

# エンタープライズ系ソフトウェアの 重大度別信頼性への質的影響要因の分析

古山恒夫<sup>1,a)</sup>

概要：IPA 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、具体的には新規開発ソフトウェアの稼働後不具合数に影響を与える質的要因を不具合の重大度別に明らかにした。いずれの重大度レベルでも稼働後不具合数がゼロの値をもつプロジェクトは 30 % 以上占めるため、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、重大度レベルによって信頼性に与える影響要因が異なることがわかった。重大な不具合を減少させるためには、計画段階での品質目標の実行可能性の検討とユーザ担当者の豊富なシステム経験が重要な役割を果たす。中度の不具合を減少させるためには、設計支援ツールの利用、デバッグ\_テストツールの利用、ユーザ担当者による設計内容の十分な理解、ユーザ担当者による受け入れ試験の関与、テストチームの高いスキルレベル、信頼性、性能・効率性、移植性への高い要求レベルが重要な役割を果たす。デバッグテストツールの利用は軽微な不具合を減少させるのに役立つ。コードジェネレータの利用は、少なくとも重大な不具合や軽微な不具合を増加させる恐れがある。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。

## Analysis of Qualitative Factors that Affect Reliability of Enterprise Software for each Severity Level of Failures

### 1. はじめに

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<sup>\*1</sup>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。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。

(b) は、テスト工数を早期に把握するのに有効であり既に多くの研究が行われている。

(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り、現在でも多くの研究が行われている。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、効率よくテストを行うことができる。

(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 5 件しかない ([1], [2], [3], [4], [5])。このうち [2], [3], [4] の研究は同一の指導者 H. Pham 氏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<sup>\*2</sup>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。

[5] の研究では目的変数として 5 つの品質特性を取り上げ

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

<sup>a)</sup> furuyama@tokai-u.jp

<sup>\*1</sup> 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。

<sup>\*2</sup> 論文では具体的な定義は行われていない。

ていてその中のひとつに信頼性がある。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個とりあげている。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)のランク及びユーザからの仕様変更要求回数に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。

これらの研究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、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。これに対して筆者は、稼働後(サービスイン後)の発生不具合数および開発プロセスに係る多くの質的変数を備えたIPAのデータリポジトリ(IPAデータ)を用いて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[1]。分析対象としたプロジェクトは新規開発プロジェクトの305件であったが、そのうちの約2/3のプロジェクトでは、稼働後の発生不具合数を、発生した不具合による影響の大きさを考慮した重大度別に分計している。そこで今回新たに3つの重大度レベル(重大、中度、軽微)ごとに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。本論文における分析対象データ、分析方法および考察の枠組みは上記論文で記載したものを踏襲しているため、本論文の1章から3章までは、特に断りなく上記論文から抜粋し、必要に応じて適宜修正して記述している。

以下、2章で分析対象データの概要を重大度別に紹介する。3章で分析方法の要点を述べる。4章で分析結果を、5章で分析結果に対する考察を述べ、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[7]のうち、次の条件を満たすものを対象とする。

1) システム稼働後の重大度別の発生不具合数が報告されている。重大度のレベルは次の3つである。

- ・重大：顧客へ損害を与え、緊急対応を要する。
- ・中度：顧客への損害はないが、緊急対応を要する。
- ・軽微：顧客への損害はなく、緊急対応も不要。

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<sup>\*3</sup>)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。

<sup>\*3</sup> 番号と名称はデータ白書[7]で定義されたもの。

### 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1ヶ月後の重大度別発生不具合現象数(5255\_発生不具合現象数(重大)1ヶ月, 5259\_発生不具合現象数(中度)1ヶ月, 5263\_発生不具合現象数(軽微)1ヶ月)とする。以下では稼働後1ヶ月後の発生不具合現象(数)を重大度別にそれぞれ単に重大な不具合(数)、中度の不具合(数)、軽微な不具合(数)と呼ぶ。

