柿元 健, 亀井 靖高, まつ本 真祐, 門田 暁人, 松本 健一, 楠本 真二, “ バグ予測で信頼性はどれだけ向上するのか？-テスト工数割り当ての観点からの従来研究の評価, ” 松下 誠, 川口 真司(編), ソフトウェア工学の基礎XV 日本ソフトウェア科学会FOSE2008, pp.63-68, November 2008.

バグ予測で信頼性はどれだけ向上するのか？
—テスト工数割り当ての観点からの従来研究の再評価

How Much Does the Bug Detection Improve Reliability?
—Re-evaluation of Previous Studies from the Viewpoint of Test Effort Allocation

柿元 健*　亀井 靖高†　松本 真佑‡　門田 暁人§　松本 健一¶
楠本 真二‖

あらまし　ソフトウェアの信頼性確保のために，数多くの fault-prone モジュール判別手法が提案されている．しかし，従来研究では fault-prone モジュール判別の判別精度のみによる評価が行われ，本来の目的である信頼性確保の効果については評価されていない．本稿では，5 つの従来研究を取り上げ，それらの判別結果に基づいてテスト工数の割り当てを実施した場合に，ソフトウェア全体の信頼性がどの程度向上するのかをシミュレーションによって明らかにする．シミュレーションの結果，判別精度と信頼性は必ずしも一致しないことが分かった．さらに，信頼性向上のためには，適合率よりも再現率を重視し，高い再現率が得られる fault-prone モジュール判別モデルを用いる，あるいは，fault-prone 判定率を適切な値に設定することが重要であることが分かった．

1　はじめに

従来，ソフトウェアの信頼性の確保，および，テスト工数の削減を目的として，ソフトウェア中の各モジュールのバグ（fault）の有無を判別する方法（fault-prone モジュール判別手法）が数多く提案されている [2] [6] [10] [12] [13]．各判別手法は，様々なモジュールデータセットを用いた性能評価実験によって，fault の有無の判別性能（精度）によって評価されている．例えば，亀井ら [5] の研究では，あるレガシーソフトウェアに含まれる fault-prone モジュールの判別性能として，再現率 0.595，適合率 0.282，F1 値 0.384 という評価結果 (5 回の施行の平均値) を得ている．

しかし，fault-prone モジュール判別手法の目的は信頼性の確保，および，テスト工数の削減であるから，これらの観点からの評価が本来必要である．すなわち,「fault あり」と判定されたモジュールにより多くのテスト工数を割り当て,「fault なし」と判定されたモジュールに少ないテスト工数を割り当てた場合に，ソフトウェア全体の信頼性がどの程度になるのか，また，全体のテスト工数がどの程度になるのか，といった評価が必要である．

このような観点から fault-prone モジュール判別の効果を評価するために，我々は，判別結果に基づいたテスト工数の割り当てとその結果得られるソフトウェア信頼性のモデルを提案している [9]．このテスト工数割り当てと信頼性のモデルでは，各モジュールの fault の有無，判別結果，テスト工数割り当てからソフトウェア信頼性を算出する．これまでに，判別性能（F1 値）と信頼性の関係を明らかとするために，著者らが設定した架空の値を用いたシミュレーションを行っている [9]．

本稿では，このテスト工数割り当てと信頼性のモデルを用いたシミュレーションによって，従来研究の fault-prone モジュール判別の事例を比較する．判別性能だけ

*Takeshi Kakimoto, 大阪大学 大学院情報科学研究科
†Yasutaka Kamei, 奈良先端科学技術大学院大学 情報科学研究科
‡Shinsuke Matsumoto, 奈良先端科学技術大学院大学 情報科学研究科
§Akito Monden, 奈良先端科学技術大学院大学 情報科学研究科
¶Ken-ichi Matsumoto, 奈良先端科学技術大学院大学 情報科学研究科
‖Shinji Kusumoto, 大阪大学 大学院情報科学研究科

でしか評価されていない各事例を信頼性によって再評価し比較することで，どの事例がソフトウェアテストにおいて効果があった（信頼性が向上した）のかを明らかにする．また，比較結果より，効果的な fault-prone モジュール判別を行うための方針について検討を行う．

以降，2 章ではテスト工数割り当てと信頼性のモデルについて説明し，3 章で，従来研究の判別事例の比較結果，および効果的な fault-prone モジュール判別を行う方針について述べる．最後に 4 章で本稿のまとめを述べる．

