

開発要員数とその誤差を考慮した工数見積もり方法

角田 雅照^{1,a)} 門田 暁人² 松本 健一²

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要因を説明変数として工数を見積もる。いくつかの先行研究では、開発要員数が工数見積もりモデルの説明変数として用いられている。要員数はプロジェクトの生産性との関連が強いことが指摘されており、要員数を説明変数として用いることにより、モデルの見積もり精度向上が期待できる。ただし、工数見積もり時における要員数は見積もり値であり、実績の要員数との誤差が生じることにより見積もり精度が低下する可能性がある。そこで本稿では要員数を用いた新たな見積もり方法を提案する。提案方法では、要員数が過去プロジェクト平均と比較して非常に小さいと見積もられた場合、要員数が小さいことを前提とするモデルを用いる。逆に、要員数が非常に大きいと見積もられた場合、要員数が大きいことを前提とするモデルを用いる。これらに当てはまらない場合、要員数を説明変数に含まないモデルを用いて工数を見積もる。評価実験において、要員数の誤差を実験的に変化させ、提案方法の見積もり精度を評価した。その結果、提案方法は要員数の誤差に対してロバストであり、特に要員数の誤差が不明な場合や誤差が大きい場合に有効に働くことがわかった。

1. はじめに

近年のソフトウェア開発プロジェクトでは、大規模かつ複雑なソフトウェアを短納期で開発することが多い。このような状況においてプロジェクトの失敗（コスト超過や納期遅延）を避けるためには、コストやスケジュールの管理といったプロジェクトマネジメントが重要となる。工数見積もりはその基礎となるものであり、プロジェクトを成功に導くために欠くことのできないものの一つとなっている。工数見積もりとは、プロジェクトの初期に総開発工数を見積もることである。

高い工数見積もり精度（実績の総開発工数と見積もりの総開発工数の差が小さいこと）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][15][16]。数学的モデルに基づく工数見積もりでは、複数の過去プロジェクトにおいて記録されたデータ（開発規模や総開発工数の実績値など）を用いて見積もりモデルを構築する。そして構築されたモデルに対し、見積もり対象のプロジェクトの既知の情報（開発規模や開発言語の種別などの生産性要因）を説明変数として与えることにより、総開発工数（目的変数）を見積もる。

いくつかの先行研究[5][11]では、開発要員数を工数見積もりモデルの説明変数として用いている。要員数はプロジェクトの生産性との関連が強いことが指摘されており[14][18]、説明変数として用いることにより、モデルの見積もり精度が向上することが期待できる。ただし、工数見積もり時（見積もり工数が決定する前）に要員数を決定することは容易ではない。一般に要員数は見積もり工数とプロジェクトの工期に基づいて決定される。よって、要員数を

見積もりモデルの説明変数として用いるためには、見積もり工数を用いずに要員数を決定する必要がある。さらに、要員数のある程度高い精度で見積もることができるとしても、実際の要員数はプロジェクトの状況によって変化しうるため、結果的に精度が低下しうる。例えば、何らかの理由でプロジェクトの進捗に遅れが生じた場合、ソフトウェアの納期遅延を避けるために、開発要員の追加投入が行われる。この場合、要員数が当初の予定よりも増加することになる。

見積もりの要員数（工数見積もり時に決定する）と実績の要員数（プロジェクト完了時に測定する）の差が大きい場合、要員数を工数見積もりモデルの説明変数に用いると、モデルの見積もり精度が低下する。これは、一般に見積もりの要員数は計測されず、実績の要員数が説明変数として用いられるためである。例えば、見積もりの要員数が5人（過去プロジェクト平均と比較して少ない）で、実績の要員数が20人（過去平均と比較して多い）の場合、要員数を説明変数に含んだモデルを用いると、工数見積もりの精度は低下すると考えられる。

要員数の見積もりと実績との差が、工数見積もりの精度に悪影響を与えることを抑えるために、本稿では要員数を用いた新たな見積もり方法を提案する。提案方法を大まかに説明する。要員数が（過去プロジェクト平均と比較して）非常に小さいと見積もられた場合、要員数が小さいことを前提とするモデルを用いて工数を見積もる。逆に、要員数が非常に大きいと見積もられた場合、要員数が大きいことを前提とするモデルを用いる。これらに当てはまらない（見積もりの要員数が非常に小さくない、または非常に大きくない）場合、要員数を説明変数に含まないモデルを用いて工数を見積もる。

