

エンタープライズ系ソフトウェアの 信頼性に影響を与える質的要因の分析

古山恒夫^{1,a)}

概要: IPA/SEC が収集したエンタープライズ系ソフトウェアプロジェクトデータを分析することにより稼働後不具合数に影響を与える質的要因を明らかにした。稼働後不具合数がゼロの値をもつプロジェクトが 30 % 程度占めるため、すべての変数を対数化して線形回帰分析を行うという方法は適用できない。そこで経済学や社会学などで用いられている負の二項回帰モデルを用いて分析した。分析の結果、新規開発プロジェクトにおいて信頼性を向上させるためには、ユーザ側のプロジェクトへの関与、要求仕様変更発生の抑制、デバッグ・テストツールの利用、テストチームの高いスキルレベル、納期・品質等の達成目標と優先度の明確化が重要な役割を果たすことがわかった。特に、要求仕様変更の発生が最も信頼性に悪影響を与えている。

キーワード: エンタープライズ系ソフトウェア、ソフトウェア信頼性、負の二項回帰モデル、不具合数、質的変数

Analysis of Qualitative Factors that Affect Reliability of Enterprise Software

Abstract: Qualitative factors affecting the reliability of enterprise software were clarified by analyzing project data collected by IPA/SEC. The ordinal linear regression model cannot be applied to the analysis of data that includes zero values, such as the number of software faults detected in operations that indicate a degree of reliability, since those values cannot be transformed logarithmically. The negative binomial regression model that is used in the area of economics or sociology was applied to the analysis. The results show that a user's commitment to projects, fixed requirement specifications, the usage of debug and test tools, maintaining the high skill level of the testing team, and a high clarity of objectives and priorities for delivery and quality are important to improve reliability of the newly developed software. In particular, requirement specification changes are the most effective factors for lowering reliability.

Keywords: Enterprise software, Software reliability, Negative binomial regression model, Number of failures, Qualitative variable

1. はじめに

他の工業製品の開発と同様にソフトウェア開発でも、品質、コスト（工数）、納期（工期）はキーとなる重要な要素であり、計画段階からこの3つの要素のバランスを考慮しながら開発を進める必要がある。工数については、プロジェクト計画時の見積もりや組織レベルのプロセス改善のため

にソフトウェア工学が提唱された早い段階からその影響要因が研究され、モデル化が行われてきた。COCOMO[1] やその改良版である COCOMOII[2] などがその代表的なものである。

品質は ISO/IEC 25000 シリーズで定義されているように多くの側面をもつが、6つの品質特性のうちのひとつである信頼性という意味で使われることも多い。実際、品質予測という言葉が信頼性予測の意味で使われていることも少なくない。これはソフトウェア開発が人々の関心を集めるようになったころから、開発者がソフトウェアに含まれる欠陥に悩まされていたことを示している。現在でも信頼

¹ 東海大学
Tokai University, Kitakaname 4-1-1, Hiratsuka City, Kanagawa, 259-1292, Japan

^{a)} furuyama@tokai-u.jp

性は品質特性の中でも重要な位置を占めている。

これまでのソフトウェアの信頼性に関する研究を目的という観点からみると、次の4つに分類できる。

- (a) テスト工程時の欠陥^{*1}数の予測
- (b) 欠陥を含むモジュールの予測または判定
- (c) 出荷時の製品に含まれる欠陥数の予測
- (d) 稼働後の信頼性に影響を与える要因の抽出

(a) は、テスト工数を早期に把握するのに有効である。初期には線形回帰分析から得られた結果をもとにプログラムの複雑さ^{*2}や規模の一次式で欠陥数を予測するモデル [3] や Halstead の尺度から欠陥数を予測するモデル [4] が提案された。最近では、ベイジアンネットの手法を用いてコード行数などの量的変数及び開発スタッフのスキルレベルなどの質的変数からテスト工程時の欠陥数を予測するモデル [5]、線形回帰分析によりコードレビュー指摘密度から単体テスト、結合テスト、システムテストそれぞれの欠陥密度を予測するモデル [6]、同じく線形回帰分析を用いて設計時のレビュー工数、レビュー指摘件数、設計文書量から下流工程の欠陥数を予測するモデル [7]、ピアレビューデータから欠陥抽出数を予測するモデル [8] などが提案されている。

(b) は最近の信頼性予測で主流となっているものである。データの分析方法は、ロジスティック回帰分析又はベイズ統計を用いたものが一般的である。説明変数としては、コード行数、サイクロマティック数などのプログラムの複雑さ、Halstead の尺度などプロダクト属性によるものが多い [9][10][11] が、開発者の特性から線形判別、ロジスティック回帰、分類木によって混入欠陥数の推定や欠陥を含むモジュールを予測する研究もある [12]。

(c) の代表的なものとして、テスト工程時の欠陥検出過程から残存欠陥数を予測する、ソフトウェア信頼度成長モデル (SRGM) がある ([13], [14], [15], [16] など)。SRGM はそれだけで大きな研究分野を形作っているが、残存欠陥数は出荷前までの欠陥検出過程を分析した結果から得られたもので、筆者の知る限り、稼働後に発見された欠陥数を用いてモデルを構築したり、検証したものはない。

(d) に関する研究の目的は、COCOMO のコストドライバーのように、稼働後のソフトウェアの信頼性 (例えば発見欠陥数) を予測するような要因を体系的に明らかにすることである。残念ながらこの分野の研究は多くなく、筆者の知る限りアンケート調査によるもの 2 件しかない ([17][18])。しかも、これら 2 件の研究は同一の指導者によって同じ方法で実施されたという点でひとつの研究であると言えることができる。すなわち、[18] の研究は、ソフトウェア開発環境の変化を考慮して [17] の研究の 15 年後に全く同じ方法で調査を実施したものである。この研究で

は、さまざまなレベルの開発担当者が 32 の“環境要因”と呼ばれる影響要因の候補に対して信頼性^{*3}への影響の程度を 8 レベルで評価することにより影響度を推定し、テスト網羅率やプログラム仕様の変更頻度が信頼性に影響を与えられることを示している。ここでの分析結果は興味深いものであるが、この研究における信頼性の評価は開発担当者の経験に基づいた主観的なものであり、例えば稼働後に発見された欠陥数などの客観的なデータで評価しているわけではない。稼働後のソフトウェアの信頼性 (例えば発見欠陥数) を予測するような要因を実測データを用いて系統的に調べることができれば、この分野の研究が進むと考えられる。

(d) に関する研究が進んでいない最も大きな理由は、そのような分析が可能な蓄積データがなかったことであろう。そのような分析を可能とするためには、目的変数とする稼働後の欠陥数のデータ及び多くの質的変数を備えた大量のプロジェクトデータが必要となる。ISBSG データリポジトリには 6,000 件以上のプロジェクトデータが揃っていて、稼働後の欠陥数のデータもあるが、[5] で述べられているようにプロセス改善に役立てられるような質的変数はほとんどない。

2 番目の理由は、稼働後の欠陥数には一般に多くのゼロデータが含まれるため、工数のように対数化して回帰分析を行うことができないことである。数%のゼロデータであれば、それらを除いてから稼働後の欠陥数を対数化して回帰分析することによりそれなりの結果を得ることはできる可能性がある。しかし、ゼロデータの割合が 30 % にも及ぶと、それらを除いたデータからの分析結果の信憑性は低下する。

