

工数見積における誤差分散がユーザ評価に与える影響

Influence of Variance of Effort Estimation to Users' Evaluation

長濱 優樹†
Yuki Nagahama

角田 雅照†
Masateru Tsunoda

1. はじめに

近年、ソフトウェアはより高機能が求められており、その要求にこたえるためにソフトウェアの規模が大規模化している。大規模なソフトウェアを開発するためには、多くの工数（コスト）が必要となる。このようなソフトウェアを開発するプロジェクトでは、スケジュールやコストに関する管理を行わずにプロジェクトを成功させることは困難である。よって、大規模な開発プロジェクトにおいてプロジェクトの失敗を避けるためには、スケジュールやコストの管理が必要不可欠となる。ソフトウェア開発工数の見積もりは、それらの管理の基礎となるものである。

スケジュールやコスト管理を正確に行うためには、ソフトウェア開発の工数を高い精度で見積もることは非常に重要であり、そのため、これまでさまざまな定量的工数見積もり手法が提案されてきた[1][10][11]。定量的工数見積もり方法として、重回帰分析やプロジェクト類似性に基づく工数見積もり[9]などがあげられる。定量的に工数を見積もるためには、過去のプロジェクトで収集されたデータを見積もりの根拠データとし、見積もり対象のプロジェクトにおいて既知のデータ、例えば開発言語や開発規模などを用いて、開発工数を見積もる。

どの工数見積もり方法がより適切かを明らかにするために、それぞれの方法を比較する研究がこれまで数多く行われてきた[8]。これらの実験では、主に見積もり誤差の平均値に着目して比較が行われてきた。例えば、3つのプロジェクトを見積もった時の相対的な誤差が10%、20%、30%の場合、誤差の平均値である20%を他の手法と比較することが行われてきた。ただし、3つのプロジェクトを見積もった時の相対的な誤差が5%、50%、5%の場合も、誤差の平均値は20%である。この場合、後者の分散は前者よりもかなり大きくなる。後者のような分散が大きい手法は、ユーザに使いにくいと評価される可能性がある。

ただし、ユーザが工数見積もりにおける分散をどの程度考慮して、見積もり手法を評価するのかは、これまでほとんど分析されていない。誤差の平均値が同じで誤差の分散が異なる見積もり方法がある場合、誤差の分散が小さい方法をユーザが高く評価する可能性が高い。これに対し、一方の手法は誤差の平均値が小さいが分散が大きく、他方の手法は誤差の平均値が大きく分散が小さい場合、どちらの手法をユーザが高く評価するのかは自明ではない。

そこで本研究では、誤差の平均と分散が異なる見積もり結果を複数用意し、ユーザがどの手法を高く評価するかを実験により確かめる。これにより、ユーザが見積もり誤差の分散をどの程度考慮しているのかを明らかにする。見積もり方法を広く普及させるためには、ユーザからその方法が高く評価される必要がある。ただし、ユーザにあるツールが選ばれるかどうかは、単純にツールから得られる効果

の期待値に基づくのではない。例えばツールの確実性が高いかどうか、ツールの選択において重視される要素の一つとなっている[5]。そのため本研究でも、ユーザによるツールの直接的な評価を重視して分析を行う。

2. 見積もり精度評価指標

工数見積もりの精度評価指標として、 AE 、 MRE (Magnitude of Relative Error)[2]、 BRE (Balanced Relative Error)[4]の3つの指標の平均値と中央値がしばしば用いられる。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。

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

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

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

それぞれの指標の値が小さいほど、工数見積も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であるといえる。ただし、 MRE は過大見積もりに対し、アンバランスな評価になるという問題がある。（見積もり工数が0以上の）過少見積もりの場合、 MRE は最大でも1にしかならない（例えば実測値が1000人時、見積もり値が0人時の場合、 MRE は1となる）。そこで本研究では、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[7]である BRE を評価指標に用いる。

3. 実験

3.1 各手法の見積もり精度

実験では被験者に、ある見積もり手法による見積もり工数と実際の工数を複数提示し、その手法が好ましいかどうかを評価してもらった。見積もり手法として、表1に示す5種類を用意した。これらは重回帰分析などの実際の見積もり方法に基づくものではなく、見積もり誤差の平均値と分散が一定となるようにランダムで値を作成したものである。例えば見積もり手法Aにより見積もり結果は、 BRE 平

表1 各見積もり方法の誤差平均と分散

見積もり方法	BRE 平均値	BRE 分散
A	25%	6%
B	25%	16%
C	32%	7%
D	50%	7%
E	50%	16%

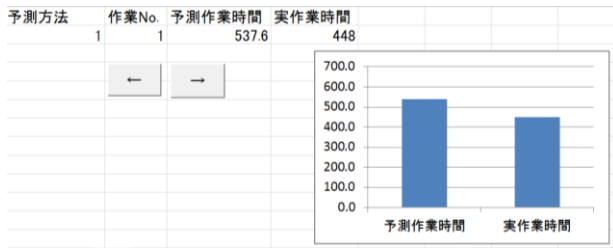


図1 見積もり結果の提示

均値を 25%, 分散が 6%となるように, 工数の実測値と見積もり値をランダムで生成した. なお, 見積もり手法 C は, 実験ツールに不具合があったため, 被験者によりわずかに BRE 平均と分散が異なっている.

