

熟練者判断を取り入れたソフトウェア開発工数見積もりモデル

角田 雅照¹ 門田 暁人² ジャッキー キョン³ 松本 健一²

概要: プロジェクトの失敗を避けるためにはプロジェクト管理が重要であり、工数見積もりはその基礎となるものである。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。ただし、定性的または定量的に測定されていなければ、生産性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、プロジェクトマネージャが定性的、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。評価実験において、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た結果、提案方法は生産性レベルの誤推測に対してロバストであり、多くの場合、従来の見積もり方法よりも高い精度となることがわかった。

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

Software Effort Estimation Models Incorporating Expert Judgment

MASATERU TSUNODA¹ AKITO MONDEN² JACKY KEUNG³ KENICHI MATSUMOTO²

Abstract: Project management is important to avoid software development failure. Effort estimation is the basis of the management. 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. In an experiment, we changed error of productivity level and compared proposed methods with a conventional model. As a result, our method is robust for the error of productivity level, and the method is more accurate than the conventional model in most cases.

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

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コスト超過や納期遅延などのプロジェクトの失敗を避けるためには、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[2][20][21]。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、例えば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、例えば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。

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、すなわち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）測定されていなければ、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。

†1 近畿大学

Kinki University

†2 奈良先端大学院大学

Nara Institute of Science and Technology

†3 香港城市大学

City University of Hong Kong

本研究では、これらの必ずしも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、熟練者であるプロジェクトマネージャ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。一般にプロジェクトマネージャは、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、プロジェクトマネージャ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はいえる。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。提案方法の評価実験においては、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. 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説明変数（開発要員の技術力など）を説明変数に用いている場合がある[2]。ただし、例えばプロジェクトマネージャ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[11]。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。

数学的モデルの構築に、熟練者の判断を利用する方法が提案されており、Baker[1]は熟練者による見積もりモデルの構築を支援するツールを提案している。ただし、そのツールでは従来法に基づく見積もりモデルを用いており、提案方法のような測定されてない情報を直接モデルに取り入れることはできない。

CoBRA 法[4]は、見積もりモデルの導出に熟練者の知識を利用する。具体的には、生産性に影響すると考えられる変数（開発チームの能力など）を熟練者が選定し、その変数を用いてモデルが構築される。構築されたモデルの利用方法は、一般的なモデルに基づく見積もり方法と同じである。それに対し提案方法では、モデル構築時には熟練者の知識を利用せず、モデル利用時（工数見積もり時）に、熟練者のプロジェクトに対する詳細な知識を利用する点が異なる。

CoBRA 法は熟練者の知識を詳細にモデルに反映できることが利点であるが、モデルを構築するためある程度の工数を要する。例えば Briand ら[4]のケーススタディでは、モデルの構築に2人月を要している。それに対し提案方法では、非常に少ない工数でモデルを構築することができる（1人日程度でモデルの構築が可能であると思われる）。

CoBRA 法、提案方法とも熟練者に依存する部分があり、どちらの見積もり精度が高いかを結論付けることは容易ではない。ただし、仮に提案方法の見積もり精度が CoBRA 法よりも低いとしても、熟練者の知識を CoBRA 法よりも少ない初期工数で利用することができ、実験において示すように、従来モデル（熟練者の知識を利用しないモデル）よりも見積もり精度が高くなることは利点であると考えられる。

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[12]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（モデル構築に用いるプロジェクトの開発規模を揃えるべきであるなど）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。Jørgensen[10]は熟練者による見積もりの誤差原因を明らかにするために、見積もり誤差を目的変数とし、説明変数にユーザの優先項目（コスト重視か、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。

3. 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、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、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。提案方法による見積もりモデルの構築手順を図1に示す。

提案方法では、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できる、すなわち生産性レベルの推測誤差は、平均的には一定の範囲に収まると仮定している。プロジェクトマネージャが生産性レベルの推測を大きく誤る可能性がないとはいえない。ただし、熟練者による見積もりとモデルによる見積もりは、精度の面で優劣は付けられないというサーベイ結果[9]や、熟練者29人による積み上げ式に基づく見積もりの実験結果（相対的な誤差の平均値が78%）[18]より、類推法を用いるならば大きく誤る頻度は高くないと考えられる。

