特別研究報告

題目

e^3 の機能拡張と追加評価

指導教員
楠本 真二 教授

報告者
古田 雄基

平成 27 年 2 月 13 日

大阪大学 基礎工学部 情報科学科

平成 26 年度 特別研究報告

e^3 の機能拡張と追加評価

古田 雄基

内容梗概

ソフトウェア開発プロジェクトでの開発工数の見積りは，プロジェクト計画作成において非常に重要な作業である．見積もられた工数を基にして，開発期間，開発人数，開発コスト等が計画される．工数を管理し，プロジェクトを円滑に遂行することにより，納期遅れやコスト超過といったプロジェクトの失敗を未然に防ぐことが可能となる．

所属研究室では，これまでに工数予測ツール e^3 を開発してきている．e^3 は予測対象プロジェクトに対して複数の工数予測手法を同時に適用し，妥当な工数予測結果を推薦するツールである．一方，e^3 には，より多くの予測手法の実装やユーザインタフェースの改良，より多くの適用評価が求められていた．

そこで，本研究では，工数予測ツール e^3 に対して複数の工数予測手法を追加し，ユーザインターフェースの改良を行った．次に，6 つのソフトウェア開発データセットを用いた追加実験を行った．結果として，これまでの評価結果と同じく，ツールが推薦する予測結果の有用性を確認した．更に，実用性の評価のため，商用見積もりツールである KnowledgePLAN との比較実験行った．比較実験にあたっては，ある企業の 42 プロジェクトデータを用いた．実験の結果，e^3 の予測精度は KnowledgePLAN の予測精度にわずかに劣っていたが，分析を通じて e^3 の予測精度向上のため方向性が確認できた．

主な用語

工数予測
予測精度
KnowledgePLAN
Leave-One-Out

目 次

表 目 次

図 目 次

1　まえがき

ソフトウェア開発において，プロジェクトを円滑に遂行するためには，工数予測が重要である．工数とは，開発期間と要員の積で算出される延べ作業時間を表す数値であり，プロジェクトを管理する上で重要な指標となる．工数を正確に予測し，必要な工数を把握することによって，スケジュールの適切な管理や開発資源の割り当てを行うことができる．結果として，納期遅れやコスト超過といったプロジェクトの失敗のリスクを抑えることが可能となる．工数を正確に予測するために開発されてる工数予測ツールの 1 つとして，e^3[3, 4, 5] が開発されている．e^3 は 1 件の現行プロジェクトに対して複数の工数予測手法を同時に適用し，Leave-One-Out(LOO)[6] を用いてデータセット中の過去プロジェクトの工数を予測したときの誤差をもとに算出した，現行プロジェクトの工数の予測値に対する期待される精度に応じた予測値の決定を支援することを目的としたツールである．

e^3 はツールが表示する期待される精度の高い工数予測手法をその都度選択することにより，特定の工数予測手法にとらわれることなく，使用するデータセットの特徴に応じた柔軟な予測値の決定が可能となる有用なツールであるが，このツールの機能に関して，インターフェースの不便さや登録されている手法の少なさに対する改善の要望があったため，それに応じたツールの機能拡張を行った．さらに，実際のデータセットに対して適用した例が少なかったため，ツールの有用性を確認する目的で，PROMISE の 6 つのデータセットに対してもこのツールを適用し，追加実験を行った．追加実験としては，このツールに対して行われた評価実験 [3] と同様に，特定の工数予測手法を使用し続けた場合と，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合とで予測誤差の比較を行う．また，e^3 の工数予測ツールとしての性能評価は今までに行われていなかったため，今回，商用見積もりツールである KnowledgePLAN[2] との性能比較を行った．KnowledgePLAN とは，1 万件を超えるプロジェクト実績データを搭載した知識データベースをもとに，プロジェクトの所要工数や期間，品質，コスト，成果物の規模などを予測するツールである．比較方法としては，ツールの表示する期待される精度の最も高い工数予測手法の予測値と，KnowledgePLAN の予測値のそれぞれに対して，実測値との誤差を求め，その結果を比較する．

以降，2 章では研究の背景となる関連研究について述べる．3 章では本研究の目的について述べる．4 章では e^3 に対して行った機能拡張について述べる．5 章では e^3 に対して行った評価実験と結果についての考察を述べる．最後に 6 章で本報告のまとめを述べる．

2　準備

2.1　工数予測ツール e^3

工数予測ツール「e^3」は，期待される制度による工数予測手法の選択を支援するツールである．1 件の現行プロジェクトに対して，複数の工数予測手法を適用し，工数予測手法ごとに工数の予測値と期待される精度を表示する．1 度の実行で複数の工数予測手法の予測結果を確認することができ，期待される精度の比較によって，特定の工数予測手法に依存しない柔軟な工数予測が可能となる．

e^3 の構成図を図 1 に示す．ユーザが現行プロジェクトと過去プロジェクトの実績データ (プロジェクトデータ) を与えると，まず，内部でプロジェクトデータの編集が行われる．その後，編集後のプロジェクトデータと実装されている工数予測手法から現行プロジェクトの工数予測と，期待される精度の計算が行われる．ユーザは，本ツールから得られた工数の予測値と期待される精度をもとに，採用する予測値を選択できる．

なお，e^3 は，汎用性を重視し，Ruby on Rails[7, 8] を用いて Web アプリケーションとして実装されている．

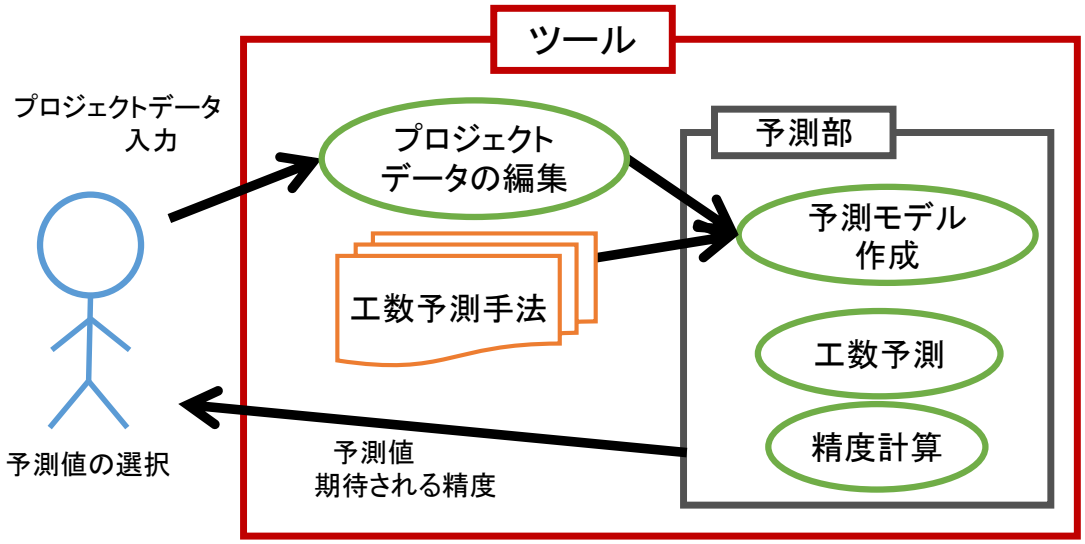

図 1: e^3 の構成図

2.1.1　期待される精度

ツールにおいて計算される期待される精度について以下で説明する．

ある工数予測手法によって得られた現行プロジェクトの工数の予測値に対する期待される精度は，同一の工数予測手法を用いて，過去プロジェクトの工数を予測したときの実測値と予測値の誤差で表現される．現行プロジェクトと違い，過去プロジェクトの工数の実測値は既知であるため，実測値と予測値の誤差を計算することが可能である．そのときの誤差が小さいほど，同一の工数予測手法で現行プロジェクトの工数を予測したときの期待される精度は高く，小さい誤差で現行プロジェクトの工数を予測できることが期待される．

過去プロジェクトの工数を予測する手順としては，Leave-One-Out(LOO)[6] を用いる．LOO は，過去プロジェクト n 件によって構成されるデータセットから 1 件を予測対象プロジェクトとして抽出し，残りの $n-1$ 件の過去プロジェクトをもとに予測対象プロジェクトの工数を予測して誤差を求めることを n 件の全過去プロジェクト分繰り返して工数予測手法の精度を計算する方法である．LOO を用いて，1 件の過去プロジェクトを抽出し，精度計算のための予測を行う場面を図 2 に示す．

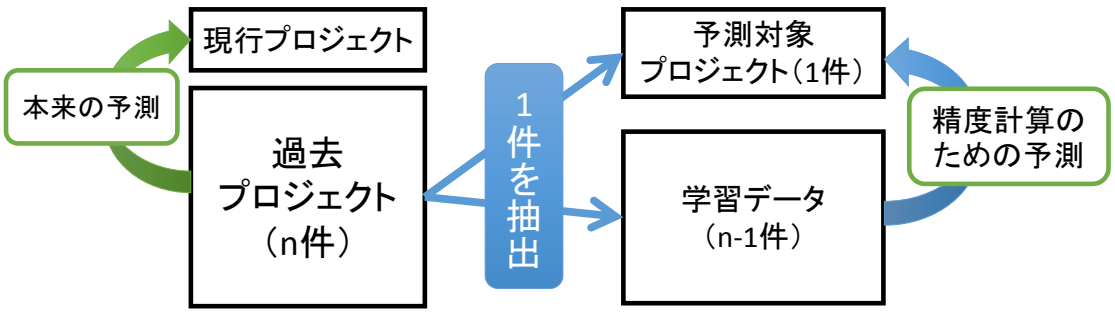

図 2: LOO を用いた過去プロジェクトの工数予測

ある工数予測手法について，LOO を用いて現行プロジェクトの予測値に対する期待される精度を求める具体的な手順は以下のとおりである．

手順 1　n 件の全過去プロジェクトから 1 件のプロジェクトを抽出する．

手順 2　抽出した 1 件の予測対象プロジェクト，残りの n-1 件を学習データ (過去プロジェクト) として，当該工数予測手法にて学習データをもとに予測対象プロジェクトの工数を予測する．

