品質工学に基づく高信頼性ソフトウェア開発のための人的要因分析

髙橋　宗雄

1. はじめに

ソフトウェアの故障は，ソフトウェア開発における多種多様な人的要因の複雑な相互関係が根元的な原因で発生する．このため，ソフトウェアの信頼性管理では，故障の発生を事前に防止することは困難であり，故障が発生してからその原因（バグと呼ばれる）を分析して解決するという，対症療法的な方法をとっているのが現状である．

しかし，このような方法ではバグそのものは解決できても，バグがなぜソフトウェアに混入したかというバグの原因までは明らかにできないので，ソフトウェアの信頼性問題の本質は未解決のままである．すなわち，バグ混入の主要な原因である人的要因の相互関係を明らかにしないかぎり，ソフトウェアの信頼性問題を根本的に解決することは困難である．

このようなソフトウェアの信頼性問題に応えうる技術の開発を目的として，最近，品質工学[1]の手法により人的要因の相互関係を解明しようとする試みがいくつか報告されている[2]-[4]．品質工学は，すでにハードウェアの世界では多くの実践を通して技術開発の手法としての有効性が確認されており，ソフトウェアの分野でも今後応用が進むものと思われる．特に，品質工学はソフトウェア工学上の研究課題の一つであるソフトウェアプロセス改善の概念と相性がよく，品質工学とソフトウェア工学との融合から新しいソフトウェア信頼性技術の創造が期待される．

本稿では，新しいソフトウェア信頼性技術の創造に向けた基礎的研究の一例として，プログラムのコードレビューにおけるエラーとその原因である人的要因との関係を品質工学の手法を用いて分析した結果について紹介する．

たかはし　むねお　桐蔭横浜大学工学部
〒225-8502 横浜市青葉区鉄町 1614

2. ソフトウェア信頼性とエラー

ソフトウェア信頼性は故障の発生についての性質であり，その源泉はエラーである．エラーとは，ソフトウェア開発の種々の局面で，ソフトウェアにバグを生じさせる人間の誤りを意味する．エラーによりソフトウェアに混入したバグをコンピュータが実行すると，その結果が故障として観測される．すなわち，エラーがバグとして顕在化し，そのバグが直接の原因となって故障が発生する（図 1）．

エラーは，人間の思考過程で発生する誤りであるから，一過性で外部から観測することは難しい．Reason はエラーに対して現象と心理学的特性を包含した定義を与えている[5]．本稿では，Reason の定義におけるエラーの現象的側面に着目して，エラーによって生ずる現象（結果）をエラーとして捉える．現象は一般に観測することが可能であり，ソフトウェア開発では，その多くはバグに相当する．本稿では以後，特に断らないかぎりエラーとバグを区別しないですべてエラーという．

3. 人的要因モデル

ソフトウェアは一般に多数の人間による共同作業で開発される．このため，エラーはソフトウェア開発において人間が行う情報変換・伝達の過程で何らかの原因によって発生する．この原因となる主要なものが多

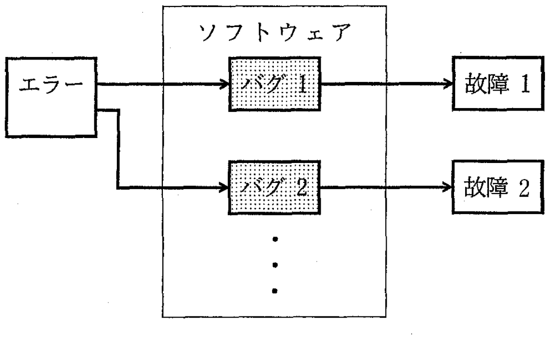
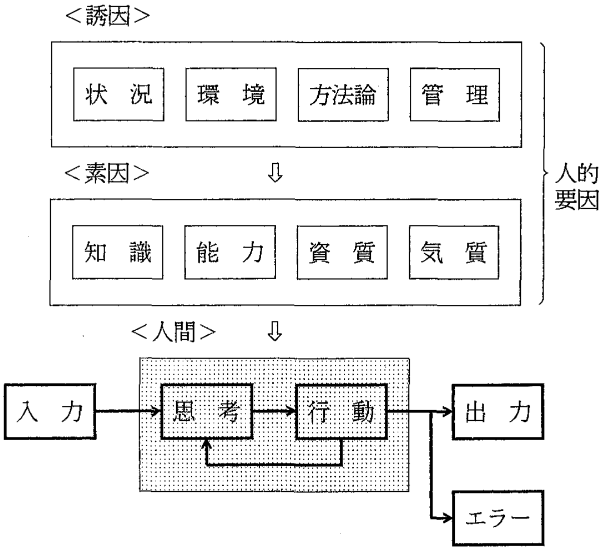


