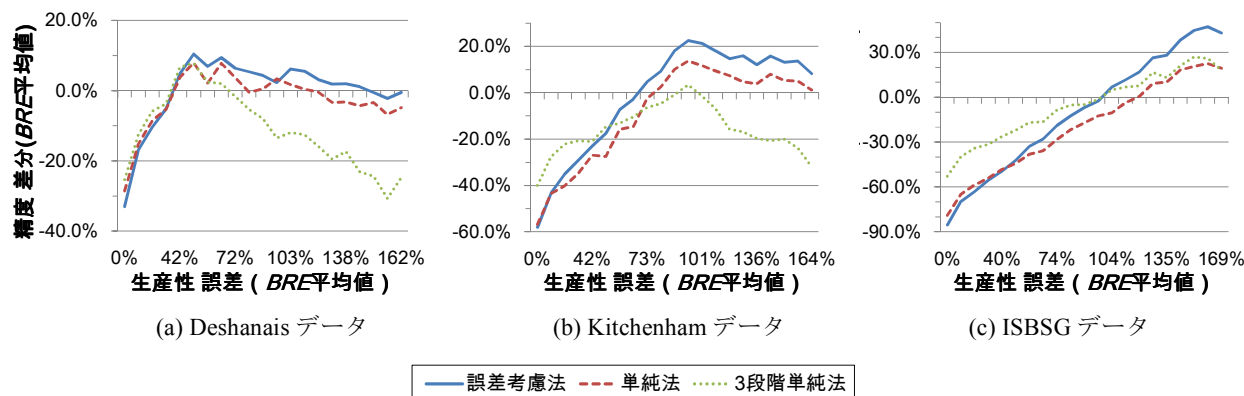


図 5 見積もり精度の差分と生産性の誤差との関係 (熟練者による見積もりを考慮した場合)

Figure 5 Relationships between difference of estimation accuracy and error of productivity considering effort estimation by experts.

去の見積もり結果に基づき精度を計算することが可能であり、生産性レベルの推測誤差よりも把握が容易であると考えられる。

## 5. 関連研究

工数見積もりモデルによっては、熟練者の判断に基づく情報（開発要員の技術力など）を用いている場合がある[1][3]。ただし、例えばPMとスキルの高い開発要員との関係に軋轢がある場合、生産性に大きく影響すると考えられるが、その関係を工数見積もりモデルに機械的に取り入れる方法は、これまで提案されていない[6]。提案方法は、そのような情報をモデルに反映させる方法であると見なすことができる。さらに、熟練者の判断に基づく説明変数には誤差が含まれることが避けられないが、従来法[1][3]は誤差を前提としておらず、その影響も不明確である。

提案方法と同様に、説明変数に誤りが含まれることに着目した研究が存在する。Kitchenhamら[7]は見積もり工数の誤差が生じる原因を複数示しており、説明変数の誤りを原因の1つとして挙げている。また、それらの原因を考慮して、誤差を小さくするためのガイドラインを示している。ただし、実データを用いた定量的な分析は行われておらず、さらに、提案方法のような見積もり方法は示されていない。

## 6. おわりに

本研究では、測定されていない要因をモデルに与える代わりに、プロジェクトマネージャが生産性のレベルを粗い粒度で推測し、工数見積もりモデルの説明変数とする方法を提案した。生産性レベルの推測に誤りが含まれることを考慮しない方法（単純法）、考慮した方法（誤り考慮法）、単純法を誤り考慮法に近づけた方法（3段階単純法）の3つの手法を提案し、実験において生産性レベルの推測誤りの程度を変化させ、従来法（生産性レベルを含まない見積もり方法）と精度を比較した。

今後の課題は、生産性レベルがある程度の正しさで推測できるという仮定の正しさを、被験者を用いて確かめるこ

と、生産性レベルの推測誤差と、熟練者による工数見積もりの誤差との関係を確かめることである。また、実験では生産性レベルの推測にバイアスがないことを前提としたが、バイアスが存在する場合の影響を分析することも今後の課題の一つである。

**謝辞** 本研究の一部は、文部科学省科学研究補助費（基盤C：課題番号25330090）による助成を受けた。

## 参考文献

- [1] Boehm, B.: *Software Engineering Economics*, Prentice Hall (1981).
- [2] Boetticher, G., Menzies, T. and Ostrand, T.: *PROMISE Repository of empirical software engineering data*, West Virginia University, Department of Computer Science (2007).
- [3] Briand, L., Emam, K., and Bomarius, F.: COBRA: a hybrid method for software cost estimation, benchmarking, and risk assessment, *Proc. international conference on Software engineering (ICSE)*, pp. 390-399 (1998).
- [4] Desharnais, J.: *Analyse Statistique de la Productivite des Projets Informatique a Partie de la Technique des Point des Fonction*, Master Thesis, University of Montreal (1989).
- [5] International Software Benchmarking Standards Group (ISBSG), *ISBSG Estimating: Benchmarking and research suite*, ISBSG (2004).
- [6] Jørgensen, M., Boehm, B. and Rifkin, S.: Software Development Effort Estimation: Formal Models or Expert Judgment? *IEEE Software*, vol. 26 no. 2, pp. 14-19 (2009).
- [7] Kitchenham, B., and Linkman, S.: Estimates, Uncertainty, and Risk, *IEEE Software*, vol. 14 no. 3, pp. 69-74 (1997).
- [8] Kitchenham, B., Pfleeger, S., McColl, B. and Eagan, S.: An Empirical Study of Maintenance and Development Estimation Accuracy, *Journal of Systems and Software*, vol. 64, no. 1, pp. 57-77 (2004).
- [9] Lokan, C. and Mendes, E.: Cross-company and single-company effort models using the ISBSG Database: a further replicated study, In *Proc. the International Symposium on Empirical Software Engineering (ISESE)*, pp. 75-84 (2006).
- [10] Miyazaki, Y., Terakado, M., Ozaki, K. and Nozaki, H.: Robust Regression for Developing Software Estimation Models, *Journal of Systems and Software*, vol. 27, no. 1, pp. 3-16 (1994).