重大度別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[8]ため(常用)対数変換後のものである。重大度のレベルが重大な不具合数には、999件という大きなものが含まれていたため、表1にはそれを除いた場合の統計量も併せて記載している。

表1から次のことがわかる。

- 重大度別のデータ件数は、いずれも合計のデータ件数(5267\_発生不具合現象数(合計)1ヶ月)の45%程度である。
- ゼロデータ件数は重大な不具合で最も多く、次いで中度の不具合である。軽微な不具合で最も少ない。データ件数に対する割合はそれぞれ75%(3/4), 47%(1/2程度), 34%(1/3)である。
- 不具合数の平均および最大の大きさは、重大(999件を除いた場合) < 中度 < 軽微 < 重大(999件を含めた場合)である。
- FP規模は、平均、分散、最大、最小のいずれも重大度別にはよらずほぼ等しい。

#### (2) 説明変数

信頼性に影響を与える可能性のある変数として、順序尺度に従う質的変数のうちから表2に示す51個、名義尺度に従う質的変数のうちから表3に示す4つの業種と5つのアーキテクチャを選んだ。名義尺度に従う質的変数の具体的な変数名は、例えば「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。

- 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 $\rho$ が $-0.7 \leq \rho \leq 0.7$ の範囲にある。

ただし、 $\rho$ 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。 $\rho$ 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	重大		中度		軽微		合計 (参考) [1]	
	不具合数	FP 規模	不具合数	FP 規模	不具合数	FP 規模	不具合数	FP 規模
データ件数 (A)	142 (141)		138		135		305 (304)	
ゼロデータ件数 (B)	106	0	65	0	46	0	99	0
(B/A)	0.75	0.0	0.47	0.0	0.34	0.0	0.32 (0.33)	0.0
平均	8.7 (1.7)	3.13	5.7	3.09	6.2	3.10	14.5 (11.2)	3.09 (3.08)
分散	7,81.6 (77.4)	0.21	190.5	0.19	164.0	0.20	4,086.1 (890.0)	0.21
最大	999 (86)	4.16	88	4.16	100	4.32	999 (415)	4.32
最小	0	2.05	0	2.06	0	2.06	0	1.93

(注) FP 規模の平均, 分散, 最大, 最小はいずれも常用対数変換後の値. 不具合数の括弧内は最大の不具合数 999 件をもつプロジェクトを除いた場合の値 (含めた場合と等しい場合は記載せず).

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 (略称)	変数	
プロジェクト全般 (P) (11 個)	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	
	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 (T) (11 個)	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 (U) (14 個)	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)
	ユーザー担当者	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_ユーザーとの役割分担_責任所在_明確度合	
	ユーザー承認有無	506_要求仕様, 508_設計
要求レベル (R) (8 個)	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 (D) (7 個)	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 [7] 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 (略称)	変数
業種 (I) (4 個)	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 (A) (5 個)	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

## 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

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 値化する分割点は複数個あるが, 偏り率  $\rho$  の絶対値が最も小さくなるものを分割点とする. 本研究では原則として下位レベルをベースになる群とみなすこととして, 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. 例えば, 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, 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 回帰モデル

回帰モデルとしては、一般化線形モデルのひとつである負の二項回帰モデルを用いる。負の二項回帰モデルは負の二項分布と呼ばれる確率関数を用いて回帰分析を行う方法である（文献 [6] の 3.3）。負の二項回帰モデルでは、説明変数（群）の線形結合と目的変数の平均  $\mu$  を対数関数で結合する。例えば、 $i$  番目のプロジェクトの説明変数の値  $x_i$  から  $\mu_i$  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）。

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

これを用いて  $N$  個のデータに対する尤度関数

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

が最大になるようにパラメータ  $(\alpha, \beta_0, \beta_1)$  を定める（最尤推定法）。

今回の分析では自然対数で変換された 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  $\mu$  を推定する。

$$\ln \mu = \beta_0 + \beta_1 \ln FP + \beta_2 x \quad (4)$$

計算は統計ツール  $R$  を用いて行う。

#### 3.2 要因の選択基準

信頼性に影響を与える要因の選択は、回帰係数の  $p$  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

