

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)}

受付日 2019年1月22日, 採録日 2019年7月3日

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信頼性, 具体的には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持つプロジェクトが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, ユーザ側の協力が重要な役割を果たすことが分かった. すなわち, ユーザ担当者の要求仕様への関与, 明確な要求仕様と要求仕様変更発生の抑制, ユーザ担当者の豊富な業務経験と設計内容の十分な理解が信頼性を向上させる. また, 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・品質などの達成目標と優先度の明確化, 計画段階での品質目標の実行可能性の検討, 高い要求レベルも信頼性向上に重要な役割を果たすことが分かった. 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. これらの要因は, 既存の研究で得られたものと差はあるものの, それらと相反するものはない. 信頼性に影響を与える要因の生産性への影響を調べると, 信頼性も生産性も改善するものは, 納期・品質などの達成目標と優先度の明確化だけであった. これらの分析結果は信頼性向上の視点からのプロセス改善に役立てることができる.

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

TSUNEO FURUYAMA^{1,a)}

Received: January 22, 2019, Accepted: July 3, 2019

Abstract: Qualitative factors affecting the reliability of enterprise software were identified by analyzing project data collected by the Information-Technology Promotion Agency in Japan. An ordinal linear regression model cannot be used to analyze data that includes zero values, such as for the number of software faults detected during operation, because zero values cannot be logarithmically transformed. Therefore, a negative binomial regression model like those used in the areas of economics and sociology was used. The results showed that the reliability of newly developed software depends on the user staff's commitment to defining requirement specifications, the number of requirement specification changes, the usage of debug and test tools, the skill level of the test team, the clarity of the objectives and priorities, and whether the plan was evaluated for quality. In particular, the number of specification changes is the main factor in reduced reliability. These results do not contradict previous results obtained using a questionnaire although there were some differences between the analyzed variables. A comparison of the factors for reliability and productivity revealed that only one factor, clarity of objectives and priorities, improved productivity while the others reduced or did not affect it.

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

¹ 東海大学
Tokai University, Hiratsuka, Kanagawa 259-1292, Japan
^{a)} furuyama@tokai-u.jp

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持つが、製品品質に関する8つの品質特性のうちの1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なかで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、大きく次の4つに分類できると考えられる。

- (a) 出荷時の製品に含まれる欠陥^{*1}数の予測
- (b) テスト工程時の欠陥数の予測
- (c) 欠陥を含む可能性の高い (fault-prone) モジュールの識別
- (d) 稼働後の信頼性に影響を与える要因の抽出

(a) に関する研究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 (文献 [3], [4], [5], [6] など)。これらのモデルにより、テスト終了時期を判定したり、推定残存バグ数からリリース製品の品質 (信頼性) を推定することができる。SRGM はそれだけで大きな研究分野を形作っていて、ソフトウェア信頼性モデルの研究論文のなかでは最も多いという報告がある [7]。

(b) は、テスト工数を早期に把握するのに有効である。モデルを構築するためのデータ分析方法としては、研究初期の段階から現在まで線形回帰分析が多く利用されている (文献 [8], [9], [10], [11], [12], [13] など) が、目的変数がゼロを含む場合にも分析できる、ポアソン回帰モデルによる分析 [14] や負の二項回帰モデルによる分析 [15] も報告されている。また、ベイジアンネットの手法を用いた分析結果 [16], [17] も報告されている。

(c) は最近の信頼性に関する研究で主流となっているものである。Fault-prone モジュールが識別されれば、そこにテスト資源を集中することにより、効率良くテストを行うことができる。データの分析方法としては、主成分分析と判別分析を組み合わせたもの [18], [19], ロジスティック回帰分析 [20], [21], [22], [23], [24] などを中心に、論理式 [25],

ベイジアンネットワーク [26], [27], ルールベース [28], 相関ルール [23], ランダムフォレスト [29] などさまざまな方法が用いられている。最近では、SRGM と融合した研究も試みられている [30]。

Hall らは 2000 年から 2010 年までの 208 件の論文のうち、十分な情報の揃っている 36 件の欠陥予測の論文を精査して、fault-prone モジュール/ファイルの識別などの欠陥予測では単純サイズとロジスティック回帰の予測性能が優れていると報告している [31]。

(b), (c) における予測子 (説明変数) としては、初期の研究ではコード行数、サイクロマティック数などのプログラムの複雑さ、Halstead の尺度などのコード属性だけであった [8], [18], [32]。その後、モジュール間の制御フローやデータフローに着目した設計メトリクス^{*2}を用いたモデル [9], [19], [25] やオブジェクト指向 (OO) (設計) メトリクスを説明変数とするモデル [20], [22], [26] が提案された。これらはいずれもプロダクトの属性に着目したものであり、プロダクトメトリクスと呼ばれる。

それに対して開発プロセスの属性に着目したものがプロセスメトリクス^{*3}である。プロセスメトリクスとしては、レビュー指摘件数などの開発時に得られるものと開発者のスキルレベルやツールの利用など開発環境にかかわるものがある。前者はおおむね比尺度に従う量的変数であり、後者は順序尺度や名義尺度に従う質的変数であることが多い。前者の研究には、レビュー指摘密度を用いたもの [11]、開発者の開発特性に着目したもの [12]、レビュー指摘件数やレビュー工数を取り上げたもの [13] などがある。後者の研究には、テストスタッフの経験などを取り上げたもの [17] などがある。過去の欠陥数、モジュール変更回数、モジュールの年齢、要求件数などのプロセス属性を要因の候補として本格的に取り上げた研究としては、[14], [15], [21] などがある。

Graves らは欠陥数予測にはプロダクト属性よりもプロセス属性が有効であると報告している [14]。また、Radjenović らは欠陥数予測にはプロダクト属性よりも OO メトリクスとプロセス属性が有効である報告している [34]。これらの報告は、従来のプロダクト属性だけでなく、プロセス属性により着目すべきであることを示唆している。

プロセスメトリクスを説明変数として広く取り入れた研究は Fenton の研究 [17] を除いてほとんどない。Fenton は、仕様化と文書化のプロセス、設計・製造プロセス、テストプロセス、プロジェクトの管理に関して多くの要因を

^{*2} メトリクス = 測定量 (変数) + 測定方法 (測定対象の属性と尺度の対応付けの方法) であるが、測定量を定めると自然と測定方法も定まることも多く、メトリクスと測定量を区別せずに使っても問題ないことが多い。本論文でも同義的に扱う。

^{*3} このほかにリソースメトリクスという分類項目もある [33] が、本論文では開発するための手段という意味でプロセスメトリクスとリソースメトリクスをプロセスメトリクスで一括りにする。

