

人的要因メタモデルの提案と  
その適用実験例

7J-2

高橋宗雄

古宮誠一

石崎 豊

桐蔭学園横浜大学工学部

I P A 技術センター

1. はじめに

ソフトウェアの信頼性に影響を及ぼす人的要因は多種多様であり、ソフトウェアの信頼性はそれらの要因の複雑な相互作用により作り込まれる。したがって、人的要因によるエラー混入のメカニズムが解明できれば、それに基づいてソフトウェアの信頼性向上技術の開発や改善を行うことができる。これまでにも、ソフトウェアの信頼性に影響を及ぼす人的要因を明らかにする試みはいくつか行われている<sup>1)・2)</sup>。しかし、そのほとんどは仮説の提案にとどまっておき、ソフトウェアの開発実験により検証されたものは少ない。仮説は検証されて始めて意味がある。そこで本稿では、品質工学的アプローチに基づいて、人的要因の相互作用を分析するための人的要因メタモデルを提案する。さらに、メタモデルのひとつの適用例として、設計レビューを対象とした実験モデルを設定し、レビューにおける人的エラーと人的要因との関係を統計的手法により調べた予備実験の結果について報告する。

2. 人的要因モデルの設定

ソフトウェアの信頼性に影響を及ぼす人的要因をKJ法により抽出し、要因連関図にまとめたものを図1に示す。これから、人的要因は多種多様であり、かつそれらの相互関係は多重構造になっていることが分かる。このように、人的要因は複雑に関連し合う多重構造をなしているが、我々は人的エラーの生起に対する影響が直接的か間接的に着目して、次のような二つの要因からなる2重構造モデルを仮定した(図2)。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り、これを素因と呼ぶ。もう一つは人の外にあって、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。素因と誘因を併せて人的要因と呼ぶ。

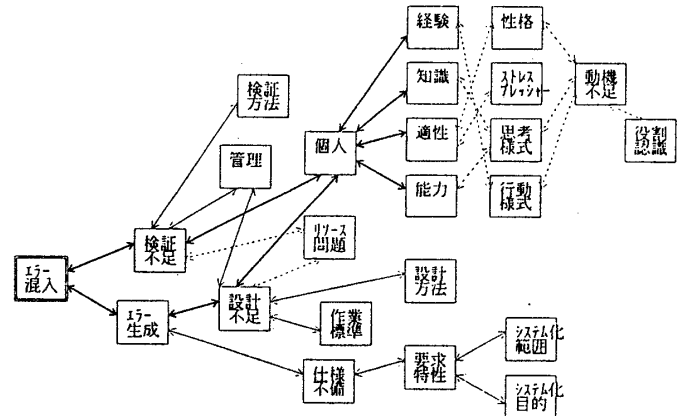
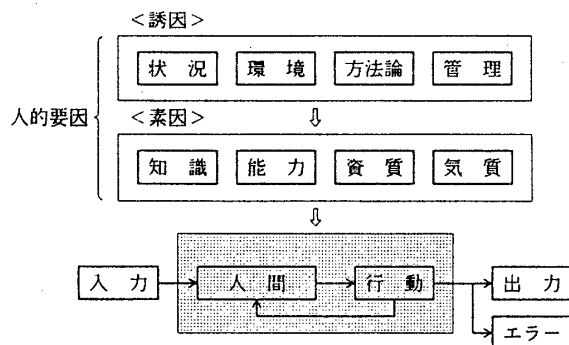


図1 人的要因の連関図(例)



- 入力 : 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 : アルゴリズム誤り、テストケースもれなど
- 知識 : 言語知識、業務知識、OSの知識など
- 能力 : 思考力、理解力、分析力、発見能力、表現力など
- 資質 : 思考の柔軟性、リーダーシップ、協調性など
- 気質 : 外向的、内向的、慎重、粘着など
- 状況 : 単調作業、きつい線表、コスト不足、プレッシャーなど
- 環境 : 作業環境(騒音など)、体調(疲労、寝不足など)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

図2 人的要因モデル

3. 実験モデルの設定

提案した人的要因モデルは、ソフトウェア開発のあらゆる局面に適用できるメタモデルであり、これを検証するためには具体的なインスタンスを設定して実験を行う必要がある。インスタンスはそれがどのように利用されるかを想定して設定されなければならない。このインスタンスを実験モデルと呼ぶ。たとえば、ソフトウェアの設計、設計レビュー、テストケース設計などを対象とした

