

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}

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。稼働後不具合数はゼロの値をもつプロジェクトが30%程度占めるため、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、信頼性を向上させるためには、ユーザ側の関与が重要な役割を果たすことがわかった。

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

TSUNEO FURUYAMA^{†1}

Abstract: The qualitative factors affecting reliability of enterprise software are clarified by analyzing the enterprise software project data collected by IPA/SEC. The ordinal linear regression model can't be applied to analysis of data including the number of software faults after operation since about 30% of them are zero that can't be transformed logarithmically. The negative binomial regression model that is used in the area of economics or sociology is applied to reliability analysis. The analysis results show user's commitment has important role for improving 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 COCOMO [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。現在でも信頼性は品質特性の中でも重要な位置を占めている。

信頼性の予測を目的という観点からみると、次の3つに分類される。

- (1) テスト工程時の欠陥数の予測
- (2) 欠陥を含むモジュールの予測または判定。
- (3) 出荷時の製品に含まれる欠陥数の予測

(1) 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が提案されていた[4]。最近では、コード行数やレビュー指摘件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ベイジアンネットの手法を用いて分析した結果をもとに、テスト工程時の欠陥数を予測するモデルも提案されている[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研究[6]、ピアレビューデータから欠陥摘出数を予測するモデル[7]もこの範疇に入ると考えられる。

(2) 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複雑さ、Halstead の尺度などプロダクト属性によるものが多い[8][9][10]。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 Fault-prone モジュールを予測する研究もある[11]。

^{†1} 東海大学
Tokai University

(3) の代表的なものに、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 ([12][13][14][15]など多数)。

これらの信頼性予測モデルは、いずれも出荷前までのプロジェクトデータを分析した結果から得られたもので、筆者に知る限り、(3)の出荷時の製品に含まれる欠陥数の予測を稼働後に発見された欠陥数で検証したものはない。また、説明変数は多くの場合、サイクロマティック数や Halstead の尺度などのプロダクト属性であり、プロセス属性としてはレビュー指摘件数などの量的変数だけであることが多い。例えば、開発担当者のスキルレベルは高いか、ツールを利用しているかなどの質的変数を説明変数として取り上げているものには、テストスタッフの経験などを含む[16]の論文、開発者の特性を説明変数としてとりあげた[11]などの論文がみられるだけである。つまり COCOMO のコストドライバのような、信頼性に影響を与える質的要因は筆者の知る限りこれまで明らかにはされていない。

その最も大きな理由は、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 (又は障害数) のデータと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、[17]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 (又は障害数) に与える規模の影響が工数に比べて小さく、規模以外の影響要因を含めても現実的な予測精度をもつ予測モデルが構築しにくいと思われることである。

3 番目の理由は、稼働後の欠陥数 (又は障害数) 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 (又は障害数) 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、収集項目の中にシステム稼働後 (サービスイン後) の発生不具合数 (現象数と原因数) だけでなく、プロセス改善に役立つと思われる多くの質的変数を含む。ここで現象数は障害数に、原因数は欠陥数に対応する。これを分析することにより、出荷後製品の信頼性に影響を与える要因を抽出できると考えられる。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[18]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[19]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 [19]のうち、次の条件を満たすものを対象とする。

開発種別が新規開発である。

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。

FP の実測値 (5001_FP 実測値_調整前) が報告されている。

システム稼働後の発生不具合数が報告されている。

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 ヶ月後の不具合現象数とする。以下では稼働後 1 ヶ月後の不具合現象数を単に不具合数と呼ぶ。

2.1 の ~ を満たすプロジェクト数は 305 である。305 件のデータの基本統計量を表 1 に示す。

- 305 のプロジェクトのうち 99 のプロジェクト (32%) で不具合数の値がゼロである。
- 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。

表 1 分析対象データの基本統計量

Table 1 Fundamental statistics of analyzed data

	不具合数	FP
データ数	305	
平均	14.5	3.09
最大	999	4.32
最小	0	1.93
ゼロデータ数	99	0
歪度	-	0.07
尖度	-	-0.33