^{*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。

取り上げて、ベイジアンネットを用いてモジュール内の欠陥数を推定し、実際の欠陥数と比較して高い精度で欠陥数を推定できることを示した。ただし、どの要因がどの程度推定に寄与しているかは明らかにしていない。

(d)に関する研究は(a)~(c)までの研究と異なり、欠陥数の予測までは踏み込んでいない。この分野の研究はまだ少なく、筆者の知る限り4件しかない(文献[35], [36], [37], [38])。このうち[35], [36], [37]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1つの研究であるということが出来る。すなわち、文献[36]の研究は、ソフトウェア開発環境の変化を考慮して文献[35]の研究の15年後にまったく同じ方法で調査を実施したものである。また、文献[37]の研究は、文献[36]と文献[35]の研究が単一のサイクルで開発・出荷するソフトウェアを対象にしたものであるのに対して、新たにアジャイル開発で行われるような複数回のサイクルで開発・出荷するソフトウェアに対して同じ方法で信頼性への影響を調査している。これらの研究では、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^{*4}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ると思われることを示している。

文献[38]の研究では目的変数として5つの品質特性を取り上げていてそのなかの1つに信頼性がある。説明変数として、組織的な変数を10個、個人的な変数を5個、技術的な変数を9個取り上げている。情報部門の役員(Executives)が目的変数と説明変数に関する質問に対して5段階評価の回答を行っている。分析の結果、特に情報部門の管理者(director)のランクおよびユーザからの仕様変更要求回数が信頼性に大きな影響を与える要因であることを明らかにした。

これらの分析結果はいずれも興味深いものであるが、これら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だけであり、たと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。実測データを用いて稼働後のソフトウェアの信頼性(たとえば発見欠陥数)を予測するような要因を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およ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(International Software Benchmarking Standards Group)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おり、稼働後の欠陥数のデータもあるが、Fentonらが述べている

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[16]。

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、それらを含むプロジェクト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^{*5}のデータリポジトリには、2004年から収集を始めた4,000件を超えるエンタープライズ系プロジェクトデータ(SECデータ)があり、収集項目のなか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、およびこれら発生不具合数に影響を与えると思われる多くのプロセス変数が含まれる。ただし、SECデータは、他の多くのリポジトリに含まれるフィールドデータと同様に、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、実際に分析に必要となる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件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)の研究に貢献できる可能性がある。

SECデータには多くのプロセス変数が含まれるが、特に質的変数の多さは他のリポジトリのデータにはないものである。ま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、プロセス改善の指針として分析結果を利用できる可能性が高い。それらを考慮して、今回は質的変数の発生不具合数への影響を分析することとした。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(文献[39]の3.3節)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[40]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。

*4 論文では具体的な定義は行われていない。

*5 2018年7月に組織変更があり、従来のSECの業務は社会基盤センターで継続して実施することとなったが、本稿ではSECの名称を用いる。

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 [40]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) 規模を示すFP（ファンクションポイント）の実測値（5001_FP実測値_調整前*6）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一定期間において発生した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)の条件はそれをふまえたものである。

工数と同様に不具合数も規模の影響を強く受けると考えられるので、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)の条件はそれをふまえたものである。

これまでの分析経験によると、生産性も生産性への影響要因も開発種別（新規開発，改良開発など）で異なる [41]。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。開発種別のなか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、分析の精度を高めるために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カ月，3カ月，6カ月の3種類，全部で6種類のデータが収集されているが、今回はデータ件数の最も多い稼働後1カ月後の発生不具合現象数（5267_発生不具

*6 番号と名称はデータ白書 [40] で定義されたもの。

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 規模
データ件数	305	
ゼロデータ件数	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

(注) FP 規模の値は、データ件数とゼロデータ件数以外は常用対数変換後の値

合現象数（合計1カ月）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節の1)~4)を満たすプロジェクト数は305である*7。305件のデータの基本統計量を表1に示す。ただし、FP規模の基本統計量は、一般にFP規模が対数正規分布に従う [42] ため*8（常用）対数変換後のものである。なお、本論文では基本的に自然対数（ln）を用いるが、図表など直観的な分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。表1から305件のプロジェクトのうちの32%（99件）のプロジェクトで不具合数の値がゼロであることが分かる。

(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。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51個を選んだ。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た。

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上げる。このなかで分析に耐えられるだけのデータ件数が揃っている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*9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。具体的な変数名は、たとえば「業種_製造業」のように表3の分類と変数を結び付けたものを用いる。

今回の分析では、1個の質的変数に加えて、不具合数に大きな影響を与えていると思われる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。

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。

*7 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。

*8 log (FP 規模) の分布に対するシャピロ・ウィルク検定の p 値は 0.341 であり、正規分布に従うといえる。

*9 データ白書 [40] の付録 A.3 参照。

表 2 影響要因の候補 (順序尺度に従うもの)

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

分類	変数	
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	
ユーザー側 (14)	501_要求仕様_明確さ	
	要求仕様変更発生状況	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)
	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	
	ユーザ承認有無	506_要求仕様, 508_設計
要求レベル (8)	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	
	601_PM スキル	
開発担当者 (7)	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 [40] で定義されたもの

表 3 影響要因の候補 (名義尺度に従うもの)

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

分類	変数
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, インターネット・イントラネット

- 1) データ件数 (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 のとりうる範囲は -1 から 1 であり, $N_1 = N_2$ の場合は $\rho = 0$ となる。この条件は, 回答数が上位レベルに偏る天井効果および下位レベルに偏る床効果を避けるために加えたものであり, 全体の回答数に占める上位 (または下位) のレベルの回答数の許容される割合は $15\% \sim 85\%$ になる。これは, 正規分布をする母集団を「平均 ± 1 標準偏差」の範囲内で分割した場合のそれぞれの部分の割合にほぼ等しい*10。天井効果も床効果もアンケートの設計に問題がある場合に生ずるもので, これが生ずると正しい分析結果が得られない。

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るが, 発生事象数が最低 10 件は必要であるという, 2) の条件をサポートする研究もある [43]。なお, 値が正規分布に従う 2 つのデータ集合から 15 件ずつのデータをサンプリングすることにより, それらの集合の母平均の差が標準偏差 (2 つの集合で等しいと仮定) よりも大きい場合に, 有意水準 10% (両側), 検出力 85% で母平均に差がないという帰無仮説を棄却できる (文献 [44] に基づいて算出)。これは 1) の条件によってどの程度の分析が可能か, ということの 1 つの目安となると考えられる。

