

## チームによるプログラム開発における デバッグ能力の実験的評価

楠本 真二 高田 義広 松本 健一 菊野 亨 鳥居 宏次

大阪大学

### 1. ま え か き

プログラム開発の現場では、通常、チームを編成してプログラム作成が行われる。そうした開発環境でのデバッグ能力の評価は基本的、かつ、重要な問題の1つである。デバッグ能力とはいかに効率よくバグを見つけて、それを直すかを測る尺度である。

ここでは、チームによるデバッグ能力が「結合テストにおけるデバッグに要する時間」で近似できるという立場をとる。その下で、デバッグに要する時間を事前に予測するための評価尺度について議論する。この評価尺度が求めれば、かなり正確なプログラム開発計画の作成が可能になると期待できる。

### 2. プログラマ性能<sup>[2,3]</sup>

ソフトウェアの生産性を測る1つの尺度にプログラマ性能がある。この尺度はデバッグ能力に対しても十分に適用可能と考えられる(詳細は3.2で述べる)。

#### 2.1 エラー寿命

ここでは、IEEE標準<sup>[6]</sup>で言うフォールトを測定し、それから推定した誤りをエラーと呼ぶ。設計、コーディング、デバッグの一連のソフトウェアの開発過程においてエラー $e_i$  ( $i=1, 2, \dots$ )がソフトウェア中に具体化してから、取り除かれるまでの時間をエラー寿命 $E_i$ と定義する。

#### 2.2 プログラマ性能の評価尺度S

プログラマの性能を評価する単純な方法はエラー数を数えることである。しかし、エラーが持つ影響度に応じた重み付けが必要である。ここではその重みとしてエラー寿命を用いる。

この考えに基づき、プログラマ性能の評価尺度Sを次のように定義する。まず、エラー寿命 $E_i$ は与えられた問題の難しさ $p$ にも依存すると考える。更に、性能の高いプログラマほどSの値が大きくなるように逆数を用いて次式(1)のように定める。

$$S = (\sum E_i / f(p))^{-1} \dots (1)$$

但し、 $f$  : 正規化関数

$p$  : 与えられた問題の難しさ

とする。更に、評価尺度Sの適用に当たっては $f(p) = L^2$  ( $L$ はプログラムの大きさとする)と置く。従って、式(1)は、次式(2)のように定まる。

$$S = (\sum E_i / L^2)^{-1} \dots (2)$$

### 3. 評価法

ここではデバッグ能力を評価する方法として3通りの方法を述べる。

#### 3.1 デバッグ能力

プログラム開発におけるチームのデバッグ能力を測

る尺度として、結合テスト時のチーム全員のデバッグ作業に要した時間を用いる。すなわち、デバッグ作業時間が短いほど、チームとしてのデバッグ能力の評価は高いものと仮定する。

#### 3.2 評価法

次の(A)~(C)に示す3通りの方法E1, E2, E3について比較検討する。この内、E2とE3は評価尺度Sに基づくものである。

表1 実験データ

Team	Member	Program size	Sum of error life spans	Total effort time(min.)	Score
#1	m1	289	1490	2009	56
	m2	263	10608	5488	7
	m3	385	7192	2652	21
	m4	137	6109	3633	3
	m5	95	6679	3523	1
#2	m1	365	7899	3999	17
	m2	278	8510	2706	9
	m3	249	6877	2730	9
	m4	155	1855	3766	12
	m5	107	101	2646	113
#3	m1	221	13329	3730	4
	m2	600	37620	3409	10
	m3	362	27689	4809	5
#4	m1	333	22972	4354	5
	m2	230	4896	3039	11
	m3	364	3970	4220	33
	m4	319	21612	3214	5
#5	m1	393	12035	3681	13
	m2	270	1569	4061	46
	m3	342	11907	4173	10
	m4	240	15147	3886	4
#6	m1	569	22470	3429	14
	m2	375	11789	3243	12
	m3	155	1818	2874	13
#7	m1	387	17634	3768	8
	m2	328	14194	3407	8
	m3	264	4747	2704	15
	m4	126	5092	3627	3
	m5	172	208	2467	142
#8	m1	583	14497	4426	23
	m2	203	2268	3211	18
	m3	233	14775	4699	4
	m4	169	25621	5366	1

Experimental Evaluation of Team Efforts in Program Integration Testing

Shinji KUSUMOTO, Yoshihiro TAKADA, Kenichi MATSUMOTO, Tohru KIKUNO and Koji TORII

Osaka University

(A)評価法E1…文献[4]の実験結果より、プログラマの「経験」と「素質」がプログラマの能力を決定すると考えられる。4.で述べる実験では、各プログラマの「経験」はほぼ同じと考えられるので、「素質」を近似的に測れる成績による評価を行う。

(B)評価法E2…チームのデバッグ能力をプログラマ性能の評価尺度Sを用いて評価する。評価法E2では、各チームをプログラマの集合であるとする。従って、チームのデバッグ能力をチームに属する各プログラマの性能Sの総和として定義する。

