

プロセスデータに基づくソフトウェア開発プロジェクトの 品質指向型定量的評価法に関する考察

鳥取大学大学院・工学研究科 青木 俊樹 (Aoki Toshiki)[†]

鳥取大学・工学部 山田 茂 (Shigeru Yamada)^{††}

[†]Graduate School of Engineering, Tottori University

^{††}Faculty of Engineering, Tottori University

1 はじめに

近年、ユーザの要求品質を満足するソフトウェアを開発できるように、開発プロジェクトの開発能力を品質の観点から向上させることが非常に重要になってきている。また、ソフトウェア開発において短納期化が進んでいることによる影響は、近年のソフトウェア開発の形態の多様化にも及んでいる。

一般に、ソフトウェアの開発方法は、その開発プロセスをいくつかの工程に分け、要求仕様定義、設計、コーディング、テストの順に作業を進めていくウォーターフォールモデル [1] とよばれる開発形態が多く取られていた。しかし、生産性をより高いレベルで要求されている近年のソフトウェア開発においては、従来型のウォーターフォールモデルに従った開発形態だけでは十分な生産性を得ることが困難である。そこで、近年では、迅速かつ適応的にソフトウェアを開発することのできるアジャイル開発手法 (agile software development) が注目されており、小・中規模なアプリケーション開発や、顧客の要件定義や要望が曖昧である場合、継続的なリリースが必要になる社内システムなど、仕様の変更が頻繁に行われる開発に多く適用されている。

アジャイル開発の主な目的は、顧客に対して迅速に価値を提供すること、変化に素早く対応することなどが挙げられ、近年のソフトウェア開発においては必要不可欠な開発手法となっている。しかしながら、アジャイル開発のプロセス計測データを採取するのは非常に困難であり、アジャイル開発手法を用いたプロジェクトでは、十分な定量的評価法が確立されていない。したがって、アジャイル開発手法を用いたソフトウェア開発の現場では、経験則・暗黙知により信頼性・品質を判断することが多く、定量的な評価を実施することなくリリースされているのが現状となっている。

本研究では、実際に P 社において収集されたアジャイル開発におけるプロセス計測データを用いて、信頼性・品質を定量的に評価するためのメトリクスを抽出し、プロジェクトの定量的評価を行う。ここで、協調フィルタリング [2-5] および重回帰分析 [6] を適用することで、ソフトウェアの製品品質の推定法に関する考察を行い、結果の精度を比較する。さらに、ソフトウェア信頼性評価法 [7] をアジャイル開発に適用することで、定量的評価における有用性も考察する。

2 分析データ

本研究では、アジャイル開発におけるプロセス計測データを用いて、プロジェクトの定量的評価を行う。アジャイル開発とは、ソフトウェア工学の考え方に基づいて、迅速かつ適応的にソフトウェアを開発する軽量な開発手法の総称を表し、従来型の開発手法であるウォーターフォールモデルなどの計画駆動型開発手法と対極に位置するものである。アジャイル開発では、プログラムを常に実行可能な状態に保ち、確認・拡充していくインクリメンタル手法 (段階的拡充手法) をとる。開発対象を多数の小さな機能に分割し、1つの反復 (イテレーション) で1機能を開発する。この反復のサイクルを繰り返していき、機能を1つずつ追加して開発していく。各反復では、それまでに開発した成果物に1つの小さな機能を追加する。要求分析、設計、コーディング、テストといったソフトウェアプロジェクトに要する全ての工程を、1つの反復内で行う。各段階に要する期間は、数日から数週間と短いのが通例であり、このようにすることによって、仕様変動のリスク、技術的実現性のリスクを回避することができる。本研究では、P 社において実際に収集されたアジャイル開発のプロセスデータを取り上げ、開発規模当りに規準化したものを対象として分析する。

表 1： 相関分析表

	X_1	X_2	X_3	X_4	X_5	Y
X_1	1					
X_2	-0.92	1				
X_3	-0.88	0.74	1			
X_4	-0.94	0.84	0.98	1		
X_5	-0.82	0.67	0.87	0.80	1	
Y	-0.29	0.53	-0.17	0.02	-0.13	1

各プロジェクトの概要は以下の通りである。

ProjectA : NET 開発環境 (Oracle-DB 使用) 下で開発した WEB システムで、開発工数の削減 (効率改善) を目指した開発支援システムである。