本章では提案方法を重回帰分析に適用することを前提

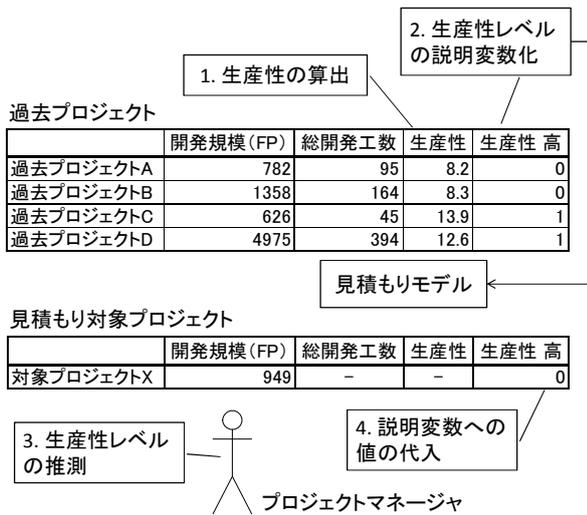


図 1 提案方法による見積もりモデルの構築手順

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

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。

3.1 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

提案方法に限らず、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、少なくとも複数の過去プロジェクトにおける開発規模と総開発工数のデータが必要であり、それらを用いて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（効率が低い）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、例えば開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。

ステップ3: 生産性レベルの推測

プロジェクトマネージャは、見積もり対象プロジェクトに関して把握している情報、例えば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）よりも大きいか小さいか、すなわち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。詳細な推測方法については3.4節で後述する。

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。

3.2 推測誤りを考慮する方法（誤り考慮法）

誤り考慮法は、直感的に説明すると生産性を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方法である。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（図2）。レベル1をレベル5と誤る、あるいは逆にレベル5を1と誤る確率は低いと考えられるため、単純法よりも誤りに対して頑健である。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

プロジェクトマネージャは、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、言い換えると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大きいか、言い換えると過去のプロジェクトの上位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、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を代入して、工数を見積もる。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。

3.3 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

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 2 誤り考慮法における生産性レベルの扱い

Figure 2 Productivity level on the error consideration model.

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、目的変数との関係が線形とは限らないため、これをダミー変数化する。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

プロジェクトマネージャは、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 パーセンタイル値より大きい、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 パーセンタイル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 パーセンタイル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。

3.4 生産性レベルの推測方法

プロジェクトマネージャは過去のプロジェクト全てを把握しているわけではないため、見積もり対象プロジェクトの生産性が、データセットの上位 20% や下位 20% に入るかどうかを直接的に判断することは容易ではない。ただし類推法に基づき、見積もり対象プロジェクトと生産性が類似したプロジェクトを特定することは可能であると考えら

れる。そこで、類似プロジェクトの生産性に基づき、生産性レベルを決定する。具体的には、以下の手順で見積もり対象プロジェクトの生産性レベルを決定する。

1. プロジェクトマネージャは、自らが関わったプロジェクトと比較して、見積もり対象プロジェクトと生産性が近いと考えられるプロジェクト（類似プロジェクト）を特定する。
 - 比較対象を増やすために、プロジェクトマネージャとしてだけでなく、システムエンジニアとしても関わったプロジェクトも含める。
2. 類似プロジェクトの生産性を計算する。
 - プロジェクトごとに記録されている FP と工数から生産性を算出する。
3. 算出した生産性と、生産性レベルの閾値（20、80 パーセンタイル値など）と比較し、見積もり対象プロジェクトの生産性レベルを決定する。

3.5 生産性レベルの扱い

生産性レベル決定時の段階数やパーセンタイルは予備実験に基づいて決定した。単純法で 5 段階の生産性レベルを設定した場合、生産性レベルの推測誤差の影響を受けやすく、誤差が大きくなると、工数見積もりの精度が大きく低下した。そこで、生産性レベルは 3 段階までとした。誤差考慮法において、25、75 パーセンタイル（4 段階）を生産性レベルの閾値とすると、生産性の推測誤差の影響を比較的受けやすい傾向があった。10、90 パーセンタイル（10 段階）の場合、工数見積もりの精度改善の効果が小さくなった。そこで、20、80 パーセンタイル（5 段階）を閾値とした。段階数やパーセンタイル値が最適であるかどうかまでは確認できていないが、それらを明らかにすることは今後の課題とする。

提案方法では生産性レベルを説明変数として扱いモデルを構築するが、生産性レベルに基づいてデータを層別し、それぞれのサブセットのデータごとに別個のモデルを構築することも可能である。予備分析を行ったところ、両者の見積もり精度にあまり差がなかったため、今回は生産性レベルを説明変数として扱った。