(注) 不具合数は変換なし, FP は常用対数変換後

(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。不具合数に対する FP 規模の影響は、工数に対する影響ほど大きくはないものの無視できないものであるため、いわゆるコントロール変数として FP 規模を説明変数に加える。

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(*1)
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 (9)	501_要求仕様_明確さ
	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)

(*1) 「 / 」が項目の区切りを表す。番号はデータ白書[19]で定義されたもの

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 46 個を選んだ。これらの 46 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が、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[19]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から代表的な変数として 4 つの業種と 5 種類のアーキテクチャを取り上げる。これらの変数では分析に耐えられるだけのデータ数が揃っているもののみを対象とする(表 3)。ただし、これらの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。

データ数(回答数)が 30 件以上ある。

各レベルに属する回答数が 10 件以上ある。

回答の内容が極端に偏っていない。具体的には、偏り率が -0.7~0.7 の範囲であることとする。ただし、偏り率は、偏り率 = (上位レベルの回答数 - 下位レベルの回答数) / 総回答数、で定義する。偏り率のとり得る範囲は -1 から 1 であり、上位レベルの回答数が多い場合はプラス、下位レベルの回答数が多い場合はマイナスとなる。上位レベルと下位レベルの回答数が等しい場合、偏り率は 0 となる。

ただし、~ 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。

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.

変数	とりうる値(*1)
201_業種 1	製造業, 情報通信業, 卸売・小売業, 金融・保険業, 左記以外の業種
308_アーキテクチャ 1	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) 値ひとつとそれ以外の値すべての 2 つのレベルで比較する。

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。対数は常用対数を用いる。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) 順序尺度に従う質的変数

順序尺度に従う変数の多くは 4 つ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 レベル以上の値をもつ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、着目する分類項目とそれ以外の分類項目すべて、の 2 つの分類にまとめる。例えば、「製造業」と「製造業以外のすべての業種」のように 2 値化する。

3. 分析方法

3.1 一般線形化モデル

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。そのようなデータに対しては、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。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。

目的変数にゼロを含む割合が多い場合は、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)などが提案されている。これらの分析手法は、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。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。

ゼロ過剰モデルもハードルモデルによる適合度の向上度は、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[20]。これらのことから、今回のデータ分析モデルとしては、負の二項回帰モデルを用いる。

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[18]。

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

この分布の平均は μ 、分散は $\mu + \alpha\mu^2$ となる。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。0 のとき負の二項分布はポアソン分布に近づく。負の二項回帰モデルを用いた分析では、プロジェクト i の説明変数の値 x_i から次のリンク関数と呼ばれる式で μ_i を推定する(説明変数が複数の場合も同様の式で表すことができる)。

$$\ln \mu_i = \beta_0 + \beta_1 x_i \quad (2)$$

これを用いて尤度関数

$$L(\alpha, \beta_0, \beta_1) = \prod_{i=1}^N f(y_i|\mu_i, \alpha) \quad (3)$$

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(3)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。