2 テスト工数割り当てと信頼性のモデル

我々は，これまで fault-prone モジュール判別の結果に基づいたテスト工数の割り当てと割り当ての結果得られるソフトウェア信頼性の関係を表したモデルを提案している [9]．このモデルでは，パッケージやサブシステム単位でなく，モジュール（ファイル）単位での予測を対象とするため，fault の有無は 2 値判別されるものとしている [11]．また，判別結果に対し，開発現場では，次のポリシーに基づいてテスト工数の割り当てを行うとしている．

> Fault を含むと判別されたモジュール（fault-prone モジュール）により多くのテスト工数を割り当てる．逆に，fault を含まないと判別されたモジュール (non-fault-prone モジュール) には少しのテスト工数しか割り当てない．
> 具体的には，fault-prone モジュールには，non-fault-prone モジュールの r 倍のテスト工数が割り当てられる．(r は 1 以上の実数)

このモデルでは，ある fault を含むモジュールにおいて発見される fault の割合は式 (1) の指数型モデルで定義する．

$$d(E) = 1 - e^{-ZE} \tag{1}$$

ここで，$d(E)$ は，ある fault-prone モジュールに工数 E を割り当てた時の fault 発見率，Z は，単位工数あたりに発見される fault の割合である．

この 1 モジュールにおける fault 発見率，工数割り当てのポリシー，各モジュールの fault の有無および判別結果から，ソフトウェア全体の信頼性を表す式 (2) のモデルが導出される．(詳細な導出は文献 [9] を参照されたい）ソフトウェア信頼性の指標としては，fault 発見率 (ソフトウェアの潜在 fault 数に対する発見された fault 数の割合) を用いている．

$$\begin{aligned} D(E_F, E_{NF}) = & \left(1 - e^{-\frac{E_{all} \times r}{N_P(r-1)+N_M} Z}\right) \times \frac{N_C}{N_F} \\ & + \left(1 - e^{-\frac{E_{all}}{N_P(r-1)+N_M} Z}\right) \times \frac{N_F - N_C}{N_F} \end{aligned} \tag{2}$$

ここで，
$D(E_F, E_{NF})$：ソフトウェアの fault 発見率，　E_{all}:全テスト工数
N_M：ソフトウェアの全モジュール数
N_P：実際に fault を含むモジュール（fault 含有モジュール）数
N_F：fault-prone と判別されたモジュール（fault-prone モジュール）数
N_C：fault-prone モジュールと判別された fault 含有モジュール（正答モジュール）数
r：Non-fault-prone モジュールに対する fault-prone モジュールに割り当てられるテスト工数の倍率
である．

このテスト工数割り当てと信頼性のモデルでは，指数型モデルに基づいて fault は発見されるモデルを拡張したモデルである．指数型モデルはソフトウェア信頼度成長モデル（SRGM）でも採用されているため，このモデルは指数型 SRGM と同程度の妥当性はあると考えられる．

表 1　再評価した判別事例

文献	実測 / 予測	なし	あり	含有率	再現率	適合率	F1 値
Gyimothy TSE2005 [3]	なし	1624	744	0.420	0.446	0.726	0.552
	あり	226	598				
亀井 IPSJ2007 [5]	なし	1883	17	0.021	0.595	0.272	0.373
	あり	67	25				
Kamei ISESE2006 [4]	なし	108	64	0.461	0.462	0.640	0.537
	あり	31	55				
Khosgoftaar IJSEKE2006 [8]	なし	2316	279	0.191	0.669	0.308	0.422
	あり	1266	564				
馬場 IEICE2008 [1]	なし	4	0	0.750	1.000	0.857	0.923
	あり	4	24				

3　従来研究の再評価

3.1　概要

従来研究の再評価は，判別性能でしか評価されていない各事例を，工数割り当てと信頼性のモデルによるシミュレーションで得られる信頼性で比較することで行う．また，比較した結果から，効果的な fault-prone モジュール判別を行う方針についても検討を行う．

工数割り当てと信頼性のモデル（式 (2)）のパラメータのうち，実測，予測に基づく各モジュール数（N_M, N_F, N_P, N_C）は各判別事例の判別結果を用いる．また，判別事例から入手できないソフトウェアテストとテスト工数の割り当てに関するパラメータについては，総テスト工数 E_{all} は全モジュール数 N_M と同じ値とし，単位工数あたりの fault 発見率 Z は 0.5 とする．そして，non-fault-prone モジュールに対する fault-prone モジュールに割り当てられるテスト工数の倍率（r）を変化させ，fault 発見率 D の値を求める．（E_{all}，Z によって fault 発見率 D の値は変化するが，r に関わらず同じ値だけ増減するためシミュレーションの結果の分析には影響しない．）