ProjectB : NET 開発環境 (Oracle-DB 使用) 下で開発した WEB システムであり、プロジェクトマネジメント用にプロジェクトを定量化するために、各種データに基づく分析結果を可視化する機能を幾つか備えた分析専用の WEB システムである。

ProjectC : Unix 環境下で開発した大型プリンタ装置の組込系システムである。

ProjectD : Linux 環境下で開発したデータ制御システムであり、ホストコンピュータに繋がる専用端末の組込系開発である。

ProjectE : Linux 環境下で開発した端末画面制御システムであり、ホストコンピュータに繋がる専用端末の組込系開発である。

以上のデータを使用し、ST (システムテスト) 検出フォールト数 (Y) を推定・予測することによりプロジェクトの定量的評価を行う。

3 重回帰分析

本研究では、取り扱うプロセス計測データが多変量であるため、多変量解析の 1 つである重回帰分析を用いる。重回帰分析とは、目的変数が説明変数の変動によってどの程度影響されるかを分析し、説明変数から目的変数を線形式により推定・予測する方法である。その関係式から、結果に大きな影響を与えている要因を明らかにすることができる。

3.1 相関分析

説明変数として扱う 5 つのメトリクス $X_i (i=1,2,3,4,5)$ と目的変数 (Y) の相関分析を行うと表 1 の結果が得られ、下記の相関関係が考えられる。

- 開発規模当り障害件数 (X_2) が、開発規模当り ST 検出フォールト数 (Y) との相関が最も高い。
- 開発規模当り ST 検出フォールト数 (Y) は、開発規模当り開発工数 (X_4) との相関が非常に低い。
- 結合数 (X_1) と開発規模当り障害件数 (X_2) との相関は高く、多重共線性の可能性がある。

以上より、開発規模当り ST 検出フォールト数 (Y) と、多重共線性がある結合数 (X_1) を除き、開発規模当り ST 検出フォールト数 (Y) と相関が高い順に選定し、開発規模当り障害件数 (X_2)、開発規模当りテストケース数 (X_3) を重回帰分析の説明変数として用いる。

表 2：回帰精度

重相関係数 R	0.986
決定係数 R^2	0.971
補正決定係数 R^2	0.943
標準誤差	0.401

表 3：分散分析表

要因	自由度	変動	分散	検定統計量 F_0
回帰	2	10.946	5.473	34.0512 *
残差	2	0.321	0.161	
計	4	11.268		

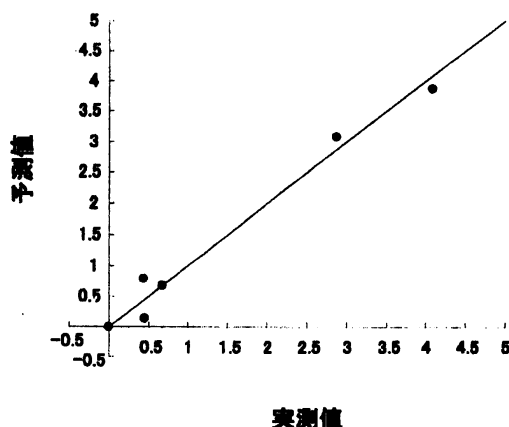


図 1：ST 検出フォールト数 (Y) の予測精度

3.2 分散分析

重回帰分析における回帰精度および分散分析表は、それぞれ表 2 および表 3 になる。表 2 の回帰精度より、決定係数 R^2 は 0.971 という非常に高い値となる。また、表 3 の分散分析表より、

$$F_0 = 34.0512 > F_2^2(0.05) = 19.00,$$

となり、危険率 5% で有意となり、得られた重回帰式が予測に役立たないという帰無仮説は棄却される。以上より、重回帰式のデータに対する適合性は高いといえる。

3.3 推定された重回帰式

重回帰分析より、式 (1) の重回帰式 Y が導出される。また、分析するデータを標準化して、回帰分析を行った標準偏回帰係数より、式 (2) の標準化重回帰式 Y^N が導出される。

$$Y = 1.574 \cdot X_2 - 0.012 \cdot X_3 - 1.552, \quad (1)$$

$$Y^N = 1.448 \cdot X_2 - 1.240 \cdot X_3. \quad (2)$$

式 (2) より、標準偏回帰係数の絶対値を比べると、説明変数の目的変数に影響を与える度合の大きさは、 $X_2 > X_3$ であることがわかる。よって、開発規模当り障害件数 (X_2) と開発規模当りテストケース数 (X_3) が目的変数である開発規模当り ST 検出フォールト数 (Y) に大きな影響を与えているといえる。