(C)評価法E3…チーム全体を一人の仮想的なプログラマと見なしてチームのデバッグ能力を定義する。すなわち、チームを構成する各プログラマを全て、同一人物と見て式(2)を適用する。従って、チームのデバッグ能力を次式(3)のように定める。

$$S = (\sum E_i / \sum L_i^2)^{-1} \dots (3)$$

ここで、 $L_i$ はプログラマiの作成したプログラムの大きさを表す。

#### 4. 実験

あるコンピュータ会社の新人研修において行われたプログラム開発よりデータを収集した。

##### 4.1 データの収集

プログラム開発の主な特徴を以下にまとめる。

- (1) 8つのチームに分かれて、各チームが典型的なファイル処理システムをCOBOLを用いて開発した。
- (2) 開発されたCOBOLプログラムの大きさは約2000行であった。
- (3) 各チームは3人～5人で構成されている。メンバーの決定は、研修時の成績に基づいて、チーム間の能力差が少なくなるように、教官によって決定された。

##### 4.2 実験データ

収集されたデータを表1に示す。表1には、左から順に、チーム番号、各チームに属するプログラマ、プログラムの大きさ(行数)、エラー寿命の合計( $\sum E_i$ )、作業時間の合計(分)、及び、プログラマ性能の値Sを示す。

#### 5. 比較・検討

3通りの評価方法E1, E2, E3に基づく評価値(デバッグ作業時間の予測に用いる値)と、実際に測定したチームのデバッグ作業時間を表2に示す。

次に、表2に基づいて、評価値とチームのデバッグ作業時間との相関を求めた。求めた相関係数の値を表3に示す。表3より、相関係数の値はE1, E2, E3がそれぞれ0.08, -0.66, -0.83となっており、評価法E3, E2, E1の順に相関が高いことが分かる。評価法E3によるチームのデバッグ能力と実測したデバッグ作業時間(D)の比較を行った結果を図1に示す。図1の直線の傾き-0.83が表3中の相関係数となっている。

#### 6. むすび

チームによるプログラム開発におけるデバッグ能力の評価の1つの試みを報告した。デバッグ能力がデバッグに要する作業時間で近似できると仮定して、プログラマ性能に基づく評価法E3でデバッグ作業時間が評価できることを実験的に検証した。

今後は、評価結果をプログラマの生産活動にフィードバックして生産性の向上を目指す試み<sup>[1, 5]</sup>についても検討していく予定である。

表2 各評価法の結果

Team	E1	E2	E3	Team debugging effort time(min.)
#1	1045	88	43	490
#2	986	160	53	460
#3	1212	19	18	2200
#4	941	54	29	2170
#5	1118	73	38	980
#6	1014	39	33	650
#7	1163	176	39	570
#8	1012	46	25	1430

表3 相関係数の比較

	E1	E2	E3
Team debugging effort time	0.08	-0.66	-0.83

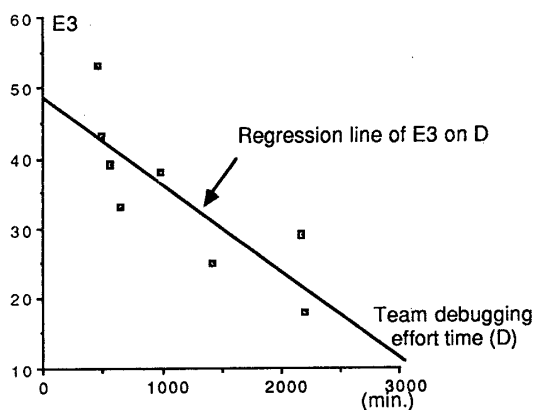


図1 評価値(E3)とデバッグ作業時間の比較

#### 文献

- [1] Basili, V.R. and Rombach, H.D. : "Tailoring the software process and project goals and environments", Proc. of 9th ICSE, pp.345-357 (1987).
- [2] 松本, 井上, 菊野, 鳥居 : "エラー寿命に基づくプログラマ性能の実験的評価 - 大学環境におけるプログラム開発-", 電子情報通信学会論文誌D J71-D, 10, pp.1959-1965 (1988).
- [3] Matsumoto, K., Kusumoto, S., Kikuno, T., and Koji, T. : "Reliable team organization for software development activities", (submitted to FTCS-19).
- [4] Moher, T. and Schneider, G.M. : "Methods for improving controlled experimentation in software engineering", Proc.5th ICSE pp.224-233 (1981).
- [5] Sackman, H., Erikson, W.J. and Grant, E.E. : "Exploratory experimental studies comparing online and offline programming performance", Commun.ACM, 11, 1, pp.3-11 (1968).
- [6] "IEEE Standard Glossary of Software Engineering Terminology", IEEE, 342 E. 47th St., New York, Rep. IEEE-Std-729-1983, (1983).