手順 3　予測対象プロジェクトの工数の実測値と予測値の誤差を計算する．

手順 4 予測対象プロジェクトとして抽出するプロジェクトを変更し，n 件の全過去プロジェクトについて，手順 1 から手順 3 を繰り返す．

手順 5 n 回繰り返された手順 3 によって求められた n 個の誤差をもとに，当該工数予測手法を使用した際の現行プロジェクトの工数の予測値に対する期待される精度を算出する．

2.1.2 類似性に基づく手法 [1]

以下に，今までに e^3 に実装されている手法の中から，今回の研究に関連する，類似性に基づく手法についてのみ記述する．

類似性に基づく手法 (Estimation by Analogy, EbA) は，過去プロジェクトの中から現行プロジェクトと類似しているプロジェクト (類似プロジェクト) を探し，類似プロジェクトの工数の実測値をもとに現行プロジェクトの工数を予測する手法である．これは，メトリクスの値の類似しているプロジェクト同士は工数の値も類似しているという仮説を前提としている．類似プロジェクトのみを用いて予測を行うため，過去プロジェクトの中に特異なプロジェクトが存在したとしても，そのプロジェクトの影響を受けずに予測が可能という長所がある．

EbA は，ダミー変数化，正規化，類似度計算，予測値計算の 4 つの手順から構成される．各手順で用いるアルゴリズムはそれぞれいくつか提案されているが，ここでは，e^3 に実装されているアルゴリズム [9] のみを示す

手順 1 ダミー変数化

データセットに名義尺度のメトリクスが含まれる場合，他の比例尺度のメトリクスと同様に扱えるようにするため，カテゴリごとにダミー変数に置き換える．プロジェクト p_i が名義尺度のメトリクス m_j においてカテゴリ c に属するかどうかを表すダミー変数 $d_{ij}(c)$ は式 (1) で定義される．

$$d_{ij}(c) = \begin{cases} 1 & \text{...カテゴリ } c \text{ に属する} \\ 0 & \text{...カテゴリ } c \text{ に属さない} \end{cases} \tag{1}$$

手順 2 正規化

一般に，データセット中の各メトリクスの値域にはばらつきがある．そこで，メトリクスごとの類似度への影響度を均等にするため，全てのメトリクスの値を 0 から 1 の範囲で正規化する．メトリクス m_j のデータセット中での最大値を $max(m_j)$，最小値を $min(m_j)$ とすると，プロジェクト p_i のメトリクス m_j の値を v_{ij} は式 (2) で定義される．

$$v_{ij}^{'} = \frac{v_{ij} - min(m_j)}{max(m_j) - min(m_j)} \tag{2}$$

手順 3 類似度計算

各過去プロジェクトについて，現行プロジェクトとの類似度を計算する．プロジェクト p_a とプロジェクト p_b の類似度 $sim(p_a, p_b)$ は式 (3) で定義される．

$$sim(p_a, p_b) = \frac{1}{dist(p_a, p_b)} \tag{3}$$

ここで，$dist(p_a, p_b)$ はプロジェクト p_a とプロジェクト p_b のユークリッド距離を表し，正規化されたメトリクスの値 $v_{ij}^{'}$ を用いて式 (4) で定義される．

$$dist\left(p_a, p_b\right) = \sqrt{\sum_{j=1}^{n} \left(v_{aj}^{'} - v_{bj}^{'}\right)^2} \tag{4}$$

なお，n はメトリクス数を表す．

手順 4 予測値計算

類似プロジェクトの工数の実測値をもとに，現行プロジェクトの工数の予測値を算出する．予測値計算のアルゴリズムとしては，Wighted SUM[10] を用いたプロジェクト p_i の工数の実測値を e_i とし，プロジェクト p_c とプロジェクト p_i の類似度 $sim(p_c, p_i)$ とすると，現行プロジェクトの p_c の工数の予測値 $\hat{e}_c$ は式 (5) で定義される．

$$\hat{e}_c = \frac{\sum_{i \in k'} \left(e_i \times sim\left(p_c, p_i\right)\right)}{\sum_{i \in k'} \left(sim\left(p_a, p_i\right)\right)} \tag{5}$$

ここで，$k^{'}$ は現行プロジェクトと類似度の高い上位 k 個のプロジェクトの集合を表す．

2.2 KnowledgePLAN[2]

KnowledgePLAN は SOFTWARE PRODUCTIVITY RESEARCH(SPR) 社の開発したソフトウェアプロジェクトの計画を支援するツールである．KnowledgePLAN を使用することにより，プロジェクトのサイジングと，作業，リソース，スケジュール，および不具合の見積もりを効果的に実施することができる．

KnowledgePLAN では 1 万件を超えるプロジェクト実績データを搭載した知識データベースを基に，プロジェクトの所要工数や期間，品質，コスト，成果物の規模などを予測できる．

2.3 工数予測手法

過去プロジェクトの実績データに基づくプロジェクトの工数見積もりに関する既存研究としては，大杉直樹ら [11] の研究がある．この研究では，複数の異なる組織で収集されたソフトウェア開発プロジェクトのデータに対して，重回帰分析，対数重回帰分析，ニューラルネット，協調フィルタリングを用いて工数見積を行い，その精度を評価している．評価の結果，協調フィルタリングが他の手法よりも高い精度で予測することが可能であることが示されている．

2.4 欠損値補完法

重回帰モデルを利用した工数予測には，モデルを構築するために使用するプロジェクトデータに欠損値が含まれる場合は，何らかの方法を用いて，欠損値のないデータセットにしなければならないという問題が存在する．

この欠損値をなくす手法に関する既存研究としては，田村晃一ら [12, 13] の研究がある．これらの研究では，欠損値のないデータセットを作成する方法として，平均値挿入法，ペアワイズ除去法，k-nn 法，CF 応用法の 4 つの欠損値補完法と無欠損データ作成法を適用することにより重回帰モデルを構築して工数予測を行い，その予測精度を評価している．評価の結果，プロジェクトのデータ件数が少ない場合には，無欠損データ作成法を，データ件数が多い場合には，類似性に基づく欠損値補完手法 (k-nn 法，CF 応用法) を用いる場合に高い精度のモデルが構築されることが示されている．

3　本研究の目的

3.1　e^3 の現状と課題

工数予測ツール 3^3 は，生方らの文献 [3] で行われた 3 つのデータセットに対する実験によって，ツールが表示する期待される精度の高い工数予測手法をその都度選択することにより，各データセットにおいて，最も精度の高い手法と同程度の精度で予測を行うことが可能であることが分かっている．

また，e^3 が開発されて以降、日本ファンクションポイントユーザ会 (JFPUG) の Web ページにて会員に公開され，実用化に向けた検討も行われている．

一方，e^3 に対する課題として以下のことが指摘されている．

- ツールの機能が不十分である．

  適用可能な工数予測手法の充実や予測結果の分析支援機能，ユーザインターフェースに改良の余地があるといった点が指摘されている．

- ツールの適用事例が少ない．

  ツールの適用が行われたのが，生方らの文献 [3] での 3 つのデータセットに対する適用のみであり，ツールの適用事例が少ないという点も指摘されている．

3.2　研究内容

そこで，本研究では，上記の e^3 の課題を解決するために，以下の二つのことを行った．

1. e^3 の機能拡張

   機能拡張として，工数予測手法の追加とユーザインターフェースの改良を行った．

2. e^3 の追加評価

   追加評価として，PROMISE[14] の 6 つのデータセットへ適用し，その結果が，生方らの文献 [3] で得られた知見と同じであるかを確認した．また，e^3 の工数予測ツールとしての性能を評価するために，某企業で使用されている商用の見積もりツール KnowledgePLAN との比較実験も行った．

4　機能拡張

4.1　概要

e^3 は有用な工数予測ツールであるが，その一方で，このツールのインターフェースに関する不便さや，登録されている手法の数が少ないといった要望も存在する．そこで，本研究では，工数予測ツール e^3 をより有用なツールにするために，これらの要望に応えた機能拡張を行う．

