

## 工程別の欠陥埋め込み件数の予測

衣簾宏和<sup>†1</sup> 野中誠<sup>†2</sup> 阿萬裕久<sup>†3</sup>

ソフトウェア方式設計, ソフトウェア詳細設計及びコード作成の各工程にて欠陥の埋め込み件数を予測する線形回帰モデルを得た. 本モデルは定量的要因である開発規模に加えてプロジェクトの定性的要因を考慮して導出した.

### A Study on Predicting Injected Defects by Phases

HIROKAZU KINUHATA<sup>†1</sup> MAKOTO NONAKA<sup>†2</sup>  
HIROHISA AMAN<sup>†3</sup>

This paper explains a linear regression model which predicts the number of defects injected in software architecture design processes, software detail design processes and coding processes. The model is derived by considering both a quantitative factor of development size and several qualitative factors.

#### 1. はじめに

ソフトウェア品質管理において, レビューや試験による欠陥検出は有効かつ重要な活動であるが, 同時にコストのかかる活動でもある. すなわち, 必要最小限の投入工数により最大限の欠陥検出を行うことが重要となってくる. 例えば, プロジェクトの計画段階で目標として設定した欠陥検出件数に対して, 実際の欠陥埋め込み数が少ない場合, 残りの存在しない欠陥を検出しようと追加レビュー及び追加試験を行うことは無駄な工数となる. また, 欠陥埋め込み数を実際より少なく予測した場合, 後工程への欠陥流出を引き起こす可能性が高まり, その処置のための工数がかかる. これらの工数を削減するには, プロジェクトの開発方針, 特性及び組織の実力に見合った適切な欠陥埋め込み数を予測する必要がある. プロジェクト固有の欠陥埋め込み数を工程ごとに精度良く設定できれば, レビュー及び試験の評価も実施し易くなり品質管理活動の促進につながる.

しかし実際のところ, ソフトウェアの開発現場では定量的要因である開発規模や定性的要因であるプロジェクトの特性などを考慮しているものの, 欠陥検出の目標件数を(工程ごとの欠陥埋め込み件数の議論をすることなく)経験的に決定しているケースが散見される. これに対し, 従来研究ではプロジェクトの定量的要因や定性的要因といったデータを入力として欠陥数を予測する数理モデルも研究・提

示されているが, 最終成果物での欠陥数や開発ライフサイクル全体での欠陥埋め込み数を予測するものが多く, 開発途中での工程の完了判定時の品質予測というニーズを満足するには課題がある<sup>1)2)</sup>. そこで筆者らはソフトウェア開発ライフサイクルのソフトウェア方式設計からコード作成までの工程を対象として工程別の欠陥埋め込み数を予測するモデルの構築・評価を行ってきた<sup>3)4)</sup>. 本稿では, これまでの研究から得られた知見に基づき, 筆者の所属する組織での実プロジェクトに対して線形回帰モデルによる工程別の欠陥埋め込み数の予測を行い, その評価を行う.

#### 2. 分析対象データ

分析対象のデータには, 筆者の所属する組織で 2008 年から 2010 年に開発された中での類似案件のソフトウェア開発の品質データを用いた. 以下, 表 1 に分析対象データの概要を示す. 尚, 表 1 に示している総開発規模にはコメント行を除外している.

筆者所属部門での開発は主としてエンタープライズ系であり, 数年間の長期スパンで, 新規開発から機能更新及び追加の開発を行っている. 今回の分析対象のデータは所属部門の中で, 1 つの製品系列から選定している. また, 分析対象のデータはソフトウェア方式設計からソフトウェア結合試験までのものである. 上位仕様のソフトウェア要求仕様ではソフトウェアをプロセス単位で分割が完了している. このため, 受注している開発では開発タスクごとにライフサイクルとして閉じており, 開発タスクをデータ分析の 1 レコードとして扱っている.