被験者は情報科学を専攻する大学の 4 年生 16 人である. 各手法では 20 個の見積もり結果を図 1 の形式で提示した. ユーザは GUI のボタンを操作することにより, それぞれの見積もり結果を閲覧できる. また, 見積もり値と実測値は棒グラフにより視覚化して提示した.

3.2 マッチング

どの手法が好ましいかを比較する方法として, 5 つの見積もり方法全てを比較し, それぞれの優劣を評価することが最も一般的である. ただし表 1 に示すように, 全ての手法間に非常に大きな見積もり誤差があるわけではいため, 全てを比較して評価することは容易ではない[4]. そこで, マッチングと呼ばれる方法を用いた. マッチングとは, ある 2 つの評価対象について, どちらが好ましいかを評価したデータを多数集めることにより, それぞれの評価対象を順位付けする方法である.

マッチングの結果を単純に順位付けに用いると, 優劣の結果を集計することになる. ただしこの場合, 見積もり精度に大きな差がある場合と, 見積もり精度にあまり差がない場合の優劣について, どちらも同様に評価するため, 必ずしも正確に評価できないという問題がある. この問題を解決するために, イロレーティング[3]を用いる. イロレーティングはチェスなどの一対一の対戦ゲームの勝敗を元に, プレイヤーの実力を順位付けする方法である.

3.3 イロレーティングによる評価

イロレーティングによる評価を表 2 に示す. 被験者が不足しているため, 見積もり方法 D については比較回数が少なくなっている. 見積もり方法 A と B については, 誤差平均は同じだが分散の小さい A のほうが, 被験者からの評価が高かった. このことから, 工数見積もり手法の評価において, 分散は無視できないと考えられる.

見積もり方法 B と C については, 誤差平均が小さく誤差

表2 イロレーティングによる評価

見積もり方法	レーティング	比較回数	選択回数	非選択回数
A	2381	33	27	6
B	2298	34	24	10
C	2153	30	13	17
D	2111	24	7	17
E	2057	39	9	30

分散が大きい B のほうが, 誤差分散の小さい C よりも被験者からの評価が高かった. B と C の誤差平均の差は大きくないが, 被験者はその違いを重視するとともに, 誤差分散の大きさについてはあまり重視しなかったといえる. このことから, 誤差分散の小ささよりも, 誤差平均の小ささのほうが重視される可能性が高いといえる.

見積もり方法 D と E については, 誤差平均は同じだが分散の大きい E のほうが, 被験者からの評価が高かった. なお, AE の平均値と最大値, 実測値の平均値, BRE の最大値は D と E でほとんど差がない, もしくは E のほうが大きかった, 評価結果が予想と逆となった理由として, E の評価回数が少ないことが影響している事が考えられる. その他の理由として, 誤差平均の大きさと比較して分散の大きさが小さい場合, 分散の違いに被験者が気づきにくかった可能性もある.

4. おわりに

本研究では, 工数見積もり方法の評価において, 誤差の分散をユーザがどの程度評価するかを分析した. ユーザによる見積もり方法の評価では, マッチングとイロレーティングを適用した. その結果, 以下の傾向が見られた.

- 誤差平均が 2 つの見積もり方法で同様に小さい場合, 分散の小さい方法の評価が高かった.
- 誤差平均が小さく分散が大きい見積もり方法のほうが, 誤差平均が大きく分散の小さい方法よりも評価が高かった.
- 誤差平均が 2 つの見積もり方法で同様に, 誤差分散が誤差平均と比較して小さい場合, 誤差分散の差は考慮されない可能性がある.

今後はさらに誤差平均と分散の組み合わせを変化させ, ユーザが重視する要因を明らかにする予定である.

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 16K00113) による助成を受けた.

参考文献

- [1] B. Boehm: *Software Engineering Economics*, Prentice Hall (1981).
- [2] Conte, S., Dunsmore, H., and Shen, V.: *Software Engineering, Metrics and Models*, Benjamin Cummings (1986).
- [3] Elo, A.: *The Rating of Chess Players, Past and Present*, Ishi Press (2008).
- [4] Hacker S., and Ahn, L.: Matchin: eliciting user preferences with an online game, *In Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1207-1216 (2009).
- [5] Kina, K., Tsunoda, M., Hata, H., Tamada, H., and Igaki, H.: Analyzing the Decision Criteria of Software Developers Based on Prospect Theory, *In Proc. of International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, pp.644-648 (2016).
- [6] 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).
- [7] Møllokken-Østfold, K., and Jørgensen, M.: A Comparison of Software Project Overruns-Flexible versus Sequential Development Models, *IEEE Trans. on Software Eng.*, Vol.31, No.9, pp.754-766 (2005).
- [8] Shepperd, M., and Kadoda, G.: Comparing Software Prediction Techniques Using Simulation, *IEEE Trans. on Software Eng.*, Vol.27, No.11, pp.1014-1022 (2001).
- [9] Shepperd, M. and Schofield, C.: Estimating software project effort using analogies, *IEEE Trans. on Software Eng.*, Vol.23, No.12, pp.736-

743 (1997).

[10] Srinivasan, K., and Fisher, D.: Machine Learning Approaches to Estimating Software Development Effort, *IEEE Trans. on Software Eng.*, Vol.21, No.2, pp.126-137 (1995).

[11] 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 (2005).