4.2　インターフェースの機能拡張

4.2.1　入力画面

今までの e^3 の入力画面を以下の図 3 に示す．今までの e^3 には，入力項目として，以下のものしかなかった．

- プロジェクトデータ

  現行プロジェクトと過去プロジェクトのデータを記述した CSV ファイルを指定する．

- 工数予測手法

  適用する工数予測手法をチェックボックスにより選択する．ユーザが追加実装した工数予測手法も含め，e^3 に実装されているすべての工数予測手法が一覧で表示される．

- 精度の計算頻度

  精度の計算頻度を指定する．本来は「1 プロジェクト毎」が望ましいが，数字を大きくすることで計算を省略し，実行時間を短縮することが可能である．

今回，この入力画面を拡張し，以下の図 4，図 5，図 6 のように 3 つの画面に分割した．それぞれの画面は大まかに以下の機能に分かれている．

- 入力ファイル画面

  入力ファイルを指定する画面

- フィルタリング画面

  入力されたプロジェクトデータのフィルタリングを指定する画面

- 選択画面

  工数予測手法と精度の計算頻度と欠損値補完法と使用する変数を選択する画面

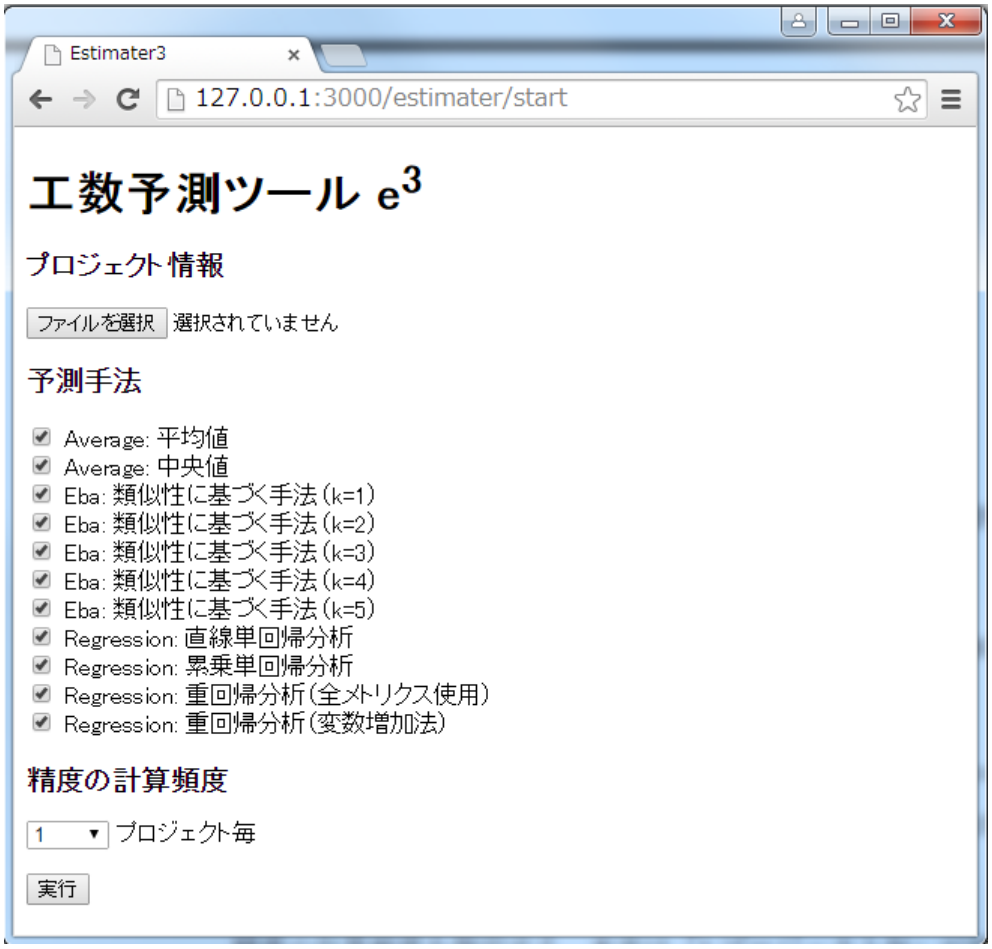
Estimater3
127.0.0.1:3000/estimater/start
工数予測ツール e3
プロジェクト情報
ファイルを選択
選択されていません
予測手法
Average: 平均値
Average: 中央値
Eba: 類似性に基づく手法(k=1)
Eba: 類似性に基づく手法(k=2)
Eba: 類似性に基づく手法(k=3)
Eba: 類似性に基づく手法(k=4)
Eba: 類似性に基づく手法(k=5)
Regression: 直線単回帰分析
Regression: 累乗単回帰分析
Regression: 重回帰分析(全メトリクス使用)
Regression: 重回帰分析(変数増加法)
精度の計算頻度
1
プロジェクト毎
実行

図 3: e^3 の入力画面

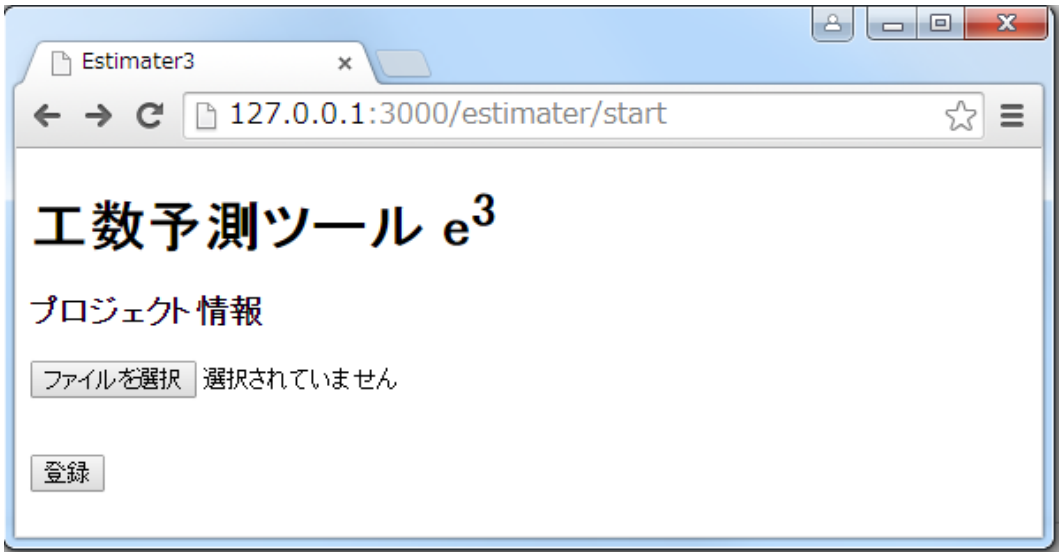

図 4: e^3 の新しい入力画面 1

4.2.2 フィルタリング機能

今までの e^3 では，プロジェクトデータの中に，実行する前に省いておきたいプロジェクトがある場合は，入力するファイルからそのプロジェクトを消さなければならなかった．そこで，今回，入力ファイルを変更せずに，プロジェクトデータから，指定した条件に反するプロジェクトを省く，フィルタリング機能を追加した．フィルタリング機能は，各メトリクスに対して，フィルタリング画面で指定した値を持たないプロジェクトを実行にかけるプロジェクトデータから取り除く．メトリクスの種類によって，指定の仕方は以下の表 1 のように分かれる．

表 1: フィルタリングの指定

順序尺度	数値の最小値と最大値を指定し，その範囲外の値を持つプロジェクトを取り除く
名義尺度	残す要素を指定し，それ以外の要素を持つプロジェクトを取り除く
その他の尺度	名義尺度と同様に扱う

