コスト見積り誤差評価の統計的仮説検定を用いた考察

水野修 †，菊野 亨 †，稲垣 勝巳 ‡，高木 徳生 ‡，坂本 啓司 ‡
† 大阪大学 大学院基礎工学研究科 情報数理系専攻
〒 560-8531 大阪府豊中市待兼山町 1-3
Tel: 06-850-6567 Fax: 06-850-6569
E-mail: o-mizuno@ics.es.osaka-u.ac.jp
‡ オムロン株式会社

要旨

本研究では，プロジェクトデータの分析に採用するコスト予測精度に関する分類基準と，その分類で使用するしきい値について統計的仮説検定を用いて考察する．

まず，分類基準に関しては，コストの見積り誤差がマイナス方向に大きくふれたプロジェクト (例えば，見積りよりも 20%少ないコストで終了したプロジェクト) を成功プロジェクトと混乱プロジェクトのいずれとみるかの分類基準について議論する．統計的仮説検定の結果，そうしたプロジェクトを混乱プロジェクトとすることの妥当性が確認できた．

次に，分類のしきい値については，幾つかの代表的なしきい値について実際にクラス分けをして，統計的仮説検定を行った．その分析の結果，±10% をしきい値として成功プロジェクトと混乱プロジェクトに分けるという開発現場で利用されている基準値の妥当性が確認できた．

なお，これらの妥当性の確認はあくまでも今回の分析の対象としたプロジェクトにおいてできたことであって，現時点では必ずしも一般化できる状況にはない．

1 研究の背景

ソフトウェア開発における品質と生産性向上の重要性が指摘されている．もっとも有望な取り組みとしてプロセス改善 [2] が注目され，多くの試みが報告されてきている [3][7][10][11].

プロセス改善をより効果的に実現するためには適切な開発計画の作成とその実施が必要となる [1][14]．もし不適切な計画を与えられり，計画は適切であるがその履行に問題があると，せっかくの改善計画も計画倒れとなり実行できないために，結局はプロセス改善に失敗すると考えられる．

開発現場にとって開発計画の作成と履行の適切さを判断する直観的で，かつ，受け入れやすい 1 つの基準は開発コストが見積り通りであったかどうかである [15]. 一般的に開発コストは働いた人間の延べ人数に近似的に等しいと考えられるので，開発期間 (作業時間) や作業者の技術レベルなども間接的に含めて評価されていると考えてもよい．

我々はこれまでに次の 3 つの命題 (P1), (P2), (P3) が成り立つことをオムロン (株) ソーシアル事業グループにおいて実施した 31 件のプロジェクトの開発データを用いて実証した [5][8].

(P1) 開発計画が標準手順に準拠している度合が高いほどその計画のコスト予測精度も高い．

(P2) コスト予測精度の高い開発計画の下で実施されたプロジェクトにおいてはフィールド品質も高い．

(P3) コスト予測精度の高い開発計画の下で実施されたプロジェクトにおいてはそのチーム生産性が高い．

2 研究の目的

本研究では，上述の 3 つの命題 (P1)(P2)(P3) に関連して，開発プロジェクトを「成功プロジェクト」と「混乱プロジェクト」に分類する基準について統計的仮説検定の手法を用いて考察する．まず開発現場で現在採用されているコスト予測精度から見たクラス分類の基準の妥当性について考察する．次に，そのクラス分類の基準となっているコスト見積りの誤差に対するしきい値の選び方の妥当性について考察する．

(A1) 開発現場で採用されている分類基準

開発現場では,「開発コストの誤差がマイナス方向に大きくふれたプロジェクトはやはり混乱 (失敗) プロジェクトであるとみなす」ことがプロジェクトの妥当な分類基準だとされている．

これに関して，本報告では次のことを示す．

もし「誤差がマイナス方向にふれた全てのプロジェクトを成功プロジェクトとみなす」ようにすると，成功プロジェクトと混乱プロジェクトの間に品質，生産性の面から有意的な差がなくなってしまう．