2.3 変数変換

(1) 量的変数

FP 規模は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。不具合数は変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 3 つ以上のレベルの値を持つが, 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どうしを合併して全体で 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 値化する分割点は複数個あるが, 偏り率 ρ の絶対値が最も小さくなるものを分割点とする。本研究では下位レベルをベースになる群と見なすこととして, 下位レベルに 0 を上位レベルに 1 の値を割り当てる。したがって, 上位レベルの不具合数が少ない場合は回帰係数はマイナスとなる。

(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, 対象とするプロジェクトが変数名に含まれる業種に該当する場合は, 該当する業種名を, 該当しない場合は「それ以外の業種」という値を割り当てる。たとえば, 変数「業種_製造業」の値は, 「製造業」と「それ (製造業) 以外の業種」の 2 つであり, 前者を上位レベル,

*10 正確には $15.9\% \sim 84.1\%$ である。

後者を下位レベルとする。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

目的変数がゼロを多く含むデータに対しては、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（文献 [39] の 3.2 節, 3.3 節）。この 2 つを比較すると、ポアソン回帰モデルはパラメータが 1 つしかないため、適合度（AIC の値）はパラメータ 2 つを持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（文献 [39] の 4.6 節）。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（文献 [39] の 4.5 節）。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いえない。

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は、もともとの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 [45] 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関数を用いて回帰分析を行う方法である（文献 [39] の 3.3 節）。

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

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 μ 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、たとえばプロジェクト 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。

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。

$$\ln \mu = \beta_0 + \beta_1 \ln FP + \beta_2 x \quad (5)$$

計算は統計ツール R を用いて行う。

3.3 説明変数の選択基準

信頼性に影響を与える要因の選択は、回帰係数の傾きの p 値に基づいて行う。今回は質的変数が 2 値しかとらないので回帰係数の傾きは、上位レベルと下位レベルの 2 つの群のそれぞれの平均値の差と等しい。判断基準とする 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。

4. 分析結果

4.1 分析条件を満たさない変数

表 2 および表 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 節の (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 1 つの条件を満たさないことが分かった。

- 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：506_要求仕様_ユーザ承認有無。
- 業種：情報通信業, 卸売・小売り業。
- 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。

これらの 12 個の変数は分析の対象外とし、以降は 48 個の変数を分析対象とする。

表 4 影響要因と見なされた質的変数
Table 4 Qualitative variables identified as effective factors.

分類	変数名	内容		データ件数			質的変数		不具合率	
		上位レベル	下位レベル	総数	上位レベル	下位レベル	偏回帰係数	p 値 (%)		
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:おおむね明確 + c:やや不明確 + d:不明確	160	55	105	-0.50	9.4	0.61	
	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	0.36	
ツール利用	408_デバッグ_テストツール利用	a:あり	b:なし	139	67	72	-0.83	1.2	0.44	
	411_コードジェネレータ利用			130	37	93	0.71	5.6	2.03	
ユーザ側	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	0.55	
	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	0.31
		5117_詳細設計			45	16	29	-1.38	0.9	0.25
		5119_結合テスト			36	19	17	-1.51	1.3	0.22
		5120_総合テスト(ベンダ確認)			42	24	18	-0.89	8.9	0.41
	ユーザ担当者	502_要求仕様関与	a:十分に関与 + b:おおむね関与	c:関与が不十分 + d:未関与	167	100	67	-0.64	0.8	0.53
504_業務経験		a:十分に経験	b:おおむね経験 + c:経験が不十分 + d:未経験	84	34	50	-0.72	6.1	0.49	
507_設計内容理解度		a:十分に理解 + b:おおむね理解	c:理解が不十分 + d:まったく理解していない	89	69	20	-0.97	3.0	0.38	
要求レベル	514_性能・効率性	a:きわめて高い + b:高い	c:中位 + d:低い	122	69	53	-0.74	2.2	0.48	
	515_保守性			88	27	61	-1.13	0.8	0.32	
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	0.44	
業種	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	0.52	

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 節の (2) で述べたデータ件数に関する 1)~3) の条件, および 3.3 節の説明変数の選択基準をすべて満たす 16 個の質的変数を表 4 に示す. なお, 2.2 節の (2) で述べたように, 回帰分析の説明変数は FP 規模と 1 つの質的変数であることから, 表 4 に示す質的変数の偏回帰係数は, FP 規模の影響を除去した後 (FP 規模が一定とした場合) の目的変数への影響の大きさを示していることに留意してほしい.

表 4 から次のことが分かる.

- プロジェクト全般では, 113_(納期・品質などの) 達成目標_優先度_明確度合が非常に明確な場合, および 121_計画の評価(品質)において品質目標が明確でその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(ベンダ確認)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, 特に詳細設計および結合テストでの変更発生の偏回帰係数の絶対値が大きく, 最も信頼性に悪影響を与えている.
- 業種では, 金融・保険業が他の業種に比べて信頼性が

高い。

- アーキテクチャでは、他のアーキテクチャと信頼性に差があるものはない。
- 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(0.5 = 8/16)からみても最も多い。

5. 考察

5.1 不具合数比率

式(5)におけるパラメータ $\beta_0, \beta_1, \beta_2$ (切片および回帰係数)の最尤推定値をそれぞれ b_0, b_1, b_2 とすると、あるFP規模における上位レベルおよび下位レベルそれぞれの不具合数の平均 μ_i^+ および μ_i^- は、質的変数が上位レベルで1、下位レベルで0の値をとることから、

$$\ln \mu_i^+ = b_0 + b_1 \ln FP + b_2 \quad (6)$$

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

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(6)と式(7)から

$$\ln \frac{\mu_i^+}{\mu_i^-} = \ln \mu_i^+ - \ln \mu_i^- = b_2 \quad (8)$$

となり、不具合数比率は

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

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、411_コードジェネレータ利用を除くすべての不具合数比率が0.7以下である。また、要求仕様変更発生状況の不具合数比率が、0.22~0.41と低い値を示している。特に5117_詳細設計で0.25、5119_結合テストで0.22と低く、仕様変更なしの場合はある場合に比べて不具合数が1/4以下になることが分かる。

5.2 外れ値の影響

不具合数が999と415という大きな値を持つプロジェクトが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持つ206件のプロジェクトにおける $\log(\text{FP 規模})$ と $\log(\text{不具合数})$ の散布図を図1に示す。 $\log(\text{不具合数})$ が3.0と2.5に位置する2つの点(図1中の Δ)が不具合数999件と415件に対応するが、これらを外れ値と見なすべきかどうかを図1から判断するのは難しい。

そこで、四分位数から計算される内境界点[46]を用いて

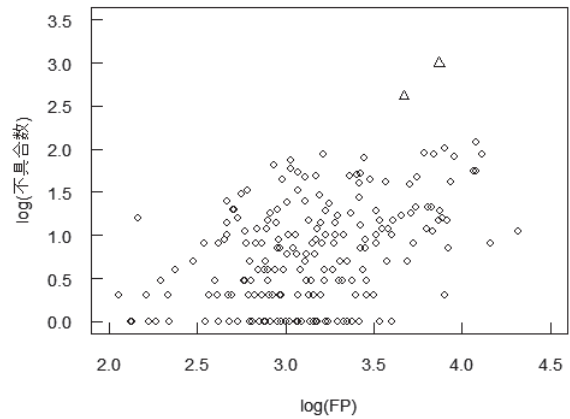


図1 FP規模と不具合数(正のもの)の散布図

Fig. 1 Scatter plot graph of log number of failures (positive) and log FP size.