3.4 重回帰式によるソフトウェア品質の予測

式 (1) にアジャイル開発下におけるプロセスデータを代入した予測値と、実測値である開発規模当り ST 検出フォールト数 (Y) との関係を図 1 に示す。図 1 より、実測値と予測値はほとんど誤差がないことがわかる。よって、開発規模当り ST 検出フォールト数 (Y) は高い精度で予測でき、データに対する適合性は高く、プロセスデータを開発規模当りに標準化したことが精度の向上に大きく関わったと考えられる。

表 4：類似度

	ProjectA	ProjectB	ProjectC	ProjectD	ProjectE
ProjectA	——	0.9243	-0.1424	-0.6412	-0.8788
ProjectB	0.9243	——	0.1051	-0.7303	-0.9496
ProjectC	-0.1424	0.1051	——	-0.2478	-0.2750
ProjectD	-0.6412	-0.7303	-0.2478	——	0.5865
ProjectE	-0.8788	-0.9496	-0.2750	0.5865	——

3.5 重回帰分析からのプロジェクト評価

- 開発規模当り障害件数 (X_2) が開発規模当り ST 検出フォールト数 (Y) に最も大きな影響を与えていることから、モジュールを結合する前の段階でモジュール内の障害件数を抑えることが重要であることがわかる。
- 開発規模当りテストケース数も開発規模当り ST 検出フォールト数 (Y) に影響を与えていることから、適切かつ適量のテストケース数の設定が重要であることがわかる。

4 協調フィルタリング

協調フィルタリングは、ソフトウェア工学の分野では、プロジェクト間の類似性を分析する定量的手法であり、予測対象のプロジェクトと最も類似性の高いプロジェクトデータから順番に抽出していき、その抽出したデータをもとに予測を行う方法である。重回帰分析ではただ 1 つの予測式（モデル）が作成されるのに対し、協調フィルタリングでは、対象プロジェクトごとに予測式が構築されるため、プロジェクトの個性をより強く反映した予測が行える。また、欠損値の量や分布（欠損値の偏り）が変化しても精度は低下しにくいという特徴があるため、データに欠損値が含まれている場合にも有効である。

4.1 協調フィルタリングの手順

本研究で適用する協調フィルタリングの手順について述べる。協調フィルタリングは、以下に示す 3 つの手順で実行される。

(1) メトリクスの正規化

メトリクスはその種類ごとに値域が異なるため、類似度の計算を行う前に正規化という補正作業が必要となる。各メトリクスが予測に対して与える影響を均一にするため、メトリクスの値を 0 から 1 までの値に統一する。

(2) プロジェクト間の類似度の算出

予測対象のプロジェクトと推定の根拠として用いる他の各プロジェクトとの類似度を求める。本研究では、adjusted cosine similarity をアルゴリズムとして使用する。類似度の値域は $[-1.0, 1.0]$ である。

(3) 類似度に基づく推定値の計算

手順 (2) で求めた類似度を用いて推定値を計算する。ここで、算出アルゴリズムは、amplified weighted sum を使用する。

4.2 協調フィルタリングの適用結果

2. で取り上げたプロセスデータに協調フィルタリングを適用すると、プロジェクト間の類似度は表 4 の結果が得られる。表 4 より、NET 開発環境下で開発された ProjectA と B、Linux 環境下で開発された ProjectD と E はプロジェクト間の類似度が高いことがわかる。よって、本研究で抽出したメトリクスにおいて、同種の開発プロジェクトはプロセスも類似しているということが確認できる。

表 5：予測精度の比較結果

	重回帰分析	協調フィルタリング ($k=4$)
MAE	0.233	1.456
MRE	0.334	1.518

協調フィルタリングを用いてプロジェクト間の類似性から予測を行った結果は比較的精度の低いものとなった。ここで、予測に使用するプロジェクト数 k は絶対誤差の平均が最も小さくなった $k=4$ を採用した。協調フィルタリングによる予測の精度が低下した原因として、予測に使用するデータ数が少なく、類似性の高いデータを十分に得ることができなかったことが考えられる。