<sup>†1</sup> 三菱電機マイコン機器ソフトウェア株式会社  
Mitsubishi Electric Micro-Computer Application Software Co., Ltd  
<sup>†2</sup> 東洋大学 経営学部  
Faculty of Business Administration, Toyo University  
<sup>†3</sup> 愛媛大学大学院 理工学研究科  
Graduate School of Science and Engineering Ehime University

表 1 分析対象データの概要

項目	値
収集期間データ	3年(2008年から2010年まで)
プロジェクト数	5件
開発タスク数	46件
ソフトウェア開発の種類	エンタープライズ系
開発言語	C++, Visual C++, Excel VBA
総開発規模(新規・変更)	88.92KLOC

### 3. 欠陥埋め込み数の分析と予測モデル

表 2 に開発タスク単位で収集している定量的メトリクス、表 3 に定性的メトリクスを示す。

定量的メトリクスは品質管理活動の過程で得られるデータである。定性的メトリクスとして、筆者の所属する組織では顧客、エンジニアリング、支援、マネジメント、組織の5つの視点でリスクを点数化(1~4の範囲)し評価・管理を行っており、これらのリスク評価を採用した。表 3 はリスク評価項目のうち、欠陥の埋め込みに影響があると経験的に考えられる評価項目を抽出し採用したものである。また取り得る値とはリスクの評価点である。

表 3 に示している定性的メトリクスの水準はリスクが低いと判断できれば高い値を取る。例えば、表 3 の M<sub>1</sub> は開発対象の事前学習を実施していれば「3」、未実施の場合「2」となる。分析データは開発タスクごとに表 2 及び表 3 のメトリクスを持っており、分析にあたっては各工程で開発タスクを1レコードとして分析を行う。また分析ツールとして Ra を使用する。

表 2 定量的メトリクス(抜粋)

メトリクス名	意味	単位
欠陥埋め込み数	該当の工程で埋め込まれた欠陥数	件
開発規模	新規開発規模と変更の規模	KLOC
流用規模	完全流用の規模	KLOC
欠陥検出数	該当工程のレビューで検出した欠陥数	件
レビュー工数	該当工程で実施したレビュー工数	人時

表 3 定性的メトリクス(抜粋)

記号	メトリクス名	水準数	取りうる値
M <sub>1</sub>	開発対象に対する事前学習の有無	2	2,3
M <sub>2</sub>	要求性能の難易度	3	2,3,4
M <sub>3</sub>	システム構造の複雑度	3	2,3,4
M <sub>4</sub>	既存システムからの流用	2	2,3
M <sub>5</sub>	仕様変更の可能性	3	2,3,4
M <sub>6</sub>	未解決課題の数	3	2,3,4
M <sub>7</sub>	直近の仕様変更の頻度	3	2,3,4

a R は統計解析向けプログラミング言語。

これ以降、定性的特性を述べる際は記号を用いて記載する。

以降に分析の手順とその結果を示す。

#### (1) Step1:各メトリクスの分布傾向の確認と対数変換

各工程のメトリクスに対して分布傾向を確認した。その結果、「欠陥埋め込み数」及び「開発規模」は偏った分布であった。データを正規分布として評価するため、これらの偏った分布をもつメトリクスに対して対数変換を行った。

(以降、対数変換を行ったメトリクスに LN を付して表す。)以下、図 1 から図 3 へ各工程の「欠陥埋め込み数」の頻度分布、図 4 へ「開発規模」の頻度分布を示す。

