Vol.2010-SE-168 No.6
Vol.2010-EMB-17 No.6
2010/6/1

プロセスおよびプロダクトメトリクスを用いたFault-Proneクラス予測の適用事例

纐纈伸子[†1] 川村真弥[†1]
野村准一[†1] 野中誠[†2]

筆者らの所属する組織では，従来のプロセスメトリクスを中心とした品質保証活動に加えて，プロダクトメトリクスを活用した精度の高い品質保証の導入を検討している．fault-prone (FP) モジュール予測技法は，このニーズに対する有効な手段と考えられる．本稿では，筆者らの組織で実際に開発された製品のデータを用いて，機能テスト工程以降に摘出される欠陥を対象としたFPクラスの予測モデルを構築した試みについて述べる．プロダクトメトリクスのみを用いてFPクラス予測モデルを構築した場合と，プロセスメトリクスを組み合わせて予測モデルを構築した場合を比較した結果，後者の方が高い予測精度が得られることがわかった．また，実際のテスト工程でFP予測を用いることで品質を早期に安定させる方法について考察を行ったのであわせて報告する．

A Case Study on Predicting Fault-Prone Classes with Process and Product Metrics

Nobuko Koketsu,[†1] Shinya Kawamura,[†1]
Jyunichi Nomura[†1] and Makoto Nonaka[†2]

The general purpose of our study is to introduce a more precise and effective quality assurance practice into our software development organization that has been applying quality assurance activities mainly focusing on process metrics. The fault-prone (FP) module prediction technique is one of promising techniques that meets our goal. In this paper, we explain our attempt to build several FP class prediction models to predict defects that would be detected in a functional testing phase by using our actual product development data. We found that the models built by using both process and product metrics showed better performance than the ones built by using only product metrics. We also discuss practical consideration how to apply the FP technique to a testing phase in practice to stabilize software quality earlier than before.

1. はじめに

大規模・複雑化するソフトウェアにおいて，その信頼性に対する顧客や社会からの要求はますます高まっている．一方で，市場が要求する機能をタイムリーかつ妥当なコストの範囲で提供することが求められており，ソフトウェアの信頼性を高いレベルで効率的に確保する技術が求められている．このような技術ニーズに対して，fault-prone (FP) モジュール予測技法は，有効な解決策の1つとして期待されている．

FP予測技法は，ソフトウェアを構成するモジュールのうち，欠陥を含んでいる可能性の高いモジュールを特定する技法である．FP予測を高い精度で実現できれば，予測結果に基づいたテスト戦略などの立案と実施ができ，品質保証コストの削減が期待できる．

FP予測に関する研究は，これまでに数多く行われてきている．初期の研究は1980年代に始まり[1)]，1990年代からはオブジェクト指向ソフトウェアを対象にCK (Chidamber & Kemerer) メトリクス[2)]などを説明変数とした予測技法の研究[3)4)5)]へと発展した．2000年以降はオープンソースソフトウェア (OSS) を対象とした研究[6)7)8)9)]や，それまで主流だった多重ロジスティック回帰モデル以外の多様な予測モデルを適用した研究[6)8)10)]などが示されており，FP予測研究は広がりを見せている．

しかし，これらのFP予測研究で分析対象とした組織またはデータの文脈は，筆者らの所属組織における状況に適合するとは言い難い．筆者らの組織では，1970年代からの継続的なソフトウェア品質改善活動[11)]の蓄積があり，品質会計制度[12)]と呼ばれる手法を中心とした品質保証を長年にわたって実施している．また，2002年にはCMMIレベル5を達成するなど，ソフトウェアライフサイクルプロセスの成熟度は比較的高い．

この状況を踏まえてFP予測技法に関する先行研究を検討すると，筆者らの組織において適用する上での課題が指摘できる．たとえばShenらの研究[1)]は，開発言語，利用メトリクス，ならびにテストおよび出荷後の欠陥密度が筆者らの組織とは大きく異なっており，提示された予測モデルの適合性は低いと考えられる．Ohlssonらの研究[13)]は，設計文書からの抽出情報を利用しているが，すでに確立された設計技法を用いている組織において，FP

†1 NEC
NEC Corporation
†2 東洋大学
Toyo University

Vol.2010-SE-168 No.6
Vol.2010-EMB-17 No.6
2010/6/1

予測の目的で設計技法を変更するのは困難である．Basili らの研究[3] は，学生が開発したプロジェクトのデータを用いており，筆者らの所属組織とはプロセス成熟度の面で文脈が大きく異なる．さらに，OSS を対象とした研究については，その成果を，品質保証体制が確立された組織における商用ソフトウェア開発に適用できるかどうかは議論の余地がある．Fenton らはプロセスの質に関する定性的データを組み合わせた残存欠陥数の予測モデルを示しているが[14]，筆者らの組織が測定している定量的データとは性質が異なる．Zhou らは CK メトリクスを用いた FP 予測研究を横断的に調査した結果，クラス規模やクラス結合度に関するメトリクスが概ね一貫して統計的有意であったと述べているが[5]，これまで述べた通り，文脈の違いを十分に考慮する必要がある．