A Meta-model of Human Factors and its Application

Muneo Takahashi

Toin University of Yokohama

1614 Kurogane, Midori, Yokohama 225, Japan

実験モデルが考えられる。これらの実験モデルの検証によりメタモデルを間接的に検証する。このための第1ステップとして、我々は設計レビューにおける人的エラー（設計エラーの見逃しと設計エラーの指摘間違い）と人的要因（レビューアの知識、能力の個人差）との関係に着目した実験モデルを設定した（図3）。このモデルを用いて、プログラムに既知の論理エラーを埋め込み、それをテスト形式で検出する実験を通して、人的エラーに有意な影響を及ぼす素因を明らかにする予備実験を試みた。

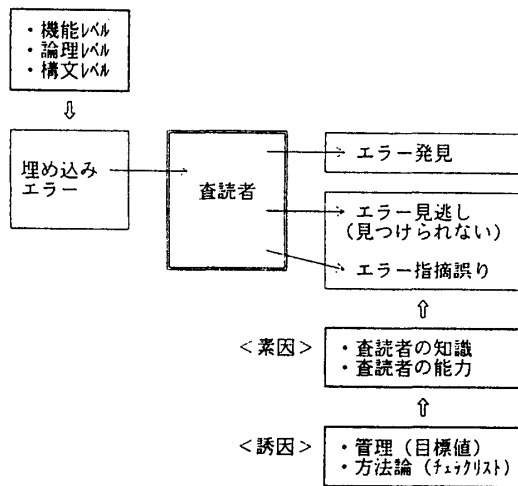


図3 実験モデル

4. 予備実験の計画

(1) 問題の選定

問題分野の知識に強く依存するような問題や複雑な言語機能を用いて記述しなければならない問題は不適切である。この実験では、日本語で記述されたプログラム仕様から、一般的な知識で問題の意味が理解できる数値の大小比較の問題を選定した。

(2) エラーの埋め込み

エラーについては、局所的な理解で分かるもの（大小比較エラー）とプログラム全体を理解しなければ分からないもの（初期設定エラー）の2種類の論理エラーと言語の構文エラーを合わせて10個（論理エラー60%）埋め込んだ。

(3) 素因の測定

被験者による自己評価に基づいて、知識（C言語、数学）および能力（思考力、理解力、問題発見力、分析力、表現力）を3段階で数値化した。

(4) 被験者の選定と実験時間

C言語の教育を1年間受けた工学系の学生を64名を対象にして、30分の時間内に何個エラーを発見できるかを調べる実験を実施した。

5. 予備実験の結果

分散分析の結果を表1に示す。表1から以下のことが推察される。

- (1) C言語の理解度は、論理エラーの見逃しに有意な影響を及ぼす。すなわち、C言語を理解していると答えた学生は論理エラーの見逃しと指摘間違いが少ない。
- (2) 数学の得意/不得意は構文エラーの見逃しに影響を及ぼす。すなわち、数学が得意であると答えた学生は構文エラーの見逃しが多い。数学が得意な学生は論理的な矛盾に興味あり、構文エラーの発見をおろそかにしたのか、あるいは単なる偶然かその理由はよく分からない。
- (3) 矛盾/欠陥発見能力は構文エラーの見逃しに影響を及ぼす。すなわち、発見能力があると構文エラーの見逃しが少ない。

表1 分散分析結果

素因		ヒューマンエラー	
		論理エラーの見逃し	構文エラーの見逃し
知識	C言語	8.49 **	0.25
	数学	0.36	3.17 *
能力	思考力	0.71	2.06
	理解力	0.71	0.91
	分析力	0.91	1.56
	発見能力	1.11	3.91 *
	表現力	0.63	0.53

\* 危険率1%有意  
\*\* 危険率5%有意

6. おわりに

予備実験の結果からは、C言語の理解度が論理エラーの見逃しに影響を及ぼすことが確かめられた。一方、構文エラーの見逃しに対しては、数学の知識と問題発見能力が有意であるという結果が得られた。しかし、素因の測定が客観性に乏しいので、これらの結果の妥当性を議論するにはさらに詳細な分析が必要である。今後は、誘因を含めて実験を継続する予定である。

[謝辞]

本研究はIPA技術センターの「ソフトウェア信頼性に及ぼす人的要因の調査研究」プロジェクトのワーキング委員会メンバの協力を得て進めたものである。

[参考文献]

- (1) B.Curtis, Eds. "Tutorial: Human Factors in Software Development," IEEE Computer Society Press (1985).
- (2) T.Nakajo and H.Kume, "A Case History Analysis of Software Error Case-Effect Relationships," IEEE Trans. on Software Eng. Vol.17, No.8 (1991).