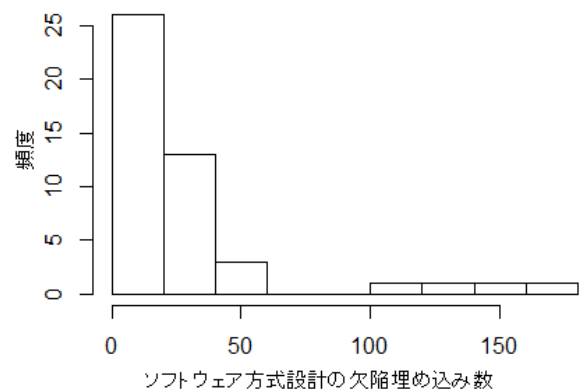


図 1 ソフトウェア方式設計の頻度分布

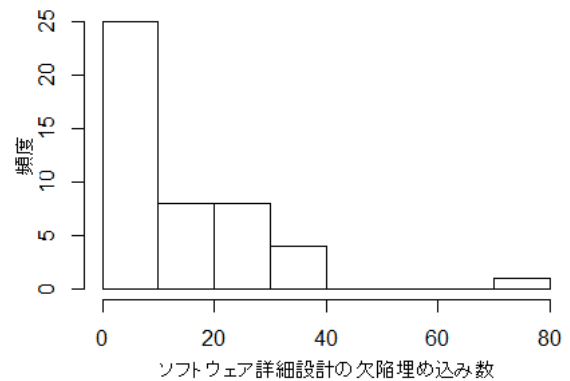


図 2 ソフトウェア詳細設計の頻度分布

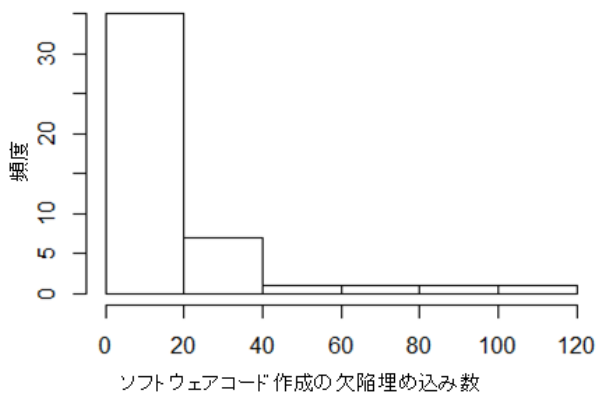


図3 ソフトウェアコード作成の頻度分布

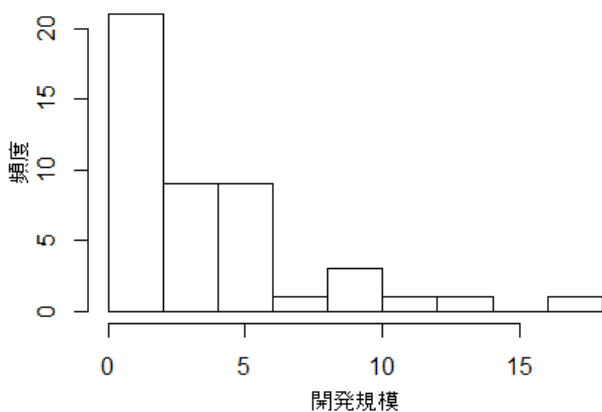


図4 開発規模の頻度分布

(2) Step2:外れ値の除外

欠損値のあるデータ, 欠陥埋め込み密度(単位:件/KLOC)が大きく異なるデータ及び開発方針が異なるデータ等を外れ値とした。この結果, 各工程で以下表4の件数を分析対象データとした。また, 外れ値とした観点を表5へ示す。

表4 工程別の分析対象のデータ件数

工程	分析対象のデータ件数
ソフトウェア方式設計	26件
ソフトウェア詳細設計	26件
コード作成	24件

表5 外れ値と判断する観点

観点	補足
データとして欠損している	
欠陥埋め込み密度[件/KL)を箱髴図により出力した場合, 外れ値となる。	欠陥埋め込み密度は, 各工程のタスクに対して欠陥埋め込み数を開発規模で除算することで導出可能。
欠陥埋め込み数が0件である。	
詳細設計書を未作成である。	
自社開発ではなく, 開発を外部へ発注している。	

(3) Step3:説明変数の候補の選定

Step2後のLN欠陥埋め込み数と各メトリクスとの関連を分析した。散布図や相関係数, 筆者らの経験的知識を考慮した上で各工程の説明変数の候補を以下, 表6のように絞り込んだ。