また，全てのメトリクス毎にデータの欠損しているプロジェクトを取り除くように指定することが可能である．

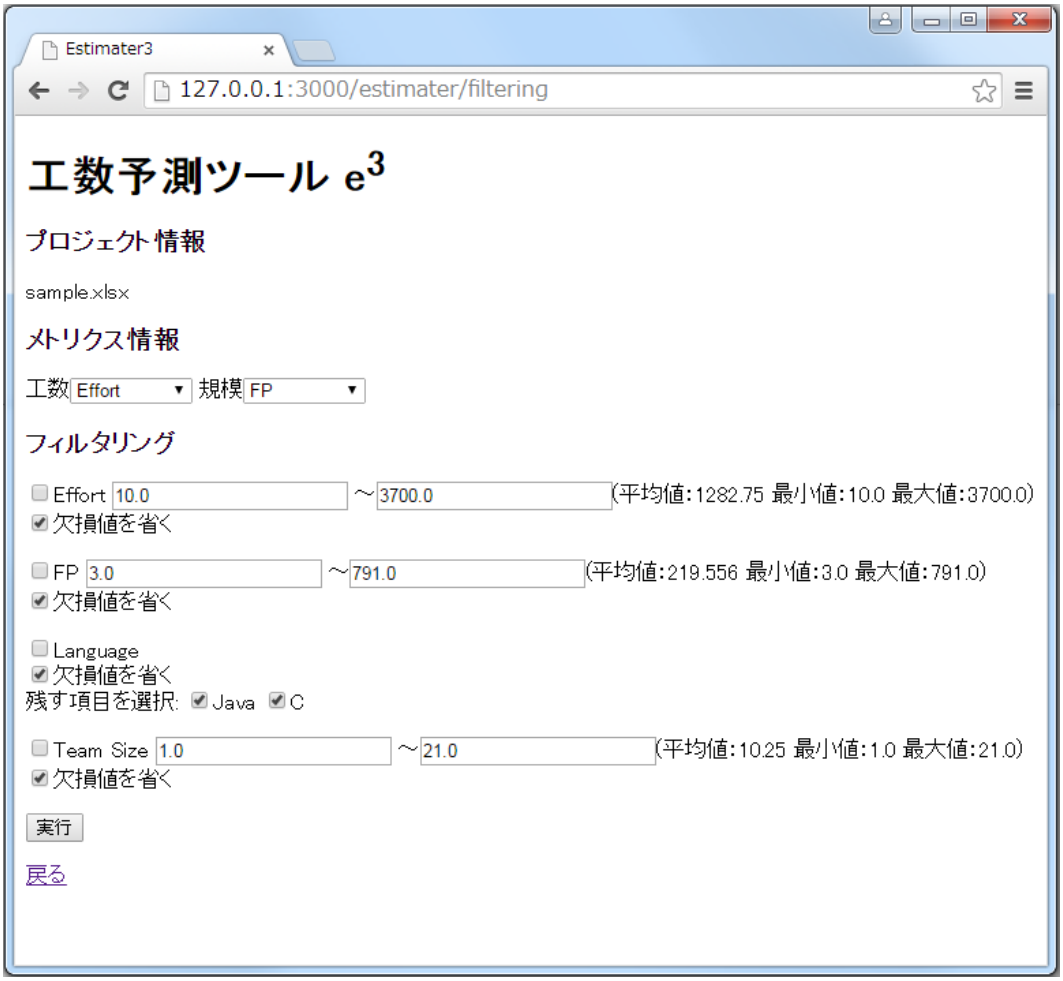

図 5: e^3 の新しい入力画面 2

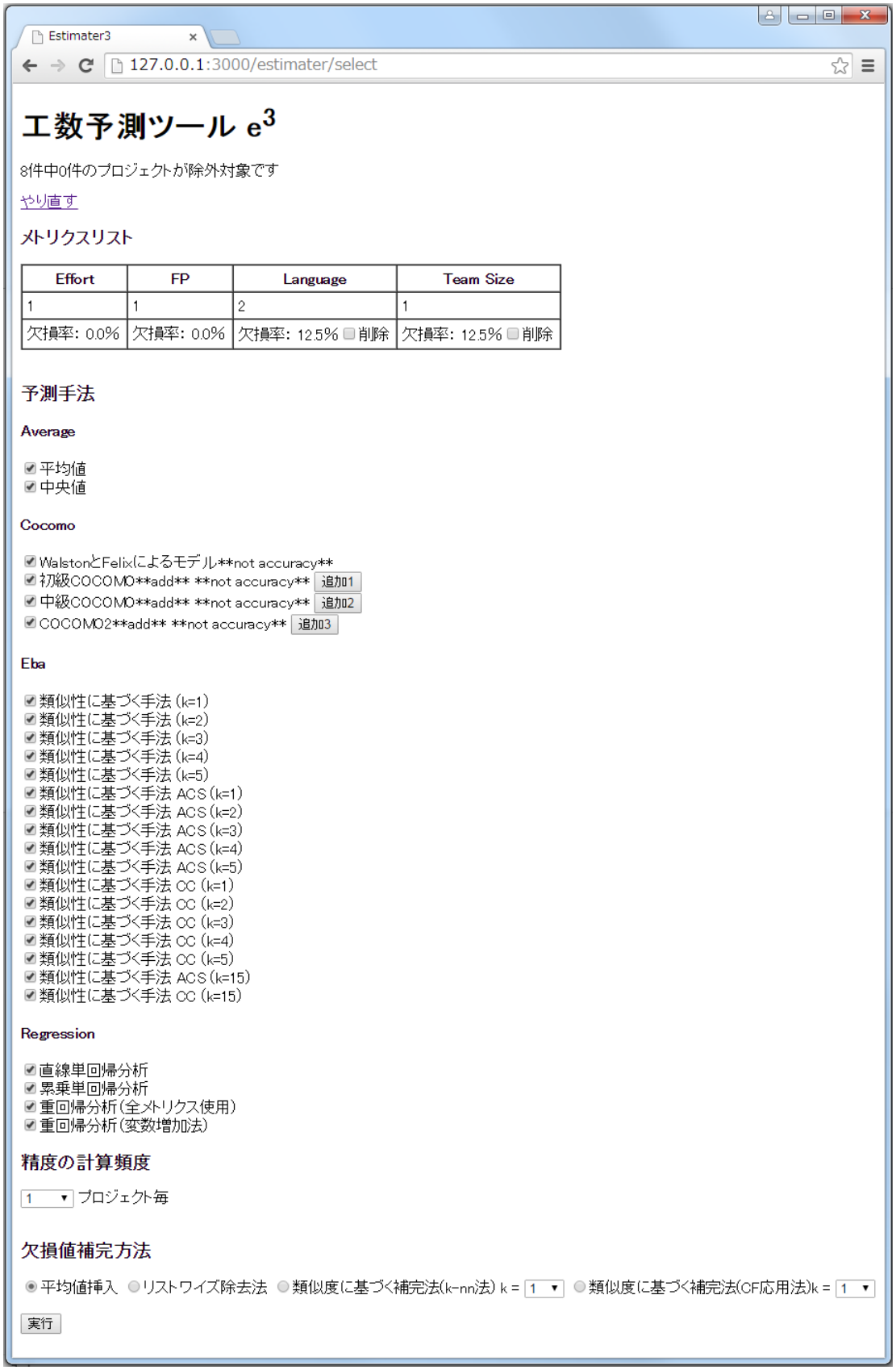

図 6: e^3 の新しい入力画面 3

4.2.3　入力ファイル形式の拡張

今までの e^3 では，入力するファイルは csv ファイルでなければならなかった．しかし，表形式のデータを作成する際には，MicrosoftOffice の Excel ファイル形式を使用することが多い．そこで，今回入力可能なファイルの形式を拡張し，csv ファイルだけでなく，Excel のファイル形式である xls ファイル，xlsx ファイル形式に対応させた．

4.2.4　メトリクスの選択機能

今までの e^3 では，不要なメトリクスを取り除く場合は，入力ファイルの第 2 行目の値を変更してその他の尺度に変更する，もしくは入力ファイルから不要なメトリクスを削除する必要があった．そこで，今回，入力ファイルを変更せずに，不要なメトリクスを取り除くためのメトリクスの選択機能を追加した．さらに，メトリクスごとにデータの欠損率を表示することにより，欠損率の高いメトリクスを取り除くことも可能である．

4.2.5　欠損値自動補完法の選択機能

今までの e^3 では，プロジェクトデータの中に欠損値が含まれている場合は自動補完が行われる．しかし，その自動補完方法は 1 つのみであり，かつ，自動補完を望まない場合は，事前に入力ファイルから欠損しているデータを取り除いておく必要があった．そこで，今回欠損値を自動補完する方法を選択する機能を追加した．選択可能な欠損値の自動補完方法は以下の 4 つである．

平均値挿入法　今までの e^3 に実装されていた方法である．メトリクスの種類によって，補完される方法が異なる．

- 比例尺度のメトリクス

  過去プロジェクトの当該メトリクスの平均値を自動補完

- 名義尺度のメトリクス

  過去プロジェクトの当該メトリクスの最頻値を自動補完

リストワイズ法　欠損値が含まれるプロジェクトを全て取り除く方法である．メトリクスの選択機能と組み合わせて，あらかじめ欠損率の高いメトリクスを取り除いておくことで，プロジェクトが少なくなりすぎないように調整することも可能である．

k-nn 法　k-nn 法は類似性に基づく手法による欠損値補完法である．欠損値に対して，類似したプロジェクトのメトリクス値を用いて欠損値を補完する．k-nn 法は以下の 3 つの手順から構成される．

- 正規化

  第 2.1 節の類似性に基づく手法で説明した正規化と同様の方法である．

- プロジェクト間の類似度計算

  各過去プロジェクトについて，現行プロジェクトとの類似度を計算する．プロジェクト p_a とプロジェクト p_b の類似度 $sim(p_a, p_b)$ は式 (6) で定義される．

$$sim(p_a, p_b) = \frac{1}{dist(p_a, p_b)} \tag{6}$$

  ここで，$dist(p_a, p_b)$ はプロジェクト p_a とプロジェクト p_b のユークリッド距離を表し，正規化されたメトリクスの値 v'_{ij} を用いて式 (7) で定義される．

$$dist\,(p_a, p_b) = \sqrt{\sum_{j \in M_a \cap M_b} \left(v'_{aj} - v'_{bj}\right)^2} \tag{7}$$

  M_a と M_b はそれぞれ，プロジェクト p_a と p_b の欠損していないメトリクスの集合を表している．

- 補完値の計算

  補完値の計算には，補完する対象のメトリクス m_b が欠損していない，プロジェクト p_a と類似度が高い上位 k 個のプロジェクトの m_b の値の平均値を補完値とする．k の値は，自動補完方法を選択する際にユーザにより指定される．

CF 応用法　CF 応用法は k-nn 法と同様に，類似したプロジェクトのメトリクス値を用いて欠損値を補完する．また，CF 応用法も以下の 3 つの手順から構成される．

- 正規化

  CF 応用法で説明した方法と同様に正規化を行う．

- プロジェクト間の類似度計算

  各過去プロジェクトについて，現行プロジェクトとの類似度を計算する．プロジェクト p_a とプロジェクト p_b の類似度 $sim(p_a, p_b)$ は式 (8) で定義される．

$$sim\left(p_a, p_b\right) = \frac{\sum_{j \in M_a \cap M_b} \left(v'_{aj} - md\left(m'_j\right)\right)\left(v'_{bj} - md\left(m'_j\right)\right)}{\sqrt{\sum_{j \in M_a \cap M_b} \left(v'_{aj} - md\left(m'_j\right)\right)^2} \sqrt{\sum_{j \in M_a \cap M_b} \left(v'_{bj} - md\left(m'_j\right)\right)^2}} \tag{8}$$

M_a と M_b はそれぞれ，プロジェクト p_a と p_b の欠損していないメトリクスの集合を表している．

- 補完値の計算

  補完値の計算には，類似したプロジェクトの対応するメトリクスの実測値を用いる．CF 応用法の類似度では，規模が異なるが傾向が似ているプロジェクト同士は類似度が高いとみなしているため，補完値の計算において，類似度 $sim(p_a, p_i)$ を重みとして，プロジェクト p_a と類似したプロジェクトのメトリクス値 v_{ib} に，プロジェクトの規模を補正する $\mathrm{amp}(p_a, p_i)$ を乗じた値で加重平均を行った値を補完値としている．よって，プロジェクト p_a のメトリクス m_b の補完値 $\hat{v}_{ab}$ は式 (9) で定義される．

$$\hat{v}_{ab} = \frac{\sum_{i \in k'} \left(v_{ib} \times amp\left(p_a, p_i\right) \times sim\left(p_a, p_i\right)\right)}{\sum_{i \in k'} \left(sim\left(p_a, p_i\right)\right)} \tag{9}$$

  ここで，k' はメトリクス m_b が欠損しておらず，かつ，プロジェクト p_a と類似度の高い上位 k 個のプロジェクトの集合を表す．k の値は，自動補完の方法を選択する際に，ユーザにより指定される．

  ここで，$amp(p_a, p_b)$ はプロジェクト p_a の規模を f_a と定義すると，式 (10) で定義される．

$$amp\left(p_a, p_b\right) = \frac{f_a}{f_b} \tag{10}$$

  $amp(p_a, p_b)$

  M_a と M_b はそれぞれ，プロジェクト p_a と p_b の欠損していないメトリクスの集合を表している．

4.2.6 ツールへの入力方法の柔軟化

今までの e^3 では，登録されている工数予測手法に与えられるデータは入力されたファイルのプロジェクトデータのみであった．そこで，今回登録されている工数予測手法毎に追加データを与える機能を追加した．これにより，さらに柔軟に工数予測手法を登録することが可能となる．