提案方法では、「要員数が（過去平均と比較して）非常に小さいと見積もられた場合、実際の要員数が（過去平均と比較して）大きくなることはない」と想定する。また、

1 近畿大学
Kinki University, Japan
2 奈良先端大学院大学
Nara Institute of Science and Technology, Japan
a) tsunoda@info.kindai.ac.jp

「要員数が（非常に小さいではなく）小さいと見積もられた場合、実際的要員数が大きくなることはある」と想定する。同様に、「要員数が非常に大きいと見積もられた場合、実際的要員数が小さくなることはない」と想定する。

以降、2章において要員数を用いた見積もり方法について説明し、3章では見積もり方法の評価実験について述べる。4章で実験結果を示し、5章で関連研究について述べ、最後に6章でまとめを述べる。

2. 提案方法

提案する工数見積もり方法について説明する。提案方法では、プロジェクトマネージャが開発要員数を見積もり、その結果をモデルの説明変数に代入することにより工数を見積もる。本稿では、開発要員数に見積もり誤差が含まれることを考慮した方法（誤差考慮法）を提案するとともに、誤差を考慮しない方法（単純法）について説明する。提案方法による見積もりの手順を図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、その他の工数見積もり方法にも適用可能である。

2.1 誤差考慮法

誤差考慮法では、開発要員数を5段階（数字が小さいほど要員数が小さい）で見積もる。以降、これを要員数レベルと呼ぶ。ここで開発要員数とは、プロジェクト存続中に従事している要員数の平均値を指す。誤差考慮法では、要員数レベルが1の場合は実績的要員数も小さくなる可能性が高く、5の場合は実績的要員数も大きくなる可能性が高いと想定する。これらに該当する場合、（順序尺度化した）要員数を説明変数に含むモデルを用いて工数を見積もる。これら以外の場合は要員数が不明確であると判断し、要員数を含まないモデルを用いて工数を見積もる。提案方法では、要員数を非常に小さい（レベル1）と見積もった場合（図2のプロジェクトV）、要員数が中央値よりも大きくなる確率は低いと想定する。同様に、要員数を非常に大きい（レベル5）と見積もった場合（図2のプロジェクトZ）、要員数が中央値よりも小さくなる確率は低いと想定する。以降では誤差考慮法の手順について詳細に説明する。

ステップ1: 要員数の閾値決定

蓄積されているデータ（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を事前に蓄積しておく必要がある）を用いて、過去のプロジェクトにおける要員数の20パーセンタイル値（要員数のレベル1の閾値）、中央値、80パーセンタイル値（要員数のレベル5の閾値）を計算する。

ステップ2: 要員数の順序尺度化

過去の各プロジェクトにおいて、要員数とその中央値を比較し、要員数が中央値よりも大きい場合は1、中央値以下の場合は0とする変数を新たに作成する（図3参照。図では「要員数大」の変数）。この変数とその他の変数（開

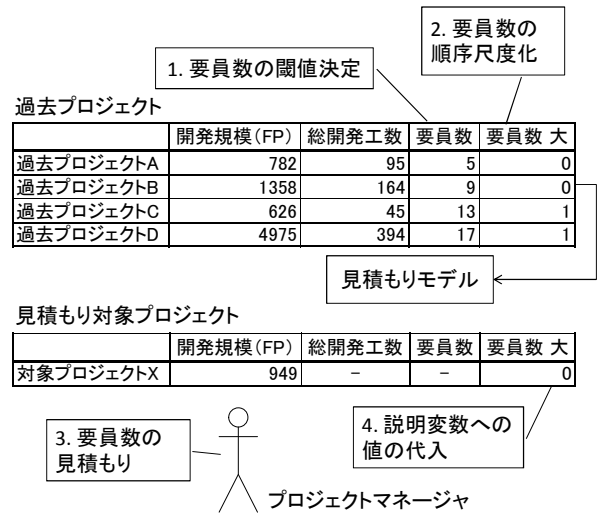


図1 誤差考慮法による見積もり手順

Figure 1 Estimation procedure based on the Error-Consideration Method.