これらの課題を踏まえ，筆者らは，品質保証体制が確立された組織において適用可能な独自の FP 予測モデルを，先行研究の知見を参考にした上で構築する取り組みを行っている．本稿では，CK メトリクスと当該組織において従来から収集しているプロダクトメトリクスを用いて，機能テスト以降に摘出した欠陥とその摘出元クラスとの関連について分析した結果を報告する．また，これらのプロダクトメトリクスを用いて FP クラス予測モデルを構築した場合と，プロセスメトリクスを組み合わせて FP クラス予測モデルを構築した場合の比較を行う．さらに，筆者らの組織のように品質保証体制が確立された組織において，FP 予測を用いることでテスト工程での品質を早期に安定させる方法について考察する．

2. 開発製品およびプロセスの概要

ここでは，筆者らの組織が開発する製品と，開発プロセスおよび品質保証プロセスの概要を述べる．これらを示すことで，本稿における FP 予測の文脈を明確にする．

2.1 開 発 製 品

筆者らの組織では，オペレーティングシステムやシステムソフトウェアの開発・保守を行っている．これは，ある特定の顧客向けのシステムではなく，汎用的な製品である．そのため，出荷後に品質問題が発生した場合の影響範囲が大きく，高いレベルでの品質確保が特に求められる．

製品の開発にあたっては，既存製品の機能強化をバージョンアップという形で繰り返し実施することが多い．完全に新規開発される製品は少ない．

2.2 開発プロセス

開発プロセスは，製品開発プロセスと品質改善プロセスの 2 つを定義している．製品開発プロセスは，**図 1** に示したウォーターフォール型モデルを基本としている．これは，基本設計からコーディングまでの 4 工程からなる上工程と，単体テストからシステムテストまでの 3 工程からなるテスト工程で構成される．それぞれの工程について，成果物，実施作業，および測定項目を定義している．測定項目のうち，代表的なプロセスメトリクスを**表 1** に示す．品質改善プロセスは，製品開発プロセスの全フェーズにわたって体系的に定義しており，欠陥の未然予防や再発防止に関する各種の活動が定義されている．これらの開発プロセスは組織的に標準化されており，開発部門が製品を開発する際に一律に適用している．

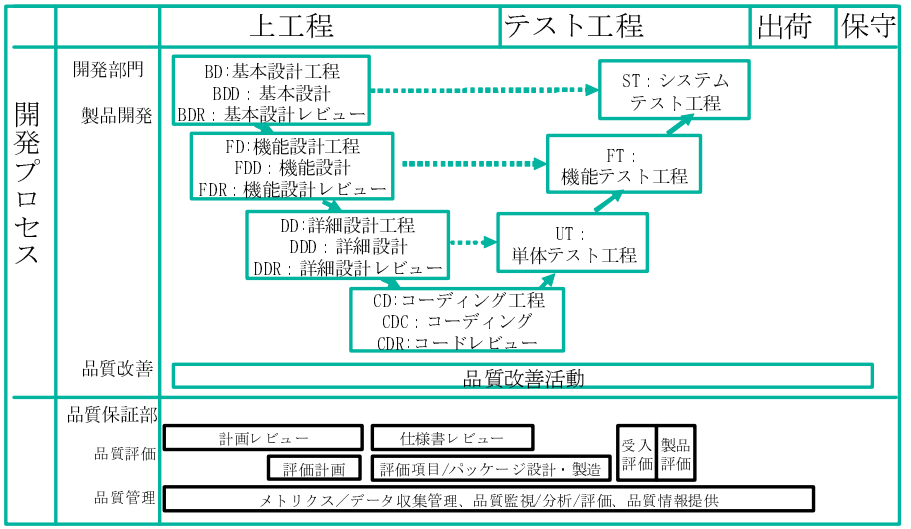


図 1 開発プロセスと品質保証活動

2.3 品質保証プロセス

品質保証プロセスは，開発部門によって作成された成果物と実施された作業の両方について，その品質を保証する活動である．このプロセスは品質保証部門が第三者の立場で実施している．品質保証プロセスは，品質評価プロセスと品質管理プロセスの 2 つを定義している．品質評価プロセスでは，開発計画や仕様書の妥当性確認など，第三者の立場から評価を行う．一方，品質管理プロセスでは，開発部門の全工程にわたって測定されたデータ（表 1 参照）を収集し，これらを元に品質状況と進捗状況を客観的かつ多角的に判断し，問題点の指摘を行う．また，複数の製品開発から収集されたデータを横断的な視点で品質分析を行い，品質向上施策の指示などを行っている．