IPA/SEC が 2004 年から収集を始めたデータリポジトリには、現在 4,000 件を超えるプロジェクトデータがあり、収集項目の中にはシステム稼働後 (サービスイン後) のソフトウェアの信頼性を客観的に評価できる発生不具合数、すなわち発生不具合現象数と呼ばれる故障発生数と発生不具合原因数と呼ばれる検出障害数、及びこれら発生不具合数に影響を与えられると思われる多くの変数が含まれる。ただし、IPA/SEC のプロジェクトデータは、他の多くのフィールドデータのリポジトリと同様に、実験計画に基づいてデータを収集したものではないため多くの欠損値を含む。そのため、実際に分析に必要な変数の値をすべてそろえたプロジェクトは多くなく、最低限必要な発生不具合数と FP 規模をともにそろえたプロジェクト数は全体の 15% 以下である。しかしそれでもそれらのプロジェクトデータはこれまで存在しなかったものであり、また統計的分析に耐えられるだけのデータ数は揃っているため、これらのデータを分析することにより (d) の分類の研究に貢献

*1 本論文では特に断らない限り、障害の同義語として使用する

*2 分岐の数と関数呼び出しの数を加えたもの

*3 論文では具体的な定義は行われていない

できる可能性がある。

(d) の分類に属する研究の究極の目的は稼働後の不具合数の予測と考えられるが、不具合数のばらつきは工数以上に大きく、例えば COCOMO のようなモデルで意味のあるものが簡単に構築できるかどうかは現時点ではわからない。今回の分析では、そのような予測モデルの構築に先立ってまずシステム稼働後の信頼性に影響を与える要因を抽出することを目的とした。

変数にはレビュー指摘件数のような量的変数とツール利用の有無などの質的変数があるが、今回は特に IPA/SEC のデータリポジトリで特徴的な質的変数の発生不具合数への影響を分析した。質的変数は開発計画時にプロジェクトでコントロールできるものが多く、いわゆるプロセス改善を行いやすいと考えられる。ゼロ過剰データに対しては、経済学や社会学などの分野で広く用いられている負の二項回帰モデル [19] を用いることにより、不具合数がゼロのプロジェクトを除外することなく分析することができる。

本論文では IPA/SEC が収集したエンタープライズ系ソフトウェアプロジェクトデータ [20] を分析することにより、信頼性（システム稼働後の発生不具合数）に大きな影響を与える質的変数を明らかにした結果を報告する。2 章で分析対象データを紹介する。3 章で分析方法を、4 章で分析結果を示す。5 章で分析結果に対する考察を、6 章でまとめを述べる。

2. 分析対象データ

2.1 分析対象プロジェクト

分析対象プロジェクトは、IPA/SEC で収集したエンタープライズ系ソフトウェアプロジェクト 4,067 件 [20] のうち、次の条件を満たすものを対象とする。

- 1) システム稼働後の発生不具合数が報告されている。
- 2) FP の実測値 (5001_FP 実測値_調整前) が報告されている。
- 3) 開発種別が新規開発である。
- 4) 開発 5 工程 (基本設計, 詳細設計, 製造, 結合テスト, 総合テスト (ベンダ確認)) をすべて実施しているプロジェクトである。

信頼性の定義としてはさまざまなものが考えられるが、本論文ではシステム稼働後の発生不具合数が少ないシステムを信頼性の高いシステムであると定義する。1) の条件はそれを踏まえたものである。

2.2 の (2) で詳述するように、システム稼働後の発生不具合数は規模の影響を強く受ける。質的変数の発生不具合数への影響度を分析するにあたっては、規模を説明変数に加えることによって規模の影響を除く必要がある。2) の条件はそれを踏まえたものである。

これまでの分析経験によると、生産性は開発種別 (新規開発, 改良開発など) で異なる。また、生産性への影響要

因も開発種別で異なる [21]。信頼性でも予備分析の結果で同様な傾向がみられることから、開発種別で層別することが望ましいと考えられる。開発種別の中では新規開発のプロジェクト数が最も多く、1) と 2) の条件を満たすもので比較するとその数は改良開発の約 3 倍あるので、分析の精度の観点から 3) の条件を加える。

生産性そのものや生産性への影響要因の分析には、プロジェクトが携わった工程を揃えて工数データを収集することは必須である。IPA/SEC のエンタープライズ系ソフトウェアプロジェクトでは開発 5 工程を生産性の分析における標準工程としている。信頼性の分析においても開発 5 工程が必須の条件となるかどうかは必ずしも明確ではないが、信頼性への影響要因と生産性への影響要因を比較するためには条件を揃えておくのが望ましいと考えて 4) の条件を加える。

2.2 目的変数と説明変数

(1) 目的変数

目的変数はシステム稼働後の発生不具合数とする。発生不具合数は、不具合現象数と不具合原因数の 2 種類、収集時期が稼働後 1 ヶ月, 3 ヶ月, 6 ヶ月の 3 種類、全部で 6 種類のデータが収集されているが、今回はデータ数の最も多い稼働後 1 ヶ月後の発生不具合現象数 (5267_発生不具合現象数 (合計) 1 ヶ月) を目的変数とする。以下では稼働後 1 ヶ月後の発生不具合現象数を単に不具合数と呼ぶ。

2.1 の 1)~4) を満たすプロジェクト数は 305 である*4。305 件のデータの基本統計量を表 1 に示す。ただし、FP は対数正規分布に従う [22] ため、(常用) 対数変換後の基本統計量である。なお、本論文では基本的に自然対数 (ln) を用いるが、図表など直観的なわかりやすさが重要と考えられる場合は常用対数 (log) も用いる。

- 305 件のプロジェクトのうち 99 件のプロジェクト (32%) で不具合数の値がゼロである。
- log (FP) の歪度・尖度の値は± 1 の範囲にあり、正規分布が否定されない。

(2) 説明変数

ソフトウェアプロジェクトで扱う変数には、比尺度に従う量的変数と、名義尺度または順序尺度に従う質的変数があるが、本論文では基本的に質的変数ひとつを説明変数とする。

量的変数の代表的なものは FP 規模で代表される規模変数である。305 件のプロジェクトに対して不具合数に対する FP 規模の影響を調べる (分析方法は 3 参照) と、係数は 0.918 (つまり不具合数は FP 規模にほぼ比例)、p 値は 0.0% であり、FP 規模は工数に対する影響と同様に不具合数にも大きな影響を与えていることがわかる。そこで FP

*4 1) と 2) の条件を満たすプロジェクト数は 573、1)~3) の条件を満たすプロジェクト数は 369 である。

規模の影響を除去するために、FP 規模をいわゆるコントロール変数として説明変数に加える。

順序尺度に従う質的変数のうちから信頼性に影響を与える可能性のある変数として表 2 に示す 53 個を選んだ。これらの 53 個の変数は、「要求レベル (の高さ)」のようにプロジェクト計画段階で早期にわかるものか、またはツールの利用や作業スペースのようにプロジェクトの遂行にあたってプロジェクト管理者や企業レベルでコントロールできるものである。表中の変数の頭に付いた番号はデータ白書 [20] で付与されているものである。データ白書では類似の変数を 100 番台が同じものでグループ化しているので、これを参考にしながら、表 2 ではその変数の意味を考慮して変数を 5 つのグループに分類している。

名義尺度に従う質的変数のうち、代表的な変数として業種とアーキテクチャに属するものを取りあげる。この中で分析に耐えられるだけのデータ数が揃っている 4 つの業種と 5 種類のアーキテクチャを表 3 に示す。業種は 201_業種_1 で示された値 (分類項目) を日本標準産業分類の大分類のレベルで分類し直したものの*5であり、アーキテクチャは 308_アーキテクチャ_1 に示された値 (分類項目) そのものである。なお、名義尺度に従う変数の値は計画段階ではわかっているもののプロジェクト管理者や組織レベルではコントロールできないものであり、分析結果は参考に過ぎない。