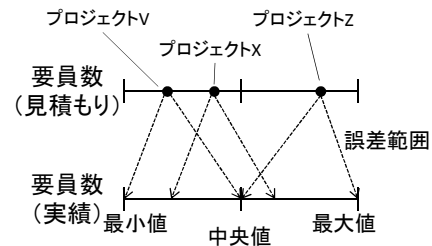


図2 要員数の見積もり誤差の範囲

Figure 2 Error range of estimated team size.

発規模など)を説明変数として、工数見積もりモデルを構築する。また、要員数を含まないモデル（従来法に基づくモデル）を構築する。

ステップ3: 要員数の見積もり

プロジェクトマネージャは、見積もり対象プロジェクトの要員数が、過去プロジェクトの20パーセンタイル値（ステップ1において算出）よりも小さいか（要員数レベル1か）、80パーセンタイル値より大きいか（要員数レベル5か）、もしくはそれ以外か（要員数レベル2から4か）を見積もる。

ステップ4: 説明変数への値の代入

ステップ3において、要員数が過去プロジェクトの20パーセンタイル値よりも小さい、または80パーセンタイル値より大きいと見積もった場合、順序尺度化した要員数を説明変数に含むモデル（ステップ2で構築）を用いて、工数を見積もる。前者の場合は順序尺度化した要員数に0を代入し、後者の場合は1を代入する。要員数が20から80パーセンタイルに含まれると見積もられた場合、要員数を含まないモデル（従来法に基づくモデル、ステップ2で作成）を用いて見積もる（図3参照）。

過去プロジェクト(ステップ2)

	要員数	要員数 レベル	要員数 大
過去プロジェクト A	4	1	0
過去プロジェクト B	7	2	0
過去プロジェクト C	10	3	0
過去プロジェクト D	11	3	1
過去プロジェクト E	14	4	1
過去プロジェクト F	25	5	1

見積もり対象プロジェクト(ステップ4)

	要員数	要員数 レベル	要員数 大
見積プロジェクト V	5	1	0
見積プロジェクト W	8	2	-
見積プロジェクト X	9	3	-
見積プロジェクト Y	13	4	-
見積プロジェクト Z	24	5	1

要員数レベルを含まないモデルで見積もる ←

図 3 誤差考慮法の詳細

Figure 3 Details of the Error-Consideration method

2.2 単純法

誤差考慮法と比較するために、本稿では4種類の単純法(要員数の誤差を考慮しない方法)を適用し、見積もり精度を比較する。各手法の詳細を以下に述べる。

直接利用法

1. 実績の要員数を説明変数として含む工数見積もりモデルを構築する。
2. プロジェクトマネージャは、見積もり対象プロジェクトの要員数を見積もり、その結果を1で作成したモデルに代入し、工数を見積もる。

5段階法

1. 過去の各プロジェクトにおいて、実績の開発要員数に基づいた、5段階の順序尺度変数を新たに作成する。各段階の閾値は実績要員数の20, 40, 60, 80パーセント値である。
2. 作成した順序尺度変数を説明変数に含んだ、工数見積もりモデルを作成する。
3. プロジェクトマネージャは、見積もり対象プロジェクトの要員数を5段階で見積もり、その結果を2で作成したモデルに代入して工数を見積もる。

2段階法

1. 過去の各プロジェクトにおいて、実績の開発要員数に基づいた、2段階の順序尺度変数を新たに作成する。閾値は実績要員数の中央値である。
2. 作成した順序尺度変数を説明変数に含んだ、工数見積もりモデルを作成する。
3. プロジェクトマネージャは、見積もり対象プロジェクトの要員数を2段階で見積もり、その結果を2で作成したモデルに代入して工数を見積もる。

3段階法

1. 過去の各プロジェクトにおいて、実績の開発要員数に基づいた、3段階の順序尺度変数を新たに作成する。各段階の閾値は実績要員数の20, 80パーセント値である。
2. 作成した順序尺度変数を2つのダミー変数に変換し、それらを説明変数に含んだ工数見積もりモデルを作成する。
3. プロジェクトマネージャは、見積もり対象プロジェクトの要員数を3段階で見積もり、その結果を2で作成したモデルのダミー変数に代入して工数を見積もる。