Vol.2010-SE-168 No.6
Vol.2010-EMB-17 No.6
2010/6/1

表 1 代表的なプロセスメトリクス

測定対称の工程	メトリクス
開発プロセス共通	開発日遅れ日数，完了日遅れ日数，欠陥摘出数/KL，欠陥摘出率
上工程のみ	作業進捗率，作業工数/KL，レビュー工数/KL，欠陥摘出数/レビュー工数
テスト工程のみ	テスト項目消化率，テスト項目数/KL，テスト工数/KL

3. 分析対象データの概要

3.1 分析対象の製品

今回は，新規に開発した汎用的なシステムソフトウェア製品を分析対象として選定した．バージョンアップ開発ではなく新規開発を選定した理由は，流用元ソフトウェアの性質がFP 予測に与える影響を排除したいと考えたためである．

製品全体の開発規模は約 105KL[⋆1]で，C および C++を開発言語として使用している．今回の分析対象としたのは，C++で開発された約 80KL のソフトウェアであり，215 個のクラスで構成されている．開発製品は機能別に 7 つのサブシステムに分かれており，それぞれのサブシステムは開発を担当する会社や所属が異なっている．

3.2 クラス単位のプロダクトメトリクス

表 1 のメトリクスに加えて，FP 予測の目的で収集したクラス単位のプロダクトメトリクスを**表 2** に示す．このうち，CK メトリクスはテクマトリクス社の Understand[15] を用いて測定した．そのほかのメトリクスは，2.3 節で述べた品質管理プロセスで継続的に収集している項目であり，社内で作成したツールを用いて測定している．

4. 予 備 分 析

FP クラス予測を行うに先立ち，収集したデータの予備分析を実施した．予備分析の視点は，プロダクトメトリクスの正規化，データの層別，および予測対象の欠陥と各種メトリクスとの関連性分析の 3 点である．

4.1 プロダクトメトリクスの正規化

表 2 に示したプロダクトメトリクスには，クラスの規模すなわち有効行数またはメソッド数と相関が高いものがある．相関が高いメトリクスの組について，予測モデルの説明変数と

⋆1 KLOC (Kilo Lines Of Code) の略であり，コメント行と空白行を除いた有効行数で測定している．

表 2 クラス単位のプロダクトメトリクス

種別	メトリクス	概要
規模	有効行数	総行数からコメント行数と空白行数を除いた値（クラス単位で集計）.
	メソッド数	クラスに含まれるメソッド数.
複雑度	サイクロマチック数	分岐命令による経路複雑度を表す数値（クラス単位で集計）.
	分岐条件数	分岐命令の条件式の数（クラス単位で集計）.
	最大ネスティング数	各メソッドの最大ネスティング数のクラス平均値.
CK[2]	WMC	メソッドの数を，それぞれの重みを考慮した上で合計した値.
	DIT	継承ツリーにおけるルートクラスまでの階層数（上位クラスの数）.
	NOC	直接のサブクラスの数.
	CBO	メソッドやインスタンス変数を参照・実行している他のクラスの数.
	RFC	オブジェクトが受け取るメッセージに応答して実行されるメソッド数の合計.
	LCOM	共通の属性を操作するメソッドの数．凝集性の欠如度合いを表す.

備考：CK メトリクスは，WMC (Weighted Methods per a Class), DIT (Depth of Inheritance Tree), NOC (Number Of Children), CBO (Coupling Between Object Classes), RFC (Response For a Class), および LCOM (Lack of Cohesion Of Methods) を意味する.

して両方とも使用すると，多重共線性の問題が生じる場合がある．この場合の対処は一方の説明変数を除外するのが一般的であるが，安易に除外すると，メトリクスの本質的な意味を見落としてしまう可能性がある．たとえば，規模と複雑度の相関が高い場合に一方の説明変数を安易に除外すると，欠陥が作り込まれる要因が規模にあるのか，複雑性にあるのかが判断できなくなる．本質的な意味の部分で説明変数として寄与しないメトリクスであれば，より汎用的なメトリクスである規模メトリクスを主体的に用いる方が望ましいと考えられる．このように，規模と相関が高いメトリクスを用いる場合には注意が必要である．

有効行数に対する他のメトリクスとの相関関係を分析した結果，サイクロマチック数，分岐条件数，および CBO の 3 つのメトリクスの相関が高いことが判明した．それぞれのメトリクスについて，正規化前後における有効行数とメトリクス同士の相関係数を**表 3** に示す．有効行数で正規化したことにより，有効行数と各メトリクスとの相関が低くなるだけでなく，メトリクス同士の相関も低くなった．

同様の分析をメソッド数に対して行った結果，有効行数と RFC の相関が高いことが示された．これらのメトリクスの正規化前後におけるメソッド数との相関係数を**表 4** に示す．メソッド数で正規化したことによりメトリクス同士の相関が低くなった．

以上の結果から，サイクロマチック数，分岐条件数，および CBO を FP クラス予測モデルの説明変数として用いる場合には，有効行数で正規化した後の数値を用いる．同様に，