3.2　判別事例

再評価する従来研究の fault-prone モジュール判別の事例として 5 件の判別事例を採用した．表 1 に，採用した 5 件の判別事例の文献，実測と予測それぞれの fault の有無ごとのモジュール数，fault を含むモジュールの割合（fault 含有率），適合率，再現率，F1 値を示す（各事例の詳細については紙面の都合上割愛する）．採用した判別事例は，比較的近年に実施された以下の条件を満たす判別事例である．条件に該当する判別事例が複数存在する従来研究においては，1 つの事例のみを採用している．

- 判別手法として，ロジスティック回帰モデル，もしくは，ロジスティック回帰モデルに基づいた手法を採用している．
- 予測，実測，それぞれの fault の有無のモジュール数が文献中で示されている，もしくは，著者らが取得可能である．
- 他の判別事例とデータセットが異なる．

3.3　結果

シミュレーションによる再評価の結果を図 1 に示す．グラフの縦軸はソフトウェア信頼性を示す指標である fault 発見率，横軸は，non-fault-prone モジュールに対する fault-prone モジュールに割り当てられるテスト工数の倍率（r）を示し，各折

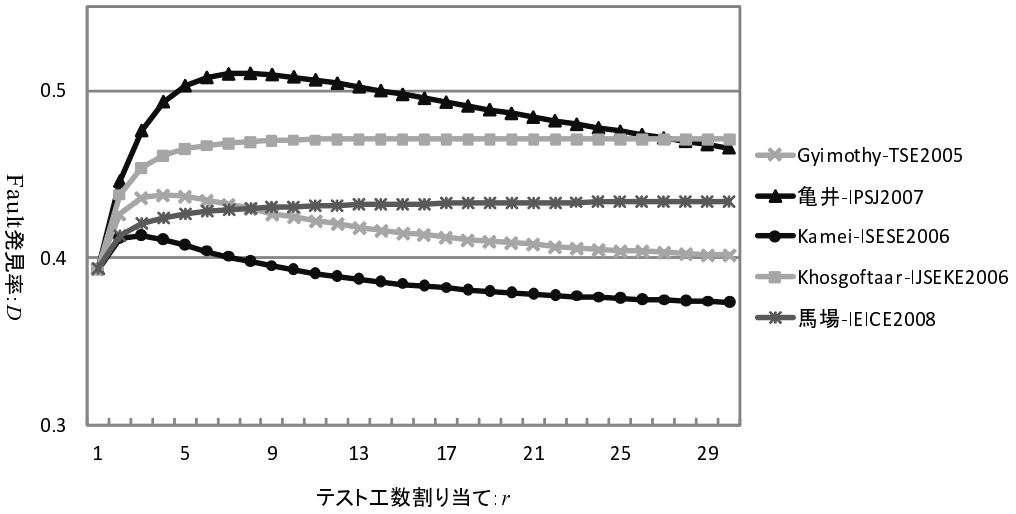

図 1　各判別事例のテスト工数割り当てによる fault 発見率

れ線は従来研究の判別事例の結果を示す．$r = 1$ の時は，fault-prone モジュールと non-fault-prone モジュールに均等にテスト工数を割り当てる，すなわち，fault-prone モジュール判別を行わない場合を表している．本稿のシミュレーションでは，全モジュール数と総テスト工数を同じとしたため，全ての判別事例で同じ値（0.393）となっている．この $r = 1$ の時との差が fault-prone モジュール判別によって向上した信頼性となる．

再評価の結果より，適切にテスト工数を割り当てれば，fault-prone モジュール判別によって信頼性は向上している．しかし，工数の割り当て方によっては fault-prone モジュール判別を行うことで信頼性が低下している場合もある．各事例によって信頼性の向上のしかたには異なっているため，信頼性の最大値，および，テスト工数の割り当てやすさの観点から比較を行った．

信頼性の最大値

再評価の結果について信頼性の最大値に着目すると，判別性能（F1 値）が最も低い事例である [亀井-*IPSJ2007*] で適切な工数を割り当てた場合に最も高い信頼性が得られている．そして，F1 値が 2 番目に低い [*Khosgoftaar-IJSEKE2006*] でも 2 番目に高い（r が 27 以上は最も高い）信頼性が得られている．一方で，F1 値が最も高い事例である [馬場-*IEICE2008*] は 3 番目，もしくは，4 番目の信頼性しか得られていない．