直接利用法を除き、各手法に対して要員数に基づく層別(ある変数の値に基づいて、データセットをサブセットに分割し、各サブセット上でモデルを作成すること)を適用することは可能である。層別はダミー変数よりも有効に働く場合があるが[18]、直接利用法との比較の条件を同一にするため、本稿では層別を適用しない。

3. 評価実験

誤差考慮法と単純法の性能を評価するために、実プロジェクトのデータセットを利用して見積もりモデルを構築した。具体的には、開発要員数の誤差を実験的に変化させ、工数見積もり精度が従来法(要員数を説明変数に含まないモデル)と比較して、どの程度改善されるのかを確かめた。

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[4]。本研究ではリリース9のデータを用いており、1989年から2004年までの3026件のプロジェクトについて、99種類の変数が記録されている。ただし、ISBSG データにはデータの信頼性が低いプロジェクトが含まれており、これらを分析から除外する必要がある。そこでLokanら[10]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、201件を実験に用いた。開発要員数として平均要員数を採用し、説明変数として、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)と平均要員数を用いた。

3.2 評価指標

工数見積もり精度の評価指標として、AE (Absolute Error), MRE (Magnitude of Relative Error)[3], MER (Magnitude of Error Relative to the estimate)[8], BRE (Balanced Relative Error)[12]の4つの指標の平均値を用いた。工数の実測値をx, 見積もり値を \hat{x} とするとき、それぞれの指標は以下の式により求められる。

$$AE = |\hat{x} - x| \quad (1)$$

$$MRE = \frac{|\hat{x} - x|}{x} \quad (2)$$

$$MER = \frac{|\hat{x} - x|}{\hat{x}} \quad (3)$$

$$BRE = \begin{cases} \frac{|\hat{x} - x|}{x}, \hat{x} - x \geq 0 \\ \frac{|\hat{x} - x|}{\hat{x}}, \hat{x} - x < 0 \end{cases} \quad (4)$$

各評価指標は、値が小さいほど工数見積りの精度が高いことを示す。直感的には *MRE* は実測値との相対誤差、*MER* は見積り値との相対誤差であるといえる。ただし、*MRE* と *MER* は過大見積りと過小見積りに対し、アンバランスな評価になるという問題がある[2][9]。そこで本稿では、*MRE*、*MER* に加え、過大見積りと過小見積りをバランスよく評価する指標[13]である *BRE* の平均値を特に重視してモデルの精度を評価した。

各見積り方法の精度を評価するために、従来法をベースラインとし、評価指標ごとにベースラインとの差分を求めた。差分が正の場合、従来法よりも精度が改善したことを示し、負の場合、精度が悪化したことを示す。従来法との差の検定にはノンパラメトリックな手法である Wilcoxon の符号付順位和検定を用い、有意水準は5%とした。

3.3 実験手順

要員数の見積りに誤差を生じさせるため、テストデータの要員数に $n\%$ の誤差を発生させ、誤差を含む要員数に基づいてテストデータの要員数レベルを決定した。要員数の実測値を x 、見積り値を \hat{x} とするとき、誤差 n は以下の式により計算される。

$$n = \begin{cases} \frac{\hat{x} - x}{x} \times 100, \hat{x} - x \geq 0 \\ \frac{\hat{x} - x}{\hat{x}} \times 100, \hat{x} - x < 0 \end{cases} \quad (5)$$

式5は実際の要員数と見積もった要員数との間に $n\%$ の誤差があることを示す。一般に計測誤差は正規分布に従うと仮定するため、実験においても要員数の誤差は正規分布に従うとした。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5から誤差を含む(見積り)の要員数を逆算した。

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。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、ダミー変数化した。工数とFPに対しては対数変換を適用した。