5 ソフトウェア製品品質の予測精度の比較

アジャイル開発におけるソフトウェア製品品質の予測結果について、重回帰分析と協調フィルタリングの精度を比較した結果を表 5 に示す。ここで、予測精度の評価には、絶対誤差の平均 (MAE) と相対誤差の平均 (MRE) を使用し、両者が低い値をとるほど予測の精度が高いといえる。表 5 より、MAE および MRE の両者において、重回帰分析の方が予測の精度が高いという結果となった。

6 ソフトウェア信頼性評価

ソフトウェア品質の計測方法として、ソフトウェア信頼性評価技術がある。その中でも、ソフトウェア信頼度成長モデルは、動的環境におけるソフトウェアの挙動を信頼度成長過程として記述するものであり、開発中のソフトウェアに含まれる不具合数やソフトウェア信頼度を推定する方法としてよく知られている。このモデルは実際の適用例も多く、ソフトウェア信頼性モデルの中でも中心的役割を担っている。

6.1 アジャイル開発における信頼性評価の定義

アジャイル開発におけるメトリクスを離散的データとして捉え、テスト時間の代替メトリクスとしてソフトウェア信頼性評価を試みる [8]。イテレーション j までに発見される ST 検出フォールト数の総数を表す計数過程 $\{N(j), j \geq 0\}$ が平均値関数 $H(j)$ をもつ非同次ポアソン過程 (以下 NHPP と略す) に従うものと仮定すると、ソフトウェア信頼度成長モデル (以下 SRGM と略す) は

$$\Pr\{N(j) = n\} = \frac{\{H(j)\}^n}{n!} \exp[-H(j)] \quad (n = 0, 1, 2, \dots), \quad (3)$$

と表現できる [7]。式 (3) において $H(j)$ は $N(j)$ の期待値であり、イテレーション j までに発見される総期待フォールト数を表す。

6.2 信頼性評価の適用結果

アジャイル開発のプロジェクトデータを使用し、テスト時間の代替メトリクスを結合数、障害件数、レビュー回数、テストケース数、開発規模、開発工数として信頼性評価を実施する。まず、最終となるインクリメントを結合した後のシステムテスト完了時を、従来の信頼性評価におけるテスト終了時期と仮定し、ST 検出可能フォールト数が有限であると考え、信頼性評価に用いるモデルを指数形 SRGM [7]、遅延 S 字形 SRGM [7] とする。次に、アジャイル開発は段階的に拡充していくインクリメンタル手法を用いるため、最終的に検出される ST 検出可能フォールト数は無限となると考えることもできる。よって、検出可能フォールト数が無限の場合を仮定した対数型ポアソン実行時間モデル [9] を信頼性評価に用いるモデルに加える。以上の 3 つのモデルを信頼性評価に使用し、モデルに対する適合性評価を行う。ここで、適合性を比較する評価基準を平均偏差平方和 (MSE) および赤池情報量基準 (AIC) とする。

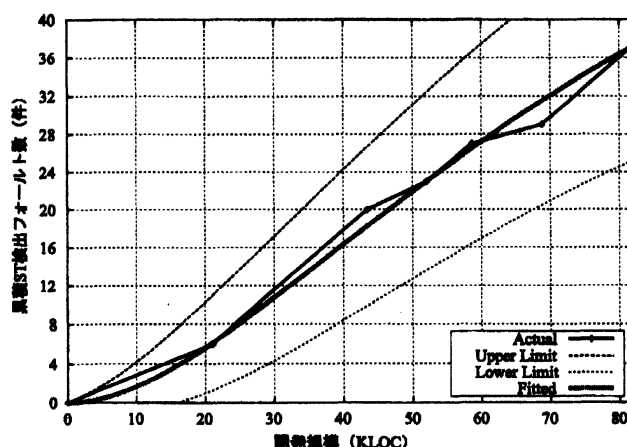
ここでは、Project A における信頼性評価の結果を一例として、MSE および AIC に基づく適合性評価の結果を表 6 および表 7 に示す。適合性評価の結果より、Project A においては、開発規模をテスト時間の代替メトリクスとした遅延 S 字形 SRGM が最良の適合性を示した。最良の適合性を示した平均値関数 $H(j)$ の推定値、発見された総フォールト数の総計、および 90% 信頼限界を図 2 に示す。