分析結果の頑健性を高めるために、各説明変数は次の 3 つの条件を満たすものとする。

- 1) データ数 (回答数) が 30 件以上ある。
- 2) 各レベルに属する回答数が 10 件以上ある。
- 3) 偏り率 ρ が $-0.7 \leq \rho \leq 0.7$ の範囲にある。

ただし、 ρ は N_1 を上位レベルの回答数、 N_2 を下位レベルの回答数としたとき

$$\rho = \frac{N_1 - N_2}{N_1 + N_2} \quad (1)$$

で定義したものである。 ρ のとり得る範囲は -1 から 1 であり、 $N_1 = N_2$ の場合は $\rho = 0$ となる。

表 1 分析対象データの基本統計量

Table 1 Fundamental statistics of analyzed data

項目	不具合数	FP (*1)
データ数	305	
ゼロデータ数	99	0
平均	14.5	3.09
分散	4,086.1	0.213
最大	999	4.32
最小	0	1.93
歪度	12.8	0.07
尖度	186.6	-0.33

(*1) データ数以外は log (FP) に対する統計量

*5 データ白書 [20] 付録 A.3 参照

表 2 影響要因の候補 (質的変数: 順序尺度に従うもの)

Table 2 Candidates of effective qualitative factors subject to ordinary scale.

分類	変数
開発プロジェクト全般 (11)	111_新技術利用, 112_役割分担_責任所在, 113_達成目標_優先度_明確度合, 1011_定量的出荷品質基準_有無, 1013_第三者レビューの有無, 5241_品質保証体制_基本設計
	114_作業スペース, 115_プロジェクト環境_騒音 計画の評価 (120_コスト, 121_品質, 122_工期)
ツールの利用 (11)	302_業務パッケージ, 403_類似プロジェクト, 404_プロジェクト管理ツール, 405_構成管理ツール, 406_設計支援ツール, 407_ドキュメント作成ツール, 408_デバッグ_テストツール, 409_ CASE ツール, 411_コードジェネレータ, 412_開発方法論利用, 422_開発フレームワーク
ユーザ側 (16)	501_要求仕様_明確さ, 要求仕様変更発生状況 (5115_要件定義, 5116_基本設計, 5117_詳細設計, 5118_製作, 5119_結合テスト, 5120_総合テスト (ベンダ確認), 5121_総合テスト (ユーザ確認)) ユーザ担当者 (502_要求仕様関与, 509_受け入れ試験関与, 503_システム経験, 504_業務経験, 507_設計内容理解度), 505_ユーザとの役割分担_責任所在_明確度合, 506_要求仕様_ユーザ承認有無, 508_設計_ユーザ承認有無
要求レベル (8)	要求レベル (512_信頼性, 513_使用性, 514_性能・効率性, 515_保守性, 516_移植性, 517_ランニングコスト要求, 518_セキュリティ), 519_法的規制
開発担当者 (7)	601_PM スキル
	要員スキル (602_業務分野経験, 603_分析・設計経験, 604_言語・ツール利用経験, 605_開発プラットフォーム使用経験) 1010_テスト体制 (スキルレベル, 要員数)

(注) 番号はデータ白書 [20] で定義されたもの

表 3 影響要因の候補 (質的変数: 名義尺度に従うもの)

Table 3 Candidates of effective qualitative factors subject to nominal scale.

分類	変数 (*1)
業種 (4)	製造業, 情報通信業, 卸売・小売業, 金融・保険業
アーキテクチャ (5)	スタンドアロン, メインフレーム, 2 階層クライアントサーバ, 3 階層クライアントサーバ, インタネット・イントラネット

1)~3) の条件は主に筆者のこれまでの分析経験に基づいて定めたものであり、必ずしも明確な根拠があるわけではない。

2.3 変数変換

(1) 量的変数

FP 規模は一般に対数正規分布に従うので対数変換をする。対数は自然対数を用いる。不具合数は、対数変換をすることのできないゼロの値をもつものが 32% を占めるた

め、対数変換を行わずそのままの値を用いる。

(2) 順序尺度に従う説明変数

順序尺度に従う変数の多くは3つ以上のレベルの値をもつが、各レベル間が等間隔であるという保証はないため、厳密にはレベルを表す数をそのまま用いて回帰分析を行うことはできない。そこで今回の分析では3つ以上のレベルをとりうる変数は隣り合うレベル同士を合併して全体で2レベルにする(2値化する)。このときレベルの若番の方を上位レベル、老番の方を下位レベルと呼ぶ。2レベル化する分割点は複数個あるが、最も偏り率の小さくなるものを分割点とする。

(3) 名義尺度に従う質的変数

名義尺度に従う変数に対しては、変数ごとに次のように2値化する。すなわち、そのプロジェクトが変数名に該当する場合は上位レベルの値を割り当て、該当しない場合は下位レベルの値を割り当てる。例えば、変数「製造業」の値は、そのプロジェクトが製造業向けの場合は上位レベル、製造業以外の場合は下位レベルの値を割り当てる。業種欄が空欄(欠損値)の場合は、業種を分析する場合に限り、そのプロジェクトを分析対象から除外する。アーキテクチャの場合も同様である。

3. 分析方法

3.1 一般化線形モデル

目的変数である不具合数はゼロを含むことが多く、対数変換して(重)回帰分析することができない。このようなデータに対しては、一般化線形モデルに属するポアソン回帰モデルや負の二項回帰モデルを用いる方法が知られている[19]。このふたつを比較すると、ポアソン回帰モデルはパラメータがひとつしかないため、適合度(AICの値)はパラメータ2つをもつ負の二項回帰モデルに比べて悪くなることが多い。

モジュールに含まれる欠陥数のように、特にゼロを含む割合が多い場合はゼロ過剰データと呼ばれ、それらを分析するためにゼロ過剰モデル(ゼロ過剰ポアソン回帰モデルやゼロ過剰負の二項回帰モデル)が提案されている[19]。ゼロ過剰モデルは、目的変数の値として本質的にゼロをとるものがあるという仮定に基づいているが、プログラムモジュールレベルとは異なり、ソフトウェア全体で不具合数がゼロであるという仮定は現実的ではない。

また、ゼロ過剰データに対しては、ハードルモデル(ハードルポアソン回帰モデルやハードル負の二項回帰モデル)も提案されている[19]。ハードルモデルでは、データを2つあるいは3つに分割して、それぞれに最適な分布を当てはめる点に特徴がある。しかし、一般にデータを細かく分割してそれぞれに別々のモデルを適用すると適合度の値が向上するので、このやり方は本質的な解決とは言い難い。

ゼロ過剰モデルやハードルモデルによる適合度の向上度

合は、ポアソン回帰モデルと負の二項回帰モデルの差ほど大きくない[23]ことも考慮して、今回は負の二項回帰モデルを用いて分析する。

3.2 負の二項回帰モデルとは

負の二項回帰モデルは、負の二項分布と呼ばれる次の式で表される確率分布関数を用いて回帰分析を行う方法である[19]。

$$f(y|\mu, \alpha) = \frac{\Gamma(y + \alpha^{-1})}{\Gamma(y + 1)\Gamma(\alpha^{-1})} \times \left(\frac{\alpha^{-1}}{\alpha^{-1} + \mu}\right)^{\alpha^{-1}} \left(\frac{\mu}{\alpha^{-1} + \mu}\right)^y, \quad \alpha \geq 0, y = 0, 1, 2, \dots \quad (2)$$