各見積り方法に対して、以下の手順で実験を行った。

1. データセットをランダムに5等分する。
2. 5等分したうちの1つをテストデータ、残りをラーニングデータとする。
3. ラーニングデータを用いて見積りモデルを構築する。
4. テストデータにおいて、誤差を含んだ要員数を計算し、それに基づき(見積りの)要員数レベルを決定する。
5. テストデータに含まれるプロジェクトの工数を未知と見なし、構築したモデルを用いて見積もる。
6. 工数見積りの結果に基づき、各評価指標を計算する。
7. 要員数の誤差の標準偏差が2を超えていない場合、標準偏差を0.1増やし、手順4~6を繰り返す。
8. 手順2~7を5回繰り返す。
9. 手順1~8を4回繰り返す。

4. 実験結果

図4から図7は、*AE* 平均値(*MAE*)、*MRE* 平均値(*MMRE*)、*MER* 平均値(*MMER*)、*BRE* 平均値(*MBRE*)それぞれの、ベースラインと各見積り方法との差分を、見積り方法ごとに示している。以降に見積り方法ごとに結果を述べる。

4.1 5段階法

要員数誤差が小さい場合、*MAE*、*MMRE*、*MBRE* の差分がほとんどの場合で最も大きく、*MMER* の差分も2番目に大きかった。また、要員数誤差が64%以下の場合、*MBRE* が従来法と比較して有意に改善しており、その他の指標も有意に改善していた。ただし、要員数誤差が大きい場合、各指標の差分は他の見積り方法よりも小さく、差分が負の値になっている(見積り精度が従来法よりも低下している)場合があった。よって、5段階法は要員数の誤差が明確であり、かつ誤差が小さい場合に有効であるといえる。誤差が不明確である場合、見積り精度が従来法よりも低下する可能性があるため、適用は避けるべきである。

4.2 誤差考慮法

誤差考慮法は、要員数誤差が小さい場合、各評価指標の差分が最も小さかった(見積り精度が最も低かった)。ただし、要員数誤差が大きい場合、各評価指標の差分が最も大きく(見積り精度が最も高く)、かつ差分が負の値になる(見積り精度が従来法よりも低下する)ことがなかった。要員数誤差が160%以下の場合、*MBRE* が従来法と比較して有意に改善していた。さらに要員数誤差が150%以下

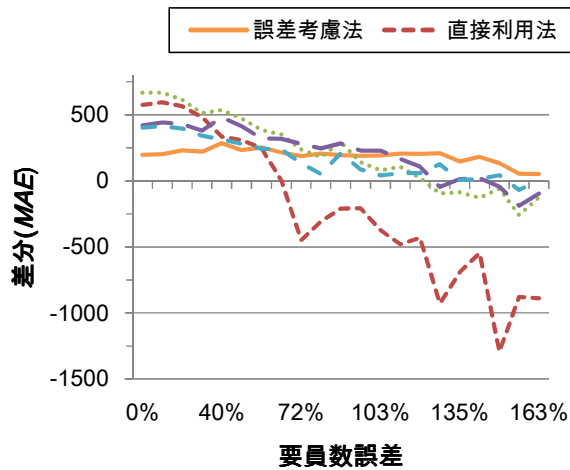


図 4 MAE と要員数誤差との関係

Figure 4 Relationship between MAE and team size error.

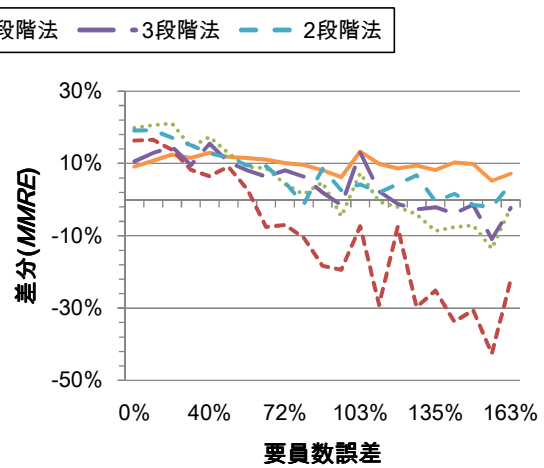


図 5 MMRE と要員数誤差との関係

Figure 5 Relationship between MMRE and team size error.