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

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 の負の二項回帰分析の R のスクリプトは次のようなものである[21]。

```
> library(MASS) # 必要なライブラリの読み込み
> fm <- glm.nb(y~x1, 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)を判断するための有意水準(p 値)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%とする。

(2) 回帰係数

質的変数の回帰係数の値が $\log 1.5 = 0.176$ 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 選定条件を満たさない変数

表 1 及び表 2 であげた説明変数の候補のうち、次の 11 個の変数が 2.2 の(2)で述べた「データの偏り率が 0.7 以下

0.7 の範囲であること」という説明変数の条件の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，インターネット・イントラネット
- ・業種：情報通信業と卸売・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
なお，スタンドアロン，メインフレーム，409_CASE ツールは，上位レベルの件数が10件未満であり，条件も満たさなかった。

これらの11件の変数は分析の対象外とし，以降は順序尺度に従う変数38個と名義尺度に従う変数4個を分析対象とする。

4.2 すべての選定条件及び選択基準を満たす変数

2.2 の(2)で述べた対象の層のデータ件数に関する～の条件，及び3.4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。

以下の分析結果はすべてFP規模が同一と仮定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ヶ月の発生不具合検出数が少ない）。
- ・アーキテクチャでは差がみられない。
- ・プロジェクト全般では，113_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には信頼性が高い。
- ・ツールを利用する方が利用しない場合よりも信頼性が高い。

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 N			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	係数 (*1)	p 値 (%)	
業種	201_業種1	金融・保険業	左記以外	305	90	215	0.29	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	0.22	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	0.44	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	0.36	1.2	2.3
	411_コードジェネレータ利用	a：有り	b：無し	130	37	93	-0.31	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	0.26	1.2	1.8
	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	0.28	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	0.31	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	0.42	3.0	2.6
要求レベル	514_要求レベル_性能・効率性	a：極めて高い+b：高い	c：中位+d：低い	122	69	53	0.32	2.2	2.1
	515_要求レベル_保守性	a：極めて高い+b：高い	c：中位+d：低い	88	27	61	0.49	0.8	3.1
開発担当者	1010_テスト体制	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	0.36	4.4	2.3

(*1) 2つの群の平均値の差に相当

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方が、502_ユーザ担当者が要求仕様に関与している方が、504_ユーザ担当者の業務経験が豊富な方が、507_ユーザ担当者の設計内容理解度が高い方が、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・効率性が高い方が、515_要求レベル_保守性の高い方が、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。

分類別でみた場合、ユーザ側に属する変数の影響度が、絶対数(4個)からみても選択された比率(0.44=4/9)からみても最も大きい。

5. 考察

5.1 不具合数比率

質的変数は上位レベルと下位レベルの2値しかとらないため、質的変数の偏回帰係数の値はFP規模が等しいと仮定した場合の2つのレベルそれぞれの群の平均値の差に相当する。不具合数の推定平均値はリンク関数によって対数変換されているので、平均値の差は元のスケールでは2つの群それぞれの中央値に相当する値の比となる。この比を「不具合数比率」と呼ぶことにする。

表4の不具合数比率は上位レベルの不具合数に対する下位レベルの不具合数の比の値を示している。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数が負の場合は、逆数で示している。表4から質的変数の不具合数比率は概ね2.0、すなわちひとつの要因でみると上位レベルと下位レベルで不具合数が倍半分の違いがあることがわかる。

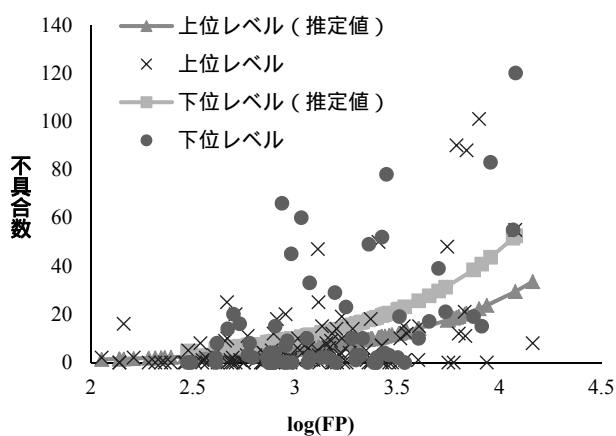


図1 FP規模と不具合数の散布図例(501_要求仕様の明確さ)

Fig.1 Scatter plot graph of FP and number of failures for clarity of requirement specifications.