4. 評価実験

単純法、誤り考慮法、3 段階単純法の性能を評価するために、3 つのデータセットを利用して、生産性レベルの推測の誤りの程度を変化させ、見積もり精度を従来の、生産性レベルを用いないモデルと比較した。

4.1 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 [8]。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

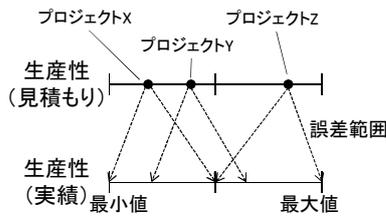


図 3 生産性の推測誤りの範囲

Figure 3 Error range of speculated productivity.

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[16]の条件、例えばデータの品質が 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[14]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。このうち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[3]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[7]。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、プロジェクトマネージャの経験年数、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。トランザクション数とエンティティ数は説明変数から除外した。これは、未調整 FP=トランザクション数+エンティティ数という関係があり、説明変数として用いることは、多重共線性の点から不適切であるためである。

4.2 評価指標

工数見積り精度の評価指標として、AE (Absolute Error), MRE (Magnitude of Relative Error)[6], MER (Magnitude of Error Relative to the estimate)[13], BRE (Balanced Relative Error)[17]の 4 つの指標の平均値と中央値を用いた。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

積もり値との相対誤差であるといえる。ただし、MRE と MER は過大見積もりと過少見積もりに対し、アンバランスな評価になるという問題がある[5][15]。そこで本研究では、MRE、MER に加え、過大見積もりと過少見積もりをバランスよく評価する指標[19]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。

提案方法の見積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。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。

4.3 生産性の推測誤差

推測の誤りを生じさせるため、テストデータの実績値に n%の誤差を発生させ、その実績値に基づいて生産性レベルを決定した。実績値を 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、実際の実績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、実験においても実績値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 から誤差を含む実績値を逆算した。

生産性の誤差が正規分布に従う、すなわち誤差の分布が対称であり、偏りが無いという仮定の妥当性について述べる。Moløkken-Østfold ら[18]は、熟練者 29 人による見積り結果の偏りを、BREbias により示している。BREbias は過大見積もりや過小見積もりに偏っているかどうかを示す指標であり、計算式は式 3 と同一である。論文の実験では熟練者の見積りによる BREbias の平均値は -0.04、中央値は 0.00 であり、ほとんど偏りがなかった。ここでの見積もりは積み上げ法に基づいており、また、データの詳細な分布は不明ではあるが、実績値の分布に偏りが無いという仮定に一定の妥当性を与えると考えられる。

生産性の誤差の設定範囲の妥当性について述べる。上記文献[18]の実験では、熟練者による見積りの BRE 平均値が 78%であった。本論文の実験で変動させた誤差の範囲は、BRE 平均値に換算すると最大で 163%となっている。すなわち、実験で設定した誤差の範囲は熟練者による見積り誤差の 2 倍以上となっている。熟練者の見積りとモデルによる見積りとは、精度の面で優劣は付けられないという