不具合の重大度ごとに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 3.2 の要因の選択基準をすべて満たす質的変数を表 4、表 5、表 6 に示す。スペースの関係で、上位レベルおよび下位レベルの具体的な内容は省略したが、いずれの変数でも上位レベルは下位レベルよりもプロセスのレベルは高い。例えば、重大な不具合への影響要因（候補）である、503\_ユーザ担当者\_システム経験の上位レベルは、「a:十分に経験+b:概ね経験」であり、下位レベルは、「c:経験が不十分+d:未経験」である。また、ツールの利用では、常に「a:あり」が上位レベル、「b:なし」が下位レベルである。2.2 の (2) で述べたように、回帰

表 4 重大度が重大な不具合の影響要因とみなされた質的変数  
Table 4 Qualitative variables identified as factors effective to failures in severe level.

分類	変数名	データ件数		偏回帰係数	p 値 (%)
		上位レベル	下位レベル		
P	121_計画の評価 (品質)	88	23	-3.61	0.0
T	404_プロジェクト管理ツール利用	49	11	2.66	1.2
	411_コードジェネレータ利用	12	38	2.64	0.1
	422_開発フレームワーク利用	39	42	2.69	0.0
U	503_ユーザ担当者_システム経験	20	15	-2.17	0.9
D	601_PM スキル	45	30	1.90	2.0
I	業種_金融・保険業	59	83	-3.20	0.0

分析の説明変数は FP 規模とひとつの質的変数であることから、各表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 1 に示したように、「重大」な不具合数の最大値は 999 件である。2 番目に大きな不具合数は 86 件であることから、999 件という数が分析結果に与える影響が過度に大きくなっている可能性がある。そこで、文献 [1] で行ったように、このプロジェクトを除いた場合の偏回帰係数の  $p$  値の変化を調べた。999 件を含めた場合の偏回帰係数の  $p$  値が 10% 以下で、999 件を除いた場合の偏回帰係数の  $p$  値が 10% を超えた変数は、112\_役割分担\_責任所在 ( $p$  値は 66.2%，以下同様)、113\_達成目標\_優先度\_明確度合 (62.3%)、121\_計画の評価 (品質) (10.8%)、5241\_品質保証体制\_基本設計 (60.5%)、403\_類似プロジェクト利用 (45.5%)、406\_設計支援ツール利用 (20.3%)、業種\_卸売・小売業 (81.0%)、アーキテクチャ\_3 階層クライアントサーバ (84.9%)、アーキテクチャ\_インターネット・イントラネット (65.6%) の 9 個であった。これらの変数のうち、999 件を除いた場合の  $p$  値が 10.8% の 121\_計画の評価 (品質) だけは要因に含めることとし、それ以外のは要因として選定することを保留することとした。表 4 はその結果を示している。

#### 4.1 重大度の高い信頼性への影響要因

表 4 から次の場合に重大な不具合が発生しにくいことがわかる。

- 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。
- 404\_プロジェクト管理ツール、411\_コードジェネレータ、422\_開発フレームワークを利用しない場合。

表 5 重大度が中度の不具合の影響要因とみなされた質的変数  
**Table 5** Qualitative variables identified as factors effective to failures in middle level.

分類	変数名	データ件数		偏回帰係数	p 値 (%)
		上位レベル	下位レベル		
P	114_作業スペース	30	41	1.15	1.1
T	403_類似プロジェクト_有無	28	24	-0.97	6.8
	406_設計支援ツール利用	26	35	-1.36	0.9
	408_デバッグ_テストツール利用	22	39	-1.02	7.1
	412_開発方法論利用	44	15	-1.33	1.7
U	507_ユーザ担当者_設計内容理解度	38	13	-1.20	3.9
	509_ユーザ担当者_受け入れ試験関与	40	31	-0.97	3.1
R	512_信頼性	41	28	-0.87	6.6
	514_性能・効率性	42	27	-0.98	4.1
	516_移植性	33	14	-1.14	7.1
D	1010_テスト体制 (スキルレベル)	29	14	-1.19	4.5
A	アーキテクチャ_2階層クライアントサーバ	22	116	-1.26	1.1

表 6 重大度が軽微な不具合の影響要因とみなされた質的変数  
**Table 6** Qualitative variables identified as factors effective to failures in slight level.