表6 工程別の説明変数の候補

工程	説明変数の候補
ソフトウェア方式設計	LN開発規模, M <sub>1</sub> , M <sub>2</sub> , M <sub>3</sub> , M <sub>4</sub> , M <sub>5</sub> , M <sub>6</sub> , M <sub>7</sub>
ソフトウェア詳細設計	LN開発規模, M <sub>2</sub> , M <sub>3</sub> , M <sub>4</sub>
コード作成	LN開発規模, M <sub>2</sub> , M <sub>4</sub>

また, 表6に示した各工程の説明変数の候補とLN欠陥埋め込み数との散布図を図5, 図6及び図7に示す。それぞれ縦軸にLN欠陥埋め込み数, 横軸に説明変数の候補を示している。横軸が定性的メトリクス(M<sub>x</sub>)であるものは, 表3に示したとり得る値の範囲である。図5, 図6及び図7ともに, LN開発規模が増えれば欠陥埋め込み数が増え, 定性的メトリクスで高評価となれば, 欠陥埋め込み数が減る傾向にあることが分かる。

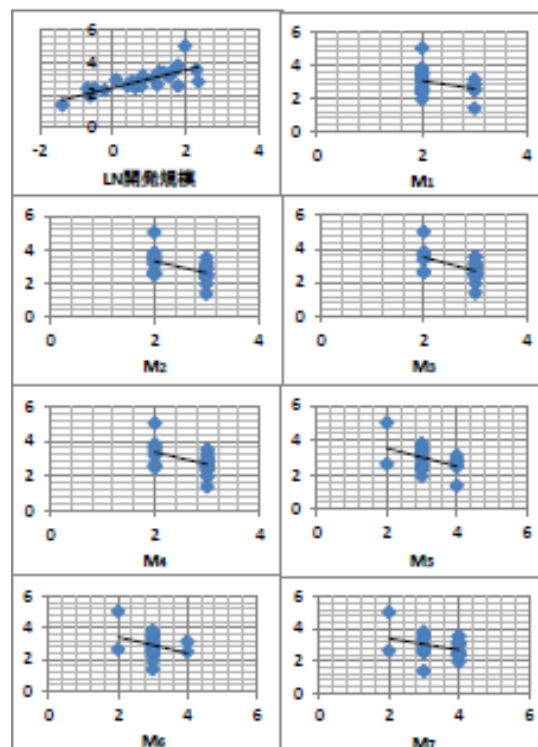


図5 ソフトウェア方式設計の散布図

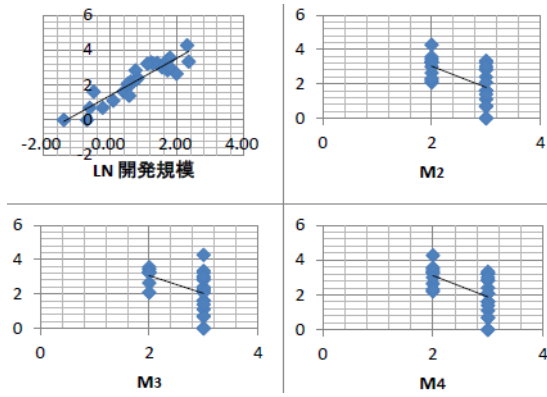


図 6 ソフトウェア詳細設計の散布図

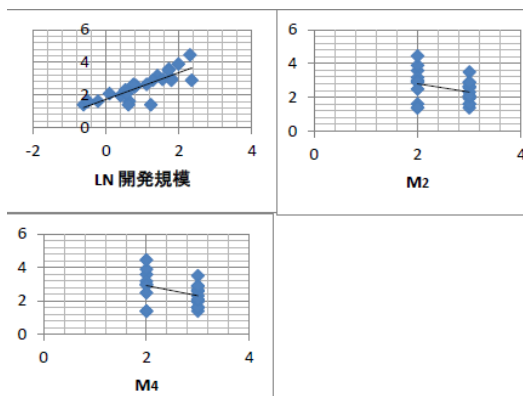


図 7 コード作成の散布図

(4) Step4:ステップワイズ変数選択による重回帰分析

ステップワイズ変数選択による重回帰分析を実施し、表 7 を得た。

表 7 重回帰分析結果

工程	項目	値 (P 値)	
ソフトウェア 方式設計	回帰統計	調整済み R <sup>2</sup>	0.675
		N	26
	説明変数	定数項	3.86 (0.000)
		LN 開発規模	0.52 (0.000)
		M <sub>5</sub>	-0.43 (0.004)