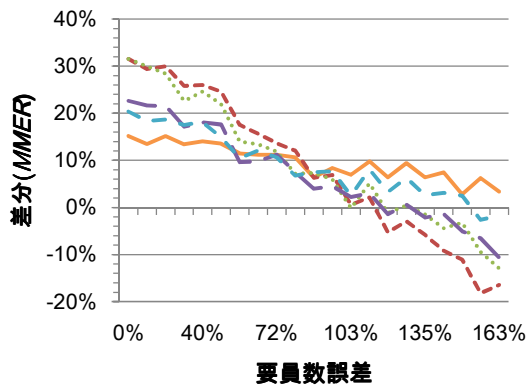


図 6 MMR と要員数誤差との関係

Figure 6 Relationship between MMR and team size error.

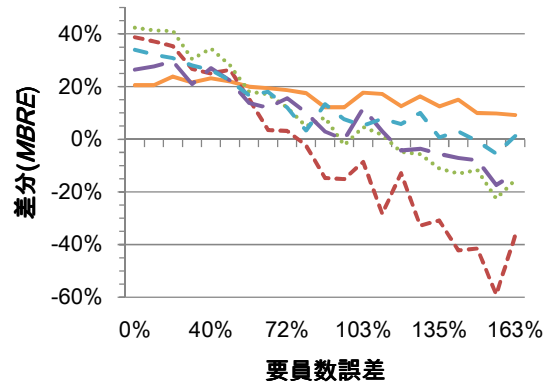


図 7 MBRE と要員数誤差との関係

Figure 7 Relationship between MBRE and team size error.

の場合、ほとんどの評価指標が有意に改善していた。よって、誤差考慮法は要員数の誤差が不明確である場合や、誤差が大きい場合に有効であるといえる。

4.3 直接利用法

直接利用法は、要員数の誤差が小さい場合、MAE と BRE の差分が 5 段階法に次いで大きく、MMER は多くの場合で最も差分が大きかった。ただし、要員数の誤差が大きい場合、各指標の差分が負の値になっており、かつ差分の絶対値が大きかった（従来法と比べて精度が大きく低下していた）。要員数の誤差が 135% 以上の場合、MBRE が従来法よりも有意に悪化しており、誤差が 160% 以上の場合、MMRE を除く評価指標全てが有意に悪化していた。従って、直接利用法は要員数の誤差が不明確である場合や、誤差が大きい場合に用いるべきでないといえる。

4.4 3 段階法, 2 段階法

要員数の誤差が小さい場合を除き、3 段階法と 5 段階法の各評価指標の差分の変化は、傾向が比較的類似していた。2 段階法とこれらを見積もり方法を比較すると、要員数誤

差が大きい場合、2 段階法の MMR, MMER, MBRE の差分はほとんどの場合で最も大きく、MAE も最も大きい場合が多かった。また、要員数の誤差が大きい場合、2 段階法では各指標の差分が負の値になっている場合が少なく、かつその場合の差分の絶対値は小さかった（従来法と比べて精度が大きく低下していなかった）。

5. 関連研究

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について説明しており、要員数の見積もり誤差は「前提の誤りに基づく誤差」に分類される。この誤差の影響を抑えるために、Kitchenham らはコンティンジェンシー（予備の工数）を設定することを提言するとともに、コンティンジェンシーを決定する方法（見積もり工数と、見積もり時の前提が誤っている確率に基づいて計算する）を説明している。ただし Kitchenham らが指摘しているように、計算時の想定内容が誤っている場合、コンティン

ジェンシーは有効に機能しない。誤差考慮法はコンテンツジェンシー計算時のような作業を必要とせず、かつ誤りに対してもロバストである。

熟練者による見積もり誤差に着目した研究が存在する。Jørgensen[6]は(モデルによる見積もりではなく)熟練者による見積もりの精度を改善することを目的として、見積もり誤差を目的変数とし、説明変数に契約形態(固定料金かなど)などを用いて重回帰分析を行なっている。ただし、モデルの説明力は低く、また、提案方法のような誤差を前提とした見積もり方法は提案していない。

6. おわりに

本研究では、開発要員数を工数見積もりモデルの説明変数として用いる方法を提案した。要員数の見積もりに誤差が含まれることを考慮した方法(誤差考慮法)を提案するとともに、誤差を考慮しない4種類の方法(単純法)について説明した。実験において要員数の誤差を変化させ、従来法(要員数を含まない見積もり方法)と見積もり精度を比較した。その結果、以下が明らかとなった。