FP規模に対する不具合数の散布図例を図1に示す。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66中15(23%)の組合せが独立ではないという結果が得られた。しかし、各要因間の(ピアソンの)相関係数を求めたところ相関係数が0.4を超えるのは、514_要求レベル_性能・効率性と515_要求レベル_保守性(0.45)とv502_ユーザ担当者_要求仕様関与と507_ユーザ担当者_設計内容理解度(0.40)の2つだけであり、相関は高いとは言えず、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36個中1個だけが-1.1であった。これらの結果から、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の判断は難しく、各変数個別に考えた方がよいと思われる。

一方、外れ値とは「それを除くと分析結果が急激に変化するもの」という考え方もできる。その観点から不具合数

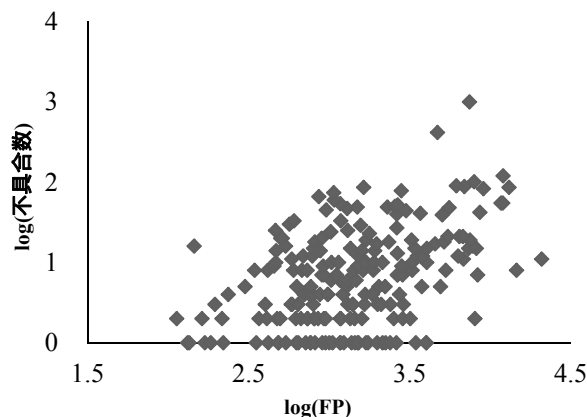


図2 FPと不具合数(正のもの)の散布図

Fig.2 Scatter plot graph of FP and positive number of failures.

が大きなプロジェクトを除いて再分析してみた結果を表5に示す。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つのプロジェクトとともにそれらの変数に対しては値が欠損し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408_テスト・デバッグツールの利用以外は、2件のデータを除くことにより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13.1%に、411_コードジェネレータ利用では5.6%から43.0%といずれもp値が大きくなっている。特に411_コードジェネレータ利用では変動が大きく1件のデータを除くとp値が43%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ておいた方がよいと思われる。

それに対して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ないものいづれも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。

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 N		質的変数	
		上位レベル	下位レベル	係数	p値 (%)
金融・保険業	含む	90	215	0.29	0.3
	除く		213	0.20	2.9
113_達成目標_優先度_明確度合	含む	55	105	0.22	9.4
	除く		104	0.17	17.6
121_計画の評価(品質)	含む	205	44	0.44	0.1
	除く		42	0.20	13.0
408_デバッグ_テストツール利用	含む	67	72	0.36	1.2
	除く	66		0.55	0.0
411_コードジェネレータ利用	含む	37	93	-0.31	5.6
	除く	36		-0.12	43.0

5.4 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6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、目的変数を工数とし、2.2(2)で述べた～の条件を満たす変数に対して線形回帰分析を適用して3.3の(1)と(2)の選択基準を満たすものを選んでいる(2014年までの一世代前のデータに対する分析結果は[22]に詳しく述べられている)。生産性における工数比率は、FP規模が同一と仮定した場合の上位レベルの工数に対する下位レベルの工数の比

の値である。ただし、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6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が信頼性を向上させるのに対し、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のデータにユーザ側の工数が計上されていないので実際には生産性という点ではマイナスではないと思われる。
- ・信頼性向上にも生産性向上にも寄与する要因は、113_達成目標_優先度_明確度合(が非常に高い場合)である。
- ・ユーザ側の関与以外に、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。

表6 信頼性と生産性の影響要因の比較

Table 6 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_設計支援ツール	-	2.4
	408_デバッグ_テストツール利用	2.3	1/1.9
ユーザ側	501_要求仕様_明確さ	1.8	-
	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	-
要求レベル	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	-
	515_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

- ・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向上に寄与する。
- ・要求レベルに関する変数のうち, 514_要求レベル_性能・効率性, 515_要求レベル_保守性が信頼性向上に寄与し, 512_要求レベル_信頼性と 518_要求レベル_セキュリティが生産性の低下を招くという結果であった。しかし, 512_要求レベル_信頼性と 518_要求レベル_セキュリティは 514_要求レベル_性能・効率性との相関係数がそれぞれ 0.61 と 0.58 と高く, また表中に「-」で示した欄も 512_要求レベル_信頼性と 518_要求レベル_セキュリティの不具合比率はいずれも 1.7, 514_要求レベル_性能・効率性と 515_要求レベル_保守性の工数比率がそれぞれ 1/1.3 と 1/1.5 とすべての要求レベルに関する変数が同じ傾向を示した。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ものに比べて, 生産性は低い(開発に工数がかかる)が信頼性は高い。

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, 信頼性への影響要因を分析した。

p 値が 10% 以下で不具合比率が 1.5 以上となる質的変数として次のものを選び出した。

- ・113_達成目標_優先度_明確度合が非常に明確な場合