4.3 追加した手法

4.3.1 類似性に基づく手法 (調整コサイン類似性)

現在，e^3 に実装されている類似性に基づく手法は，類似度としてユークリッド距離の逆数を採用しているもののみである．そこで，今回，類似度として調整コサイン類似性 (Adjusted Cosine Similarity)[11] を追加した．以下に手法の手順を示す．

手順 1 ダミー変数化

第 2.1 節の類似性に基づく手法で示したダミー変数化と同様の方法である．

手順 2 正規化

第 4.2.5 節の k-nn 法で説明した正規化と同様の方法である．

ここまでの手順はすでに実装されている類似性に基づく手法と同様である．

手順 3 類似度計算

各過去プロジェクトについて，現行プロジェクトとの類似度を計算する．プロジェクト p_a とプロジェクト p_b の類似度 $sim(p_a, p_b)$ は式 (11) で定義される．

$$sim\left(p_a, p_b\right) = \frac{\sum_{j \in M_a \cap M_b} \{\left(v_{aj'} - \overline{m}_{j'}\right) \times \left(v_{bj'} - \overline{m}_{j'}\right)\}}{\sqrt{\sum_{j \in M_a \cap M_b} \left(v_{aj'} - \overline{m}_{j'}\right)^2} \sqrt{\sum_{j \in M_a \cap M_b} \left(v_{bj'} - \overline{m}_{j'}\right)^2}} \tag{11}$$

M_a と M_b はそれぞれ，プロジェクト p_a と p_b の欠損していないメトリクスの集合を表している．また $\overline{m}_{j'}$ はメトリクス $m_{j'}$ の正規化された特性値の平均値を表す．類似度 $sim(p_a, p_b)$ の値域は $[-1.0, 1.0]$ である．

手順 4 予測値計算

類似プロジェクトの工数の実測値をもとに，現行プロジェクトの工数の予測値を算出する．プロジェクト p_i の工数の実測値を e_i とし，プロジェクト p_c とプロジェクト p_i の類似度 $sim(p_c, p_i)$ とすると，現行プロジェクトの p_c の工数の予測値 $\hat{e}_c$ は式 (12) で定義される．

$$\hat{e}_c = \frac{\sum_{i \in k'} \left(e_i \times amplifer\left(p_a, p_i\right) \times sim\left(p_c, p_i\right)\right)}{\sum_{i \in k'} \left(sim\left(p_a, p_i\right)\right)} \tag{12}$$

ここで，k' は現行プロジェクトと類似度の高い上位 k 個のプロジェクトの集合を表す．

また，e^3 には，調整コサイン類似性を類似度として用いた類似性に基づく手法として，k の値の異なるの 6 つの新たな工数予測手法として追加した．

4.3.2 類似性に基づく手法 (相関係数)

現在，e^3 に実装されている類似性に基づく手法は，類似度としてユークリッド距離の逆数を採用しているもののみである．そこで，今回，類似度として相関係数 (Correlation Coeffcient) を採用した手法 [11] を追加した．以下に手法の手順を示す．

手順 1 ダミー変数化

第 2.1 節の類似性に基づく手法で示したダミー変数化と同様の方法である．

手順 2 正規化

第 4.2.5 節の k-nn 法で説明した正規化と同様の方法である．

ここまでの手順はすでに実装されている類似性に基づく手法と同様である．

手順 3 類似度計算

各過去プロジェクトについて，現行プロジェクトとの類似度を計算する．プロジェクト p_a とプロジェクト p_b の類似度 $sim(p_a, p_b)$ は式 (13) で定義される．

$$sim\left(p_a, p_b\right) = \frac{\sum_{j \in M_a \cap M_b} \{\left(v_{aj'} - \overline{p}_{j'}\right) \times \left(v_{bj'} - \overline{p}_{j'}\right)\}}{\sqrt{\sum_{j \in M_a \cap M_b} \left(v_{aj'} - \overline{p}_{j'}\right)^2} \sqrt{\sum_{j \in M_a \cap M_b} \left(v_{bj'} - \overline{p}_{j'}\right)^2}} \tag{13}$$

M_a と M_b はそれぞれ，プロジェクト p_a と p_b の欠損していないメトリクスの集合を表している．また $\overline{p}_{j'}$ はプロジェクト $p_{j'}$ の正規化された特性値の平均値を表す．類似度 $sim(p_a, p_b)$ の値域は $[-1.0, 1.0]$ である．

手順 4 予測値計算

第 4.3.1 節の調整コサイン類似性の手順 4 と同様の方法である．

また，e^3 には，相関係数を類似度として用いた類似性に基づく手法として，k の値の異なる 6 つの新たな工数予測手法として追加した．

4.3.3 COCOMO 法

開発に必要とする段階の数，各工程の難易度，あるいはチームの開発能力などを補正係数として掛け合わせて工数を見積もる手法である COCOMO 法を追加した．COCOMO 法はデータセットを用いた工数予測手法ではないため，データセットを用いた工数予測手法を適用することを目的とする e^3 の趣旨からは外れる．しかし，e^3 の複数の工数予測手法の結果を比較するという利用方法から，その比較の一つの基準とするために，広く一般に知られている手法である COCOMO 法を追加した．

COCOMO[15]　COCOMO は階層化されたソフトウェア推定モデルで工数や開発期間などの推定式を導出しており，その階層は以下のようになっている．

1. 初級 COCOMO

   LOC により推定されたプログラム規模の関数としてソフトウェア開発労力や開発コストを計算する静的な単一変数モデル．

2. 中級 COCOMO

   プログラム規模および開発特性 (ソフトウェア自体，ハードウェア，開発要因，プロジェクト) に対する主観的評価を反映したコスト要因の関数として開発労力を計算するモデル．

3. 上級 COCOMO

   ソフトウェア開発過程の各工程 (要求仕様定義，設計，コーディング，テスト) に与えるコスト誘因の影響の評価を中級 COCOMO に組み込んだモデル．

今回新たに e^3 に実装したのは，初級 COCOMO と中級 COCOMO のみである．

COCOMO 法では，これらの階層において，プロジェクトの開発形態を次の 3 つのモードに分類している．

1. organic モード

   比較的小規模な開発チームで，自社開発のような熟練度の高い開発形態を指し，在庫管理システムや科学技術計算用パッケージなどを開発する場合に使用するモード

2. embedded モード

   開発チームが大規模であり，種々の厳しい制約条件の下でハードウェア，ソフトウェア，操作手順などの複雑な開発環境や，要求仕様への厳しい一致性や，開発技法とテスト技法に関する規制が求められ，革新的な OS(オペレーションシステム) を開発する場合などに使用するモード

3. semi-detached モード

   organic モードと embedded モードの中間に位置する開発環境の場合に使用するモード

COCOMO の全階層に対して適用される開発工数 (人月) の見積もり値の基本式は E を開発工数，L を開発規模 (KLOC) とすると以下の式 (14) である．

$$E = a\,(L)^b \tag{14}$$

ここで，a, b は定数である．また，開発期間 (月) の見積もり値の基本式は，D を開発期間とすると，以下の式 (15) である．

$$D = c\,(E)^d \tag{15}$$

ここで，c, d は定数である．

また，上記の式の定数 a, b, c, d は COCOMO の階層と開発モードに対応して決められ，その組み合わせは表 2 のとおりである．

表 2: 開発工数と開発期間の定数の対応表

		開発モード	organic モード	semi-detached モード	embedded モード
階層	初級 COCOMO	a	2.4	3.0	3.8
		b	1.05	1.12	1.20
		c	2.5	2.5	2.5
		d	0.38	0.35	0.32
	中級 COCOMO	a	3.2	3.0	2.8
		b	1.05	1.12	1.20
		c	2.5	2.5	2.5
		d	0.38	0.35	0.32

中級 COCOMO および上級 COCOMO では表 3 で示した修正係数を掛け合わせることにより，プログラムの開発環境を考慮した開発工数の見積もり値を算出する．修正係数は 15 のコスト誘因に対するランク付けをすることにより決まる修正係数が基本となる．ランク付けは VL，L，N，H，VH，EH という 6 種類の評価レベルがあり，各コスト誘因に対する修正係数が用意されている．

COCOMOII[16, 17]　COCOMOII は 2000 年に提案された COCOMOII2000 のことであり，COCOMO を改良したモデルである．COCOMOII における工数見積もり式は以下の式 (16) である．

$$PM = 2.94 \times Size^E \times \prod EM_i + PM_{AUTO} \tag{16}$$

$$Size = (1 + REVL/100) \times (KNSLOC + KASLOC)$$

$$E = 0.91 + 0.01 \times \sum SF_j$$

表 3: コスト誘因に対する修正係数表

	ランク付け	VL	L	N	H	VH	EH
ソフトウェア属性	信頼性の要求度	0.75	0.88	1.00	1.15	1.40	
	データベースの規模		0.94	1.00	1.08	1.16	
	プロダクトの複雑性	0.70	0.85	1.00	1.15	1.30	1.65
ハードウェア属性	実行時間制約			1.00	1.11	1.30	1.66
	主記憶制約			1.00	1.06	1.21	1.56
	仮想計算機の複雑度		0.87	1.00	1.15	1.30	
	ターンアラウンドタイム		0.87	1.00	1.07	1.15	
開発要因属性	分析者能力	1.46	1.19	1.00	0.86	0.71	
	アプリケーション経験度	1.29	1.13	1.00	0.91	0.82	
	プログラミング能力	1.42	1.17	1.00	0.86	0.72	
	仮想計算機経験度	1.21	1.10	1.00	0.90		
	プログラミング言語経験度	1.14	1.07	1.00	0.95		
プロジェクト属性	近代的プログラミング手法	1.24	1.10	1.00	0.91	0.82	
	ソフトウェアツールの使用頻度	1.24	1.10	1.00	0.91	0.83	
	開発スケジュールの要求度	1.23	1.08	1.00	1.04	1.10	