- 誤差考慮法は要員数の誤差が大きい場合に有効であり、従来法よりも精度が低くなることがなかった。誤差考慮法は誤差が不明確である場合や、誤差が大きい場合に適用するとよい。
- 5段階法(単純法の一つ)は要員数の誤差が小さい場合に有効であった。5段階法は要員数の誤差が明確であり、かつ誤差が小さい場合に適用するとよい。
- 直接利用法(単純法の一つ)は、要員数の誤差が大きい場合、従来法よりも精度が大きく低下していた。直接利用法は要員数の誤差が不明確である場合や、誤差が大きい場合には適用すべきでない。

今後の課題は、実際にプロジェクトマネージャが要員数を見積もった場合の誤差を明らかにすることと、他のデータセットに提案方法を適用した場合の効果を確かめることである。

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] Burgess, C. and Lefley, M.: Can genetic programming improve software effort estimation? A comparative evaluation, *Journal of Information and Software Technology*, vol. 43, no. 14, pp. 863-873 (2001).
- [3] Conte, S., Dunsmore, H., and Shen, V.: *Software Engineering, Metrics and Models*, Benjamin/Cummings (1986).
- [4] International Software Benchmarking Standards Group (ISBSG), *ISBSG Estimating: Benchmarking and research suite*, ISBSG (2004).
- [5] Jeffery, R., Ruhe, M., and Wiczorek, I.: Using Public Domain Metrics To Estimate Software Development Effort, *Proc. International*

Symposium on Software Metrics (METRICS), pp. 16-27 (2001).

- [6] Jørgensen, M.: Regression Models of Software Development Effort Estimation Accuracy and Bias, *Empirical Software Engineering*, Vol. 9, Issue 4, pp. 297-314(2004).
- [7] Kitchenham, B., and Linkman, S.: Estimates, Uncertainty, and Risk, *IEEE Software*, vol. 14 no. 3, pp. 69-74 (1997).
- [8] Kitchenham, B., MacDonell, S., Pickard, L., and Shepperd, M.: What Accuracy Statistics Really Measure, *Proc. IEE Software*, vol. 148, no. 3, pp. 81-85 (2001).
- [9] Lokan, C.: What Should You Optimize When Building an Estimation Model? *Proc. International Software Metrics Symposium (METRICS)*, pp. 34 (2005).
- [10] 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).
- [11] Mendes, E., Di Martino, S., Ferrucci, F., and Gravino, C.: Cross-company vs. single-company web effort models using the Tukutuku database: An extended study, *Journal of Systems and Software*, vol. 81, issue 5, pp. 673-690 (2008).
- [12] 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).
- [13] Møllokken-Østfold, K. and Jørgensen, M.: A Comparison of Software Project Overruns-Flexible versus Sequential Development Models, *IEEE Transaction on Software Engineering*, vol. 31, no. 9, pp. 754-766 (2005).
- [14] Rodríguez, D., Sicilia, M., García, E., and Harrison, R.: Empirical findings on team size and productivity in software development, *Journal of Systems and Software*, vol. 85, issue 3, pp. 562-570 (2012).
- [15] Shepperd, M. and Schofield, C.: Estimating software project effort using analogies, *IEEE Transaction on Software Engineering*, vol. 23, no. 12, pp. 736-743 (1997).
- [16] Srinivasan, K. and Fisher, D.: Machine learning approaches to estimating software development effort, *IEEE Transaction on Software Engineering*, vol. 21, no. 2, pp. 126-137 (1995).
- [17] Tsunoda, M., Amasaki, S., and Monden, A.: Handling categorical variables in effort estimation, *Proc. International Symposium on Empirical software engineering and measurement (ESEM)*, pp. 99-102 (2012).
- [18] Tsunoda, M., Monden, A., Yadohisa, H., Kikuchi, N., and Matsumoto, K.: Software development productivity of Japanese enterprise applications, *Information Technology and Management*, vol. 10, issue 4, pp. 193-205 (2009).