ソフトウェア 詳細設計	回帰統計	調整済み R <sup>2</sup>	0.847
		N	26
	説明変数	定数項	2.23 (0.000)
		LN 開発規模	0.99 (0.000)
		M <sub>2</sub>	-0.29 (0.153)
コード作成	回帰統計	調整済み R <sup>2</sup>	0.640
		N	24
	説明変数	定数項	1.71 (0.000)
		LN 開発規模	0.82 (0.000)

説明変数はいずれも統計的に有意であり、予測モデルの説明力も高い。得られたモデル式を以下、式(1)、式(2)及び式(3)へ示す。なお、式(1)はソフトウェア方式設計、式(2)はソフトウェア詳細設計、式(3)はコード作成である。

$$\log_e D = 3.86 + 0.52 * \log_e S - 0.43 * M_5 \quad (1)$$

$$\log_e D = 2.23 + 0.99 * \log_e S - 0.29 * M_2 \quad (2)$$

$$\log_e D = 1.71 + 0.99 * \log_e S \quad (3)$$

ここで D は欠陥埋め込み数、S は開発規模である。

4. 予測モデルの評価

得られたモデル式を用いて、予測値と実績値の比較評価を行った。予測誤差の評価にあたっては、予測値と実測値の差を実測値で割った MRE (Magnitude of Relative Error) を用いた。表 8 に各工程の MRE の平均値 (MMRE, Mean of MRE), MRE の中央値 (MdMRE, Median of MRE) 及び MRE の標準偏差を示す。

表 8 得られたモデル式の予測の MRE の結果

工程	項目	値
ソフトウェア 方式設計	MMRE	0.076
	MdMRE	-0.064
	MRE の標準偏差	0.458
ソフトウェア 詳細設計	MMRE	0.090
	MdMRE	0.042
	MRE の標準偏差	0.485
コード作成	MMRE	0.140
	MdMRE	-0.121
	MRE の標準偏差	0.734