- ・121_計画の評価(品質)において品質目標が計画で実行可能性を検討済みの場合
- ・408_デバッグ_テストツールを利用する場合
- ・501_要求仕様が明確な場合
- ・502_ユーザ担当者が要求仕様に関与する場合
- ・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
- ・514_性能・効率性への要求レベルが高い場合
- ・515_要求レベル_保守性が高い場合
- ・1010_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 4 割を占め, 信頼性向上にはユーザ側の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。

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, p. 767.
- [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, 1971, vol. 71, p. 353-379.
- [4] Halstead, M. H.: Elements of Software Science, Elsevier, North-Holland, 1975.
- [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, 情処論文誌, 2009, vol. 50, no. 3, p. 1144-1155.
- [7] 小室睦, 薦田憲久: ピ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, 2011, vol. J94-D, no. 2, p. 439-449.
- [8] Khoshgoftaar T. M., and Gao, K.,: Count Models for Software Quality Estimation, IEEE Tr. Reliability, 2007, vol. 56, no. 2, p. 212-222.
- [9] 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, 情処処理学会論文誌, 2008, vol. 49, no. 12, p. 3954-3966.
- [10] 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, 2008, vol. 81, p. 823-839.
- [11] 裕本真佑, 亀井靖高, 門田暁人, 松本健一: 開発者メトリック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, 2010, vol. J39-D, no. 8, p. 1576-1589.
- [12] Goel, A.L. and Okumoto, K.: Time -Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures, IEEE Trans. Rel., 1979, vol. R-28, no. 3, p. 206-211.
- [13] Yamada, S., Ohba, M. and Osaki, S.: S-Shaped Reliability Growth Modeling for Software Error Detection, IEEE Trans. Rel., 1983, vol. R-32, no. 5, p. 475-478.
- [14] Furuyama, T. and Nakagawa, Y.: A Manifold Growth Model that Unifies Software Reliability Growth Models, Int. J. of Reliability, Quality and Safety Engineering, 1994, vol. 1, no. 2, p. 161-184.
- [15] 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, 2004, vol. J-87-D-I, no. 8, p. 805-814.
- [16] Fenton, N., Neil, M., Marsh W., Hearty, P., Marquez, D., Krause, P. and Mishra, R.: Predicting Software Defects in Varying Development Lifecycles using Bayesian Nets, Information and Software Technology, 2007, vol. 49, p. 32-43.
- [17] Fenton, N., Neil, M., Marsh W., Hearty, P., Radlinski, L., and Krause, P.: Project Data Incorporating Qualitative Factors for Improved Software Defect Prediction, Third Int. Workshop on Predictor Models in Software Engineering (PROMISE'07), 2007.
- [18] Cameron, A. C., and Trivedi, P. K.: Regression Analysis of Count Data, 2nd ed., Cambridge Uni. Press, 2013, p. 566.
- [19] 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.
- [20] Gao, K., and Khoshgoftaar T. M.,: A Comprehensive Empirical Study of Count Models for Software Fault Prediction, IEEE Tr. Rel., 2007, vol. 56, no. 2, p. 223-236.
- [21] Zeileis, A., Kleiber, C., and Jackman, S.: Regression Models for Count Data in R, J. Statistical Software, 2008, vol. 27, Issue 8, p. 1-21.
- [22] 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, 2016, 第 11 巻, 第 4 号 (通巻 47 号), p. 40-47.