Vol.2010-SE-168 No.6
Vol.2010-EMB-17 No.6
2010/6/1

表 3 有効行数と相関の高いメトリクスの正規化前後における相関係数

メトリクス		有効行数	サイクロマチック数	分岐条件数	CBO
正規化前	サイクロマチック数	0.96	−	0.96	0.64
	分岐条件数	0.91		−	0.60
	CBO	0.71			−
有効行数による	サイクロマチック数	−0.28	−	0.19	0.24
正規化後	分岐条件数	0.29		−	−0.18
	CBO	0.11			−

備考：正規化後の数値は，表頭のメトリクスも正規化した上での相関係数を示している．

表 4 メソッド数と相関の高いメトリクスの正規化前後における相関係数

メトリクス		メソッド数	有効行数	RFC
正規化前	有効行数	0.84	−	0.63
	RFC	0.84		−
メソッド数による	有効行数	0.04	−	−0.03
正規化後	RFC	−0.27		−

備考：正規化後の数値は，表頭のメトリクスも正規化した上での相関係数を示している．

RFC と有効行数を FP クラス予測モデルの説明変数として用いる場合には，メソッド数で正規化した後の数値を用いる．

4.2 サブシステムの層別

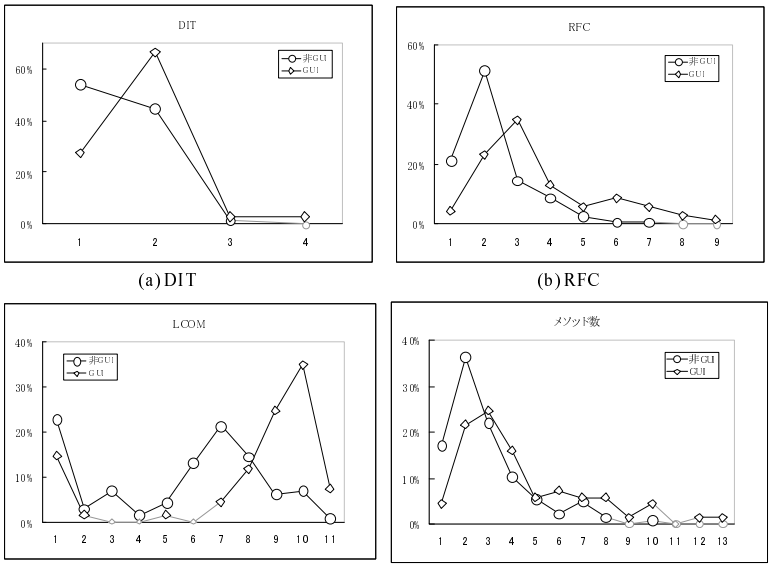
次に，対象データの質的な違いに着目して層別した場合に，メトリクスの値に特徴的な差異が生じるかどうかを分析した．いくつかの質的変数に着目して分析を行った結果，GUI と非 GUI によってサブシステムを層別した場合に顕著な差異が見られた．したがって，この層別に基づいて FP クラス予測を行う．以下に，その根拠となる分析データを示す．

図 2 は，(a)DIT，(b)RFC，(c)LCOM，および (d) メソッド数について，それぞれの測定値を 4〜18 個の階級に分割し，各階級に属するクラス数が全体のクラス数に占める割合（以下，クラス出現率と呼ぶ）を GUI と非 GUI に分けて表した図である．図 2 に示した各グラフの分布形状から，GUI に属するサブシステムのクラスは，DIT，RFC，LCOM，およびメソッド数の分布が非 GUI に比べて右側に位置していることがわかる．このように，GUI と非 GUI の違いによってクラス出現率が異なるメトリクスが複数あるため，この層別を採用するのが望ましいと判断した．

4.3 メトリクスと機能テスト欠陥クラスとの関連

各メトリクスと，機能テスト工程以降に欠陥が摘出されたクラス（以下，機能テスト欠陥クラスと呼ぶ）との関連について分析した．**図 3** に，表 2 に示したメトリクスのうち，機能テスト欠陥クラスとの関連性が高かった有効行数/メソッド数の対数をとった値と分岐条件数/有効行数について，ならびに複雑度を表す一般的なメトリクスであるサイクロマチック数についての分析結果を示す．図 3 では，クラス全体を各メトリクスの値によって 8〜14 個の階級に分割し，各階級に属するクラス数を 100 とした場合の機能テスト欠陥クラスが占める比率（以下，機能テスト欠陥クラス比率）を網掛け部分で表している．また，図 3 の折れ線は，全体のクラス数を 100 とした場合の，それぞれの層に属するクラス数の比率を表している．