(A2) 開発現場で採用されているしきい値

開発現場では，成功プロジェクトと失敗プロジェクトを分ける基準として,「誤差が ±10% 以上になると混乱プロジェクトであるとみなす」が採用されている．

これに関して，本報告では次のことを示す．

しきい値を ±10% 以外の値 (例えば，±5%, ±15%) に設定してしまうと，成功と混乱の間に品質，生産性の面から有意的な差がなくなってしまう．

3 メトリクスの定義

3.1 開発プロセスモデル

オムロン (株) ソーシアル事業グループでは，ATM(Automated Teller Machines)，POS(Point Of Sales) 端末，券売機などの多くのソフトウェア組込み型機器を製造している．このような組み込み型ソフトウェアの開発は図 1に示す開発プロセスに従って行われる．

開発プロセスは設計フェーズとディバッグフェーズに分けられる．設計フェーズの 1 つの特徴は各設計作業工程，コーディング作業工程の後に必ずレビュー作業工程が配置されていることである．一方，ディバッグフェーズの特徴は単体，統合，機能，検証の 4 種類の異なる目的を持ったテスト，ディバッグが繰り返されていることである．

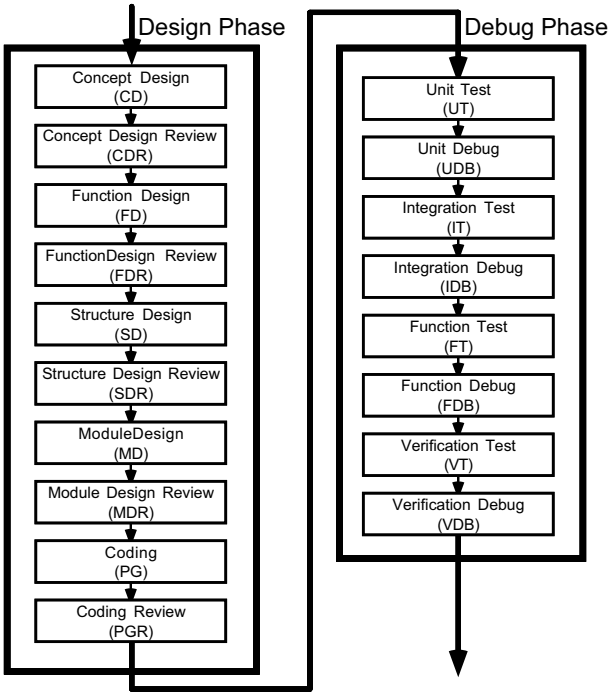

図 1: 開発プロセスモデル

3.2 ソフトウェア開発に関するメトリクス

本研究ではソフトウェア開発を品質と生産性の両面から分析，評価するため次のメトリクス [1][9][14] を用いる．

1. 開発規模 SLC

   再利用分も含めて設計，コーディングしたソースコードの行数を表す．但し，コメント行は全て除いて集計する．なお，再利用分についてはその修正の度合に応じて行数を計算している．

   以降では次の記号を用いる．

   SLC ··· 開発規模 (単位は Kstep である)．

   $newSLC$ ··· 新規開発した部分のソースコードの行数．

   $slgSLC$ ··· 若干の修正を行って再利用した部分のソースコードの行数．

   $extSLC$ ··· 大幅な修正を行った上で再利用したソースコードの行数．

   α, β ··· 補正係数．

   このとき開発規模 SLC は次式で定義される．

   $$SLC = newSLC + \alpha \times slgSLC + \beta \times extSLC$$

2. フィールド品質 FQ

   ソフトウェア出荷後の一定期間 (6ヵ月) 以内に発見されるフォールトの件数をその開発規模で正規化したもので，作成されたソフトウェアプロダクトの品質を測る．

   以降では次の記号を用いる．

   FQ ··· ソフトウェアのフィールド品質 (単位は件/Kstep である)．

   FD ··· 出荷後の 6 ヵ月間に発見されたフォールトの総数．

   このときフィールド品質 FQ は次式で定義される．

   $$FQ = \frac{FD}{SLC}$$

3. チーム生産性 TP

   プロジェクトに参加した開発チームの開発者全員の生産性の平均値を測る．具体的には，その開発チームで作成した全てのソフトウェアの開発規模をその開発に要した延べ工数で割って求める．

   以降では次の記号を用いる．

   TP ··· チーム生産性 (単位は Kstep/人月 である)．

$EFT \cdots$ 開発に要した人月の総和.