図 1　エラー，バグ，故障の関係

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．



入　力：ニーズ，要求仕様，プログラムなど
出　力：設計仕様，プログラム，テストケースなど
エラー：曖昧な要求，プログラムの論理誤りなど
知　識：業務知識，言語知識，ＯＳ知識など
能　力：思考力，理解力，分析力　表現力など
資　質：思考の柔軟性，リーダシップ，協調性など
気　質：外向的，内向的，慎重，粘着質など
状　況：単調作業，厳しい線表，コスト不足など
環　境：作業環境（騒音など），体調など
方法論：技法，ツール，プロセスなど
管　理：チーム編成，動機付け，役割認識など

図２ 人的要因モデル

種多様な人的要因の相互関係である．

人的要因の相互関係は多重構造になっていることが経験的に知られている．たとえば，エラーの発生は個人の経験や知識が関係しており，さらにそれらは作業環境などによって影響を受ける．このように，人的要因の相互関係は複雑な多重構造をしていると考えられるが，ここでは，エラーの発生に対する影響が直接的か間接的かに着目して，図２に示すような二つの要因から成る２重構造モデルを仮定する[6]．一つは，人の内部にあって人手による作業にエラーを起こさせる直接の原因となるものである．これを素因と呼ぶ．もう一つは，人の外部にあって，素因の影響を強めたり弱めたりする働きをするものであり，これを誘因と呼ぶ．素因は直接制御することが不可能または困難な要因であり，誘因は制御可能な要因である．素因と誘因を総称して人的要因と呼ぶ．

4. 品質工学を適用した実験

4.1 実験モデルの設定

３章で述べた人的要因モデルは，人的要因の相互関係を素因と誘因でモデル化したものであり，ソフトウェア開発のあらゆる局面に適用できるいわばメタモデルである．このメタモデルを用いて人的要因の仮説を検証するには，エラー，素因および誘因を具体的に設定して実験を行う必要がある．このためのモデルを実験モデルと呼ぶ．

表１ エラーの例

工　程	エラーの例
要求分析	・曖昧な要求 ・不完全な要求 ・矛盾のある要求
設　計 （コーディング）	・機能設計の不足 ・機能設計の誤り ・初期設定誤り ・データ定義誤り ・論理誤り ・パラメータの受渡し誤り
レビュー	・エラー見逃し ・エラー指摘誤り
テ ス ト	・テストケースの誤り ・値域や境界条件の抽出不足 ・結果の確認誤り

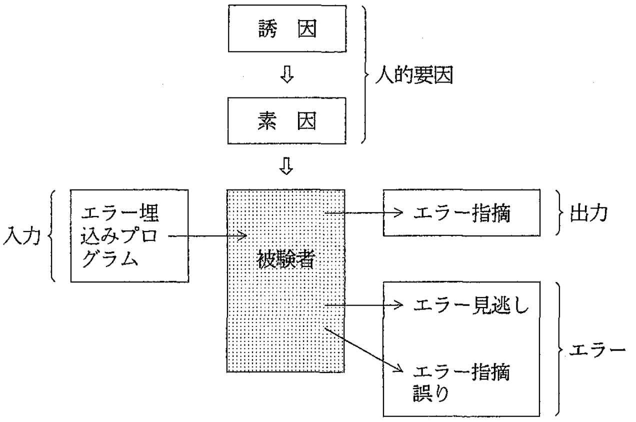


図３ 実験モデル

ソフトウェア開発は，要求分析，設計，コーディング，レビュー，テストなどの一連の工程に従って行われる．エラーはこれらの工程によって，表１に示すようにその現象が異なる．さらにエラーの現象によって素因および誘因は異なる．それゆえ，どの工程を対象にして実験を行うかは，実験モデルを設定する上で重要な考察要因である．