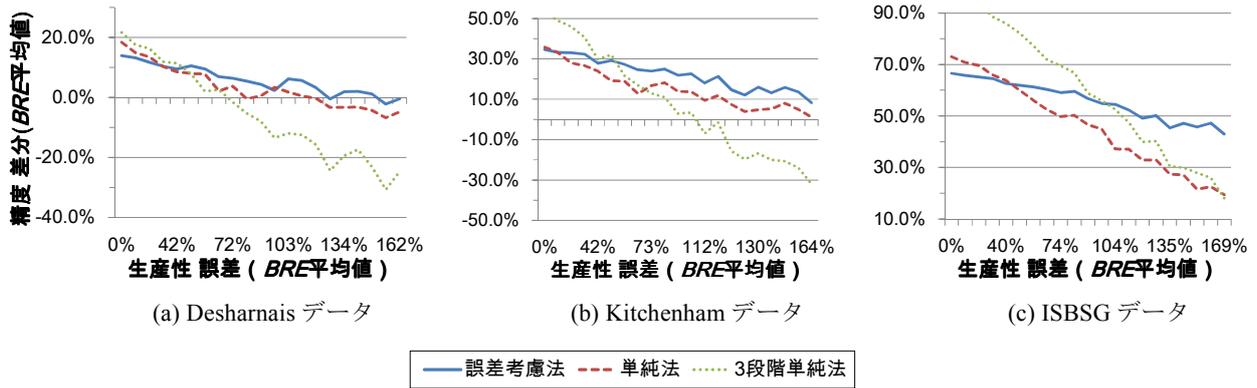


図 4 見積り精度の差分と生産性の誤差との関係

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

サーベイ結果[9]もあるため、類推法を用いれば荒い粒度の推測でも誤差はある程度の範囲に収まると考えられる。

なお、生産性レベルの推測時には、生産性の具体的な推測値を決定する必要はなく、3章で述べた程度の推測でよい。上記は図3に示すような、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。

4.4 実験手順

実験は 5-fold cross validation に基づいて行った。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。

実験は各見積り方法に対し、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 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、誤差を含んだ生産性を計算し、それに基づき生産性レベルを決定する。
5. 構築したモデルを用いて、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て見積もる。
6. 見積り結果に基づき、各評価指標を計算する。

表 1 従来法の見積り精度

Table 1 Estimation accuracy of the conventional model.

データセット		AE	MRE	MER	BRE
ISBSG	平均値	2990	103.6%	93.3%	152.0%
	中央値	1262	53.7%	59.1%	86.0%
Kitchenham	平均値	1746	70.5%	57.5%	92.6%
	中央値	628	38.8%	41.3%	53.6%
Desharnais	平均値	1724	37.5%	37.4%	46.9%
	中央値	1066	32.5%	30.9%	38.5%

7. 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 2~6 を繰り返す。
8. テストデータを変更し、手順 2~7 を 5 回繰り返す。
9. 手順 1~7 を 4 回繰り返す。

5. 実験結果及び考察

5.1 従来法との比較

従来法と提案方法との見積り精度 (BRE 平均値) の差分と、生産性の誤差との関係を図 4 に示す。紙面の都合上、BRE 平均値以外の評価指標のグラフは省略する。横軸の生産性誤差は、生産性の BRE 平均値を算出したものである。BRE は、式 3 の分母を絶対値に置き換えた式により求める。すなわち、誤差 n の絶対値をとり、平均値を求めたものが BRE 平均値となる。そのため、誤差 n の平均値は 0 であるが、BRE 平均値は 0 とならない。また、従来法の見積り精度を表 1 に示す。

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、すなわち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、実験において変化させた誤差の範囲 (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 を下回るものがほとんどなく、下回った場合もごくわずかな程度であった。

従来法と比べて評価指標が改善した数 (最大 8 個) と生産性の誤差との関係を図 5 に示す。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、改善した指標の数が 4 個を下回る、すなわち悪化した指標の数が半数を超える場合があった。これに対し、誤り考慮法の場合、Desharnais データにおいて生産性の誤差が最も大きかった場合を除いて、改善した指標の数は 4 以上、すなわち悪化した指標の数は最大でも 4 個であった。

従来法と比べて、評価指標が統計的に有意に改善したのかどうかを確かめた。図 6 は 8 個の評価指標うち、有意に

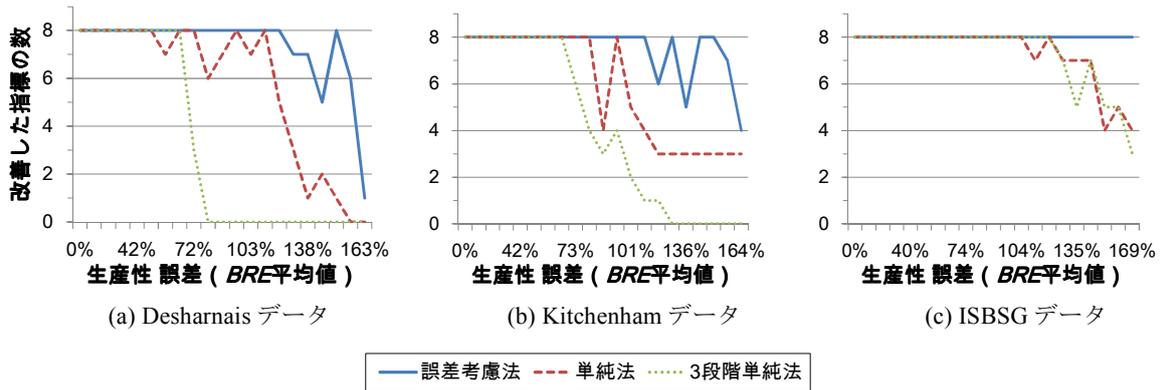


図 5 評価指標が改善した数と生産性の誤差との関係

Figure 5 Relationships between the number of improved indices and error of productivity.