このときチーム生産性 TP は次式で定義される.

$$TP = \frac{SLC}{EFT}$$

3.3 開発計画に関するメトリクス

コスト予測精度 RE

開発計画の適切さ，あるいは，妥当さを直観的に測るためのもので，計画段階で想定した開発に要するコストと実際にかかったコストの差を評価する．ここでは，コストの単位は人月とする.

以降では，次の記号を用いる.

$RE \cdots$ 開発コストの見積りと実績の差を見積りで正規化したもの (単位は % とする).

$actCOST \cdots$ プロジェクトで実際にかかったコスト (単位は人月とする).

$estCOST \cdots$ 開発コストの見積りの値 (単位は人月とする).

このときコスト予測精度 RE は次式で定義される.

$$RE = \frac{actCOST - estCOST}{estCOST} \times 100$$

3.4 プロジェクトの成功と混乱

ここでは，成功プロジェクトと混乱プロジェクトをそれぞれ次のように定義する.

– 成功プロジェクト

成功プロジェクトとは，品質が高く生産性も同時に高いプロジェクトのことである．すなわち，品質 FQ の値が小さく，生産性 TP の値は大きいプロジェクトを「成功プロジェクト」と呼ぶ.

– 混乱プロジェクト

混乱プロジェクトとは，品質と生産性がともに低いプロジェクトのことである．すなわち，品質 FQ の値が大きく，生産性 TP の値は小さいプロジェクトを「混乱プロジェクト」と呼ぶ.

4 開発データの特性

1節や 3.1節の開発プロセスでも説明したように，本研究で分析の対象とするプロジェクトはいずれもオムロン (株) で行っている組込みソフトウェアの開発である.

詳細は省略するが，これらの組込みソフトウェアは非常に複雑な機能を実現している．更に，出荷後に発見されたフォールトを修正するには高い費用を要するため，開発チームに対しては特に高い信頼性が要求されることになる.

ソフトウェアの開発はシステムハードウェアの開発と並行して行われるため，全体的なプロジェクトの管理が必要となる．一般にこの種の開発においては，製品の仕様はハードウェア設計の制約を強く受けることになる.

オムロン (株) の場合には，画面構成，帳票の構成，処理の流れ，処理速度などについては顧客からの要求仕様の変更が発生することもある．しかし，ハードウェアとのインタフェース部分や OS(Operating System) などは社内で決定できるために，開発途中で変化することは少ない [7].

今回の分析で対象とした 31 件のプロジェクトは次の 3 種類に分類される.

(1) 銀行関連の 6 プロジェクト $\cdots$ ATM，振込，記帳など

$$PB_1, PB_2, \cdots, PB_6$$

(2) 鉄道関連の 24 プロジェクト $\cdots$ 自動改札，券売機，清算機など

$$PR_1, PR_2, \cdots, PR_{24}$$

(3) 流通関連の 1 プロジェクト $\cdots$ POS ターミナルなど

$$PS_1$$

分析に用いた各プロジェクトに対しては，開発規模 SLC，出荷後 6ヵ月間に発見されたフォールトの総数 FD，開発に要した人月の総和 EFT，開発コストの見積り $estCOST$ とその実績 $actCOST$ などのメトリクスの値が収集されている.

これらのメトリクスの値そのものは企業の秘密に属するものなので，残念ながらここでは公表できない.

5 分析

この節では，2節で説明した 2 つの項目 (A1)，(A2) の順に分析して行く.

5.1 記法

以降では次の記法を用いる.

まず，コスト予測精度のデータだけに基づく判断で成功したと見なされるプロジェクトの集合を成功プロジェクトのクラスと呼び，C_S と表す.