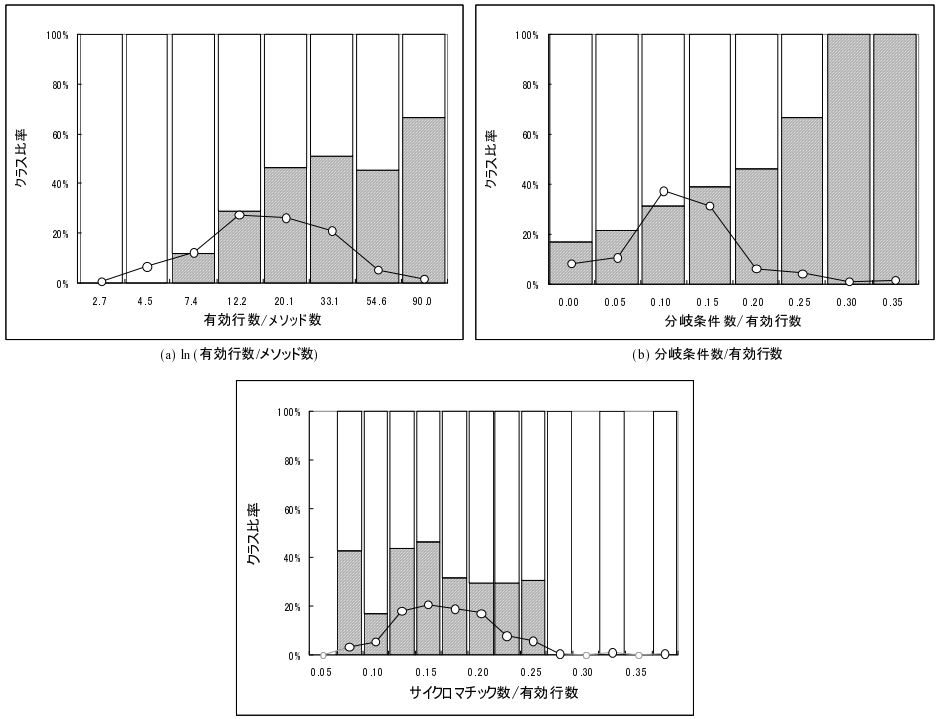
図 3(a) および (b) をみると，有効行数/メソッド数と分岐条件数/有効行数は数値が高くなるほど機能テスト欠陥クラス比率が高くなっていることがわかる．一方，複雑度を表す代表的なメトリクスであるサイクロマチック数を有効行数で正規化した値は，機能テスト欠陥クラス比率との関連が見られなかった（図 3(c) 参照）．このように，表 2 に示したプロダクトメトリクスのうち有効行数/メソッド数と分岐条件数/有効行数は，機能テスト欠陥クラ



(a) DIT　(b) RFC　(c) LCOM　(d) メソッド数

図 2 GUI と非 GUI でのメトリクス別のクラス出現率

Vol.2010-SE-168 No.6
Vol.2010-EMB-17 No.6
2010/6/1



(a) ln (有効行数/メソッド数)　(b) 分岐条件数/有効行数

(c) サイクロマチック数/有効行数

図 3　機能テスト欠陥クラス比率

スとある程度の関連があることがわかった.

5. Fault-prone クラスの予測

予備分析の結果を踏まえて FP クラス予測モデルの構築と評価を行った．機能テスト欠陥が観測されたクラスに 1（欠陥有り）を，観測されなかったクラスに 0（欠陥無し）を割り当て，目的変数のデータとした．FP クラス予測に用いる標本データは，C++で開発されたクラス全体（215 個），非 GUI クラス（146 個），GUI クラス（69 個）の 3 種類とした.

5.1 FP 予測の評価指標

はじめに，評価手順と評価指標を整理しておく．予測モデルは leave-one-out 法を用いて評価した．評価指標は，文献 8) を参考に**表 5** に示した 5 つの指標を用いた．このうち F 値は，FP 予測の評価指標としてよく利用される指標であり，一般にトレードオフの関係にある再現率と適合率を統合して評価できる．F 値は，(1) 式で与えられる[8)].

$$\text{F 値} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{1}$$

表 5　FP クラス予測の評価指標

評価指標	定義
正確度 (accuracy)	全モジュールのうち，FP と判別して実際に欠陥があったモジュール数と，FP でないと判別して実際に欠陥がなかったモジュール数の合計の割合.
再現率 (recall)	実際に欠陥があったモジュールのうち，FP と正しく判別できた割合.
適合率 (precision)	FP と判別したモジュールのうち，実際に欠陥があったモジュールの割合.
特異度 (specificity)	FP でないと判別したモジュールのうち，実際に欠陥がなかったモジュールの割合.
F 値	再現率と適合率の調和平均．値が大きいほど，判別の精度が高いことを表す.

5.2 プロダクトメトリクスによる FP クラス予測

まず，表 2 に示したプロダクトメトリクスのみを用いて，FP クラス予測モデルの構築を試みた．予測には，(2) 式で示した多重ロジスティック回帰モデル

$$P = \frac{1}{1 + e^{-(C_0 + C_1 X_1 + \ldots + C_n X_n)}} \tag{2}$$

を使用した．ここで，P は機能テスト欠陥クラスと判定された確率，X_i は説明変数として選択されたメトリクス，C_i は標準化偏回帰係数を表す．説明変数の選択にはステップワイズ法を使用した.