分類	変数名	データ件数		偏回帰係数	p 値 (%)
		上位レベル	下位レベル		
T	408_デバッグ_テストツール利用	24	33	-1.56	0.1
	411_コードジェネレータ利用	11	41	1.12	2.9
R	516_移植性	27	14	-0.91	8.1
	518_セキュリティ	38	26	-0.68	6.1
I	業種_金融・保険業	55	80	-0.83	0.2

- 503\_ユーザ担当者のシステム経験が豊富な場合.
- 601\_PM のスキルが低い場合.
- 金融・保険業の場合.

#### 4.2 重大度が中度の信頼性への影響要因

表 5 から次の場合に中度の不具合が発生しにくいことがわかる.

- 114\_作業スペースが狭い場合.
- 403\_類似プロジェクト, 406\_設計支援ツール, 408\_デバッグ\_テストツール, 412\_開発方法論を利用する場合.
- 507\_ユーザ担当者の設計内容理解度が高い場合, 509\_

ユーザ担当者が受け入れに十分関与する場合.

- 512\_信頼性, 514\_性能・効率性, 516\_移植性への要求レベルが高い場合.
- 1010\_テスト体制においてスキルレベルが高い場合.
- アーキテクチャが 2 階層クライアントサーバの場合.

#### 4.3 重大度が軽微な信頼性への影響要因

表 6 から次の場合に軽微な不具合が発生しにくいことがわかる.

- 408\_デバッグ\_テストツールを利用する場合, 411\_コードジェネレータを利用しない場合.
- 516\_移植性や 518\_セキュリティへの要求レベルが高い場合.
- 金融・保険業の場合.

### 5. 考察

#### 5.1 常識に反する結果と他の要因の影響分析

偏回帰係数が正, すなわち上位レベルが下位レベルよりも不具合数が多く, 信頼性が低くなる要因を次に示す. これらはいずれも常識に反するように思える.

- 重大な不具合に関しては, 404\_プロジェクト管理ツールの利用, 411\_コードジェネレータの利用, 422\_開発フレームワークの利用, 601\_PM スキル.
- 中度の不具合に関しては, 114\_作業スペース.
- 軽微な不具合に関しては, 411\_コードジェネレータの利用.

これらの変数が真に要因であるのか, それとも何らかの理由によって要因のように見えていたのかを明らかにするために, 他の要因の影響を分析した.

線形回帰分析の場合, 変数  $x_1$  が目的変数に影響を与えていない場合でも, それが目的変数に影響を与えている変数  $x_2$  と相関がある場合には,  $x_1$  の単回帰係数が見かけ上有意となることがある. このような場合,  $x_1$  と  $x_2$  を説明変数とした重回帰分析を行うと,  $x_1$  の偏回帰係数の絶対値は小さくなって  $x_1$  が見かけ上の影響要因であったということがわかる.

負の二項回帰分析の場合にも, 理論的な解析は難しいものの, 同様な可能性が考えられるので, ここまで要因 (の候補) とみられてきた変数 ( $x_1$  に相当) に対して, それと相関のある他の要因 (の候補) ( $x_2$  に相当) およびコントロール変数としての FP を加えた重回帰分析を行ってここまで要因 (の候補) とみられてきた変数の回帰係数の p 値の変化を調べた.

次のすべての条件を満たす場合に  $x_1$  を要因として選定することを保留する.