成功プロジェクトのクラス： C_S

このクラスに属するプロジェクトの品質，生産性の平均値を以下のように表す.

品質の平均値：μ_S^{FQ}

生産性の平均値：μ_S^{TP}

注 1 この成功プロジェクトのクラス C_S に属するプロジェクトの品質，生産性は共に高いことが期待される．

混乱プロジェクトのクラス：C_C

また，このクラスに属するプロジェクトの品質，生産性の平均値を以下のように表す．

品質の平均値：μ_C^{FQ}

生産性の平均値：μ_C^{TP}

注 2 この混乱プロジェクトのクラス C_C に属するプロジェクトの品質，生産性は共に低いことが期待されている．

5.2 分類基準の分析 (A1)

成功プロジェクトと混乱プロジェクトを区別する基準として，次のような考えが受け入れられている．

あるプロジェクトの実際にかかったコストが見積もりに対してマイナスの方向に大きくずれた (つまり，予定していたコストよりも相当に安いコストでプロジェクトが完了した) 場合，それを混乱プロジェクトと見なす．その主な理由は，コストが見積もりから大きく外れたプロジェクトでは，何らかの混乱が内部でおきている恐れがあるからである．

この考え方でプロジェクトのクラス分けを行うものをクラス分け CL_{sym} と呼ぶ．今，ここでは RE が $\pm 10\%$ 以内に抑えられていると成功プロジェクト，そうでないと混乱プロジェクトと定める．

クラス分け CL_{sym}

C_S ：$-10\% \leq RE < +10\%$

C_C ：$RE \geq +10\%$, $RE < -10\%$

一方，プロジェクトにかかったコストの大小だけを問題にするクラス分けも考えられる．つまり予定していたコストを大きく上回らない限り (従って，安いコストで終了したものも)，終了したプロジェクトはすべて成功プロジェクトであるとする．今ここでは RE が $+10\%$ より小さいものをすべて成功プロジェクト，そうでない (つまり，RE が $+10\%$ を超える) ものだけが混乱プロジェクトであると定める．

この考え方でプロジェクトのクラス分けを行ったものをクラス分け CL_{asym} と呼ぶ．

クラス分け CL_{asym}

C_S ：$RE < +10\%$

C_C ：$RE \geq +10\%$

この 2 つのクラス分けの概念図を図 2に示す．

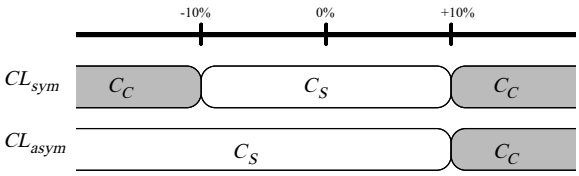

図 2: (A1) におけるクラス分け

以降では 2 つのクラス分け CL_{sym} と CL_{asym} に対し，クラス C_S と C_C に含まれるプロジェクトの品質と生産性に統計的に有意な差があるかどうかを検定する．

1. フィールド品質 (FQ) について

   クラス分け CL_{sym} と CL_{asym} のそれぞれのクラス C_S と C_C について，FQ の値に有意差があるかを検定する．

   クラス分け CL_{sym}

   クラス分け CL_{sym} の下でクラス C_S と C_C について統計的仮説検定を行う．検定したい命題は，「C_S の品質は C_C の品質よりも高い」なので，次の仮説を立てる．

   仮説　　H_0：$\mu_S^{FQ} = \mu_C^{FQ}$

   対立仮説　H_1：$\mu_S^{FQ} < \mu_C^{FQ}$

   ここで有意水準 $\alpha = 0.05$ として検定すると，この仮説 H_0 は棄却される．よって対立仮説 H_1 が支持される．

   つまり，開発計画のコスト予測精度が -10% 〜10% の範囲に入っている (クラス C_S) の成功プロジェクトとそれが 10%以上となってしまう (クラス C_C) の混乱プロジェクトとでは，そこで作成されたソフトウェアプロダクトの品質に有意的な差がある．