上記2つの値が外れ値か否かを調べることにした。ただし、ヒンジ幅 = 上ヒンジ - 下ヒンジとすると、

$$\text{上側の内境界点} = \text{上ヒンジ} + \text{ヒンジ幅} \times 1.5$$

$$\text{下側の内境界点} = \text{下ヒンジ} - \text{ヒンジ幅} \times 1.5$$

である。これら内境界点を超える値を外れ値と見なすこととする。

なお、正規分布を対象とした場合、上下のヒンジは $\pm 0.6745\sigma$ 、上下の内境界点は $\pm 2.698\sigma$ に相当し、内境界点相当を超える部分は上下それぞれ0.35%である。

今回分析対象としたデータでは、各要因ともゼロデータ件数がデータ件数の30%を超えるため意味のある下ヒンジの値を得ることができない。そのため、

$$\text{上ヒンジ} - \text{中央値} = \text{中央値} - \text{下ヒンジ}$$

であると仮定して

$$\text{上側の内境界点} = \text{上ヒンジ} + (\text{上ヒンジ} - \text{中央値}) \times 3$$

とした。

各影響要因の上側の内境界点を調べた結果を表5に示す。表5に記載されていない変数は、大きな不具合数を持つ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から、113_達成目標_優先度_明確度合と121_計画の評価(品質)の内境界点はそれぞれ3.41と3.86でいずれも999の不具合数の常用対数3.00よりも大きく、999の値を外れ値と見なす必要がないことを示している。実際、不具合数が999と415のプロジェクトを除いても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節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。

408_デバッグ_テストツール利用と411_コードジェネレータ利用の内境界点はそれぞれ1.91と2.41で999の不具合数の常用対数3.00よりも小さい。999の不具合数を外れ値

表 5 大きな不具合数を持つ 2 つのプロジェクトの影響
Table 5 Effects of two projects with large number of failures.

変数名	不具合数		データ件数		上側内境界 点	質的変数	
	999	415	上位レベル	下位レベル		偏回帰係数	p 値 (%)
113_達成目標_優先 度_明確度合	含む	なし	55	105	3.41	-0.50	9.4
	除く			104	—	-0.38	17.6
121_計画の評価 (品質)	含む		205	44	3.86	-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	なし	67	72	1.91	-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	なし	37	93	2.41	0.71	5.6
	除く		36		—	0.28	43.0
業種_金融・保険業	含む		90	215	2.71	-0.66	0.3
	除く			214	—	-0.55	1.0

と見なすことにより、408_デバッグ_テストツール利用では偏回帰係数は -0.83 から -1.02 と絶対値が大きくなる（効果が強調される）が、411_コードジェネレータ利用では偏回帰係数が 0.71 から 0.28 と小さくなるだけでなく、p 値が 5.6% から 43.0% と大幅に大きくなる。このことから 411_コードジェネレータ利用を影響要因と見なすことは保留した方がよい。

業種_金融・保険業の内境界点は 2.71 で、999 の不具合数の常用対数 3.00 よりも小さいが 415 の不具合数の常用対数 2.62 よりも大きい。999 の値のみを外れ値と見なしても、偏回帰係数が -0.66 から -0.55 に、p 値も 0.3% から 1.0% に変化するだけなので 4.2 節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。

以上のことから以下では、411_コードジェネレータ利用を除いた 15 件を信頼性に影響を与える要因と見なして議論を進める。

5.3 要因間の相関と多重共線性

表 4 で示した要因どうしは必ずしも独立ではない。15 の影響要因に対して各要因どうしの分割表を作成して独立性の検定を行ったところ、105 の組合せ中 13 (12%) 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。13 の組合せの ϕ 係数 (2 × 2 の分割表ではピアソンの相関係数と等しい) は 0.287 から 0.837 の範囲にあるが、そのうちの 3 つだけが 0.5 を超えていてそれらはすべて要求仕様変更発生状況に関する要因どうしのものである。すなわち、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 である。

15 の影響要因のうち、113_達成目標_優先度_明確度合、408_デバッグ_テストツール、504_ユーザ担当者_業務経験、1010_テスト体制 (スキルレベル) は、いずれも「独立でない」という組合せを持たない。これら 4 つの要因を除いた 11 の影響要因について多重共線性を調べたところ (デー

タ件数は 33 件)、最も VIF (Variance Inflation Factor) の大きかったのは 5119_要求仕様変更発生状況_結合テストの 4.5、次いで 5120_要求仕様変更発生状況_総合テスト (ベンダ確認) の 3.7 であり、他の要因の VIF はいずれも 2.5 以下であった。いずれも多重共線性はないという目安となる VIF < 10 の基準を満たしていることから、15 の影響要因に冗長性はないといえる。

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 [36]。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 の環境要因のなか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%) 含むなどの特徴が見られる。SEC データに対応するまた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 (= 結合テストおよび総合テスト (ベンダ確認) の工数/開発 5 工程工数) およ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、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およびプログ

表 6 Zhu らの分析結果との比較

Table 6 Comparison of results from SEC data to those of 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 [36]	
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*	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	
ユーザ側	<417_コード再利用率*>	コード再利用率	
	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>	プログラマのスキル	
	1010_テスト体制 (スキルレベル)	—	
業種	業種_金融・保険業	—	

(注) *: 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もの。—: 分析対象となる項目がないか、または項目があってもデータ件数が少ないため分析対象外となったもの。< >: 項目はあるが有意でないか、または上位 10 の影響要因と判断されなかったもの

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 影響要因の生産性への影響

信頼性への影響要因の生産性への影響を調べた結果を表 7 に示す。生産性への影響要因の計算方法は、信頼性への影響要因の計算方法と同様の方法を用いている。すなわち、まず 2.1 節で述べた 1)~4) までの条件 (ただし、1) は不具合数の代わりに工数) 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 節 (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 規模ともに対数変換を行った後、線形回帰分析を行って 3.3 節の選

表 7 影響要因の信頼性と生産性に与える影響の比較

Table 7 Comparison of variables affecting reliability and productivity.