- $x_1$  と  $x_2$  の共通集合においてもそれぞれの単回帰係数 (ただし, FP 規模をコントロール変数として加えるため, 厳密に言えば偏回帰係数) が有意である.

- (b-1)  $x_2$  が質的変数の場合は,  $x_1$  と  $x_2$  が独立でない ( $2 \times 2$  のクロス集計表で  $\chi^2$  検定の  $\phi$  係数が有意である).
- (b-2)  $x_2$  が量的変数の場合は,  $x_1$  と  $x_2$  に相関がある ( $x_1$  と  $x_2$  の相関係数の  $p$  値が有意である).
- (c)  $x_1$  と  $x_2$  を説明変数とする重回帰分析 (コントロール変数としての FP 規模も説明変数に加える) を行った結果,  $x_1$  の偏回帰係数は有意でないが,  $x_2$  の偏回帰係数は有意である.

ここで有意水準はいずれの場合も 10% とする.

$x_2$  が質的変数の場合の分析結果を表 7 に示す. なお,  $x_2$  が量的変数の場合には該当する組合せは存在しなかった.

#### (1) 重大な不具合の場合

重大な不具合では, 404\_プロジェクト管理ツールの利用と 422\_開発フレームワークの利用は, 真の要因でない可能性があることがわかった. 411\_コードジェネレータを利用しているプロジェクトはほとんどの場合, 404\_プロジェクト管理ツールと 422\_開発フレームワークを利用している. そのことが, プロジェクト管理ツールや開発フレームワークが信頼性低下要因ではないにも拘わらず, 不具合数を増加させる真の要因であるコードジェネレータの影響を受けてあたかも不具合数を増加させる要因であるかのような結果が得られた理由となっている.

以上の結果から, 重大な不具合に影響を与える要因として, 121\_計画の評価 (品質), 411\_コードジェネレータの利用, 503\_ユーザ担当者\_システム経験, 601\_PM スキル, 業種\_金融・保険業の 5 つを選定する.

#### (2) 中度の不具合の場合

中度の不具合では, 114\_作業スペース, 403\_類似プロジェクトの有無, 412\_開発方法論の利用は, 真の要因ではない可能性があることがわかった. 114\_作業スペースは, 理由は不明であるが, 中度レベルの信頼性向上要因である 406\_設計支援ツールの利用との  $\phi$  係数が  $-0.34$  と逆相関の関係がある. すなわち, 作業スペースが広いプロジェクトは設計支援ツールの利用度が低い. このことが表面上の信頼性低下要因として選定された理由と考えられる. 406\_設計支援ツールを利用しているプロジェクトはほとんどの場合 412\_開発方法論も利用している. 開発方法論の信頼性向上効果の大きな部分を設計支援ツールが占めているものと思われる.

以上の結果から, 中度の不具合に影響を与える要因として, 406\_設計支援ツールの利用, 411\_デバッグ\_テストツールの利用, 507\_ユーザ担当者\_設計内容理解度, 509\_ユーザ担当者\_受け入れ試験関与, 512\_要求レベル\_信頼性, 514\_要求レベル\_性能・効率性, 516\_要求レベル\_移植性, 1010\_テスト体制 (スキルレベル), アーキテクチャ\_2 階層クライアントサーバの 9 個を選定する.

#### (3) 軽微な不具合の場合

軽微な不具合では, 516\_要求レベル\_移植性と 518\_要求

レベル\_セキュリティは, それぞれ真の要因ではない可能性があることがわかった. 518\_要求レベル\_セキュリティの高いプロジェクトは, ほとんどの場合 516\_要求レベル\_移植性も高い. 移植性への要求レベルの高さが信頼性向上に寄与しているというよりは, セキュリティへの要求レベルの高さが信頼性向上に寄与していると言える. さらに, 金融・保険業のソフトウェアは, ほとんどの場合 518\_要求レベル\_セキュリティが高い. 金融・保険業のソフトウェアは他の業種よりさまざまな面で要求レベルが高く, それを満たすために多くの技術を駆使した高い開発プロセスでソフトウェアを開発していると考えられる. 信頼性の向上はセキュリティへの要求レベルの高さもさることながら, 金融・保険業のソフトウェアであることによる部分が大きいと言える.