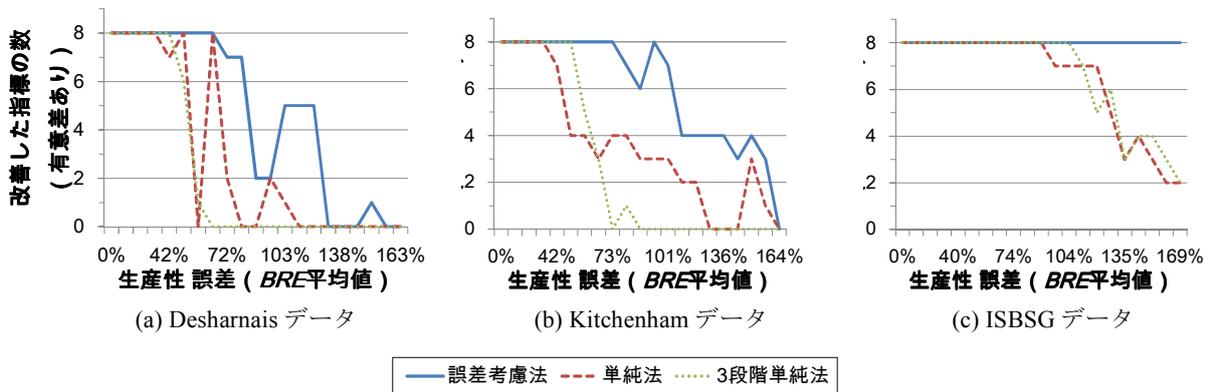


図 6 評価指標が有意に改善した数と生産性の誤差との関係

Figure 6 Relationships between the number of significantly improved indices and error of productivity.

改善した個数を示す。どの手法を用いた場合でも、生産性の誤差が小さい場合は全ての評価指標が有意に改善していた。生産性の誤差が大きい場合、有意に改善した指標の数は誤り考慮法が最も多かった。

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、すなわち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。

5.2 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。図5においても、生産性の誤差が大きい場合、改善した指標の数が最も多い方法は誤り考慮法であった。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（図4参照）。

5.3 熟練者による工数見積もりとの比較

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4.4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、すなわち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BRE平均値は100%となり、

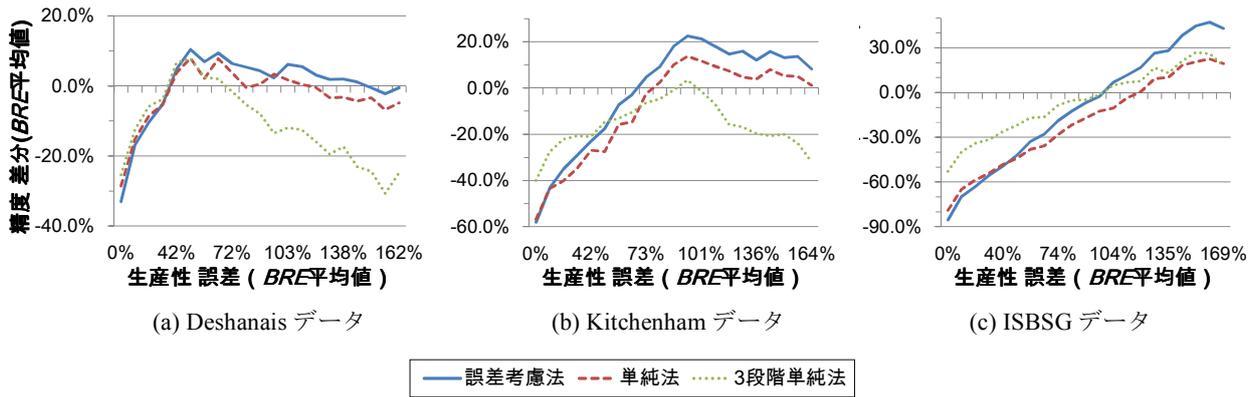


図 7 見積もり精度の差分と生産性の誤差との関係 (熟練者による見積もりを考慮した場合)

Figure 7 Relationships between difference of estimation accuracy and error of productivity considering effort estimation by experts.