表 6: MSE に基づく適合性評価

	指数形 SRGM	遅延 S 字形 SRGM	対数型ポアソン 実行時間モデル
結合数	7.58	11.43	7.50
障害件数	7.28	5.54	7.72
レビュー回数	2.97	2.32	3.10
テストケース数	7.82	6.42	—
開発規模	—	1.58	—
開発工数	2.72	4.62	—

表 7: AIC に基づく適合性評価

	指数形 SRGM	遅延 S 字形 SRGM	対数型ポアソン 実行時間モデル
結合数	45.18	45.50	45.1618
障害件数	36.03	33.73	36.09
レビュー回数	30.39	30.13	30.41
テストケース数	39.84	36.50	—
開発規模	—	29.36	—
開発工数	30.11	32.04	—

図 2: 推定された平均値関数 $\hat{H}(j)$

イテレーション数が少ない ProjectE を除き, ProjectB, C, D においても同様に適合性評価を行った結果, ProjectB および C において ProjectA と同様の結果が得られ, 開発規模をテスト時間の代替メトリクスとした遅延 S 字形 SRGM が最良の適合性を示した. また, ProjectD においては, テストケース数をテスト時間の代替メトリクスとした遅延 S 字形 SRGM が最良の適合性を示した.

7 おわりに

本研究では, アジャイル開発のプロセス計測データに重回帰分析および協調フィルタリングを適用することで, ソフトウェア開発プロジェクトの定量的評価を行った. また, アジャイル開発のプロセス計測データを用いたソフトウェア信頼性評価を試みた.

重回帰分析を適用した結果, プロセス計測データを開発規模当りに規準化したことが分析精度の向上に大きく関わった [10]. 推定された重回帰式より, ソフトウェア製品品質に大きな影響を与えていると考えられるメトリクスを分析することができた. 協調フィルタリングを適用した結果から, 予測に使用するプロジェクト数を増やし, 類似性が高いデータを十分に抽出することが予測精度の向上のために重要となることがわかった. 様々なメトリクスをテスト時間の代わりに用いてソフトウェア信頼性評価を行った結果, 多くの場合において, 開発規模を代替メトリクスとした遅延 S 字形 SRGM に良い適合性がみられた.

今後の課題として, 協調フィルタリングに使用するアルゴリズムを開発規模を考慮したものに変更していく必要がある. さらに, ソフトウェア信頼性評価において, イテレーション数がさらに増加した状態のプロジェクトデータを用いて, 推定値の適合性を確かめる必要がある.

謝辞

本研究の一部は、日本学術振興会科学研究費補助金 基盤研究 (C) (課題番号 18510124) の援助を受けたことを付記する。

参考文献

- [1] 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, 東京 (1993).
- [2] 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).
- [3] 柿本健, 角田雅照, 大杉直樹, 門田暁人, 松本健一:“協調フィルタリングに基づく工数見積り手法のデータ欠損に対するロバスト性の評価”, 電子情報通信学会論文誌, Vol. J89-D, No. 12, pp.2602-2611(2006).
- [4] 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り”, SEC Journal, No. 5(2006).
- [5] 瀧進也, 柿本健, 角田雅照, 大杉直樹, 門田暁人, 松本健一:“プロジェクト類似性に基づく工数見積りのための変数選択”, 電子情報通信学会ソフトウェアサイエンス研究会技術研究報告, pp.1 – 6(2006).
- [6] 永田靖, 棟近雅彦:「多変量解析法入門」, サイエンス社, 東京 (2001).
- [7] 山田茂:「ソフトウェア信頼性モデル」, 日科技連出版社, 東京 (1994).
- [8] 藤原隆次, 山田茂:“アジャイル開発環境におけるソフトウェア信頼性評価に関する一考察”, 日本オペレーションズ・リサーチ学会春季研究発表会アブストラクト集, pp.40 – 41(2007).
- [9] 山田茂, 大寺浩志:「ソフトウェアの信頼性」, ソフト・リサーチ・センター, 東京 (1990).
- [10] 山田茂, 福島利彦:「品質指向ソフトウェアマネジメント」, 森北出版, 東京 (2007).