分類	変数名	信頼性	生産性	
		不具合数比率	工数比率	
プロジェクト全般	113_達成目標_優先度_明確度合	0.61	0.59	
	121_計画の評価 (品質)	0.36	—	
ツール利用	408_デバッグ_テストツール	0.44	1.85	
	501_要求仕様_明確さ	0.55	1.31	
ユーザ側	要求仕様変更発生状況	5116_基本設計	0.31	
		5117_詳細設計	0.25	
		5119_結合テスト	0.22	
	ユーザ担当者	5120_総合テスト (ベンダ確認)	0.41	1.52
		502_要求仕様関与	0.53	—
	要求レベル	504_業務経験	507_設計内容理解度	0.38
			514_性能・効率性	0.48
515_保守性			0.32	1.45
開発担当者	1010_テスト体制 (スキルレベル)	0.44	—	
業種	業種_金融・保険業	0.52	1.85	

択基準を満たすものを選ぶ。

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。したがって、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上位レベルと下位レベルそれぞれの中央値の比に等しい。表 7 における工数比率は、下位レベルに対する上位レベルの比なので、工数比率が 1 より大きいものは工数が増えることを意味している。

表 7 から次のことが分かる。

- ユーザ側の開発へのさまざまな関与およ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。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になっている可能性もある。
- 5120_要求仕様変更発生状況_総合テスト (ベンダ確認) で工数比率が 1 より大きく (仕様変更ありよりなしの方が工数が多い)、直感とは合わない。さらに分析を進めたところ、この要因の上位レベル (仕様変更なし) には、生産性の低い金融・保険業のプロジェクトが有意に多いことが分かった (他の工程では有意な差はみられない)。そこで、FP 以外に業種に関する質的

変数, すなわち金融・保険業を上位レベル, その他の業種を下位レベルとした2値の変数を説明変数に加えて回帰分析をしたところ, 業種に関する質的変数の偏回帰係数のp値は3.5%であったのに対し, この要因の偏回帰係数のp値は36.5%で有意にはならなかった. 各セルの件数が10件以上というデータ件数の条件を満たしてはいないので断定はできないが, 表7で見られるこの要因の生産性の低下は業種の違いによる影響があるものと考えられる.

- 信頼性向上だけでなく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), および 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. デバッグ・テストツールの利用は生産性向上にも寄与するのではないかと思われるが, 分析結果は逆であった. 目的変数をテスト工程での検出バグ数の対数 ($\ln(5253_検出バグ現象数結合テスト+5254_検出バグ総合テスト(ベンダ確認))$), 説明変数を $\ln(FP)$ とデバッグ・テストツール利用 (の有無) として, 153 件のデータに対して線形回帰分析をすると, デバッグ・テストツールの偏回帰係数は -0.302 (p 値は 8.7%) となる. 不具合数比率にならって「検出不具合数比率」を計算すると 1.35 が得られる. ツールを利用することにより平均的に 1.35 倍多くのバグを検出し, その修正のため多くの工数を費やして生産性が低下しているものと思われる.
- 要求レベルが高いと工数は増大 (生産性は低下) するが信頼性は向上する.
- 金融・保険業のソフトウェアは他の業種に比べて, 信頼性は高いがそれを開発する際の生産性は低い.

5.6 影響要因に基づくプロセス改善の可能性

15 個の影響要因に基づいてプロセス改善を図ることにより信頼性の向上が期待できる. しかしながら, すべての影響要因をベンダ側 (だけ) でコントロールできるものではない. プロセス改善に向けた影響要因のコントロールの可能性は次に示すように分類別で異なる.

- 「開発担当者」, 「プロジェクト全般」および「ツール利用」に属するものは, さまざまな制約はあるもののプロジェクトの開始時点またはプロジェクト遂行中にベンダ側のプロジェクト管理者 (PM) や企業レベルでコントロールできるものである. 113_達成目標_優先度_明確度合を高めるために, Q (品質), C (コスト), D (納

期) の数値目標および優先度を文書で明確に定義すること (文献 [40] の p.317) が重要である. また, 121_計画の評価 (品質) を最上位レベルにすること, すなわち基本設計開始時点で品質目標を明確にしてその実行可能性を検討すること (文献 [40] の p.317) が重要である. さらに, テストに関しては, プロジェクトメンバによるか専任のテスト部隊によるかを問わず, 1010_スキルの高い要員を十分に揃えたテスト体制にすること, 408_デバッグ_テストツールを活用することが重要である.

- 「ユーザ側 (の協力)」については直接ベンダ側でコントロールすることはできないが, たとえば, ユーザ側の協力が信頼性向上に寄与することを示す何らかの根拠を示すことにより, ユーザ側を動かすことができる可能性はあると考えられる. 502_ユーザが要求仕様に関与して 501_要求仕様を明確にし, 5116, 5117, 5119, 5120_要求仕様変更の発生を抑制すること, 504_ユーザ側に業務経験の豊富な担当者を割り当ててもらうこと, 507_ユーザ担当者に設計内容を理解してもらい設計段階で抜けや誤りを指摘してもらうことなどが重要となる.
- 「要求レベル (の高さ)」は, コストや納期の兼ね合いで交渉の余地が残されている可能性もあるが, 一般にコントロールは難しい. 信頼性が高くなるのは単に要求レベルが高いからではなく, その要求に応えるためにさまざまなアクションをとるからと考えられる.
- 「業種」 (および「アーキテクチャ」) はプロジェクト管理者や企業レベルではコントロールできない. 金融・保険業のソフトウェアの信頼性が高いのは, 高い要求レベルなどに対応するためさまざまなアクションをとるからと考えられる.

5.3 節で行った独立性の検定結果のうち, 業種_金融・保険業, 514_要求レベル_性能・効率性および 515_要求レベル_保守性の各要因に対して, 有意水準 1% で独立でない判断された他の分類に属する要因を表 8 に示す.

表 8 から業種 (金融・保険業) → 要求レベル → 他の分類の要因という流れをみることができるが, 501_要求仕様_明確さと 5120_要求仕様変更発生状況_総合テスト (ベンダ確認) はこの流れだけでなく, 業種 (金融・保険業) と直結するパスがあることも示している. これについては 5.8 節の今後の課題で触れる.