式 (16) に現れる記号の意味を以下の表 4 にまとめる．

表 4: COCOMOⅡ 工数見積もり式用語集

PM	工数見積もり値 (人月)
PM_{AUTO}	ソースコードの自動生成・変換に要する工数
$REVL$	要件の変動率．要件変更により破棄されるソースコードの割合 (%)
$KNSLOC$	新規作成されるソースコード規模 (KLOC)
$KASLOC$	再利用や流用により得られるソースコード規模 (KLOC)
SF_j	規模要因値
EM_i	コスト要因値

今回新たに e^3 に実装した手法内では，あまり複雑な入力を必要としないために，$REVL$ を用いず，PM_{AUTO} や $KASLOC$ を算出せず，さらに，ソースコード規模が未調整ファンクションポイントである場合も考慮したソフトウェア規模換算係数 S_{conv} も導入した以下のような式 (17) で実装した．これにより，与えるべき入力は，プロジェクトの規模，規模要因，コスト要因，ソフトウェア規模換算係数のみとなる．

$$PM = 2.94 \times Size^E \times \prod EM_i \tag{17}$$

$$Size = S_{conv} \times S$$

$$E = 0.91 + 0.01 \times \sum SF_j$$

また，式 (17) で S とはソフトウェアの規模である．ソフトウェア規模換算係数 S_{conv} の定義はいくつかあるが，今回新たに e^3 に実装した手法で用いている定義のみを以下の表 5 に示す．また，S_{conv} は規模が KLOC である場合は「1.00」の値を取る．

表 5: ソフトウェア規模換算係数

Java	COBOL	VB.NET
0.068	0.088	0.079

COCOMOⅡ には，計画工程や要件定義工程など，プロジェクトの初期に用いるための「Early Design モデル」と基本設計工程以降で用いる「PostArchitecture モデル」の 2 つが定義されているが，今回新たに e^3 に実装したのは,「PostArchtecture モデル」のみである．よって，ここでは,「PostArchtecture モデル」のコスト要因と規模要因のみを表 6 に示す．

表 6: コスト要因と規模要因の修正係数表

規模要因	ランク付け	VL	L	N	H	VH	EH
	プロダクトの先例性	6.20	4.96	3.72	2.48	1.24	0.00
	開発の柔軟性	5.07	4.05	3.04	2.03	1.01	0.00
	アーキテクチャ／リスクの早期解決の必要性	7.07	5.65	4.24	2.83	1.41	0.00
	チーム凝集度	5.48	4.38	3.29	2.19	1.10	0.00
	プロセス成熟度	7.80	6.24	4.68	3.12	1.56	0.00
コスト要因	ランク付け	VL	L	N	H	VH	EH
プロダクト要因	信頼性の要求	0.82	0.92	1.00	1.10	1.26	
	データベースの規模		0.90	1.00	1.14	1.28	
	プロダクトの複雑性	0.73	0.87	1.00	1.17	1.34	1.74
	再利用性の要求		0.95	1.00	1.07	1.15	1.24
	文書化の要求	0.81	0.91	1.00	1.11	1.23	
プラットフォーム要因	実行時間の制約			1.00	1.11	1.29	1.63
	主記憶容量の制約			1.00	1.05	1.17	1.46
	プラットフォームのバージョン変更頻度		0.87	1.00	1.15	1.30	
要員の要因	分析者の能力	1.42	1.19	1.00	0.85	0.71	
	プログラマの能力	1.34	1.15	1.00	0.88	0.76	
	要員の継続性	1.29	1.12	1.00	0.90	0.81	
	アプリケーションの経験	1.22	1.10	1.00	0.88	0.81	
	プラットフォームの経験	1.19	1.09	1.00	0.91	0.85	
	言語およびツールの経験	1.20	1.09	1.00	0.91	0.84	
プロジェクト要因	ソフトウェアツールの使用	1.17	1.09	1.00	0.90	0.78	
	複数拠点開発	1.22	1.09	1.00	0.93	0.86	0.80
	開発期間の要求	1.43	1.14	1.00	1.00	1.00	

5　実験

5.1　実験 1:追加実験

5.1.1　概要

生方らの文献 [3] で行われた実験によって確認された，期待される精度の比較による工数予測手法の選択の有効性をより様々なデータセットにおいて確認するために追加実験を行った．実験の内容は，

- 常に同じ工数予測手法を使用し続けた場合
- ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合

とで現行プロジェクトの工数の実測値と予測値の誤差を比較する．実験に使用する工数予測手法は，生方らの文献 [3] で行われた実験に使用された e^3 に実装されていた 11 の工数予測手法に加え，今回新たに実装した，12 の類似性に基づく手法である．

5.1.2　使用したデータセット

実験には，以下の 7 つのデータセットを使用した．

- ISBSG DATA Release 11[18] ISBSG DATA release 11 は The International Software Benchmarking Standards Group（ISBSG）によって世界 24 か国から収集された 5052 件の過去プロジェクトによって構成されるデータセットである．本実験では，生方らの実験 [3] に用いたものと同じ 122 件のプロジェクトを使用した．
- Desharnais Software Cost Estimation

  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
- CM1

  NASA のデータを収集し，処理する，宇宙船の計器の開発のデータセット
- JM1

  NASA の予測を生成するために，シミュレーションを用いる，実時間地上システムの開発のデータセット

- KC1

  NASA の地上データを受信して処理するためのストレージ管理を実装しているシステムのデータセット

- KC2

  NASA の KC1 と同じプロジェクトの別の部分のデータセット

- PC1

  NASA の地球周回軌道衛星のためのフライトソフトウェアのデータセット

CM1，KC1，KC2，PC1，JM1 のデータセットは 500 件以上のデータセットであったため，実験は，精度の計算頻度を減らしたものと，ランダムに 100 件抽出したものの 2 通りの実験を行った．また，JM1 は 10000 件を超える膨大なデータセットであったため，ランダムに 100 件抽出したものでのみ実験を行った．

5.1.3 手順

工数の実測値が既知であるプロジェクトを現行プロジェクトとして e^3 を実行することにより，現行プロジェクトの工数の実測値と予測値の誤差を求めることが可能である．実験には，生方らの実験 [3] と同様に，Leave-One-Out(LOO)[6] を用いる．

計測する誤差は，実測値から見た予測値の相対誤差である MRE(Magnitude of Relative Error) と予測値から見た実測値の相対誤差である MER(Magunitude of Error Relative)[19] を用いる．MRE と MER は以下の式 (18)，式 (19) になる．

$$MRE = \frac{|\text{実測値} - \text{予測値}|}{\text{実測値}} \tag{18}$$

$$MER = \frac{|\text{実測値} - \text{予測値}|}{\text{予測値}} \tag{19}$$

また，これらの尺度による誤差を LOO によって計算した後，全過去プロジェクト分での誤差平均 MMRE と MMER を計算する．各プロジェクト i での誤差を MRE_i，MER_i とすると，MMRE と MMER は以下の式 (20)，式 (21) になる．

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i \tag{20}$$

$$MMER = \frac{1}{n}\sum_{i=1}^{n} MER_i \tag{21}$$

この MMRE と MMER を各工数予測手法ごとに求める．

5.1.4 結果

以下に，適用したそれぞれのデータセットでの結果を示す．また，best MMRE や best MMER はツールが表示する各手法の期待される精度の尺度の一つである，MMRE や MMER が最もよかった手法を用い続けた場合の予測誤差のことであり，best rank はツールが表示する各手法の期待される精度の順位が 1 番高かった手法を用い続けた場合の予測誤差のことである．[3]

ISBSG のデータセット　ISBSG のデータセットにおいて，特定の工数予測手法を使用し続けた場合いい結果を出したのは k の大きい EbA や累乗単回帰分析であった．MMRE の小さい工数予測手法は，順に EbA(k=3)(0.748)，EbA(k=5)(0.751)，累乗単回帰分析 (0.762) であり，MMER の小さい工数予測手法は，順に EbA(CC)(k=15)(0.495)，EbA(ACS)(k=15)(0.521)，EbA(CC)(k=5)(0.521) である．一方，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合も，best MMRE で MMRE が 0.869，best MMER で MMER が 0.495，best rank で MMRE が 0.890，MMER が 0.660 と，比較的いい結果が得られている．

Desharnais のデータセット　Desharnais のデータセットにおいて，特定の工数予測手法を使い続けた場合，いい結果を出したのは，重回帰分析 (変数増加法) や k の大きい EbA であった．MMRE の小さい工数予測手法は，順に重回帰分析（変数増加法）(0.508)，EbA(CC)(k=2)(0.596)，累乗回帰分析 (0.627) であり，MMER の小さい工数予測手法は，順に EbA(CC)(k=4)(0.378)，EbA(ACS)(k=5)(0.383)，EbA(CC)(k=5)(0.384) である．一方，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合も，best MMRE で MMRE が 0.508，best MMER で MMER が 0.382，best rank で MMRE が 0.654，MMER が 0.483 と，使用し続けた場合に良い結果を出した重回帰分析 (変数増加法) や EbA(CC)(k=4) と近い精度で予測が行えている．

CM1

- 精度の計算頻度を「10 件ずつ」に減らした場合

  CM1 のデータセットにおいて，特定の工数予測手法を使用し続けた場合，良い結果を出したのは重回帰分析であった．MMRE の小さい工数予測手法は，順に重回帰分析 (全メトリクス使用)(0.000278)，重回帰分析 (変数増加法)(0.0124)，EbA(CC)(k=1)(0.266) であり，MMER の小さい工数予測手法は，順に重回帰分析 (全メトリクス使用)(0.000278)，重回帰分析 (変数増加法)(0.00208)，EbA(k=3)(0.186) である．一方，ツールの表示す

る期待される精度の最も高い工数予測手法をその都度選択して使用した場合も，best MMRE で MMRE が 0.0123，best MMER で MMER が 0.00201，best rank で MMRE が 0.0123，MMER が 0.00201 と，使用し続けた場合に良い結果を出した重回帰分析と近い精度で予測が行えている．