ソフトウェア開発で人的要因の影響が大きいと想定される工程の一つはレビューである．レビューを対象とした実験は，素因と誘因の具体的なイメージが描きやすく，実験モデルの設定が容易である．ここでは，品質工学の手法を導入した第１ステップの実験である

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

ことを考慮し，コードレビューを対象として，図3に示すような実験モデルを設定した．

4.2 実験の準備

(1) プログラムの選定

問題の知識に強く依存するような，あるいは複雑な言語機能を用いなければ記述できないようなプログラムは，実験の目的から適切ではない．そこで，日本語で記述されたプログラム仕様から一般的な知識だけでプログラムの意味が理解できる「数値の大小比較」の問題を選定し，C言語で記述した．

(2) エラーの埋め込み

プログラムの論理誤り（論理エラー）とC言語の文法誤り（構文エラー）の2種類のエラーを合計10個埋め込んだ．

(3) 素因の選定と測定

コードレビューにおけるエラー，すなわち埋め込んだエラーの見逃しと指摘誤りに対する直接の原因と想定される言語能力を素因として選定した．素因の測定は，被験者の自己評価に基づいて，“高い”，“低い”の2水準で行った．

(4) 誘因の選定と測定

コードレビューの経験に基づいて，エラーの見逃しや指摘誤りへの響度が大きいと想定されるエラー指摘目標値とチェックリストを誘因として選定し，それぞれ“あり”，“なし”の2水準で測定した．エラー指摘目標値は，従来の経験ではレビューにおけるエラー指摘数を増大させる効果があると言われており，エラーの見逃しが少なくなることが期待できる．チェックリストも，エラーの指摘に効果のあることが経験的に知られている．

(5) 被験者の選定と実験時間

被験者として，言語コンパイラ，OSなどの実用プログラムの開発経験が3〜10年の人を20名選定した．実験時間は，プログラムの行数がC言語で26行であり，複雑でないことと，被験者が初心者でないことを考慮して15分とした．

4.3 実験データの収集

プログラムに埋め込んだエラーの種類（論理エラーと構文エラーの2水準）をL_8直交表の外側に割り付け，言語能力の高低(A)，エラー摘出目標値の有無

表2 実験計画とデータ

因子 / No.	言語能力 A	目標値 B	A×B	CL C	A×C	B×C	e	エラー（R） 論理エラー（R_1）	構文エラー（R_2）
1	1	1	1	1	1	1	1	1.0	0
2	1	1	1	2	2	2	2	2.0	1.5
3	1	2	2	1	1	2	2	0.5	0.5
4	1	2	2	2	2	1	1	3.0	2.5
5	2	1	2	1	2	1	2	3.0	3.3
6	2	1	2	2	1	2	1	3.7	3.0
7	2	2	1	1	2	2	1	3.0	2.7
8	2	2	1	2	1	1	2	3.5	4.0
							計	19.7	17.5

CL：チェックリスト

表3 分散分析表

要因	変動 S	自由度 f	分散 V	分散比 F	純変動 S'	寄与率 ρ(%)
言語能力 A	14.440	1	←	73.9**	14.235	60.2
目標値 B	0.303	1				
CL C	5.290	1	←	27.0**	5.085	21.5
A×B	0.203	1				
A×C	1.440	1	←	7.4*	1.235	5.2
B×C	0.723	1				
e_1	0.023	1				
e_1'(e_1, B, A×B, B×C)	1.250	4				
エラー R	0.303	1	←	1.5°		
A×R	0.203	1	←	1.0°		
B×R	0.160	1				
C×R	0.003	1				
e_2	0.543	4				
e_2'(e_2, C×R)	0.705	6				
計	23.630	15			23.630	100.0
e_0(e_1'と e_2'をﾌﾟｰﾙ)	1.955	10	0.196			
e(e_0と°印をﾌﾟｰﾙ)	(2.460)	(12)	(0.205)		(3.075)	(13.1)