5.7 分析結果の妥当性

(1) SEC データ特有の影響はないか

SEC データは国内の比較的良質なプロジェクトから得られたものが多いという点で偏っているのではないかとみられている. たとえそうだとした場合, 今回のようなプロジェクト間の差に着目した分析結果に対しては, たとえば

表 8 金融・保険業および要求レベルの要因と他の要因の関連

Table 8 Relationships between factors of industry and requirement levels and those of other categories.

変数名		データ件数					独立性の検定		
変数 1 (x_1)	変数 2 (x_2)	総数	$x_1 = 1, x_2 = 1$	$x_1 = 1, x_2 = 0$	$x_1 = 0, x_2 = 1$	$x_1 = 0, x_2 = 0$	ϕ 値	p 値 (%)	
業種_金融・保険業	514_要求レベル_性能・効率性	122	27	7	42	46	0.29	0.2	
	515_要求レベル_保守性	88	11	7	16	54	0.33	0.2	
	501_要求仕様_明確さ	193	50	12	80	51	0.19	0.7	
	要求仕様変更発生状況	5120_総合テスト (ベンダ確認)	42	13	2	11	16	0.44	0.4
514_要求レベル_性能・効率性	501_要求仕様_明確さ	118	56	13	26	23	0.30	0.1	
515_要求レベル_保守性	121_計画の評価 (品質)	87	26	1	43	17	0.28	0.6	
	要求仕様変更発生状況	5117_詳細設計	45	10	5	6	24	0.46	0.3
		5120_総合テスト (ベンダ確認)	42	12	2	12	16	0.41	0.9

生産性のレベルそのものを議論する場合と異なり、その影響はあったとしても小さいものと思われる。

(2) データのばらつきの影響はないか

稼働後 1 カ月におけるシステムの利用状況はプロジェクトによって異なる可能性が高い。また、目的変数であるシステム稼働後の不具合数のカウント基準にもプロジェクト間で等しいとは限らない。今回はそれらのばらつきはランダムにノイズとして加えられているものと仮定して分析した。もし、上位と下位のレベル間で利用状況やカウント基準に偏りがある場合は分析結果の信頼性は低下することになるが、極端な偏りが生ずるとは考えにくく、分析結果の信頼性の低下はあったとしても限定的と考えられる。

(3) 影響要因は本当に信頼性に影響を与えるか

今回の分析結果は厳密に言えば、仮説として提唱した因果関係（各質的変数の値の違いが信頼性に影響を与える）のうち否定されないものを確認したというべきものであって、因果関係そのものを立証したものではない。本来、因果関係の立証は定量的分析の外にあり、さまざまな観点から積み上げていくものである。今回得られた結果は筆者にとって納得感があり、各影響要因が信頼性に影響を与える（信頼性向上に寄与する）といってもよいと考えている。

(4) 改善効果はどれくらい期待できるか

今回の分析結果は信頼性の視点からのプロセス改善の指針の 1 つとして利用できるものと考えられるが、上位および下位レベルそれぞれに属するプロジェクト集合内での不具合数のばらつきも大きく、ある影響要因を改善したとしても不具合数比率で示した改善効果が必ず得られるわけではない。また、有効と思われる 2 つの影響要因を同時に採用したとしても、その効果が単純に加算されるわけではないことに留意する必要がある。

5.8 今後の課題

今後の課題には、a) 線形回帰モデルに基づく分析でも起こりうるものと、b) 今回の分析特有のものがある。後者はさらに b-1) 欠損値を含むいわゆる不完全データに起因するもの b-2) ゼロ過剰データに対して用いた負の二項回帰モデルに関するものに分けられる。

(1) 負の二項回帰モデルにおける効果量の推定

説明変数の変動によって目的変数の変動をどの程度説明できたかを示すことは、適用した回帰モデルの有効性を示す重要な指標となる。この指標は一般的に (r 族の) 効果量と呼ばれ、総変動 (目的変数の総平方和: S_T) に占める回帰による変動 (回帰平方和: S_R) の割合 $\eta^2 (=S_R/S_T)$ で表される [47]。

線形回帰モデルの場合、 R を実測値とモデルによる予測値 (回帰推定値) の対の集合の相関係数とすると、 η^2 は決定係数 R^2 と等しい [48] *11が、負の二項回帰モデルのように予測式が説明変数の線形結合だけで表すことができないモデルでは、一般に $R^2 = S_R/S_T$ が成立する保証はない。

また、線形回帰モデルでは、実測値と予測値の誤差の分散はすべてのプロジェクトで等しいという仮定 (等分散性の仮定) が設けられているが、負の二項回帰モデルでは 3.2 節で述べたように、この仮定は成り立たない。負の二項回帰モデルでは、形式的に求めた η^2 や R^2 の値は予測値や実測値が大きいものの影響を受けやすく、線形回帰モデルにおける効果量と同じ意味でとらえることはできない。

実際、SEC データのうち、新規開発、開発 5 工程、FP 規模と工数の両方のデータが揃っている 633 件のプロジェクトに対して工数を目的変数、FP 規模を説明変数として

*11 バイアスを除くために、 R^2 の値をデータ件数とパラメータ数 (説明変数の数) で補正した、「自由度調整済決定係数」を用いることが多い。

R^2 を求めたところ、(等分散の仮定が成立しない) 対数変換前の形式的な R^2 は 0.477、(等分散の仮定が成立すると考えられる) 対数変換後の R^2 は 0.708 となって両者の間にはかなりの差が見られた。

今後、負の二項回帰モデルなどの線形回帰モデル以外の効果量の定義とそれに従った厳密な計算方法を模索する必要がある (b-2 の問題)。

(2) 複数の質的変数による影響要因の分析

今回の分析における説明変数は、コントロール変数として加えた FP 規模を除くと質的変数 1 つである。複数の質的変数を説明変数として分析することにより、質的変数の信頼性に与える影響を詳しく知ることができる。しかしながら、今回 SEC データを対象に実際に分析を試みて次のような問題に直面した。

- データ件数が十分でない組合せが多い。
- 組合せが有意となる理由を説明することが難しい。
- 有意な組合せを体系的に説明することが難しい。

一般に説明変数を増やすと各セル (個々の説明変数の組合せ要素) の回答数が減り、2.2 節の (2) で示したデータ件数の 2 番目の条件: すべてのセルの回答数が 10 件以上という条件を満たすことが難しくなる。特に SEC データは典型的な不完全データのため、説明変数の数を増やしていくとデータ件数の条件を満たすデータ集合の数は完全データの場合よりも速く減る。不具合数と FP 規模を必須として、質的変数を 3 個にすると、すべてのセルの回答数が 10 件以上という条件を満たす組合せはゼロとなる。質的変数が 2 個の場合でも、1,126 件 ($= {}_{48}C_2 - 2$; マイナス 2 件は業種どうしの組合せとアーキテクチャどうしの組合せを除く処理) のすべての組合せのうちデータ件数の条件を満たすものは 365 (全体の 32.4%) しかなく、分析結果が偏ったものとなる可能性が高い。