この分布の平均は μ 、分散は $\mu + \alpha\mu^2$ となる。 α の値を変えることにより分散の大きさを変えることができる。 $\alpha \rightarrow 0$ のとき負の二項分布はポアソン分布に近づく。

一般化線形モデルでは、説明変数(群)の線形結合と目的変数の平均をリンク関数で結合する。負の二項回帰モデルのリンク関数は対数関数であり、例えばプロジェクト*i*の説明変数の値 x_i から μ_i を推定する場合は次の式を用いる(説明変数が複数の場合も同様の式で表すことができる)。

$$\ln \mu_i = \beta_0 + \beta_1 x_i \quad (3)$$

これを用いて*N*個のデータに対する尤度関数

$$L(\alpha, \beta_0, \beta_1) = \prod_{i=1}^N f(y_i|\mu_i, \alpha) \quad (4)$$

が最大になるようにパラメータ $(\alpha, \beta_0, \beta_1)$ を定める。実際には式(4)の両辺の対数をとった対数尤度関数を最大にするようにパラメータを定める。

$$\ln L(\alpha, \beta_0, \beta_1) = \sum_{i=1}^N \ln f(y_i|\mu_i, \alpha) \quad (5)$$

今回の分析では対数変換されたFP規模と、0と1で2値化された質的変数 x の2つの説明変数から次の式で不具合数の平均 μ を推定する。

$$\ln \mu = \beta_0 + \beta_1 \ln FP + \beta_2 x \quad (6)$$

計算は統計ツール*R*を用いて行う。例えば目的変数が y 、説明変数が x_1 と x_2 の負の二項回帰分析の*R*のスク립トは次のようなものである[23]。

```
> library(MASS) # 必要なライブラリの読み込み
> fm <- glm.nb(y~x1+x2, data=dt)
# 負の二項回帰モデルの実行, dtは対象データ
> summary(fm) # 結果の出力
```

3.3 説明変数の選択基準

信頼性に影響を与える質的変数の選択基準を次のように

定め、これらふたつの基準をともに満たすものを目的変数に対する影響要因とする。

(1) p 値

係数がゼロでないかどうか（2つの群の平均値に差があるかどうか）を判断するための p 値の有意水準は、影響の可能性を幅広く確認するため、統計学で標準的に使われている 5% より高い 10%（両側検定）とする。

(2) 回帰係数

質的変数の回帰係数の値が $\ln 1.5 = 0.405$ 以上のものを選択基準とする。これは 2つの群の不具合数比率（5.1 で詳述）が 1.5 倍以上であることを意味する。なお、本論文では、上位レベルを 0、下位レベルを 1 として分析しているため、係数がプラスの場合は、上位レベルの群の方が下位レベルのものよりも不具合数が少ないことを表す。

4. 分析結果

4.1 分析条件を満たさない変数

表 2 及び表 3 であげた説明変数の候補のうち、次の 12 個の変数が 2.2 の (2) で述べた説明変数が満たすべき 3 つの条件のうち、少なくともひとつの条件を満たさないことがわかった。

- アーキテクチャ：スタンドアロン、メインフレーム、2 階層クライアントサーバ。
- 業種：情報通信業、卸売・小売業。
- プロジェクト全般：115_プロジェクト環境_騒音、120_計画の評価（コスト）、122_計画の評価（工期）、1013_第三者レビューの有無。
- ツールの利用：405_構成管理ツール、409_CASE ツール。
- ユーザ側：5115_要求仕様発生変更状況_要件定義。

これらの 12 個の変数は分析の対象外とし、以降は順序尺度に従う変数 46 個と名義尺度に従う変数 4 個を分析対象とする。

4.2 信頼性への影響要因

信頼性への影響要因として選択された変数、すなわち 2.2 の (2) で述べたデータ件数に関する 1)~3) の条件、及び 3.3 の説明変数の選択に関する 1) と 2) の選択基準をすべて満たす質的変数を表 4 に示す。

表 4 から次のことがわかる。なお、以下の分析結果はすべて各質的変数から FP 規模の影響を除去した後（FP 規模が一定とした場合）のものである。

- 業種別では、金融・保険業は他の業種に比べて信頼性が高い（稼働後 1 ヶ月の発生不具合現象数が少ない）。
- アーキテクチャでは差がみられない（表 4 にリストアップされるものはない）。
- プロジェクト全般では、113_（納期・品質等の）達成目標_優先度_明確度合が非常に明確な場合、及び 121_計画の評価（品質）で品質目標が明確で実行可能性を

検討済みの場合は信頼性が高い。

- ツールを利用することにより信頼性が高くなるものは、408_デバッグ_テストツールである。逆に 411_コードジェネレータの利用は信頼性を低下させる。
- ユーザ側では、501_要求仕様が明確である場合、5116_基本設計、5117_詳細設計、5119_結合テスト、5120_総合テスト（ペンダ確認）の各工程それぞれで要求仕様変更が発生しない場合、502_ユーザ担当者が要求仕様に関与している場合、504_ユーザ担当者の業務経験が豊富な場合、507_ユーザ担当者の設計内容理解度が高い場合が、いずれも信頼性が高い。
- 要求レベルでは、514_要求レベル_性能・効率性が高い場合、515_要求レベル_保守性の高い場合が、いずれも信頼性が高い。
- 開発担当者に関しては、1010_テスト体制でテスト要員のスキルが高い方が信頼性は高い。
- 要求仕様変更の発生、特に詳細設計及び結合テストでの変更発生の回帰係数が大きく、最も信頼性に悪影響を与えている。
- 業種を除いた分類別では、ユーザ側に属する変数の数が、絶対数（8 個）からみても比率（ $0.53 = 8/15$ ）からみても最も大きい。

5. 考察

5.1 不具合数比率

式 (6) におけるパラメータ $\beta_0, \beta_1, \beta_2$ の最尤推定値（切片及び回帰係数）をそれぞれ b_0, b_1, b_2 とすると、ある FP 規模における上位レベル及び下位レベルそれぞれの不具合数の平均 μ_i^+ 及び μ_i^- は、質的変数が上位レベルで 0、下位レベルで 1 の値をとることから、

$$\ln \mu_i^+ = b_0 + b_1 \ln FP \quad (7)$$

$$\ln \mu_i^- = b_0 + b_1 \ln FP + b_2 \quad (8)$$

となる。

ここで FP 規模が等しいと仮定した場合の μ_i^- と μ_i^+ の比を「不具合数比率」と呼ぶことにすると、式 (7) と式 (8) より

$$\ln \mu_i^- - \ln \mu_i^+ = \ln \frac{\mu_i^-}{\mu_i^+} = b_2 \quad (9)$$

となり、不具合数比率は

$$\frac{\mu_i^-}{\mu_i^+} = e^{b_2} \quad (10)$$

で表すことができる。

表 4 に影響要因の候補の不具合数比率を示す。表 4 によると、例えば 501_要求仕様_明確さでは、明確な場合に比べて明確でない場合は不具合数の平均が 1.8 倍多くなることを示している。411_コードジェネレータの利用のように係

表 4 影響要因として選択された質的変数
 Table 4 Qualitative variables selected as effective factors.