[亀井-*IPSJ2007*]，[*Khosgoftaar-IJSEKE2006*] において高い fault 発見率が得られた要因として，両者の共通した特徴である，fault 含有率が低い，かつ，再現率は比較的高いものの適合率が低いため F1 値が低下していることが挙げられる．fault 含有率が低い場合には，fault-prone モジュールに割り当てるテスト工数の方が non-fault-prone モジュールに割り当てられるテスト工数よりも自由度が高く，r を大きくした際により多くのテスト工数が fault-prone モジュールに割り当てられる．また，再現率が高いと，判別誤りによって non-fault-prone と判定したモジュールに fault が含まれていることが少ない（見逃しが少ない）ため，fault-prone モジュールに多くのテスト工数を割り当てることがより効果的となる．従って，F1 値は低いが，fault 含有率が低く，再現率が高い，[亀井-*IPSJ2007*]，[*Khosgoftaar-IJSEKE2006*] において高い fault 発見率が得られたと考えられる．

テスト工数の割り当てやすさ

再評価の結果についてテスト工数の割り当てやすさに着目すると，[*Gyimothy-TSE2005*]，[亀井-*IPSJ2007*]，[*Kamei-ISESE2006*] は，fault 発見率が最大値となる r 以降は，fault-prone モジュールに割り当てるテスト工数の倍率 r を大きくすることで fault 発見率は低下している．従って，これらの事例では，割り当てるテスト工数が適切な値から少しでも外れると高い fault 発見率は得られないため、テスト工数の割り当ては慎重に行わなければならないといえる．一方，[*Khosgoftaar-IJSEKE2006*]，[馬場-*IEICE2008*] では，fault 発見率が最大値付近となった以降は r を変化させても fault 発見率はほとんど変化しない．従って，これらの事例では，fault-prone モジュールにある程度以上のテスト工数を割り当てることでほぼ最大の fault 発見率が得られるため，テスト工数の割り当ては容易であるといえる．

テスト工数の割り当てが容易と考えられる [*Khosgoftaar-IJSEKE2006*]，[馬場-*IEICE2008*] に共通した特徴として， 再現率が高く，かつ，fault-prone モジュールと判定したモジュールの割合（fault-prone 判定率）が高いことが挙げられる．再現率が高いと見逃しが少ないため，fault-prone モジュールに多くのテスト工数を割り当てても，non-fault-prone モジュールに含まれる見逃した fault による信頼性の低下は少ない．また，fault-prone 判定率も高いため，より再現率が高くなっていると考えられる．従って，再現率が高く，かつ，fault-prone 判定率が高い，[*Khosgoftaar-IJSEKE2006*]，[馬場-*IEICE2008*] において，fault 発見率が最大値付近でほとんど変化しないと考えられる．

3.4 考察

シミュレーションの結果より，データセットが異なる fault-prone モジュール判別の事例の結果を判別精度で比較しても，データセットの fault 含有率が異なると，fault-prone モジュール判別の効果（信頼性）とは一致しないことがわかった．一般的に fault 含有率が低いデータセットでは，高い再現率（少ない見逃し）で高い判別精度を得ることは難しい [7] が，fault 含有率が低い場合でも，再現率が高く fault の見逃しが少なければ，判別精度が低い場合でも高い信頼性が得られている．従って，異なるデータセットで行われた fault-prone モジュール判別の比較には判別精度だけでは不十分であるといえる．

高い信頼性を得るための条件とテスト工数の割り当てが容易となるための条件に共通している点は再現率が高いことである．しかし，割り当てを容易とするために fault-prone 判定率を高くすることで再現率を向上させると，特に fault 含有率が低い場合には，fault を含まない多くのモジュールに対して fault-prone モジュールに割り当てる（r 倍の）テスト工数が割り当てられるため，fault 含有モジュールひとつあたりに割り当てられるテスト工数が減少し，結果として信頼性は低下する．従って，高い信頼性を得ることとテスト工数割り当ての容易さはトレードオフの関係にあり，テスト工数割り当てと信頼性のモデルを利用するなどし，適切な信頼性と割り当ての容易さを選択することが重要である．

具体的には，高い再現率が得られると推測される fault-prone モジュール判別モデルを用いて fault-prone モジュール判別を行う，もしくは，fault-prone 判定率が調整可能な fault-prone モジュール判別モデルにおいて，fault-prone 判定率を適切な値に設定し（推定される）再現率を変化させることで，高信頼性の確保，および，容易なテスト工数の割り当てが実現できると考えられる．また，再現率を向上させることは，信頼性向上，および，工数割り当ての容易さのどちらにもつながるため，再現率を重視して fault-prone モジュール判別を実施するべきであるといえる．