ロジスティック回帰分析により選択された説明変数を**表 6** に示す．表中の対数変換が有となっているメトリクスは，データの分布が右に歪んでいたために対数変換を行って正規分布に近づける処理を施したことを表している．標準化偏回帰係数を見ると，選択されたメトリクスの中では規模を表すメトリクスがもっとも影響度が大きいことがわかる．GUI については，規模以外のメトリクスは選択されなかった.

構築された予測モデルの評価結果を**表 7** に示す．F 値に着目すると，GUI がもっとも高い値だが，いずれのモデルも必ずしも高い精度とはいえない.

5.3 プロセスメトリクスを組み合わせた FP クラス予測

次に，プロダクトメトリクスに加えて，表 1 に示したプロセスメトリクスを組み合わせて

Vol.2010-SE-168 No.6
Vol.2010-EMB-17 No.6
2010/6/1

表 6 FP クラス予測モデル（プロダクトメトリクスのみを使用）

標本データ	選択された説明変数	対数変換	標準化偏回帰係数	p 値
全体	LCOM	無	0.33	0.0164
	メソッド数	有	0.32	0.0201
	有効行数/メソッド数	有	0.93	0.0000
	サイクロマチック数/有効行数	有	0.37	0.0260
非 GUI	LCOM	無	0.51	0.0004
	有効行数/メソッド数	有	0.81	0.0000
GUI	メソッド数	有	1.31	0.0000

備考：定数項は非掲載.

表 7 FP クラス予測モデルの評価結果（プロダクトメトリクスのみを使用）

参照記号	標本データ		正確度	再現率	適合率	特異度	F 値
A1	全体	(n = 215)	68.4%	39.5%	57.7%	71.8%	0.469
A2	非 GUI	(n = 146)	68.5%	75.4%	55.9%	72.3%	0.452
A3	GUI	(n = 69)	75.4%	57.7%	71.4%	77.1%	0.638

FP クラス予測モデルを構築した．ロジスティック回帰分析により選択された説明変数を**表 8** に示す．表 6 に比べて，多くのプロセスメトリクスが選択されていることが分かる．ここで，プロダクトメトリクスのデータはクラス単位で与えられるのに対して，分析に用いたプロセスメトリクスのデータはサブシステム単位で与えられていることに注意が必要である．

係数が負の説明変数は，数値が大きいほど機能テスト欠陥クラスである確率が低くなることを表している．たとえば，非 GUI で UT 工数/KL の係数が −1.10 だが，これは，単体テスト工程での工数密度が高いほど機能テスト工程以降に欠陥が摘出される確率が低いことを表しており，意味のある係数と考えられる．ただし，全体と非 GUI では，UT 工数/KL の係数の正負が逆転しており，解釈にあたっては注意が必要である．

構築された予測モデルの評価結果を**表 9** に示す．表 9 の B11 と B12 の評価結果は，予測モデル構築時にすべてのデータを使用し，FP クラス予測の際の評価データを非 GUI に限定した場合（B11）と GUI に限定した場合（B12）を表している．

5.4 予測モデルの比較

ここでは，5.2 節と 5.3 節で示した評価結果を比較する．プロセスメトリクスを用いたか否かにかかわらず，GUI と非 GUI に分けて FP クラス予測モデルを構築した場合の方が，全体のデータを用いた場合に比較して評価指標の値が概ね高かった（A1 対 A2 および A3, B1 対 B2 および B3）．したがって，GUI と非 GUI を層別して予測モデルを構築したほうが有効と思われる．

表 8 FP クラス予測モデル（プロセスメトリクスとプロダクトメトリクスを使用）

標本データ	選択された説明変数	対数変換	標準化偏回帰係数	p 値
全体	GUI	無	−1.59	0.0003
	DIT	無	−0.50	0.0110
	RFC/メソッド数	有	0.54	0.0233
	メソッド数	有	0.97	0.0000
	有効行数/メソッド数	有	0.76	0.0000
	上工程欠陥摘出数/KL	有	3.26	0.0000
	上工程レビュー工数/KL	有	−3.07	0.0000
	UT 欠陥数/KL	有	−2.70	0.0001
	UT 工数/KL	有	2.54	0.0000
非 GUI	DIT	無	−0.86	0.0013
	RFC/メソッド数	有	0.86	0.0049
	CBO/有効行数	有	0.54	0.0511
	メソッド数	有	1.09	0.0002
	有効行数/メソッド数	有	1.13	0.0000
	上工程欠陥摘出数/KL	有	−0.97	0.0000
	UT 欠陥数/KL	有	−0.95	0.0001
	UT 工数/KL	有	−1.10	0.0000
GUI	メソッド数	有	1.31	0.0000

備考：定数項は非掲載.