図 8 から図 10 までに工程別の従来予測による MRE とモデル式の MRE を示す。

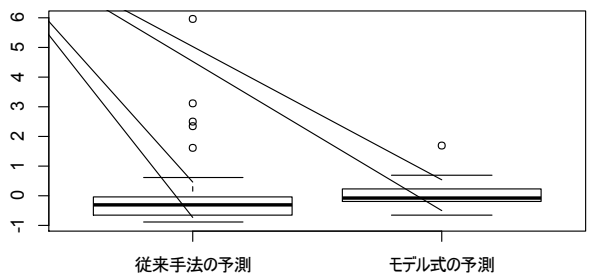


図 8 ソフトウェア方式設計の MRE 分布の比較

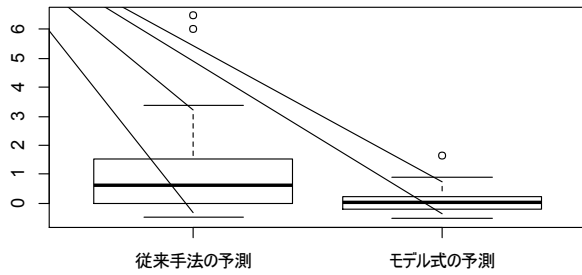


図 9 ソフトウェア詳細設計の MRE 分布の比較

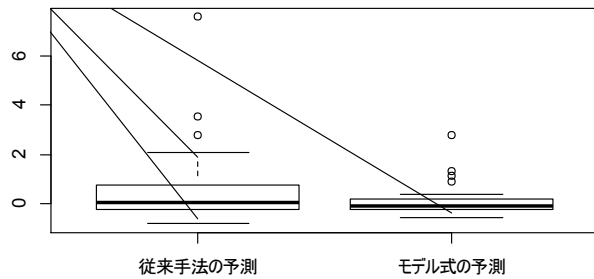


図 10 コード作成の MRE 分布の比較

モデル式による予測で MRE の四分位点の幅が減少しており、中央値が 0 近傍へ移動していることから従来手法と比較して今回得られたモデル式による予測のほうが精度向上していることが分る。尚、ここで言う従来予測とは従来から組織で利用されてきた欠陥埋め込み密度の指標値による予測である。従来手法による工程別の欠陥数の予測値は、工程別に指標値として設定している欠陥埋め込み密度と開発規模の乗算で得ている。筆者の所属する部門では各工程の欠陥埋め込み密度を以下の値を指標値として設定している。

表 9 欠陥埋め込み密度の従来指標値

工程	欠陥埋め込み密度
ソフトウェア方式設計	5.70 件/KL
ソフトウェア詳細設計	7.03 件/KL
コード作成	5.30 件/KL

また、図 11 に工程毎の MRE 分布の箱髭図を示す。各工程とも MMRE がいずれも 0.25 以下であることから十分な精度<sup>9)</sup>を持っていると考えられる。

<sup>b</sup> 従来手法では、例えば開発規模が 10KL の場合ソフトウェア方式設計では 5.70 件/KL×10KL=57 件の欠陥が埋め込まれると予測する。

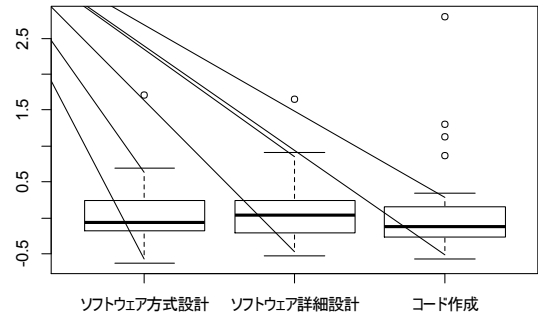


図 11 工程別の MRE の分布

各工程とも外れ値<sup>c</sup>が発生している。外れ値となったデータについてソフトウェア方式設計では、規模が小さい開発に対して欠陥の埋め込みを過大に予測し、外れ値になっていることが判明している。しかしその他の工程の外れ値に関しては、予測モデル導出に使用したメトリクスからは特徴を見出すことが出来なかった。

## 5. 考察

### 5.1 モデル式について

全工程とも係数及び説明変数に違いはあるが、全モデル式に共通して「LN 開発規模 ( $\log_e S$ )」が選択されている。開発規模の増加が欠陥埋め込み数に影響があることは容易に予測できる。従来研究からも開発規模と欠陥埋め込み数との関係が報告されており、従来研究を裏付けるものと言える。また、定性的な説明変数の係数の符号が負であることから表 3 について 3 章で述べた高い評価点を得たものほど欠陥の削減に効果があることを説明できている。

ソフトウェア方式設計の定性的な説明変数は「仕様変更の可能性 ( $M_3$ )」が選択された。「仕様変更の可能性 ( $M_3$ )」は欠陥埋め込み数へ影響しているものと考えられる。また、 $M_3$  と他のメトリクスとの間には正の相関があり、 $M_3$  を説明変数とすることで他のメトリクスを説明できた。このため、定性的メトリクスの 7 つの候補の内、モデル式の説明力をもっとも高くなる  $M_3$  が選択されている。