以上の結果から, 軽微な不具合に影響を与える要因として, 408\_デバッグ\_テストツールの利用, 411\_コードジェネレータの利用業種\_金融・保険業の 3 つを選定する.

## 5.2 重大度による影響要因の違い

表 8 に不具合の重大度別の信頼性に関する要因の偏回帰係数を示す. 表 8 から次のことがわかる.

- 重大な不具合を減らすためには, 121\_計画の評価 (品質), 具体的には計画段階での品質目標の実行可能性の検討, と 503\_ユーザ担当者の豊富なシステム経験が重要な役割を果たす.
- 中度の不具合を減らすためには, 406\_設計支援ツールの利用, 408\_デバッグ\_テストツールの利用, 507\_ユーザ担当者の設計内容の十分な理解, 509\_ユーザ担当者の受け入れ試験への十分な関与, 1010\_スキルレベルの高いテストチームが重要な役割を果たす. また, 信頼性, 性能・効率性, 移植性の要求レベルが高いと結果的に中度の不具合数が減少する. さらに, 2 階層クライアントサーバは中度の不具合を減少させる.
- 軽微な不具合を減らすためには, 408\_デバッグ\_テストツールの利用が重要な役割を果たす.
- 重大度別で共通となる要因は, 408\_デバッグ\_テストツールの利用, 411\_コードジェネレータの利用, 業種\_金融・保険業である.
- 408\_デバッグ\_テストツールの利用は中度および軽微な不具合を減らす効果はみられるが, 重大な不具合を減少させる効果はみられない.
- 411\_コードジェネレータの利用は, 重大な不具合および軽微な不具合を増加させる (中度の不具合に対してはデータ件数が条件を満たさないため不明).
- 411\_コードジェネレータの利用以外に信頼性にマイナスの影響を及ぼす要因としては, 601\_PM スキル (の高さ) がある.
- 金融・保険業では重大な不具合および軽微な不具合が

表 7 他の質的要因の影響を受けた質的要因の候補

Table 7 Candidate qualitative factors affected by other qualitative factors

重大度	変数名		データ件数				回帰分析				独立性の検定		
	被影響要因 ( $x_1$ )	影響要因 ( $x_2$ )	総数	$x_1=1,$	$x_1=1,$	$x_1=0,$	$x_1=0,$	偏回帰係数		p 値 (%)		$\phi$ 係数 (注)	p 値 (%)
				$x_2=1$	$x_2=0$	$x_2=1$	$x_2=0$	$x_1$	$x_2$	$x_1$	$x_2$		
重大	404_プロジェクト管理ツールの利用	411_コードジェネレータの利用	48	10	28	0	10	1.29	2.58	19.5	0.1	0.26	7.2
	422_開発フレームワークの利用	404_プロジェクト管理ツールの利用	48	27	4	11	6	0.89	2.00	30.2	7.0	0.26	7.5
	422_開発フレームワークの利用	411_コードジェネレータの利用	47	10	20	1	16	0.39	2.68	64.6	0.2	0.31	3.2
中度	114_作業スペース	406_設計支援ツールの利用	49	9	19	14	7	0.68	-1.30	23.2	2.2	-0.34	3.5
	403_類似プロジェクトの有無	412_開発方法論の利用	52	24	4	15	9	0.29	-2.19	54.9	0.0	0.27	5.4
	412_開発方法論の利用	406_設計支援ツールの利用	56	25	16	1	14	-0.74	-1.01	23.4	8.1	0.48	0.1
軽微	516_要求レベル_移植性	518_要求レベル_セキュリティ	41	13	14	2	12	-0.34	-1.34	51.0	1.0	0.33	3.3
	518_要求レベル_セキュリティ	業種_金融・保険業	64	21	17	3	23	-0.40	-0.79	31.0	5.3	0.44	0.0

(注)  $\phi$  係数を  $2 \times 2$  の分割表におけるクラメールの連関係数とした場合は正の値しかとらないが、四分点相関係数（ピアソンの相関係数と等しい）を  $\phi$  係数とした場合は負の値もとりえる。ここでは後者の定義に従う。

表 8 不具合の重大度別の信頼性に関する要因の偏回帰係数

Table 8 Partial regression coefficients of factors in terms of reliability for each severity level.