表 9 FP クラス予測モデルの評価結果（プロセスメトリクスとプロダクトメトリクスを使用）

参照記号	標本データ		正確度	再現率	適合率	特異度	F 値
B1	全体	(n = 215)	75.3%	56.6%	68.3%	78.3%	0.619
B11	うち非 GUI	(n = 146)	78.1%	64.0%	69.6%	82.0%	0.667
B12	うち GUI	(n = 69)	69.6%	42.3%	64.7%	71.2%	0.512
B2	非 GUI	(n = 146)	80.8%	70.0%	72.9%	84.7%	0.714
B3	GUI	(n = 69)	75.4%	57.7%	71.4%	77.1%	0.638

全体と非 GUI に関して，プロダクトメトリクスのみを使用した予測モデルの評価結果と，プロセスメトリクスを組み合わせた予測モデルの評価結果を比較すると，プロセスメトリクスを使用したモデルの方が非 GUI の再現率を除いた全ての指標で評価結果が高かった（A1 対 B1，A2 対 B2）．一方，GUI ではメソッド数以外の変数が選択されず，プロセスメトリクスが選択されなかった．結果として同一の予測モデルが得られたため，評価結果も同一に

Vol.2010-SE-168 No.6
Vol.2010-EMB-17 No.6
2010/6/1

なっている（A3 対 B3）．全体および非 GUI のモデルについては，プロセスメトリクスを組み合わせて予測モデルを構築した方が良い結果となった．

プロダクトメトリクスのみを使用した予測モデルの説明変数（表 6）と，プロセスメトリクスを組み合わせた予測モデルの説明変数（表 8）を比較すると，後者の全体を標本データとしたものでは，GUI という層別化のための質的変数が有意な説明変数として選択された．また，プロダクトメトリクスから選択された説明変数の種類が異なっていた．ただし，メソッド数はプロセスメトリクスがない場合の非 GUI を除き全てのモデルで選択されている．また 4.3 節で機能テスト工程以降に修正された欠陥と関連があると思われた分岐条件数/有効行数などのメトリクスがモデルの説明変数に選択されていない．このように，プロダクトメトリクスの中で有効な説明変数がどれなのかは，今回の分析の範囲では明確化できていない．今回の分析では 1 製品のデータのみを用いたため，製品数を増やして分析するなどの取り組みが必要である．

このように，今回の分析ではいくつかの課題も明らかになった．とはいえ，表 9 の B2 に示した非 GUI における評価結果を見ると，比較的高い精度で FP 予測を行えていることがわかる．Kaur らの FP 予測では F 値が 0.79 であり[8]，亀井らの FP 予測では 0.449 であった[6] ことを考えても，筆者らが得た 0.714 という F 値は従来研究に比べて高い精度であると考えられる．このことから，CK メトリクス，従来から収集していたプロダクトメトリクスおよびプロセスメトリクスは，FP 予測にそれぞれ有効であると思われる．

6. 実際の開発での利用方法

ここでは，FP クラス予測モデルを実際に利用する場合の活用方法について考察する．筆者らの組織では，2 章で述べたような品質保証体系が確立している．品質管理の中心となるのは収集したメトリクスを用いた管理である．管理は原則としてサブシステム単位で収集されたデータをもとに実施される．今回分析を実施した製品は 7 つのサブシステムに分割されている．そのため，これまで数値で判断できていたのはサブシステム全体の品質までであった．これよりも細かい粒度での品質は，従来は各開発者が個別に実施する品質分析に委ねていた．テスト工程では品質会計制度をもとに残存する欠陥数を管理していたが，ここでも同様に，管理はサブシステム単位であった．

一方，FP クラス予測モデルではソースコードから収集するプロダクトメクトメトリクスを用いるため，クラス単位での FP 予測が可能となる．今回の製品に関しては，従来は 7 つのサブシステム単位で管理していたものが，215 個のクラスの単位で品質分析を行えることになる．FP クラス予測をコーディング工程完了時あるいは単体テスト工程完了時に実施することで，機能テスト工程以降の品質を早期に安定させることができると思われる．

一例として，非 GUI のサブシステムで機能テスト工程以降に欠陥が摘出された/されなかったの結果に対し，プロセスメトリクスを組み合わせた非 GUI の FP クラス予測モデルでの予測結果の関係を**表 10** に示す．

表 10 FP クラス予測と機能テスト工程以降の欠陥発生状況

実績	FP クラス予測結果		
	欠陥無しと予測	欠陥有りと予測	合計
機能テスト工程以降に欠陥無し	83	13	96
機能テスト工程以降に欠陥有り	15	35	50
合計	98	48	146

表 10 に示すとおり，146 あるクラスのうち，機能テスト工程以降に欠陥の修正が発生したクラスは 50 個であった．一方，欠陥が発生する可能性があると予測されたクラスは 48 個であり，この 48 個のクラスで実際に欠陥による修正が発生したクラスが 35 個であった．テストをどのクラスにも均等に実施し，テスト項目に対して欠陥が発生する確率が均一であったと仮定した場合，35 個のクラスから欠陥を摘出するには機能テスト以降のテスト項目を 70% (35/50) 実施することが必要となる．しかし FP クラス予測で欠陥が発生する可能性が高いと予測されたクラスからテストを実施すれば，全体の 33% (48/146) のテスト項目を消化した時点で 35 個のクラスから欠陥を摘出することが可能となり，テストの早い段階から品質を向上させることができる．