分類	変数名	内容		データ数			質的変数		不具合数比率	
		上位レベル	下位レベル	総数	上位レベル	下位レベル	回帰係数	p 値 (%)		
業種	金融・保険業	金融・保険業	左記以外の業種	305	90	215	0.66	0.3	1.9	
プロジェクト全般	113_達成目標_優先度_明確度合	a: 非常に明確	b: 概ね明確 + c: やや不明確 + d: 不明確	160	55	105	0.50	9.4	1.6	
	121_計画の評価(品質)	a: 品質目標が明確で実行可能性を検討済み	b: 品質目標が不明確, または実行可能性を未検討 + c: 計画なし	249	205	44	1.02	0.1	2.8	
ツール利用	408_デバッグ_テストツール利用	a: 有り	b: 無し	139	67	72	0.83	1.2	2.3	
	411_コードジェネレータ利用			130	37	93	-0.71	5.6	1/2.0	
ユーザー側	501_要求仕様_明確さ	a: 非常に明確 + b: かなり明確	c: ややあいまい + d: 非常にあいまい	193	130	63	0.60	1.2	1.8	
	要求仕様変更発生状況	5116_基本設計	a: 変更なし	b: 軽微な変更が発生 + c: 大きな変更が発生 + d: 重大な変更が発生	46	14	32	1.17	3.2	3.2
		5117_詳細設計			45	16	29	1.38	0.9	4.0
		5119_結合テスト			36	19	17	1.51	1.3	4.5
		5120_総合テスト(ベンダ確認)			42	24	18	0.89	8.9	2.4
	502_ユーザ担当者_要求仕様関与	a: 十分に関与 + b: 概ね関与	c: 関与が不十分 + d: 未関与	167	100	67	0.64	0.8	1.9	
504_ユーザ担当者_業務経験	a: 十分に経験	b: 概ね経験 + c: 経験が不十分 + d: 未経験	84	34	50	0.72	6.1	2.1		
507_ユーザ担当者_設計内容理解度	a: 十分に理解 + b: 概ね理解	c: 理解が不十分 + d: 全く理解していない	89	69	20	0.97	3	2.6		
要求レベル	514_性能・効率性	a: 極めて高い + b: 高い	c: 中位 + d: 低い	122	69	53	0.74	2.2	2.1	
	515_保守性			88	27	61	1.13	0.8	3.1	
開発担当者	1010_テスト体制(スキルレベル)	a: スキル要員ともに十分 + b: スキルは十分, 員数は不足	c: スキルは不足, 員数は十分 + d: スキル, 員数ともに不足	62	43	19	0.82	4.4	2.3	

数が負の場合は、逆数で示している。要求仕様変更発生状況の不具合数比率が、2.4~4.5 と高い値を示している。特に 4117_詳細設計で 4.0, 4119_結合テストで 4.5 と高く、仕様変更のあるなしで不具合数が 4 倍以上違うことがわかる。

FP 規模に対する不具合数の散布図例を図 1 に示す。図 1 において同一 FP 規模の実線及び破線の高さの比が不具合数比率である。推定した平均値の差が有意であっても実測値のばらつきは大きい。

5.2 要因間の相関

表 4 で示した要因同士は必ずしも独立ではない。各要因同士の分割表を作成して独立性の検定を行ったところ 120 の組合せ中 13 (11%) の組合せが有意水準 1% で独立ではないという結果が得られた。それら 13 の組合せのうち 3 つの組合せの ϕ 係数が、関連が大あるいは強い関連があると言われる 0.5 を超えていた。いずれも要求仕様変更発生状況に関する 4 つの変数間の組合せであり、 ϕ 係数はそれ

ぞれ 5116_基本設計と 5117_詳細設計間で 0.75, 5117_詳細設計と 5119_結合テスト間で 0.52, 5119_結合テストと 5120_総合テスト(ベンダ確認)間で 0.84 であった。これら 4 つの変数に対して主成分分析を行ったところ、第一主成分の各変数の重みは 4 つの変数でほぼ等しく、第一主成分だけで総分散の 67% を説明できることから、要求仕様変更発生状況に関する変数はどれかひとつで代表させてよいと思われる。それ以外の要因同士で相関係数が 0.5 を超えるものはなく、いずれかの要因を除く必要はないと考えられる。

5.3 外れ値の影響

(1) FP 規模

表 4 にリストアップされた変数のすべての群(上位レベル, 下位レベル, その和集合)について FP 規模の(対数の)歪度と尖度を求めたところ、歪度はすべての群で ± 1 の範囲に入っていたが、尖度は 48 個中 4 個が ± 1 の範囲外であった。しかし絶対値が最も大きな尖度は -1.5 であり、

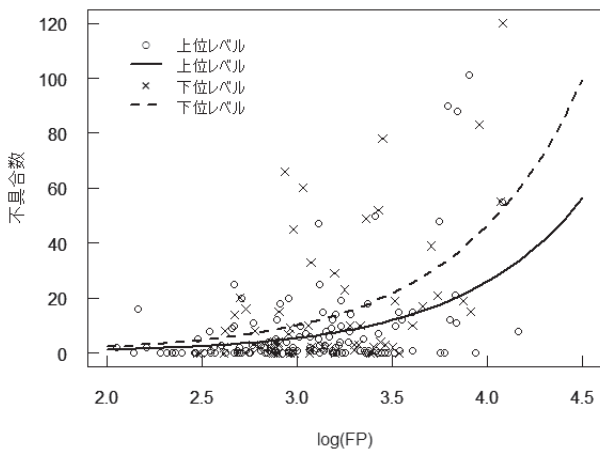


図 1 FP 規模と不具合数の散布図例 (501_要求仕様の明確さ)
Fig. 1 Scatter plot graph of FP and number of failures for clarity of requirement specifications.

どの群においても FP 規模は正規分布から大きく逸脱している可能性は低いと考えられる。

(2) 不具合数

不具合数が 999 と 415 という大きな値をもつプロジェクトが 2 つある。それ以外の 303 件のプロジェクトの不具合数はすべて 120 件以下であることから、この 2 件のデータは外れ値である可能性がある。

不具合数が正の値をもつ 206 件のプロジェクトにおける log(FP) 規模と log(不具合数) の散布図を図 2 に示す。log(不具合数) が 3.0 と 2.5 に位置する 2 つの点 (図 2 中の △) が不具合数 999 件と 415 件に対応するが、これらを外れ値とみなすべきかどうかを図 2 から判断するのは難しい。

そこで、305 件のデータのうち不具合数が大きい方から半分の 153 件のデータ集合を選択し、四分位から外れ値を求める方法で上記 2 つの値が外れ値か否かを調べた。まず 153 件のデータを常用対数で変換して最も小さい値を全体集合 (305 件) の中央値とし、選択した 153 件のデータ集合の中央値を全体集合の上ヒンジの値とする。外れ値の上側境界点は一般に、上ヒンジ + (上ヒンジ - 下ヒンジ) × 1.5 の点とするが、今回は下ヒンジの値がわからないので、上ヒンジ + (上ヒンジ - 中央値) × 3 の点を境界点とする。今回のデータでは、中央値 = 0.301、上ヒンジの値 = 1.041、境界点 = 3.262 となり、境界点は log 999 = 3.000 より大きい。すなわち、415、999 の不具合数はいずれも外れ値とみなす必要はない。

しかし、これら 2 つの値が異常値すなわち「しかるべき理由によりデータの他の部分と異質と考えられるので修正又は削除すべきと判定される値」[24] である可能性も否定できないので、これら 2 つの値を持つプロジェクトを除い

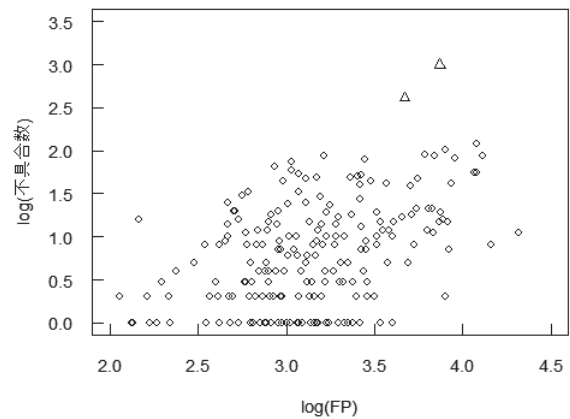


図 2 FP と不具合数 (正のもの) の散布図
Fig. 2 Scatter plot graph of FP and positive number of failures.