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

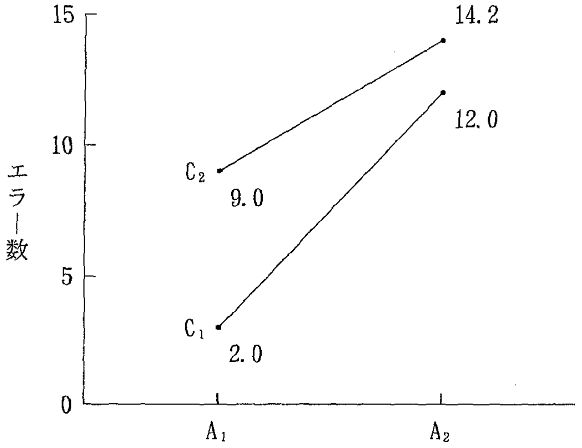


図4　言語能力とCLとの交互作用

(B)およびチェックリストの有無(C)の3因子をいずれも2水準としてL_8直交表に割り付けて実験を行い，データを収集した（表2）．

4.4　分析結果

A，B，Cの3因子について，直交表の外側に割り付けられたエラー(R)を論理エラー(R_1)と構文エラー(R_2)の2水準にとって，分散分析を実施した．また，3因子間の交互作用A×B，A×C，B×Cの効果についても分析した（表3）．分析の結果から以下のようなことが推察される．

(1)　言語能力は論理エラーおよび構文エラーの指摘に高度に有意な影響を与える．すなわち，言語の能力が低い人ほどエラーを指摘できないか，あるいは指摘誤りを起こしやすいといえる．しかし，Rとの交互作用はないので，論理エラーと構文エラーに対する効果は同じといえる．

(2)　エラー指摘目標値は論理エラーおよび構文エラーの指摘に有意な影響を与えるとはいえない．目標値はエラーの指摘に効果があると経験的には言われているが，実験の結果はそうならなかった．このような結果になった一つの原因は，レビュー時間が関係しているのではないかと推察される．実験では，レビュー時間が15分と短かったので，目標値を達成しようとするエラー指摘の努力が十分ではなかったのではないかと考えられる．

(3)　チェックリストは，論理エラーおよび構文エラーの指摘に有意な影響を与える．チェックリストも言語能力の場合と同様，Rとの交互作用はないので，論理エラーと構文エラーに対する効果は同じである．しかし，言語能力とチェックリストとの間に交互作用があるので，言語能力の高い人に対するチェックリストの有効性は，言語能力の低い人に対するそれよりも大きいといえる（図4）．

(4)　言語能力とチェックリストが論理エラーおよび構文エラーの指摘に与える影響度は，それぞれの寄与率から60%および22%程度である．

5.　おわりに

ソフトウェアの信頼性問題に根本的に応えるためには，信頼性の源泉であるエラーの発生原因を明らかにすることが必要である．このための新しいアプローチの一例として，プログラムのコードレビューにおけるエラーと，これの発生に影響を与える人的要因との関係を，品質工学の手法により分析した基礎的研究の結果を示した．この研究の目的は，エラーがどのような人的要因のどのような相互関係から発生するかを明らかにして，ソフトウェアプロセスを改善することにある．この意味で，研究から得られた知見は，最終的にはソフトウェア信頼性管理のガイドラインや，ソフトウェア技術者の教育・訓練カリキュラムなどに反映されることになる．

品質工学の適用例はハードウェア以外の分野では未だ少ないが，今後ハードウェア以外の分野でも適用が進むことを期待したい．本稿がその第1歩となれば幸いである．

参考文献

[1] 田口玄一，吉沢正孝，"品質工学講座1—開発・設計段階の品質工学"，日本規格協会，1988.

[2] 高橋宗雄，高田圭，山内成志，矢野宏，"ソフトウェア誤り検出能力の評価に関する基礎的研究"，品質工学，Vol. 4, No. 3, pp. 45-53, 1996.

[3] 高橋宗雄，"ソフトウェアの信頼性に影響を及ぼす人的要因の解明"，JISA会報，No. 45, pp. 71-80, 1997.

[4] 山田茂，影山高章，木村光宏，高橋宗雄，"コードレビューにおける人的エラーと人的要因に関する考察"，信学論A, Vol. J 81-A, No. 9, pp. 1238-1246, 1998.

[5] Reason, J., "Human Error," Cambridge University Press, 1990.『林喜男監訳：ヒューマンエラー，海文堂出版，1994.』

[6] 高橋宗雄，古宮誠一，三宅武司，"人的要因メタモデルの提案とその適用実験例"，第47回情報処理学会全国大会講演論文集，7 J-2, 1993.

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．