ソフトウェア詳細設計の定性的な説明変数は「要求性能の難易度 ( $M_2$ )」が選択された。ソフトウェア仕様の実現の難易度が欠陥の埋め込み数へ影響することを考えれば妥当と判断する。 $M_2$  はソフトウェア方式設計の時と同じく他の説明変数の候補と相関性が高く、 $M_2$  を選択すれば他のメトリクスを説明できた。このため  $M_2$  が 1 つのみ選択され

<sup>c</sup> 本稿では分析に R を使用した。R では第 1 四分位点 (25%点, Q1) と第 3 四分位点 (75%点, Q3) の差 IQR とした時、データが Q1-IQR より小さい、もしくは Q3+IQR より大きいとき外れ値としている。

ている。

コード作成では定性的な説明変数が選択されていない。これは定量的な説明変数である「LN 開発規模 ( $\log_e S$ )」と定性的な説明変数の候補との間に相関があるためである。また開発工程がコード作成まで進むとプロジェクト管理のマトリクスの影響より、ソースコードそのものを分析する視点が重要になってくるのではないかと考える。

## 5.2 モデル式の活用と課題

各工程で選択されたマトリクスは欠陥埋め込み数の削減を目的とした改善活動のヒントになっていると言える。すなわち開発において、欠陥埋め込み数の削減に最も効果のある方法は開発規模の削減であることを示している。また、定性的要因の説明変数については、欠陥の埋め込みが増える評価となっている場合、それを是として開発を続行するのではなく、欠陥の埋め込みが削減できるようにマネジメントを行う必要がある。欠陥埋め込み数の削減は出荷後の欠陥削減にもつながるため重要である。

本研究で導出した欠陥埋め込みのモデル式は(1), (2)及び(3)の定性的な説明変数は全てのソフトウェア開発において欠陥の埋め込みに影響があると考えられる。しかし、モデル式はソフトウェア業界で共通的に使用出来るわけではない。何故なら、組織の事情、実力、方針などにより組織間に違いがあり、欠陥埋め込みに影響があるマトリクス(特に定性的マトリクス)やモデル式の係数にも違いがあると考えられるからである。

また、前工程までの欠陥検出などの情報をモデルのインプットとする事でモデルの予測精度、すなわち MRE のバラつきだけでなく、外れ値の削減も改善する可能性があるため、今後も検討を行う必要がある。

## 6. おわりに

本研究では、各工程の欠陥埋め込み数のばらつきを説明する予測モデルを得ることが出来た。今後は、工程としての設計の作り込みの安定度合いの分析・評価と各工程であった懸案事項の分析を行っていく予定である。また本研究で取り組んだ手法は従来の予測手法より有効であるため、所属部門へ導入のための体系化、ルール化等を検討していく予定である。

## 参考文献

- 1) Fenton, N., et. al., : Project Data Incorporating Qualitative Facts for Improved Software Defect Prediction, Proc. Of Int'l Workshop on Predictor Models in Softw. Eng., pp.2(2007)
- 2) Oliver Laitenberger, et. al., : An Industrial Case Study to Examine a Non-Traditional Inspection Implementation for Requirements Specifications, Empirical Software Engineering, 7, pp.345-374, (2002)
- 3) 衣簀宏和, 野中誠, 阿萬裕久: 工程別の欠陥埋め込み

件数の予測—ソフトウェア方式設計を対象として—, ウィンターワークショップ 2012・イン・琵琶湖論文集, pp.35-36 (2012)

4) 衣簀宏和, 野中誠, 阿萬裕久: 工程別の欠陥埋め込み件数の予測—ソフトウェア詳細設計, コード作成を対象として—, 第 74 回情報処理学会全国大会講演論文集, 4B-4(2012)

5) Ingunn Myrtveit, Erik Stensrud, and Martin Shepperd: Reliability and Validity in Comparative Studies of Software Prediction Models, IEEE Trans. Softw. Eng., Vol. 31, No. 5, pp.380-391 (2005).