表 5 大きな不具合数をもつプロジェクトの影響

Table 5 Effects of projects with large number of failures.

変数名	2 件のデータ	データ数		質的変数	
		上位レベル	下位レベル	回帰係数	p 値 (%)
金融・保険豪	含む	90	215	0.66	0.3
	除く		213	0.47	2.9
113_達成目標_優先度_明確度合	含む	55	105	0.50	9.4
	除く		104	0.38	17.6
121_計画の評価 (品質)	含む	205	44	1.02	0.1
	除く		42	0.46	13.0
408_デバッグ_テストツール利用	含む	67	72	0.83	1.2
	除く	66		1.28	0.0
411_コードジェネレータ利用	含む	37	93	-0.71	5.6
	除く	36		-0.28	43.0

て再分析し、結果を比較した (表 5)。表 4 に記載されていて表 5 に記載されていない変数は、大きな不具合数をもつ 2 つのプロジェクトがともにそれらの変数に対しては値が欠損していて再分析の対象とならなかったものである。

表 5 の p 値をみると 2 件のデータを除くことにより、408_デバッグ_テストツールの利用は p 値が小さくなり、それ以外の要因は p 値が大きくなっている。すなわち、113_達成目標_優先度_明確度合では 9.4% から 17.6% に、121_計画の評価 (品質) では 0.1% から 13.0% に、411_コードジェネレータ利用では 5.6% から 43.0% となっている。このうち、113_達成目標_優先度_明確度合と 121_計画の評価 (品質) は p 値が大きくなってはいるもののいずれも 20% 以下であり、4.2 で得られた結果のまま影響要因と考えてよいと思われる*6。しかし、411_コードジェネレータ利用では

*6 筆者の経験では工数予測等で p 値が 20% 以下の説明変数を加えると一般に自由度調整済みの決定係数は増加する。つまり、そのような説明変数は有効であることが多い。

変動が大きく1件のデータを除くとp値が43.0%になっていることから、4.2で得られた結果すなわちコードジェネレータの利用は信頼性を低下させるという結果は保留した方がよいと思われる。

5.4 Zhuらの結果との比較

Zhuらは実際にソフトウェアを開発している人たちからのアンケートにより、32の環境変数の信頼性への影響度を調査した[18]。32の環境要因とSECデータの収集項目は必ずしも十分な対応はとれていない。32の環境要因の中にはテストに関連する項目を6項目(19%)、プロセッサや記憶装置などの開発環境に関する項目を5項目(16%)、プログラムの複雑さや要件と詳細設計書の関連などのプロダクトの内容に関する項目を2項目(6%)含むなどの特徴がみられる。SECデータに対応する又は類似の項目が存在するものは、全体の半分程度である。

今回の分析結果を、Zhuらの分析で影響度が大きいと判断された上位10個の環境変数との比較した(表6)。なお、Zhuらの結果と比較するために、量的変数であるテスト工数比率及びコード再利用率について追加分析した。

- テスト工数比率が高い(テスト工数が多い)、要求仕様変更がない(頻度が少ない)、ユーザの業務経験が豊富(分野知識が豊富)である場合は、それぞれ信頼性が向上するという点では分析結果が一致した。
- 今回の分析結果では、デバッグ_テストツールを利用する場合は信頼性が高くなるという結果であったが、テストツールの利用はZhuらの分析結果では上位10位には入らなかった。
- Zhuらの分析結果ではコードの再利用率及びプログラムのスキルが上位10位に入っていたが、今回の分析では有意な結果は得られなかった。

Zhuらの結果と今回の分析結果では影響要因と判断された変数に差はあるものの、少なくとも相反するような結果は得られていない。

5.5 信頼性と生産性への影響要因の比較

信頼性への影響要因と生産性への影響要因の比較結果を表7に示す。生産性への影響要因の選択方法は、信頼性への影響要因の選択法と同様の方法を用いている。すなわち、まず2.1で述べた1)~4)までの条件を満たすプロジェクトを選択する。次に、目的変数を工数とし、2.2(2)で述べた1)~3)の条件を満たす各質的変数に対して工数、FPともに対数変換を行った後、線形回帰分析を行って3.3の1)と2)の選択基準を満たすものを選ぶ(2014年までの一世代前のデータに対する分析結果は[21]に詳しく述べられている)。

生産性の分析では、工数が対数正規分布に従うという前提で分析を行っている。対数化したあとの分布は正規分布

表6 Zhuらの分析結果との比較

Table 6 Comparison of analysis results from IPA/SEC data and by Zhu et al.

分類	IPA/SEC データの分析に基づく影響要因	Zhu らの上位10の影響要因 [18]	
プロジェクト全般	113_達成目標_優先度_明確度合	-	
	121_計画の評価(品質)	-	
	テスト工数比率(*1)	テスト工数	
	-	テスト網羅率 要件と詳細設計の関係 プログラムの複雑さ テスト環境	
ツール利用	408_デバッグ_テストツール	<テストツール>	
	-	テスト方法論	
	<コード再利用率>	コード再利用率	
ユーザ側	501_要求仕様_明確さ	-	
	要求仕様発生変更状況	5116_基本設計	仕様変更頻度
		5117_詳細設計	
		5119_結合テスト	
		5120_総合テスト(ベンダ確認)	
	ユーザ担当者	502_要求仕様関与	-
504_業務経験		対象領域知識	
507_設計内容理解度		-	
要求レベル	514_性能・効率性	-	
	515_保守性	-	
開発担当者	<602~605 要員スキル>	プログラムのスキル	
	1010_テスト体制(スキルレベル)	-	

(注)“-”は分析対象となる項目がないか、又は項目があってもデータ数が少ないため分析対象外となったもの、<>内は項目はあるが有意でないか、又は上位10の影響要因と判断されなかったもの、(*1)は量的変数のため当初は分析対象としていなかったが、比較のために追加分析した結果有意であったもの

に従い、その平均値は元の対数正規分布では中央値に等しい。従って、生産性における工数比率は、FP規模の影響を除いた(FP規模が同一と仮定した)場合の元のスケールにおける下位レベルと上位レベルそれぞれの中央値の比に等しい。なお、表7では、係数がマイナスのものに対しては比較しやすいように工数比率を逆数で示している。

表7から次のことがわかる。

- 最も特徴的なことは、ユーザ側の開発へのさまざまな関与及び要求仕様変更のないことが信頼性を向上させるのに対し、総合テスト(ベンダ確認)での要求仕様変更以外は生産性への影響要因とはなっていないことである。しかし、IPA/SECのデータにはユーザ側の工数が計上されていないので、開発へのユーザ側のさまざまな関与は実際には生産性という点ではマイナスではないかと思われる。
- 信頼性向上にも生産性向上にも寄与する要因は、113_達

表 7 信頼性と生産性の影響要因の比較

Table 7 Comparison of effective factors for reliability and productivity.