こちらの精度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。

比較した結果を図 7 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平均的に低い場合、モデルによる見積もり、熟練者による見積もりのどちらかを採用するよりも、提案方法を採用するほうが、見積もり精度が改善されるといえる。例えば Deshanais データでは、モデルによる見積もりの BRE 平均値は 46.9% である。熟練者の見積もりの BRE 平均値が 47.7% の場合、誤り考慮法による見積もりの BRE 平均値は 36.5% となり、モデルによる見積もりや熟練者の見積もりを採用するよりも、誤り考慮法の見積もりを採用するほうが、見積もり精度が改善される (図 7 において精度の差分が 0 を超えている場合に該当する)。

熟練者の見積もり精度が Molokken-Østvold ら [18] の実験結果と同じである、すなわち BRE 平均値が 78% であるとする、Deshanais データと Kitchenham データでは誤り考慮法を利用することにより、見積もり精度の改善が見込める。ISBSG データに関しては、複数組織のデータが混在してい

ることもあり、モデルによる見積もり精度が低いため、熟練者の見積もりを利用したほうがよいといえる。このように、仮に熟練者が工数を見積もるとしても、その見積もり精度が平均的に高くない場合、提案方法は見積もり精度の面で有用であるといえる。

5.4 生産性レベルの推測誤差の把握

生産性レベルの推測誤差は、以下の 2 つの方法でおおよそ把握できると考えられる。

1. 自分が関わっていない過去プロジェクトの生産性を類推法に基づいて推測し、それと実際の生産性に基づいて推測誤差を計算する。
 - ▶ 自分が関わっている過去プロジェクトの場合、実際の生産性を把握していると考えられるので、関わっていないものを用いる。
2. 過去プロジェクトにおいて見積もり工数が記録されている場合、それを規模で除したものを推測された生産性とみなし、それと実際の生産性に基づいて推測誤差を計算する。

1 の場合、実際の推測誤差は求めたものより小さめになるとみなす。これは、自分が関わっているプロジェクトの場合、把握している情報がより多くなるためである。2 の場合、実際の推測誤差は求めたものより大きめになるとみなす。これは、工数見積もりよりも生産性レベルの推測のほうが簡易的に行われる可能性が高いためである。

6. 考察

提案方法は工数見積もりモデルに熟練者の判断を取り入れているが、Boehm はモデルによる見積もりは熟練者の主観に左右されないことを利点として挙げている [11]。そこで、工数見積もりモデルに熟練者の判断を取り入れることの妥当性と利点について考察する。

Jørgensen は、「モデルによる見積もりで説明変数として用いられる、開発要員のスキルや顧客の要求などの決定には熟練者の判断が必要であり、これは主観により左右され

うる」と指摘している[11]. すなわち、モデルによる見積もりでも、COCOMO[2]などの一部のモデルでは、熟練者の判断が含まれていることになる。さらに CoBRA 法[4]では、見積もりモデルの導出に熟練者の知識を利用するとともに、COCOMO と類似した説明変数を用いる。このように、程度の違いはあるが、従来のモデルによる見積もりにおいても、熟練者の判断は含まれているといえる。

モデルによる見積もりにおける大きな問題点の一つは、熟練者の判断がモデルに組み込まれていないと、そもそも熟練者がモデルを利用しないということである。Jørgensen は「モデルによる見積もりに、プロジェクトの詳細な情報（例えば誰が開発メンバーに入るか）が含まれないため、熟練者がモデルによる見積もり結果を採用しない」ことを問題点として挙げている[11]. 提案方法は、プロジェクトの詳細な情報を生産性として間接的にモデルに反映することができるため、モデルによる見積もりの活用を促すことが期待される。これが熟練者による判断をモデルに組み込む最大の利点といえる。

また、Boehm の調査[11]によると、モデルによる見積もりは、開発規模などの前提条件を変更した場合の工数の変化をシミュレートする際にも用いられている（熟練者による見積もりは、この目的のためにはほとんど使われない）。提案方法を用いることにより、熟練者の判断を反映したシミュレーションが容易に実施できることも利点のひとつである。

7. 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、従来法である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。さらに、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し、熟練者による工数見積もりと精度を比較した。その結果、以下が明らかとなった。