   クラス分け CL_{asym}

   次に，クラス分け CL_{asym} の下でクラス C_S と C_C について統計的仮説検定を行う．検定したい命題は，CL_{sym} の場合と同様「C_S の品質は C_C の品質よりも高い」である．従って，仮説も上と同じになる．

   有意水準 $\alpha = 0.05$ として検定すると，今回は仮説 H_0 は棄却されない．そのため，CL_{asym} におけるクラス C_S と C_C の品質の間に有意な差は見られない．

2. チーム生産性 (TP) について

   次に，チーム生産性を表すメトリクス TP について同様な統計的仮説検定を行う．仮説検定したい命題は「C_S の生産性は C_C の生産性よりも高い」ということなので，仮説を次のように立てる．

仮説　　$H_0 : \mu_S^{TP} = \mu_C^{TP}$
対立仮説　$H_1 : \mu_S^{TP} > \mu_C^{TP}$

有意水準 $\alpha = 0.05$ として検定する．CL_{sym} においては，仮説 H_0 は棄却されるが，CL_{asym} においては棄却されない．

上述の 2 つの分析の結果を表 1にまとめる．表 1で○印は有意な差が見られたことを，×印は見られなかったことを表す．この結果よりクラス分け CL_{sym} では，コストの面から成功プロジェクトと判定されたプロジェクトは，品質，生産性の面からも成功したとみなすことができる．一方，クラス分け CL_{asym} では，コスト面で分類した成功プロジェクトと混乱プロジェクトの間で品質，生産性の面からは有意的な差がないことが分かる．

表 1: クラス分け別の検定結果

	クラス分けの方法	
	CL_{sym}	CL_{asym}
品質 (FQ)	○	×
生産性 (TP)	○	×

これらの分析結果より,「開発コストの見積り誤差がマイナス方向に大きくふれたプロジェクトは (単に開発コストだけを見ると成功しているように思えるが) 実は混乱プロジェクトであるとする」ことが妥当であったということが分かる．

5.3　しきい値の分析 (A2)

次に，クラス分類の際に採用されているしきい値の妥当性について考える．(A1) の分析ではしきい値を 10%としてクラス分けを行っていたが，その 10%という値の妥当性を議論する．

そのため，次のようなクラス分けを考える．つまり，RE が $\pm\theta\%$ 以内に抑えられていれば成功プロジェクト，そうでないと混乱プロジェクトと定める．このクラス分けの様子を図 3に示す．

クラス分け $CL_{sym}(\pm\theta\%)$
C_S　：$-\theta\% \leq RE < +\theta\%$
C_C　：$RE \geq +\theta\%, RE < -\theta\%$

このクラス分け $CL_{sym}(\pm\theta\%)$ について，θ の値を 5〜15 の範囲で動かして，それぞれの場合にクラス C_S と C_C の品質，生産性の間に有意な差が見られるかを検定する[1]．検定の方法は，5.2節での C_{sym} の場合と同様に，有意水準 5%の統計的仮説検定を用いる．

[1] θ が 8%,9%,11%,13%,14% となるプロジェクトが存在しないため，これらの値でのクラス分けの結果は直前のクラス分けの結果と同様になる．そのため，ここでは省略している．

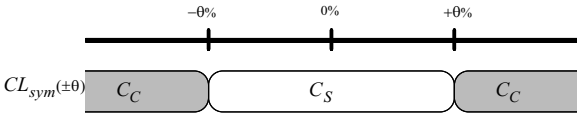

図 3: (A2) におけるクラス分け

表 2に品質と生産性のそれぞれについての検定結果を示す．表 2で○印は有意な差が見られたことを，×印は見られなかったことを表す．

表 2: 各クラス分け検定結果

	クラス分けの基準 (θ)					
	5	6	7	10	12	15
品質 (FQ)	×	×	○	○	×	×
生産性 (TP)	×	×	×	○	×	○

この表より，品質と生産性のどちらについてもクラス C_S と C_C の間で有意な差が現れたのは $\theta = \pm 10\%$ のみであった．

この結果より，少なくとも今回対象としたプロジェクトにおいてではあるが，コストの見積り誤差 10%を基準値とみて成功プロジェクトと混乱プロジェクトに分類するという方法が品質，生産性に有意な差のあるクラス分けをする上で妥当なものであったということが確認された．