分類	変数名	信頼性	生産性
		不具合数比率	工数比率 (*1)
業種	製造業	-	1.9
	金融・保険業	1.9	1/1.9
プロジェクト全般	112_役割分担_責任所在	-	1.7
	113_達成目標_優先度_明確度合	1.6	1.7
	121_計画の評価 (品質)	2.8	-
	5241_品質保証体制_基本設計	-	1.9
ツール利用	404_プロジェクト管理ツール	-	1/1.9
	405_構成管理ツール	-	1/1.6
	407_ドキュメント作成ツール	-	2.4
	408_デバッグ_テストツール利用	2.3	1/1.9
ユーザー側	501_要求仕様_明確さ	1.8	-
	5116_要求仕様変更発生状況_基本設計	3.2	-
	5117_要求仕様変更発生状況_詳細設計	4.0	-
	5119_要求仕様変更発生状況_結合テスト	4.5	-
	5120_要求仕様変更発生状況_総合テスト (ベンダ確認)	2.4	1/1.5
	502_ユーザ担当者_要求仕様関与	1.9	-
	504_ユーザ担当者_業務経験	2.1	-
	507_ユーザ担当者_設計内容理解度	2.6	-
	512_要求レベル_信頼性	-	1/1.7
	514_要求レベル_性能・効率性	2.1	-
要求レベル	515_要求レベル_保守性	3.1	-
	518_要求レベル_セキュリティ	-	1/2.3
開発担当者	1010_テスト体制 (スキルレベル)	2.3	-

(*1) 逆数は係数がマイナスのもの

成目標_優先度_明確度合 (が非常に高い場合) である。

- ユーザー側の関与以外の要因で、生産性には影響を与えないが信頼性向上に寄与する要因は、121_計画の評価 (品質) (品質目標が明確で実行可能性を検討済み)、及び 1010_テスト体制 (スキルレベルが高い場合) である。
- 408_デバッグ・テストツールを利用すると生産性は低下するが信頼性は向上する。
- 要求レベルに関する変数のうち、514_性能・効率性と 515_保守性が信頼性向上に寄与し、512_信頼性と 518_セキュリティが生産性の低下を招く。有意水準には達していないものの、性能・効率性及び保守性の工数比率はそれぞれ 1/1.3 と 1/1.5、信頼性及びセキュリティの不具合数比率はいずれも 1.7 であり、すべての要求レベルに関する変数が同じ傾向を示している。このことから、要求レベルが高いと生産性は低下するが信頼

性は向上すると言える。

- 金融・保険業のソフトウェアは他の業種に比べて、生産性は低い (開発に工数がかかる) が信頼性は高い。

5.6 線形回帰モデルによる分析結果との比較

今回分析対象としたデータに対して線形回帰モデルを用いて分析した結果を表 8 に示す。ただし、不具合数がゼロの値をもつプロジェクトは分析対象から除外し、不具合数及び FP ともに対数変換をした後で線形回帰分析を行った。

負の二項回帰モデルを適用した場合に影響要因の候補として選択された 16 個の質的変数 (表 4) のうち、次の 9 個は線形回帰モデルによる分析でも影響要因の候補となった。

- 金融保険業
- 408_デバッグ_テストツール利用
- 501_要求仕様_明確さ
- 5116_要求仕様変更発生状況_基本設計
- 5117_要求仕様変更発生状況_詳細設計
- 5119_要求仕様変更発生状況_結合テスト
- 502_ユーザ担当者_業務経験
- 516_要求レベル_保守性
- 1010_テスト体制 (スキルレベル)

このうち、502_ユーザ担当者_業務経験と 1010_テスト体制 (スキルレベル) は、負の二項回帰モデルによる分析結果よりも係数・p 値ともに向上した。すなわち係数は大きくなり、p 値は小さくなった。

また、

- 603_要員スキル_分析・設計経験

は負の二項回帰モデルによる分析では影響要因の候補から外れたが、線形回帰モデルでは影響要因の候補となった。ただし、負の二項回帰モデルによる分析でも、603_要員スキル_分析・設計経験の p 値は 17.2% であり、影響要因の候補にかなり近い質的変数であった。

負の二項回帰モデルを適用することにより、影響要因の候補として選択されたが、線形回帰モデルによる分析では p 値が有意水準として定めた 10% を超えた質的変数は次の 7 個である。

- 113_達成目標_優先度_明確度合
- 121_計画の評価 (品質)
- 411_コードジェネレータ利用
- 5120_要求仕様変更発生状況_総合テスト (ベンダ確認)
- 502_ユーザ担当者_要求仕様関与
- 502_ユーザ担当者_設計内容理解度
- 516_要求レベル_性能・効率性

以上のことから、今回の 305 件のプロジェクトデータに対して不具合数がゼロの 99 件 (32%) のプロジェクトを除いて線形回帰モデルで分析すると、 $(9+1)/16 = 62.5\%$ すなわち 2/3 程度しか影響要因の候補を選択することができないことがわかる。

表 8 線形回帰モデルによる分析結果
 Table 8 Results of a linear regression analysis.

分類	変数名	データ数 (*1)		質的変数		影響 要因 (*2)	
		上位 レベル	下位 レベル	回帰 係数	p 値 (%)		
業種	金融保険業	59 (31)	147 (68)	0.44	2.4	▽	
プロ ジェ クト 全般	113_達成目標_ 優先度_明確度合	32 (23)	76 (29)	0.19	48.5	-	
	121_計画の評価 (品質)	125 (80)	32 (12)	0.41	10.6	-	
ツール 利 用	408_デバッグ_ テストツール利用	35 (32)	52 (20)	0.51	8.0	▽	
	411_コードジェネ レータ利用	30 (7)	53 (40)	-0.37	23.2	-	
ユー ザ側 状 況	501_要求仕様_ 明確さ	88 (42)	48 (15)	0.50	2.2	▽	
	要求 仕様 変 更 発 生	5116_基本 設計	10 (4)	21 (11)	0.92	8.2	▽
		5117_詳細 設計	11 (5)	19 (10)	1.02	4.0	▽
		5119_結合 テスト	10 (9)	12 (5)	1.14	6.2	▽
		5120_総合 テスト (ベン ダ確認)	13 (11)	14 (4)	0.29	55.9	-
	502_ユーザ担当者 _要求仕様関与	60 (40)	58 (9)	0.27	21.0	-	
	504_ユーザ担当者 _業務経験	21 (13)	35 (15)	0.77	3.3	△	
	507_ユーザ担当者 _設計内容理解度	39 (30)	17 (3)	0.46	23.2	-	
要求 レ ベ ル	514_性能・効率性	46 (23)	31 (22)	0.41	16.6	-	
	515_保守性	18 (9)	37 (24)	0.66	9.1	▽	
開発 担 当 者	603_要員スキル_ 分析・設計経験	70 (32)	15 (11)	0.63	7.2	○ (*3)	
	1010_テスト体制 (スキルレベル)	33 (10)	14 (5)	1.00	1.9	△	

(*1) 括弧内は不具合数がゼロのため分析対象から除外したデータ数 (別掲)。

(*2) - : 影響要因と判断されなかったもの, ▽ : 影響要因と判断されたが係数・p 値とも負の二項回帰モデルの分析結果より悪化したもの, △ : 係数・p 値とも向上したもの, ○ : 新たに影響要因と判断されたもの。