- ランダムに 100 件抽出した場合

  CM1 のデータセットにおいて，特定の工数予測手法を使用し続けた場合，良い結果を出したのは重回帰分析 (変数増加法) であった．MMRE の小さい工数予測手法は，順に重回帰分析 (変数増加法)(0.000279)，EbA(k=5)(0.312)，EbA(k=4)(0.323) であり，MMER の小さい工数予測手法は，順に重回帰分析 (変数増加法)(0.000279)，EbA(k=5)(0.260)，EbA(k=4)(0.265) である．一方，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合も，best MMRE で MMRE が 0.000279，best MMER で MMER が 0.000279，best rank で MMRE が 0.000279，MMER が 0.000279 と，使用し続けた場合に良い結果を出した重回帰分析 (変数増加法) とほぼ同じ精度で予測が行えている．

JM1

- ランダムに 100 件抽出した場合

  JM1 のデータセットにおいて，特定の工数予測手法を使用し続けた場合，良い結果を出したのは重回帰分析であった．MMRE の小さい工数予測手法は，順に重回帰分析 (全メトリクス使用)(0.0000341)，重回帰分析 (変数増加法)(0.000186)，EbA(CC)(k=1)(0.603) であり，MMER の小さい工数予測手法は，順に重回帰分析 (全メトリクス使用)(0.0000341)，重回帰分析 (変数増加法)(0.000185)，EbA(ACS)(k=5)(0.354) である．一方，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合も，best MMRE で MMRE が 0.0000341，best MMER で MMER が 0.0000341，best rank で MMRE が 0.0000341，MMER が 0.0000341 と，使用し続けた場合に良い結果を出した重回帰分析 (全メトリクス使用) とほぼ同じ精度で予測が行えている．

KC1

- 精度の計算頻度を「200 件ずつ」に減らした場合

  KC1 のデータセットにおいて，特定の工数予測手法を使用し続けた場合，良い結果を出したのは重回帰分析であった．MMRE の小さい工数予測手法は，順に重回帰分析 (変数増加法)(0.00237)，重回帰分析 (全メトリクス使用)(0.00346)，EbA(k=3)(0.0882) であ

り，MMER の小さい工数予測手法は，順に重回帰分析 (全メトリクス使用)(0.00226)，重回帰分析 (変数増加法)(0.00234)，EbA(k=5)(0.0844) である．一方，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合も，best MMRE で MMRE が 0.00347，best MMER で MMER が 0.00227，best rank で MMRE が 0.00347，MMER が 0.00227 と，使用し続けた場合に良い結果を出した重回帰分析 (変数増加法) とほぼ同じ精度で予測が行えている．

- ランダムに 100 件抽出した場合

  KC1 のデータセットにおいて，特定の工数予測手法を使用し続けた場合，良い結果を出したのは重回帰分析 (変数増加法) であった．MMRE の小さい工数予測手法は，順に重回帰分析 (変数増加法)(0.00150)，EbA(k=1)(0.268)，EbA(ACS)(k=2)(0.278) であり，MMER の小さい工数予測手法は，順に重回帰分析 (変数増加法)(0.00151)，EbA(ACS)(k=2)(0.296)，EbA(k=5)(0.309) である．一方，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合も，best MMRE で MMRE が 0.00150，best MMER で MMER が 0.00151，best rank で MMRE が 0.00150，MMER が 0.00151 と，使用し続けた場合に良い結果を出した重回帰分析 (変数増加法) とほぼ同じ精度で予測が行えている．

KC2

- 精度の計算頻度を「10 件ずつ」に減らした場合

  KC2 のデータセットにおいて，特定の工数予測手法を使用し続けた場合，良い結果を出したのは k の小さい EbA であった．MMRE の小さい工数予測手法は，順に EbA(k=3)(0.209)，EbA(k=2)(0.218)，EbA(ACS)(k=2)(0.225) であり，MMER の小さい工数予測手法は，順に EbA(k=3)(0.192)，EbA(k=5)(0.196)，EbA(k=4)(0.197) である．一方，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合も，best MMRE で MMRE が 0.211，best MMER で MMER が 0.202，best rank で MMRE が 0.237，MMER が 0.202 と，使用し続けた場合に良い結果を出した EbA(k = 3) とほぼ同じ精度で予測が行えている．

- ランダムに 100 件抽出した場合

  KC2 のデータセットにおいて，特定の工数予測手法を使用し続けた場合，良い結果を出したのは EbA であった．MMRE の小さい工数予測手法は，順に EbA(CC)(k=3)(0.506)，EbA(CC)(k=4)(0.509)，EbA(CC)(k=5)(0.513) であり，MMER の小さい工数予測手法は，順に EbA(ACS)(k=4)(0.284)，EbA(ACS)(k=1)(0.295)，EbA(ACS)(k=3)(0.299)

である．一方，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合，best MMRE で MMRE が 13.4，best MMER で MMER が 0.337，best rank で MMRE が 13.4，MMER が 0.286 となっており，MMRE の指標においては，使用し続けた場合に良い結果を出した EbA(CC)(k=3) とかなり精度に差があった．

PC1

- 精度の計算頻度を「100 件ずつ」に減らした場合

  PC1 のデータセットにおいて，特定の工数予測手法を使用し続けた場合，良い結果を出したのは重回帰分析であった．MMRE の小さい工数予測手法は，順に重回帰分析 (全メトリクス使用)(0.000455)，EbA(CC)(k=5)(0.227)，EbA(CC)(k=4)(0.227) であり，MMER の小さい工数予測手法は，順に重回帰分析 (変数増加法)(0.000428)，重回帰分析 (全メトリクス使用)(0.00114)，EbA(k=3)(0.174) である．一方，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合，best MMRE で MMRE が 0.00741，best MMER で MMER が 0.000429，best rank で MMRE が 0.000456，MMER が 0.000429 と，使用し続けた場合に良い結果を出した重回帰分析とほぼ同じ精度で予測が行えている．

- ランダムに 100 件抽出した場合

  PC1 のデータセットにおいて，特定の工数予測手法を使用し続けた場合，良い結果を出したのは重回帰分析 (変数増加法) であった．MMRE の小さい工数予測手法は，順に重回帰分析 (変数増加法)(0.000305)，EbA(k=5)(0.386)，EbA(k=4)(0.396) であり，MMER の小さい工数予測手法は，順に重回帰分析 (変数増加法)(0.000302)，EbA(ACS)(k=2)(0.396)，EbA(ACS)(k=5)(0.406) である．一方，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合，best MMRE で MMRE が 0.000305，best MMER で MMER が 0.000302，best rank で MMRE が 0.000305，MMER が 0.000302 と，使用し続けた場合に良い結果を出した重回帰分析 (変数増加法) とほぼ同じ精度で予測が行えている．

5.1.5 考察

実験の結果, データセットごとに有効な工数予測手法は異なり，ツールの表示する期待される精度をもとに工数予測手法を選択する方法で，それぞれのデータセットで有効な工数予測手法と同程度の誤差で予測が行えることが分かった．

ISBSG のデータセットに対しては，生方らの文献 [3] で得られた結果と同様に k の値が大きい EbA が有効であった．また，生方らの文献 [3] で得られた結果と比べて，MMER の指標において，予測精度が向上していた．このことから，今回機能拡張として，新たに手法を追加したことは有益であったと考えられる．

また，Desharnais のデータセットは，k の大きい EbA や重回帰分析 (変数増加法) などが有効であった．Desharnais のデータセットは，特定の企業のみから集めたようなデータセットではなく，複数の企業から集められたデータセットであるため，過去の知見 [11] に基づけば，EbA が有効に働くと考えられる．しかし，このデータセットは，データの収集された場所，年代などが固定されているため，工数を式で表現しやすく，重回帰分析 (変数増加法) も有効に働いたのではないかと考えられる．

それ以外のデータセットについては，NASA の同一の開発プロジェクト内で収集されたデータセットであるので，工数を式で表現しやすく，ほぼ全ての NASA のデータセットにおいて，重回帰分析が有効に働いたのではないかと考えられる．

次に，NASA のデータセットにおいて，唯一例外的に，EbA が重回帰分析よりも有効に働いた KC2 のデータセットについて考察する．このデータセットでは，EbA の方が重回帰分析よりも有効に働いてはいるが，EbA の予測精度自体は他の NASA のデータセットでの予測精度とほぼ同じ程度の精度であった．ゆえに，EbA がこのデータセットに対して有効に働いたというよりも，このデータセットに対しては重回帰分析が有効に働かなかったということが考えられる．また，重回帰分析が有効に働かなかった原因として考えられるのは，KC2 は NASA の同一の開発プロジェクト内で収集されたデータセットであるので，本来工数を式で表現しやすいはずであるが，一部，式で表現しきれない特異なプロジェクトが含まれていたため，重回帰分析があまり有効に働かなかったという可能性がある．

次に，KC2 のデータセットからランダムに 100 件抽出したデータセットにツールを適用した場合に，ツールの表示する期待される精度の最も高い手法をその都度選択して使用した場合の精度と，使用し続けた場合に良い結果を出した手法の精度に差があった原因について考察する．ツールの表示する期待される精度の最も高い手法をその都度選択して使用した場合の精度が悪くなった原因は，ツールが表示する期待される精度の最も高い手法が，実際には予測値が大きく間違っていた手法である重回帰分析になってしまっているプロジェクトが存在したためであった．このようなことが起きてしまった理由としては，この問題のプロジェクトが KC2 のデータセットにおいて特異なプロジェクトであったということが考えられる．そのため，このプロジェクトを現行プロジェクトとして実行した際に，このプロジェクト以外のデータセットからツールが算出した期待される精度が最も高い手法が重回帰分析になってしまったという可能性がある．