4 まとめ

本稿では，テスト工数割り当てと信頼性のモデルによるシミュレーションによって従来研究で実施された fault-prone モジュール判別事例の再評価を行った．シミュレーションの結果，判別精度と信頼性は必ずしも一致しないことが分かった．さらに，各事例を比較した結果，fault-prone モジュール判別を効果的に実施する（信頼性を向上させる）ためには以下のことが重要であることがわかった．

- 高い再現率が得られると推測される fault-prone モジュール判別モデルを用いる．
- Fault-prone 判定率が調整可能な fault-prone モジュール判別モデルが利用可能な場合は，適切な fault-prone 判定率に設定する．
- 再現率を重視して fault-prone モジュール判別を行う．

本稿では，データセットが異なるロジスティック回帰モデル，および，ロジスティック回帰モデルに基づいた手法の判別事例について信頼性によって再評価を行った．他の判別手法の事例や，テスト工数削減についても再評価を行うこと，および，実プロジェクトにおいて工数割り当てと信頼性のモデルを用いてテスト工数を割り当て fault-prone モジュール判別の効果を確認することが今後の課題である．

謝辞 本研究の一部は，文部科学省「次世代 IT 基盤構築のための研究開発」の委託に基づいて行われた．

参考文献

[1] 馬場慎太郎, 吉田則裕, 楠本真二, 井上克郎, “Fault-Prone モジュール予測へのコードクローン情報の適用,” 電子情報通信学会論文誌 D,（採録決定）.

[2] A.R. Gray and S.G. MacDonell, “Software metrics data analysis–exploring the relative performance of some commonly used modeling techniques,” Empirical Softw. Eng., Vol.4, No.4, pp.297–316, 1999.

[3] T. Gyimothy, R. Ferenc, and I. Siket, “Empirical Validation of Object-Oriented Metrics on Open Source Software for Fault Prediction,” IEEE Trans. on Softw. Eng., Vol.31, No.10, pp.897-910, 2005.

[4] Y. Kamei, A. Monden and K. Matsumoto, “Empirical Evaluation of Svm-Based Software Reliability Model,” Proc. of the 5th ACM-IEEE Int’l. Symposium on Empirical Softw. Eng. (ISESE2006), Vol.2, pp.39–41, Brazil, 2006.

[5] 亀井靖高, 松本真佑, 柿元健, 門田暁人, 松本健一, “Fault-Prone モジュール判別におけるサンプリング法適用の効果,” 情報処理学会論文誌, Vol.48, No.8, pp.2651-2662, 2007.

[6] T.M. Khoshgoftaar, and E.B Allen, “Modeling software quality with classification trees,” Recent Advances in Reliability and Quality Eng., World Scientific, pp.247–270, Singapore, 1999.

[7] T.M. Khoshgoftaar, K. Gao, and R.M. Szabo, “An Application of Zero-inflated Poisson Regression for Software Fault Prediction,” Proc. of 12th Int’l Symposium on Softw.Reliability Eng. (ISSRE’01), pp.66–73, China ,2001.

[8] T.M. Khoshgoftaar, V. Joshi, and N. Seliya, “Detecting Noisy Instances with the Ensemble Filter: A Study in Software Quality Estimation,” Int’l Journal of Softw. Eng. and Knowledge Eng. (IJSEKE), Vol.16, No.1, pp.53.76, 2006.

[9] 柿元健, 門田暁人, 亀井靖高, 松本真佑, 松本健一, “Fault-Prone モジュール判別における F1 値とソフトウェア信頼性の関係,” ソフトウェア工学の基礎 XIV, 日本ソフトウェア科学会 FOSE2007, pp.75-83, 2007.

[10] P.L. Li, J. Herbsleb, M. Shaw and B. Robinson, “Experiences and results from initiating field defect prediction and product test prioritization efforts at ABB Inc,” Proc. 28th Int’l Conf. on Softw. Eng. (ICSE’06), pp.413–422, China, 2006.

[11] R. Moser, W. Pedrycz and G. Succi, “A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction,” Proc. 30th Int’l Conf. on Softw. Eng. (ICSE’08), pp.181–190 ,2008.

[12] J.C. Munson and T.M. Khoshgoftaar, “The detection of fault-prone programs,” IEEE Trans. Softw. Eng., Vol.18, No.5, pp.423–433, Germany, 1992.

[13] N. Ohlsson and H. Alberg, “Predicting fault-prone software modules in telephone switches,” IEEE Trans. Softw. Eng., Vol.22, No.12, pp.886–894, 1996.