重回帰分析と単回帰分析の結果の違いは、説明変数間での相関の高さと交互作用からおおむね説明することができる。しかし、すべての組合せに対して相関の高さや交互作用が生じた理由を、明確な根拠とともに示すことは難しい (a の問題)。2 つの説明変数の相関は疑似相関である可能性もある。不完全データの場合は、一般に単回帰分析で対象とするデータ集合と重回帰分析で対象とするデータ集合が異なるため、それら 2 つの集合間での説明変数の相関および交互作用の影響がさらに加わることになり、問題が一層難しくなる (b-1 の問題)。

有意となる組合せを単に羅列するだけでは意味がなく、要因どうしの因果関係を含めてそれらを体系付けて初めて分析結果は意味を持つものとなる。今回の重回帰分析から得られた有意な 2 つの質的変数の組合せは 20 件あったが、それらは全体としてみると脈絡がなく、なぜそれらの組合せだけが有意となるのかをソフト工学的観点から体系的に説明するのは困難であった (a および b-1 の問題)。

また、5.6 節で述べたように、業種、要求レベル、それ以外の分類の間には、おおむねこの順序で因果関係があると考えられる。ただし、表 8 で示したように、実際には影響要因どうしはより複雑なネットワーク構造をなしている。この構造を具体的にモデル化し、その妥当性を検証する有力な方法として共分散構造分析がある。しかし、共分散構造分析は線形回帰モデルをベースにしているため、この節の (1) で述べたように等分散の仮定が成立しない負の二項回帰モデルによる分析では、この方法を適用することができない (b-2 の問題)。

これらの問題を解決するためには、必要な情報を網羅するデータの一層の蓄積、現場での開発プロセスの詳細な把握、さらにはゼロ過剰データに対しても分析可能な共分散構造分析の開発などが必要である。

6. おわりに

信頼性に影響を与える要因が明らかになれば、それらの要因をコントロールすることにより開発プロセスを改善して信頼性を向上させることができる。SEC データには信頼性の指標としてのシステム稼働後の不具合数のデータだけでなく、開発プロセスに関連するさまざまな質的変数のデータが揃っているため、それらを分析することにより、信頼性への影響要因を明らかにすることができる。

しかしながら、システム稼働後の不具合数は、稼働後の早い段階では多くのプロジェクトでゼロであり、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線形回帰分析を行うことができない。そこで、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。

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、ユーザ側の協力が重要な役割を果たすことが分かった。すなわち、ユーザ担当者の要求仕様への関与、明確な要求仕様と要求仕様変更発生の抑制、ユーザ担当者の豊富な業務経験と設計内容の十分な理解が信頼性を向上させる。

また、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、プロジェクトの達成目標とその優先度を非常に明確にした場合、具体的には QCD の数値目標および優先度を文書で明確に定義した場合、品質に対する計画の評価をきちんと行った場合、具体的には基本設計開始時点で品質目標を明確にしてその実行可能性を検討した場合も信頼性向上に重要な役割を果たすことが分かった。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。

また、信頼性の影響要因の生産性への影響を調べた結果、次のことが明らかとなった。

- 生産性も信頼性も向上するのは達成目標とその優先度を非常に明確にした場合だけである。
- 信頼性は向上するが生産性の低下が見られないのは、品質に対する計画の評価をきちんと行った場合、テスト体制のスキルレベルが高い場合、要求仕様変更を抑制した場合である。
- 信頼性は向上するが生産性は低下するのは、要求レベルが高い場合、要求仕様を明確にした場合、デバッグ・テストツールを利用した場合である。

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 が連携して実施したものである。IPA/SEC の松本前所長、山下リーダーなら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.W. et al.: *Software Cost Estimation with Cocomo II*, Prentice-Hall, Inc. (2000).
- [3] Goel, A.L. and Okumoto, K.: Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures, *IEEE Trans. Reliability*, Vol.R-28, No.3, pp.206–211 (1979).
- [4] Yamada, S., Ohba, M. and Osaki, S.: S-Shaped Reliability Growth Modeling for Software Error Detection, *IEEE Trans. Reliability*, Vol.R-32, No.5, pp.475–478 (1983).
- [5] Furuyama, T. and Nakagawa, Y.: A Manifold Growth Model that Unifies Software Reliability Growth Models, *Int. J. Reliability, Quality and Safety Engineering*, Vol.1, No.2, pp.161–184 (1994).
- [6] 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).
- [7] Febrero, F., Calero, C. and Moraga, M.Á.: A Systematic Mapping Study of Software Reliability Modeling, *Informat. Softw. Technol.*, Vol.56, No.8, pp.839–849 (2014).
- [8] Akiyama, F.: An Example of Software System Debugging, *Information Processing*, Vol.71, pp.353–359, North-Holland (1972).
- [9] Agresti, W.W. and Evancho, W.M.: Projecting Software Defects from Analyzing Ada Designs, *IEEE Trans. Softw. Eng.*, Vol.18, No.11, pp.988–997 (1992).
- [10] Ohlsson, N. and Alberg, H.: Predicting Fault-prone Software Modules in Telephone Switches, *IEEE Trans. Softw. Eng.*, Vol.22, No.12, pp.886–894 (1996).
- [11] 角田雅照, 玉田春昭, 森崎修司, 松村知子, 黒崎 章, 松本健一: コードレビュー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).
- [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] 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).
- [14] Graves, T.L., Karr, A.F., Marron, J.S. and Siy, H.: Predicting Fault Incidence Using Software Change History, *IEEE Trans. Softw. Eng.*, Vol.26, No.7, pp.653–661 (2000).
- [15] Ostrand, T.J., Weyuker, E.J. and Bell, R.M.: Predicting the Location and Number of Faults in Large Software Systems, *IEEE Trans. Softw. Eng.*, Vol.31, No.4, pp.340–355 (2005).
- [16] 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).
- [17] 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*, Vol.49, pp.32–43 (2007).
- [18] Munson, J.C. and Khoshgoftaar, T.M.: The Detection of Fault-Prone Programs, *IEEE Trans. Softw. Eng.*, Vol.18, No.5, pp.423–433 (1992).
- [19] Khoshgoftaar, T.M., Allen, E.B., Kalaichelvan, K.S. and Goel, N.: Early Quality Prediction: A Case Study in Telecommunications, *IEEE Softw.*, Vol.13, No.1, pp.65–71 (1996).
- [20] Basili, V.R., Briand, L.C. and Melo, W.L.: A Validation of Object-Oriented Design Metrics as Quality Indicators, *IEEE Trans. Softw. Eng.*, Vol.22, No.10, pp.751–761 (1996).
- [21] Khoshgoftaar, T.M. and Allen, E.B.: Logistic Regression Modeling of Software Quality, *Int. J. Reliability, Quality and Safety Engineering*, Vol.6, No.4, pp.303–317 (1999).
- [22] Shatnawi, R. and Li, W.: The Effectiveness of Software Metrics in Identifying Error-Prone Classes in Post-release Software Evolution Process, *J. Syst. Softw.*, Vol.81, No.11, pp.1868–1882 (2008).
- [23] 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合せた fault-prone モジュール判別方法, 情報処理学会論文誌, Vol.49, No.12, pp.3954–3966 (2008).
- [24] Yang, Y., Zou, Y., Lu, H., Chen, L., Chen, Z., Xu, B., Leung, H. and Zhang, Z.: Are Slice-Based Cohesion Metrics Actually Useful in Effort-Aware Post-Release Fault Proneness Prediction? An Empirical Study, *IEEE Trans. Softw. Eng.*, Vol.4, No.4, pp.331–357 (2015).
- [25] Briand, L.C., Basili, V.R. and Hetmanski, C.J.: Developing Interpretable Models with Optimized Set Reduction for Identifying High-Risk Software Component, *IEEE Trans. Softw. Eng.*, Vol.19, No.11, pp.1028–1044 (1993).
- [26] Pai, G.J. and Dugan, J.B.: Empirical Analysis of Software Fault Content and Fault Proneness Using Bayesian Methods, *IEEE Trans. Softw. Eng.*, Vol.33, No.10, pp.675–686 (2007).
- [27] Dejaeger, K., Verbraken, T. and Baesens, B.: Toward Comprehensive Software Fault Prediction Models Using Bayesian Network Classifiers, *IEEE Trans. Softw. Eng.*, Vol.39, No.2, pp.237–257 (2013).
- [28] Vandecruys, O., Martens, D., Baesens, B., Mues, C., Backer, M.D. and Haesen, R.: Mining Software Repositories for Comprehensive Software Fault Prediction Models, *J. Syst. Softw.*, Vol.81, pp.823–839 (2008).
- [29] Moeyersoms, J., de Fortuny, E.J., Dejaeger, K. and Baesens, B. and Martens, D.: Comprehensive Software