6　むすび

本研究では，プロジェクトデータを分析する際に開発現場で採用されているコスト予測の精度に関する分類基準と，その分類に用いるしきい値の妥当性について統計的仮説検定を用いて考察した．

その結果，分類基準については，もし「誤差がマイナス方向にふれたプロジェクトを成功プロジェクトとみなす」と，成功プロジェクトと混乱プロジェクトの間に品質と生産性の上で有意な差がなくなることを確認した．

また，しきい値については現在使われている ±10% 以外の値に設定してしまうと，成功プロジェクトと混乱プロジェクトの間に有意な差が失われてしまうことを確認した．

最後に本研究のこれらの成果の解釈について述べる．既に説明したように，これらはあくまで今回の分析対象のプロジェクトに限定した話であって，現時点では必ずしも一般化できる状況にはない．

謝辞

日頃から我々の研究活動に対し多大の御協力を賜わり，また今回のデータ分析について貴重なご意見を頂いた大阪大学大学院基礎工学研究科 楠本真二講師に深謝致します.

また，本研究の成果の解釈と発表形式について貴重な意見を賜わった (財) 電力中央研究所高橋光裕氏に感謝致します.

参考文献

[1] Fenton N. E. and Pfleeger S. L. : "Software Metrics : A rigorous & practical approach," PWS Publishing (1997).

[2] Humphrey W. S. : "Managing the Software Process," Addison Wesley, Reading, MA (1989).

[3] Humphrey W. S., Snyder T. and Willis R. : "Software process improvement at Hughes Aircraft," IEEE Software, 8(4), pp.11–23 (1991).

[4] 池田央 : "統計ガイドブック," 新曜社 (1989).

[5] 稲垣勝巳，高木徳生，坂本啓司，水野修，菊野亨 : "ソフトウェア開発プロジェクトにおける開発計画の分析 – 品質，生産性との関連性–," 信学技報 SS97-27, pp.15–22 (1997).

[6] 石村貞男 : "統計解析のはなし," 東京図書 (1989).

[7] 岸田孝一監修 (アルフォンゾ・フェジッタ，アレキサンダー・ウルフ編) : "ソフトウェアプロセスのトレンド," 海文堂 (1997).

[8] Mizuno O., Kikuno T., Inagaki K., Takagi Y. and Sakamoto K. : "Analyzing effects of cost estimation accuracy on quality and productivity," Proc. of the 20th ICSE, pp.410–pp.419, (1998).

[9] Möller K. H. and Paulish D. J. : "Software Metrics : A practitioner's guide to improved product development," IEEE Press (Chapman & Hall Computing) (1993).

[10] 新原直樹，高木徳生，稲垣勝巳 : "計画精度から見たプロセス改善活動の分析," ソフトウェアシンポジウム'96 予稿集, pp.153-159 (1996).

[11] Tanaka T., Sakamoto K., Kusumoto S. and Kikuno T. : "Improvement of software process by process visualization and benefit estimation," Proc. of the 17th ICSE, pp.123–132 (1995).

[12] Takagi Y., Tanaka T., Niihara N., Sakamoto K., Kusumoto S. and Kikuno T. : "Analysis of review's effectiveness based on software metrics," Proc. of the 6th International Symposium on Software Reliability Engineering, pp.34–39 (1995).

[13] Tausworthe R. C. : "The Work Breakdown Structure in software project management," Journal of Systems and Software, vol.1, pp.181–186 (1980).

[14] 山田茂，高橋宗雄 : "ソフトウェアマネジメントモデル入門," 共立出版社 (1993)

[15] Yourdon E. : "Death March : The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects," Prentice Hall Computer Books (1997).