分類	変数名	偏回帰係数		
		重大	中度	軽微
P	121_計画の評価 (品質)	-3.61	-	-
	406_設計支援ツール	-	-1.36	-
T	408_デバッグ_テストツール	-	-1.02	-1.56
	411_コードジェネレータ	2.64	*	1.12
U	ユーザ担当者	503_システム経験	-2.17	-
		507_設計内容理解度	*	-1.20
		509_受け入れ試験関与	-	-0.97
R	512_信頼性	-	-0.87	-
	514_性能・効率性	-	-0.98	-
	516_移植性	-	-1.14	-
D	601_PM スキル	1.90	-	-
	1010_テスト体制 (スキルレベル)	-	-1.19	-
I	業種_金融・保険業	-3.20	-	-0.83
A	アーキテクチャ_2 階層クライアントサーバ	*	-1.26	-

(凡例) \* : データ件数の条件を満たさない。- : p 値が 10 % を越える。

他の業種に比べて少ない。

### 5.3 常識に反する結果のさらなる検討

常識に反する結果として 5.1 でリストアップしたもののうち、411\_コードジェネレータの利用と 601\_PM スキルの

偏回帰係数が依然として表 8 に残っている。これらの結果の妥当性を再度検討する。

#### (1) コードジェネレータの利用

コードジェネレータの利用は重大な不具合に対しても軽微な不具合に対しても信頼性にはマイナスの影響を与えるという結果であった。中度の不具合に対してはデータ件数の条件を満たさないことから分析対象外としたため現時点では信頼性に与える影響の有無、ある場合にはそれがプラスかマイナスかは不明である。また、合計の不具合数に対する分析（文献 [1]）では、最大の不具合数 999 件を含めた場合も含めない場合も偏回帰係数は正の値であった<sup>\*4</sup>。いずれの結果も偏回帰係数は正の値であり、コードジェネレータが信頼性に対してマイナスであるという結果は常識外ではない可能性が高い。

#### (2) PM スキル

変数「601\_PM スキル」のレベルは、アルファベットの若番ほどスキルレベルは高い、すなわち上位レベルである。変数のレベル (a~d) ごとの重大な不具合数と FP 規模それぞれの平均を表 9 に示す。FP 規模の平均は常用対数変換後のものであり、その値を対数逆変換した値（元のスケールで中央値に相当）を表 9 に併せて示す。

レベル a の PM が開発したソフトウェアは平均 5.57 件の不具合を発生しているが、FP 規模も大きい。一方、レベル d の PM が開発したソフトウェアは平均 0.13 件の不

<sup>\*4</sup> 999 件を含めた場合の p 値は有意であったが、含めない場合の p 値が 43% だったため要因の選定は保留している。

表 9 PM スキルのレベル別の不具合数と FP の平均

Table 9 Mean values of number of failures and FP sizes for each PM skill level.