(*3) 負の二項回帰モデルによる分析では, 係数が 0.68, p 値が 17.2%であった。

6. おわりに

システム稼働後の不具合数は, 早い段階では多くのプロ

ジェクトでゼロである。そのため, 生産性分析における工数のように, 稼働後の不具合数を目的変数にとって対数変換をして回帰分析を行うことができない。

本論文では, 経済学や社会学で用いられている負の二項回帰モデルを用いることにより, 新規開発プロジェクトにおけるソフトウェアの信頼性 (システム稼働後の不具合数) への影響要因を分析した。今回の分析では生産性への影響要因の候補と比較するために, 開発 5 工程に携わっているプロジェクトのみを対象とした。その結果, 次のような場合に信頼性が向上することが明らかとなった。

- (納期・品質等の) 達成目標や優先度が非常に明確な場合
- 計画段階で品質目標の実行可能性を検討済みの場合
- デバッグ_テストツールを利用する場合
- 要求仕様が明確な場合
- ユーザ担当者が要求仕様に関与する場合
- 要求仕様変更が発生しない場合
- ユーザ担当者の業務経験が豊富な場合
- ユーザ担当者の設計内容理解度が高い場合
- 性能・効率性への要求レベルが高い場合
- 保守性への要求レベルが高い場合
- テスト体制においてスキルレベルが高い場合

ユーザ側と分類した質的変数が影響要因全体の半分以上を占め, 信頼性向上にはユーザ側の豊富な業務経験や適切な関与が重要な役割を果たすことが明らかとなった。

また, 信頼性と生産性の影響要因を比較した結果, 次のことが明らかとなった。

- 達成目標_優先度が非常に明確な場合は, 生産性も信頼性も向上する。
- 要求仕様変更が発生しない場合は, 生産性への影響は見られないが, 信頼性は各段に向上する。
- デバッグ_テストツールを利用すると生産性は低下するが, 信頼性は向上する。
- 要求レベルが高いと生産性は低下するが, 信頼性は向上する。
- ユーザ担当者の関与は, ユーザ工数を含めたトータルな生産性ではマイナスになっている可能性もあるが, 信頼性は向上する。

さらに, 不具合数がゼロのプロジェクトを除外してから不具合数及び FP 規模を対数変換して線形回帰モデルで質的変数の影響要因を分析したところ, 負の二項回帰モデルで得られた 16 個の影響要因に対して新たに 1 個の影響要因を抽出したが 7 個の要因は影響要因とみなされなかった。このことから不具合数がゼロのプロジェクトを除外すると抽出される影響要因が少なくなる (今回の例では 2/3 に減少する) ことがわかった。

今回の分析では係数の値は設定した基準に達しているものの, p 値が有意水準に達しなかった変数があった。これ

らは検出力が不足しているケースと考えられ、今後データの充実により、さらに多くの影響要因を抽出できる可能性がある。

謝辞 本研究は東海大学とIPA/SECが共同で実施したものである。IPA/SECの松本所長、山下リーダー並びに研究員の方々のご協力に深く感謝致します。

参考文献

- [1] Boehm, B. W.: Software Engineering Economics, Prentice-Hall, Inc.(1981).
- [2] Boehm, B. et al.: Software Cost Estimation with Cocomo II, Prentice-Hall, Inc.(2000).
- [3] Akiyama, F.: An Example of Software System Debugging, Information Processing 71, North-Holland, pp.353-359(1972).
- [4] Halstead, M. H.: Elements of Software Science, Chap.11, Elsevier, North-Holland(1977).
- [5] Fenton, N., Neil, M., Marsh W., Hearty, P., Radlinski, L., D. and Krause, P : Project Data Incorporating Qualitative Factors for Improved Software Defect Prediction, *Int. Workshop on Predictor Models in Software Engineering (PROMISE '07)*(2007).
- [6] 角田雅照, 玉田春昭, 森崎修司, 松村知子, 黒崎章, 松本健一: コード指摘密度を用いたソフトウェア欠陥密度予測, 情報処理学会論文誌, Vol. 50, No. 3, pp. 1144-1155 (2009).
- [7] 角田雅照, 門田暁人, 松本健一: 組込みソフトウェア開発における設計関連メトリクスに基づく下流試験欠陥数の予測, SEC Journal, Vol.11, No.2, pp.16-23 (2015).
- [8] 小室睦, 薦田憲久: ビアレビューデータに基づく品質予測モデル, 電子情報通信学会誌 D, Vol. J94-D, No. 2, pp. 439-449 (2011).
- [9] Khoshgoftaar T. M., and Gao, K.,: Count Models for Software Quality Estimation, IEEE Tr. Reliability, Vol.56, No.2, pp.212-222(2007).
- [10] 亀井靖高, 森崎修二, 門田暁人, 松本健一: 相関ルールとロジスティック回帰分析を組み合わせた fault-prone モジュール判別方法, 情報処理学会論文誌, Vol. 49, No. 12, p.p. 3954-3966 (2008).
- [11] Vandecruys, O., Martens, D., Baesens, B., Mues, C., Backer, M. D. and Haesen, R. : Mining Software Repositories for Comprehensible Software Fault Prediction Models, J. Systems and Software, Vol.81, pp.823-839(2008).
- [12] 松本真佑, 亀井靖高, 門田暁人, 松本健一: 開発者メトリクスに基づくソフトウェア信頼性の分析, 電子情報通信学会論文誌 D, Vol. J39-D, No. 8, pp. 1576-1589 (2010).
- [13] Goel, A.L. and Okumoto, K.: Time -Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures, IEEE Trans. Rel., Vol.R-28, No.3, pp.206-211(1979).
- [14] Yamada, S., Ohba, M. and Osaki, S.: S-Shaped Reliability Growth Modeling for Software Error Detection, IEEE Trans. Rel., Vol.R-32, No.5, pp.475-478(1983).
- [15] Furuyama, T. and Nakagawa, Y.: A Manifold Growth Model that Unifies Software Reliability Growth Models, Int. J. of Reliability, Quality and Safety Engineering, Vol.1, No.2, pp.161-184(1994).
- [16] 岡村寛之, 安藤光昭, 土肥正: 一般化ガンマソフトウェア信頼性モデル, 電子情報通信学会論文誌 D-I, Vol. J-87-D-I, No. 8, pp. 805-814 (2004).
- [17] Zhang, X. and Pham, H.: An Analysis of Factors affecting Software Reliability, J. Sys. Software, Vol.50, No.1, pp.43-56 (2000).
- [18] Zhu, M., Zhang, X. and Pham, H.: A Comparison Analysis of Environmental Factors Affecting Software Reliability, J. Sys. Software, Vol.109, pp.150-160 (2015).
- [19] Cameron, A. C. and Trivedi, P. K.: Regression Analysis of Count Data, 2nd ed., Cambridge Uni. Press, p.566(2013).
- [20] 独立行政法人情報処理推進機構 (IPA) ソフトウェアエンジニアリングセンター (SEC) 監修: ソフトウェア開発データ白書 2016-2017 (2016).
- [21] 古山恒夫: 工数に影響を与える質的変数とその影響度, SEC journal, 第 11 巻, 第 4 号 (通巻 47 号), pp. 40-47 (2016).
- [22] 古山恒夫: ソフトウェアプロジェクトデータの量的変数に関する分析の一指針と分析事例, SEC journal, 第 7 巻, 第 3 号 (通巻 26 号), pp. 105-111 (2011).
- [23] Zeileis, A., Kleiber, C. and Jackman, S.: Regression Models for Count Data in R, J. Statistical Software, Vol.27, Issue 8, pp.1-21(2008).
- [24] 竹内 啓編集: 統計学辞典, p. 480, 東洋経済新聞社 (1989).