- Fault and Effort Prediction: A Data Mining Approach, *J. Syst. Softw.*, Vol.100, pp.80–90 (2015).
- [30] Okamura, H. and Dohi, T.: A Novel Framework of Software Reliability Evaluation with Software Reliability Growth Models and Software Metrics, *Proc. International Symposium on High-Assurance Systems Engineering (HASE'14)*, pp.97–104 (2014).
- [31] Hall, T., Beecham, S., Bowes, D., Gray, D. and Counsell, S.: A Systematic Literature Review on Fault Prediction Performance in Software Engineering, *IEEE, Trans. Softw. Eng.*, Vol.38, No.6, pp.1276–1304 (2012).
- [32] Halstead, M.H.: *Elements of Software Science*, Chap. 11, Elsevier, North-Holland (1977).
- [33] Marciniak, J.J. (Ed.), 片山卓也, 土居範久, 鳥居宏次 (監訳): ソフトウェア工学大事典, pp.1420–1430, 朝倉書店 (2007).
- [34] Radjenović, D., Heričko, M., Torkar, R. and Živković, A.: Software Fault Prediction Metrics: A Systematic Literature Review, *Informat. Softw. Technol.*, Vol.55, No.8, pp.1397–1418 (2013).
- [35] Zhang, X. and Pham, H.: An Analysis of Factors Affecting Software Reliability, *J. Sys. Softw.*, Vol.50, No.1, pp.43–56 (2000).
- [36] Zhu, M., Zhang, X. and Pham, H.: A Comparison Analysis of Environmental Factors Affecting Software Reliability, *J. Sys. Softw.*, Vol.109, pp.150–160 (2015).
- [37] Zhu, M. and Pham, H.: Environmental Factors Analysis and Comparison Affecting Software Reliability in Development of Multi-Release Software, *J. Syst. Softw.*, Vol.132 (C), pp.72–84 (2017).
- [38] Gorla, N. and Lin, S.-C.: Determinants of Software Quality: A survey of Information Systems Project Managers, *Informat. Softw. Technol.*, Vol.52, No.6, pp.602–610 (2010).
- [39] Cameron, A.C. and Trivedi, P.K.: *Regression Analysis of Count Data*, 2nd ed., Cambridge University Press, p.566 (2013).
- [40] 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).
- [41] 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*, Vol.11, No.4, pp.40–47 (2016).
- [42] 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*, Vol.7, No.3, pp.105–111 (2011).
- [43] Peduzzi, P., Concate, J., Feinstein, A.R. and Holford, T.R.: Importance of Events per Independent Variable in Proportional Hazards Regression Analysis II. Accuracy and Precision of Regression Estimates, *J. Clin. Epidemiol.*, Vol.48, No.12, pp.1503–1510 (1995).
- [44] 永田 靖: サンプルサイズの決め方, 朝倉書店, pp.113–114 (2018).
- [45] Zeileis, A., Kleiber, C. and Jackman, S.: Regression Models for Count Data in R, *J. Statistical Software*, Vol.27, No.8, pp.1–21 (2008).
- [46] 渡部 洋, 鈴木規夫, 山田文康, 大塚雄作: 探索的データ解析入門, pp.33–37, 朝倉書店 (1991).
- [47] 大久保街亜, 岡田謙介: 伝えるための心理統計, pp.69–92, 勁草書房 (2012).
- [48] 奥野忠一, 久米 均, 芳賀敏郎, 吉澤 正: 多変量解析法, pp.44–45, 日科技連出版 (1978).



古山 恒夫 (正会員)

1945年生。1968年東京大学工学部計数工学科卒業。1973年同大学大学院博士課程修了。同年日本電信電話公社入社。横須賀電気通信研究所で言語処理プログラムの研究開発に従事。日本電信電話(株)ソフトウェア研究所でソフトウェアプロジェクト管理法, ソフトウェア品質保証法, ソフトウェア見積り法等の研究実用化に従事。1996年東海大学教授。2015年同大学客員教授。1994年度情報処理学会山下記念研究賞受賞。工学博士。IEEE, 電子情報通信学会各会員。本会シニア会員。