

ソフトウェア見積りの可視化手法に関する一考察

Study of visualization technique of software estimation

志田 剛
Tsuyoshi Shida

津田 和彦
Kazuhiko Tsuda

筑波大学大学院 ビジネス科学研究科
Graduate School of Business Sciences, University of Tsukuba

Ordering software development, it is necessary to share the estimation result and process of the project between users and IT vendors. It is the success of the project is essential to accept each other. However, IT vendors it is difficult to describe the complexity of the functional development to the user doesn't have expertise. As a result, there is a problem that consent of estimates can't be obtained. So, I do discussion regarding technique for visualizing the complexity in feature development in this paper.

1. はじめに

ソフトウェア開発の受発注においては、発注者と IT ベンダとの間でプロジェクトの見積りの過程と結果をお互いが共有し、両者が納得してからプロジェクトを進めることが不可欠である。ところが、IT ベンダが専門知識を持たない発注者に対して、機能開発における難易度や複雑度を説明することは困難である。一方、発注者は IT ベンダから見積根拠が理解できないまま予測以上の工数が提示され、それが不満となっている。JUAS の IT 動向調査では、発注者の満足度について、2011 年の調査 [JUAS 2011]では「見積金額の妥当性」、「価格」がともに満足しているが 14%という低い結果が出ている。つまり、発注者は見積りへの納得感が得られないという問題が発生している。

そこで、本研究では機能開発における難易度や複雑度を可視化することを目的とする。具体的には、見積根拠となるプログラム構造や変動要因など過去事例を使い、これまで発注者にとってブラックボックスとなっている部分の明示化を行う。

2. ソフトウェアの見積り手法

システム開発において、ユーザからソフトウェアの見積りを依頼されたベンダはまず、RFP(Request For Proposal)や要件定義書を元に規模見積りを行う。この時、基準となる開発ライフサイクルモデルについて、IPA の調査 [IPA 2015]では、ウォーターフォール型が 96%以上と高い割合になっている。

ソフトウェア開発における見積り手法としては、過去の類似プロジェクトの実績を基礎に見積り「類推法」や、工数を目的変数として説明変数に規模や要因などを設定し数学的な関数として表す「パラメトリック法」などが有名である。最近では、熟練者の経験とソフトウェア開発の定量データとの組み合わせにより見積りを実現する CoBRA(Cost estimation Benchmarking and Risk Assessment)法というハイブリッド型の方見積りモデルが注目されている。

CoBRA 法の effort(工数)定義 [石谷 2006]では、

$$Effort = \alpha \times (Size) \times (1 + \sum co_i) \quad (式1)$$

と示されて、 α は単位時間当たりの必要理想工数を示す定数、 $Size$ は SLOC や FP 法を使い算出するプロジェクト開発規

模、 $CO(cost\ overhead)$ はプロジェクト毎に存在する変動要因であり、複数の要因を許容する。

以上のように、ソフトウェアの開発見積りの精度を向上させるための研究や取り組みは多く存在する。けれども、見積もった工数をソフトウェアに関する知識に乏しい発注者に納得させることを目的とした研究は発見できなかった。

3. ソフトウェア開発見積りモデルの抽出

3. 1 過去プロジェクトの分析

CoBRA 法における CO の要因選定はプロジェクト毎に異なる。例えば図 1 のようにプロジェクトに取り巻く 10 の変動要因を定義した場合、この変動要因はプロジェクトリーダーやメンバとブレインストーミング法で要素を決める。そして、それぞれの要因レベルを 0 から 3 までの基準で作成する。今回は過去 10 組のプロジェクトの工数と規模のデータを収集し、各プロジェクトに携わったリーダーやプロジェクトマネージャから変動要因のレベルを評価してもらい、それぞれの変動要因とした。

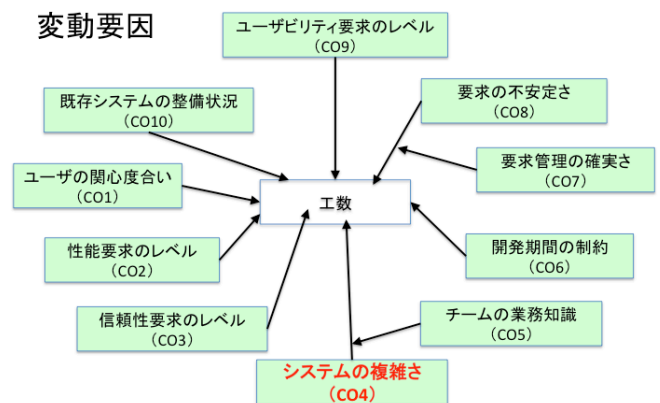


図 1 変動要因の例

次にそれぞれのプロジェクトに対しモンテカルロシミュレーションを使い、 $\sum CO_i$ 多数回計算することで $\sum CO_i$ の安定した分布を作る。その分布の中央値を当該プロジェクトにおける増加割合とする。これで式 1 の $(Size) \times (1 + \sum co_i)$ と実際のコストの値の組が 10 組得られ、これに対して回帰分析を行い、その係数が α となり CoBRA モデルが作成される。今回の実験では CoBRA 法の評

価として IPA-SEC が提供している「統合見積もりモデルツール」を用いた。そして CO の相関をとり、項目間の関係性を調査した。また、見積もり工数を目的変数、変動要因を説明変数として重回帰分析を行った。分析ツールは R を使いステップワイズ法より最も当てはまりのよいモデルを抽出した。