このような現象が，KC2 のデータセットからランダムに 100 件抽出したデータセットに

ツールを適用した場合にのみ発生した原因としては，500 件近くのデータセットから 100 件のみを抽出したため，この特異なプロジェクトの影響が大きくなってしまったということが考えられる．実際，この特異なプロジェクトを除いてツールを適用すると，MMRE の小さい工数予測手法は，順に重回帰分析 (全メトリクス使用)(0.212)，EbA(CC)（k = 3)(0.471)，EbA(CC)(k = 4)(0.473) であり，MMER の小さい工数予測手法は，順に重回帰分析 (全メトリクス使用)(0.0104)，EbA(ACS)(k=4)(0.280)，EbA(ACS)(k=1)(0.290) である一方，ツールの表示する期待される精度の最も高い工数予測手法をその都度選択して使用した場合，best MMRE で MMRE が 0.213，best MMER で MMER が 0.0125，best rank で MMRE が 0.213，MMER が 0.0131 と，使用し続けた場合に良い結果を出した重回帰分析 (全メトリクス使用) とほぼ同じ精度で予測が行えている．

5.2 実験 2:KnowledgePLAN との比較

5.2.1 概要

e^3 の工数予測ツールとしての性能を評価するために，商用見積りツール KnowledgePLAN との性能比較実験を行った．実験には以下の予測値を用いて，某企業のプロジェクトデータの実測値との誤差の平均を比較する．

- e^3 の表示する期待される精度の最も高い手法の予測値
- KnowledgePLAN を用いた予測値

但し，KnowledgePLAN を用いた予測値は，某企業の使用していた方法で予測したものを用いる．また，e^3 に実装されている手法の内，全メトリクスを用いた重回帰分析については，e^3 に実装されている方法ではこのデータセットに対して，実行することが出来なかったので，それ以外の手法を用いた．

5.2.2 結果

比較実験の結果は以下の表 7 のようになった．

e^3 よりも KnowledgePLAN の方が，MMRE，MMER のどちらの尺度においても，また，データ件数の量に関わらず，性能がいいという結果であった．

5.2.3 考察

実験の結果，KnowledgePLAN よりも e^3 の方が性能が悪かった原因の 1 つとして考えられるのは，予測に使用するプロジェクトデータの数である．KnowledgePLAN は 1 万件を超

表 7: 実験 2：KnowledgePLAN との比較結果

MMRE			
データ件数	KnowledgePLAN	Best MMRE	Best Rank
42	0.226	0.379	0.300
37	0.228	0.297	0.383
32	0.257	0.480	0.439
27	0.216	0.457	0.427
22	0.169	0.460	0.593

MMER			
データ件数	KnowledgePLAN	Best MMER	Best Rank
42	0.267	0.324	0.310
37	0.239	0.314	0.316
32	0.271	0.406	0.379
27	0.273	0.397	0.434
22	0.282	0.450	0.524

える実績データに基づいて予測を行う一方，e^3 は某企業の 42 件のプロジェクトデータのみを用いて予測を行っている．また，某企業のプロジェクトデータを 37 件，32 件，27 件，22 件に減らして実行した場合に，件数が多くなるほど，e^3 の見積もりの精度は向上していった．これらのことから，実績データが蓄積されることで e^3 の予測精度が向上する可能性が考えられる．

また，実験に使用した某企業のデータセットにおいて，いくつかのプロジェクトについては，e^3 の方が，誤差が小さいプロジェクトが存在し，KnowledgePLAN が過小評価しているもので e^3 の MER が低くなっているものと，KnowledgePLAN が過大評価しているもので，e^3 の MRE が低くなっているものがあった．また，それぞれのプロジェクトにおいて，e^3 の表示する最も精度の高い手法は類似性に基づく手法であった．今回使用した 42 個のプロジェクトデータはドメインが限定されているので，生方らの文献 [3] の知見に基づけば，回帰分析などが有効に働くとされている．ゆえに，KnowledgePLAN の 1 万件を超える実績データに基づく予測が有効に働いていると考えられる．しかし，プロジェクトデータの中に，ドメインが限定されているものの一部特異なプロジェクトが少数存在し，それらについては e^3 の類似性に基づくモデルが有効に働いたという可能性が考えられる．

6　あとがき

本研究では，工数予測ツール e^3 の機能拡張として，e^3 に工数予測手法を追加し，インターフェースの改善を行った．また，追加実験として，6 つのデータセットにツールを適用し，生方らの実験 [3] で得られた，ツールの表示する期待される精度を基に使用する工数予測手法を選択することで，データセットごとに最も有効な工数予測手法を使用し続けた場合と同程度の精度で予測が行えることが確認できた．また，e^3 と KnowledgePLAN との比較実験を行い，e^3 の工数予測ツールとしての性能を評価した．その結果としては，現状では，KnowledgePLAN の方が，工数予測ツールとしての性能がよいが，ツールを適用したある企業の 42 のデータセットの内の一部のプロジェクトに対しては，e^3 の方が性能がよいという結果が得られた．本研究の今後の課題は以下のとおりである．

- より多くのデータへの適用を通じた有効性の評価
- 予測結果の分析支援機能などのツールの更なる改良

謝辞

本研究の全過程を通し，理解あるご指導を賜り，的確なご助言を頂きました 楠本 真二 教授に心より感謝申し上げます.

本研究を行うにあたり，終始丁寧かつ適切なご指導を頂きました 岡野 浩三 准教授に深く感謝申し上げます.

本研究を行うにあたり，的確なご指導及びご助言を頂きました 井垣 宏 特任准教授に心より感謝申し上げます.

本研究を行うにあたり，多大なるご助言を頂きました 肥後 芳樹 助教に心より感謝申し上げます.

本研究を進めるにあたり，適切なご助言及び多大なるご助力いただきました 大阪大学大学院情報科学研究科コンピュータサイエンス専攻博士前期課程 1 年の 江川 翔太 氏に心より感謝申し上げます.

本研究を進めるにあたり，様々な形で励まし，ご助言を頂きました その他の楠本研究室の皆様のご協力に心より感謝致します.

本研究を進めるにあたって，様々な点でご協力頂いた　鈴鹿 久佳 氏に深謝致します.

最後に，本研究に至るまでに，講義，演習，実験等でお世話になりました大阪大学基礎工学部情報科学科の諸先生方に，この場を借りて心から御礼申し上げます.

参考文献

[1] 角田雅照大杉直樹, 門田暁人, 松本健一, 佐藤慎一. 協調フィルタリングを用いたソフトウェア開発工数予測方法. 情報処理学会論文誌, Vol. 46, No. 5, pp. 1155–1164, May 2005.

[2] 唐沢正和. ソフトウェア開発の高精度な見積もり･計画策定を支援する「knowledgeplan」. http://techtarget.itmedia.co.jp/tt/news/1005/19/news02.html.

[3] 生方克馬, 柿元健, 楠本真二. 期待される精度の比較による適切な工数予測手法の判別を支援するツール. 電子情報通信学会論文誌 D, Vol. J95-D, No. 4, pp. 885–894, 2012.

[4] 生方克馬, 柿元健, 楠本真二. 複数の手法による予測結果が比較可能な工数予測ツールの開発と評価. 電子情報通信学会技術研究報告, Vol. 110, No. 336, pp. 7–12, Dec 2010.

[5] 生方克馬, 柿元健, 楠本真二. 複数の手法による予測結果が比較可能な工数予測ツールの開発. 2010 年電子情報通信学会総合大会講演論文集, 情報・システム講演論文集 1, No. D-3-2, p. 19, Dec 2010.

[6] A.De Lucia, E.Pompella, and S.Stefanucci. Effort estimation for corrective software maintenance. In *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering*, pp. 15–19, Italy, Jul 2002.

[7] 黒田努, 佐藤和人. 改訂新版基礎 Ruby on Rails. 土田米一, 2012.

[8] arton. かんたん Ruby on Rails で Web アプリケーション. 佐々木幹夫, 2006.

[9] 中村哲彬, 柿本健, 楠本真二. 類似性に基づく工数予測における予測回避の効果. ソフトウェア信頼性研究会第 6 回ワークショップ論文集, pp. 37–44, Mar 2010.

[10] B.Sarwar, G.Karypis, J.Konstan, J.Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295, Hong Kong, May 2001.

[11] 大杉直樹, 角田雅照, 門田暁人, 松村知子, 松本健一, 菊池奈穂美. 企業横断的収集データに基づくソフトウェア開発プロジェクトの工数見積もり. *SEC journal*, Vol. 5, No. 1, pp. 16–25, 2006.

[12] 田村晃一, 柿元健, 戸田航史, 角田雅照, 門田暁人, 松本健一, 大杉直樹. 工数予測における類似性に基づく欠損値補完法の実験的評価. コンピュータソフトウェア, Vol. 26, No. 3, pp. 44–55, 2009.

[13] 田村晃一, 門田暁人, 松本健一. 欠損値処理法を用いた工数予測におけるデータ件数と予測精度の関係. コンピュータソフトウェア, Vol. 27, No. 2, pp. 100–105, 2010.

[14] PROMISE software engineering repository. http://promise.site.uottawa.ca/SERepository/.

[15] B.W.Boehm. *Software Engineering Economics.* Prentice Hall, 1981.

[16] B.W.Boehm, C.Abts, A.W.Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. J. Reifer, and B. Steece. *Software Cost Estimation with Cocomo II.* Prentice Hall, 2000.

[17] 松本健一, 大岩佐和子, 押野智樹. COCOMOⅡによる工数見積もり「経済調査会ソフトウェア開発データリポジトリ」を用いた検証. 経済調査研究レビュー, Vol. 14, No. 4, pp. 88–98, 3 2014.

[18] ISBSG estimating, benchmarking and research suite release 11., 2009. International Software Benchmarking Standards Group (ISBSG).

[19] T.Foss, E.Stensrud, B.Kitchenham, and I.Myrtveit. A simulation study of the model evaluation criterion mmre. *IEEE Transactions on Software Engineering*, Vol. 29, No. 11, pp. 985–995, Nov 2003.