- 生産性レベルの推測誤差に対し、誤り考慮法が最もロバストであり、従来法よりも精度が悪化する可能性が低い。誤差が不明確な場合、誤り考慮法を適用することにより、見積もり精度を高められる。
- 生産性レベルの推測誤差を把握できる場合、誤差が小さいならば3段階単純法を適用し、大きいならば誤り考慮法を適用することにより、見積もり精度を高められる。
- 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場合、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ると、見積もり精度を高められる。

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。なお 5.3 節で述べたように、もし生産性レベルの推測が困難としても、誤り考慮法は有効に働く。

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。

著者紹介

角田雅照（正会員）

平成 9 年和歌山大学経済学部卒業。平成 16 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了。平成 19 年同大学同研究科博士後期課程修了。同年同大学同研究科特任助教。平成 24 年東洋大学総合情報学部助教。平成 25 年近畿大学理工学部情報科学科講師。博士（工学）。ソフトウェアメトリクスの研究に従事。電子情報通信学会，日本ソフトウェア科学会，ヒューマンインタフェース学会，教育システム情報学会，IEEE 各会員。

門田暁人（正会員）

平成 6 年名古屋大学工学部電気学科卒業。平成 10 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。同年同大学同研究科助手。平成 16 年同大学助教授。平成 19 年同大学准教授。平成 15～16 年 Auckland 大学客員研究員。博士（工学）。ソフトウェアメトリクス，ソフトウェアプロテクション，ヒューマンファクタ等の研究に従事。電子情報通信学会，日本ソフトウェア科学会，IEEE，ACM 各会員。

Jacky Keung

香港城市大学コンピュータサイエンス学科助教。オーストラリア ニューサウスウェールズ大学にて博士号を取得後，オーストラリア NICTA のソフトウェア工学研究グループにて博士研究員。ソフトウェア工学に活発に取り組んでいる研究者であり，工数見積り，複雑システムの定量的モデリング及び評価，ソフトウェア開発データを用いたアンサンブル学習とその結果のプロジェクト管理への適用に興味を持つ。

松本健一（正会員）

昭和 60 年大阪大学基礎工学部情報工学科卒業。平成元年同大学大学院博士課程中退。同年同大学基礎工学部情報工学科助手。平成 5 年奈良先端科学技術大学院大学助教授。平成 13 年同大学教授。工学博士。

エンピリカルソフトウェア工学，特に，プロジェクトデータ収集／利用支援の研究に従事。電子情報通信学会，日本ソフトウェア科学会，ACM 各会員，IEEE Senior Member.

参考文献

[1] D. Barker, *A Hybrid Approach to Expert and Model Based Effort Estimation*, Master Thesis, West Virginia University (2007).

[2] Boehm, B.: *Software Engineering Economics*, Prentice Hall (1981).

[3] Boetticher, G., Menzies, T. and Ostrand, T.: *PROMISE Repository of empirical software engineering data*, West Virginia University, Department of Computer Science (2007).

[4] 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).

[5] 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).

[6] Conte, S., Dunsmore, H., and Shen, V.: *Software Engineering, Metrics and Models*, Benjamin/Cummings (1986).

[7] 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).

[8] International Software Benchmarking Standards Group (ISBSG), *ISBSG Estimating: Benchmarking and research suite*, ISBSG (2004).

[9] Jørgensen, M.: A review of studies on expert estimation of software development effort, *Journal of Systems and Software*, vol. 70, issues 1-2, pp. 37-60 (2004).

[10] Jørgensen, M.: Regression Models of Software Development Effort Estimation Accuracy and Bias, *Empirical Software Engineering*, Vol. 9, Issue 4, pp. 297-314(2004).

[11] 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).

[12] Kitchenham, B., and Linkman, S.: Estimates, Uncertainty, and Risk, *IEEE Software*, vol. 14 no. 3, pp. 69-74 (1997).

[13] 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).

[14] 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).

[15] Lokan, C.: What Should You Optimize When Building an Estimation Model? *Proc. International Software Metrics Symposium (METRICS)*, pp. 34 (2005).

[16] 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).

[17] 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).

[18] Moløkken-Østfold, K., Haugen, N., and Benestad, H.: Using planning poker for combining expert estimates in software projects, *Journal of Systems and Software*, vol. 81, no. 12, pp. 2106-2117 (2008).

[19] Møløkken-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).

[20] Shepperd, M. and Schofield, C.: Estimating software project effort using analogies, *IEEE Transaction on Software Engineering*, vol. 23, no. 12, pp. 736-743 (1997).

[21] 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).