4. 評価実験

統合見積もりモデルツールの結果からは、3つのプロジェクトから見積もり誤差率として「極めて過大」な誤差が評価された。この誤差率は表1より、50%以上の誤差がある場合に極めて過大と評価される。そしてこの「極めて過大」と評価された3つのプロジェクトの CO を確認したところ CO₄(システムの最利用度)、CO₇(信頼性要求)、CO₉(システムの複雑さ)の評価が全体的に高いことが分かった。

表1 見積もり誤差率

基準	メッセージ
-50%未満	極めて過小
-25%未満	過小
0%未満	やや過小
25%未満	やや過大
50%未満	過大
50%以上	極めて大

また重回帰分析については回帰式 AIC を算出し、フルモデルの場合、AIC=78.87 となり AIC の最小値は 2 変数で表現することができる。そして当てはまりの一番良いモデル工数は $effort = CO_6 + CO_9$ となり、式2に示す回帰式を抽出することができた。

$$-35.97 = 37.73X_6 + 16.08X_9 \quad (式 2)$$

そして 17 要因の CO 間の相関係数を計測したところ、相関係数 0.7 以上の強い相関がみられ、その中で 0.8 以上の相関が 6 個程度あったが CO6, CO9 間の相関は-0.58 となり、相関性は高くなく多重共線性は発生していないと考えられる。

今回の重回帰分析や見積もりツールを使った結果から、CO₆ と CO₉ つまりシステムの複雑さと性能要求が要因として関連深い事が分かった。その中でも特にソフトウェア工数の見積もりを実施する際、複雑さが一番のボトルネックになっていることが原因の1つとして考えた場合、数ある要因の中からこのシステムの複雑度を可視化するすることによって見積もりの可視化が出来ると考えられる。システムの複雑さを可視化することは、システム内部処理を可視化し表現する必要がある。

Web システムの構成では、「プレゼンテーション(P)層」、「ファンクション(F)層」、「データ(D)層」の 3 層構成とする構成が主流である。入力が多くなればファンクション層とデータ層の間で処理する量が多くなり、その処理に対するテストケース数が増加することで開発工数は多くなる。またこの P 層 D 層の 2 層間では、ユーザからは見えない部分となり、見積もりにどのような影響があるのか見えにくくなっており、複雑さを増す結果になっているとも考えられる。そこでこれをユーザでも理解出来るように考えたのが図2の規模可視化モデルである。これを使い見積もり規模の可視化につながると考えた。

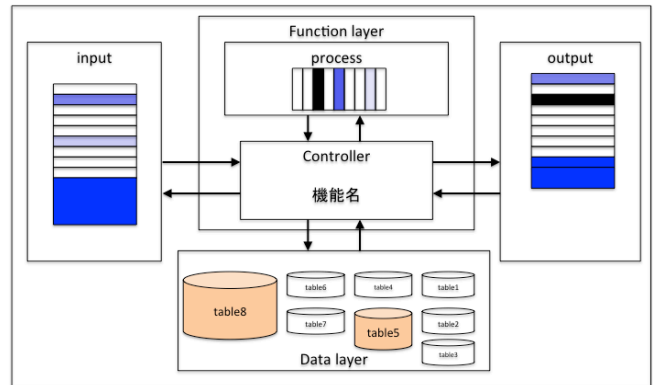


図2 規模可視化モデル

図2の規模可視化モデルでは機能毎に表現し、入出力処理を行う P 層、プログラムのメイン処理を行う F 層、データを管理する D 層それぞれの層ごとに大きさや色の濃さで表現する。箱の大きさは追加する量を表現し、処理量の大小を表現する。そして、透明度は処理の複雑度を表し、色が濃くなる程処理は複雑となり、透明になるほど処理は単純になると考える。これを見積もり規模と作成することで処理やデータの追加・更新・削除を表現する事ができ、見積もりを作成と同時に見積もり可視化モデルを表現することでユーザは IT ベンダが提示する見積もり規模を納得するが出来る。

5. おわりに

今後の予定として、欠損のない見積もりデータをプロジェクト別で収集し、より多くの欠損のない見積もりデータを用いて変動要因のモデル検証を行っていく。そして変動要因の中で今回検証した複雑さ以外の要因がないのかも検証していく。また、システムの複雑さとテストケースの量も関係性は強く、複雑な処理になるほど、テストのケース数は必然と増えてくる。つまりテストケースが多くなれば、基本的にソフトウェアの品質は高くなるが、工数が増えるため見積もり工数も比例して上がることになる。そこで見積もりの規模とテストケース数についての研究も行っていく。見積もり規模とテストケースについての可視化についても今後提案していく。

参考文献

[JUAS 2011] “企業 IT 動向調査 2011”, 日本情報システム・ユーザー協会 2011
 [IPA 2015] ”ソフトウェア開発白書データ白書”, IPA-SEC2014-2015
 [石谷 2006] 石谷靖: “ハイブリッドなコスト見積もりモデルの反復的な構築方法について”, SEC journal, vol7, 2006
 [酒井 2012] 酒井大: “CoBRA 法を使った見積もりモデル構築ポイント”, SEC journal, vol26, 2011.
 [Briand 1998] C.Briand” COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking and Risk Assessment. Proceedings of the 20th International Conference On Software Engineering”, pp.390-399,1998