レベル	データ件数	平均		中央値
		不具合数	log(FP)	
a	14	5.57	3.50	3,150
b	13	0.23	3.23	1,710
c	18	1.17	3.06	1,160
d	30	0.13	3.06	1,140
全体	75	1.41	3.17	1,480

表 10 PM スキルの各レベル間の信頼性の差

Table 10 Differences of reliability between PM skill levels

	a	b	c	d
a	–	0.05	-1.65	2.12
b	97.1	–	-1.80	0.50
c	18.1	6.6	–	2.19
d	7.3	54.6	1.1	–

(注) 右上段：偏回帰係数，左下段：p 値 (%), いずれも上位レベルを 1, 下位レベルを 0 として重回帰分析で求めたもの

具合しか発生していないが，FP 規模は中央値と比較するとレベル a の半分以下である．FP 規模で正規化してもレベル d の方がレベル a よりも信頼性が高い．

そこで，PM スキルの a~d の各レベル同士すべての組合せに対して，対象とする 2 つのレベルのいずれかに属するプロジェクトを対象に FP 規模と 601\_PM スキルを説明変数とする（重）回帰分析を行って，601\_PM スキルの偏回帰係数が有意であるかどうかを調べた（表 10）．ただし，いずれの組み合わせにおいても上位レベルを 1, 下位レベルを 0 としている．信頼性の高さは，常識的には  $a > b > c > d$  の順が期待されるが，表 10 からは， $d > b > a > c$  の順となっている．ただし，レベル a とレベル b の差はほとんどない．レベル d は IT スキル標準のレベル 3 であり，定義によると「プロジェクト管理の経験なし」である．FP 規模が小さいだけでなく，恐らく，易しいプロジェクトを任せられたため不具合数が少なくなったものと思われる．レベル d のデータ件数が最も多いため，最適分割点が bc 間ではなく cd 間となったことで，PM スキルレベルが低いプロジェクトの方が信頼性が高いという結果が得られたものであり，601\_PM スキルを要因と選定するのは保留した方がよい．

## 6. おわりに

IPA の収集データには，開発プロセスに関連するさまざまな質的変数のデータだけでなく信頼性の指標としてのシステム稼働後の重大度別の不具合数のデータがある．今回それに基づいて不具合の重大度別にその影響要因を調べた．

分析の結果，影響要因は不具合の重大度によって大きく異なることがわかった．重大な不具合に対しては，計画段

階での品質の評価とユーザ担当者のシステム経験が，中度の不具合に対しては，設計支援ツールの利用，デバッグ・テストツールの利用，ユーザ担当者の設計内容の十分な理解と受け入れ試験への関与，高いスキルレベルのテストチームおよび高い要求レベルが，軽微な不具合に対してはデバッグ・テストツールの利用が重要な役割を果たす．コードジェネレータの利用は，中度の不具合に対してはデータ件数の不足から分析できなかったが，重大な不具合と軽微な不具合の増加をもたらす．どの施策がどの重大度レベルの不具合の減少に役立つかを知ることはプロジェクト運営にとって何らかの役に立つと思われる．

とは言え，今回の分析では要因の検出力は高くない．したがって，ある重大度レベルで要因として選定された変数が，他の重大度レベルで要因として選択されなくても実は要因である可能性も十分ありえる．現実のプロジェクトでは，重大・中度・軽微な不具合が混在していて，ある重大度レベルの不具合だけを狙い撃ちするような施策を講ずることはできないことから，実際のプロジェクト運営にあたっては，重大度レベルによらず，文献 [1] での結論も含めてあらゆる要因をできる限り施策に取り込むことが望ましいと考えられる．

謝辞 本研究は東海大学と IPA が連携して実施したものである．東海大学理学部数学科の山本義郎教授の貴重なアドバイスに感謝するとともに，IPA の山下リーダおよび五味研究員をはじめとする歴代研究員の方々のご協力に深く感謝致します．

## 参考文献

- [1] 古山恒夫：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変数の分析，情報処理学会論文誌，Vol.60, No.11, pp.2048–2062 (2019)．
- [2] Zhang, X. and Pham, H.: An Analysis of Factors Affecting Software Reliability, *J. Sys. Softw.*, Vol.50, No.1, pp.43–56 (2000)．
- [3] 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)．
- [4] 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)．
- [5] 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)．
- [6] Cameron, A. C. and Trivedi, P. K.: *Regression Analysis of Count Data*, 2nd ed., Cambridge Uni. Press, p.566 (2013)．
- [7] 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)．
- [8] 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)．