7. おわりに

今回は，機能テスト工程以降の欠陥の発生状況を製品の品質ととらえ，品質とツールで収集しているプロダクトデータの関連を分析した．その結果，品質を示す指標として有効であると思われるプロダクトメトリクスを特定することができた．また，プロダクトメトリクスだけではなく，従来測定していたプロセスメトリクスを組み合わせることで FP クラス予測の正確度や F 値が向上したことから，双方のメトリクスを用いることでより有効な FP クラス予測が可能となり，テスト工程の早い段階から品質を向上させる方法を提案することができた．

まだ 1 製品への適用結果のみであるため，統計的な分析が他の製品にも適用可能なのか

Vol.2010-SE-168 No.6
Vol.2010-EMB-17 No.6
2010/6/1

については検討できていない．また，個別には機能テスト以降の欠陥摘出率と相関が高いと思われる CBO や分岐条件数/有効桁数が実際の FP クラス予測に有効な変数として採用されていない，など分析や調査が必要な箇所がある．今後も引き続きデータを蓄積・分析を続けていく予定である．

しかし，静的解析ツールを自動的に適用，結果を返却することができるしくみを整えつつある中で，実際の開発現場にプロダクトメトリクスを有効に用いるための具体的な方法を考案することができた．今後は実際の開発に本モデルを適用し，その結果について検証を実施していきたい．

参 考 文 献

1) Shen, V., Yu, T., Thebaut, S. and Paulsen, L.: Identifying Error-Prone Software: An Empirical Study, *IEEE Trans. Softw. Eng.*, Vol.11, No.4, pp.317–324 (1985).
2) Chidamber, S. and Kemerer, C.: A Metrics Suite for Object Oriented Design, *IEEE Trans. Softw. Eng.*, Vol.20, pp.476–493 (1994).
3) Basili, V.R., Briand, L.C. and Melo, W.L.: A validation of object-oriented design metrics as quality indicators, *IEEE Trans. Softw. Eng.*, Vol.22, No.10, pp.751–761 (1996).
4) Briand, L.C., Wüst, J., Daly, J.W. and Porter, D.V.: Exploring the relationship between design measures and software quality in object-oriented systems, *J. Syst. Softw.*, Vol.51, No.3, pp.245–273 (2000). 347278.
5) Zhou, Y. and Leung, H.: Empirical Analysis of Object-Oriented Design Metrics for Predicting High and Low Severity Faults, *IEEE Trans. Softw. Eng.*, Vol.32, No.10, pp.771–789 (2006).
6) 亀井靖高，森崎修司，門田暁人，松本健一：相関ルール分析とロジスティック回帰分析を組み合わせた fault-prone モジュール判別方法，情報処理学会論文誌，Vol.49，No.12，pp.3954–3966 (2008).
7) Gyimothy, T., Ferenc, R. and Siket, I.: Empirical validation of object-oriented metrics on open source software for fault prediction, *IEEE Trans. Softw. Eng.*, Vol.31, No.10, pp.897–910 (2005).
8) Kaur, A. and Malhotra, R.: Application of Random Forest in Predicting Fault-Prone Classes, *Intl. Conf. Advanced Computer Theory and Eng.* (2008).
9) Shatnawi, R. and Li, W.: The effectiveness of software metrics in identifying error-prone classes in post-release software evolution process, *J. Syst. Softw.*, Vol. 81, No.11, pp.1868–1882 (2008).
10) Menzies, T., Greenwald, J. and Frank, A.: Data Mining Static Code Attributes to Learn Defect Predictors, *IEEE Trans. Softw. Eng.*, Vol.33, pp.2–13 (2007).
11) 水野幸男監修，全社 SWQC 活動調整委員会 (編)：ソフトウェアの総合的品質管理: NEC の SWQC 活動，日科技連出版社 (1990).
12) SQuBOK 策定部会 (編)：ソフトウェア品質知識体系ガイド—SQuBOK Guide，オーム社 (2007).
13) Ohlsson, N. and Alberg, H.: Predicting fault-prone software modules in telephone switches, *IEEE Trans. Softw. Eng.*, Vol.22, No.12, pp.886–894 (1996).
14) Fenton, N., Neil, M., Marsh, W., Hearty, P. A. H.P., Radlinski, L. A. R.L. and Krause, P. A. K.P.: Project Data Incorporating Qualitative Facts for Improved Software Defect Prediction, *Proc. 3rd Int'l. Workshop on Predictor Models in Softw. Eng. (PROMISE'07)*, 10 pages (2007).
15) `http://www.techmatrix.co.jp/quality/understand/`