演習コースII（実践コース）

リリース後不具合発生予測モデルに基づく，効果的なプロセス改善への仕掛けの提案

研 究 員：星野　智彦（アイシン精機株式会社）
主　査　：小池　利和（ヤマハ株式会社）
副 主 査：小室　睦（株式会社プロセス分析ラボ）
副 主 査：柏原 一雄（株式会社デンソークリエイト）

研究概要

ソフトウェアのリリース後に発生する不具合数を低減するために，できるだけ上流で欠陥を除去することが効果的であることはよく知られている．本論文での対象組織においては，上流での欠陥除去の効果を定量的に把握できていないため，最終工程のテストによる品質作り込みが定常化していた．

本研究ではリリース後不具合発生予測モデルを構築し，リリース後の不具合発生要因となり得る指標を特定できた．「ソフトウェアアーキテクチャ設計におけるレビュー欠陥と統合テストにおける欠陥の除去比率を向上させ，テスト工数比率を低減させる」ことで不具合発生確率が低減するというモデルを構築した．

これにより上流での品質作り込みの重要性を対象組織のデータを用いて，定量的かつ客観的に示すことができた．このモデルにより開発現場の理解を得ながら従来以上に上流工程で欠陥除去を促すプロセス改善が推進可能となる．また，プロジェクト計画時にリリース後不具合発生確率を低減するための具体的な改善目標設定が可能となることも期待できる．

1. はじめに

筆者の携わっている車載ソフトウェアにおいては，製品の特性上，品質の確保が重要であり信頼性が非常に重要視されている．パソコンのアプリケーションのように一度組み込んだソフトをタイムリーにアップデートすることが難しいため，リリース後の品質要求は非常に高い[1]．

ソフトウェア開発においては，より上流で欠陥を除去することで，下流のテスト工程での品質を向上し，手戻りコストやリリース後の不具合の低減に効果があることがよく知られている[2]．従って，信頼性の向上のために上流工程の強化を試みることが重要であると考えられる．

しかし，スケジュール，工数，技術上の問題点などは開発途中で明らかになるのに対し，品質状況は開発の最終盤もしくはリリース後にならないと正確に把握することが難しい．このため，開発現場は，上流工程の重要性を理解していたとしても十分な品質作り込みをせずに次工程に進んでしまうことがある．また，プロセス改善の観点から見ると，どのプロセスをどのように改善すれば十分な品質向上が実現できるか分からないため，開発現場への動機付けが十分にされないことがある[3]．本研究では，リリース後の不具合発生確率低減を目的として，上流工程での品質作り込みを開発現場に意識付けることを課題とする．開発現場のデータから，リリース後の不具合発生の原因となる指標を特定し，重回帰分析を用いてリリース後の不具合発生予測モデルを構築する．更に，開発現場の動機付けの仕掛けとして，構築したモデルをもとに各指標の目標値と改善効果を示す．これにより，上流工程での品質作り込みの重要性を定量的かつ客観的に示すことができ，開発現場を適切に動機付け，改善を効果的に進めることが期待できる．

2. 組織の背景

2.1.組織における役割

筆者の所属企業は自動車向けサプライヤとして，車載部品を製造している．筆者はその中で，全社のソフトウェア開発を横断的に支援する組織（以下，自組織）に所属し，プロセスの制定および改善業務に従事している．顧客からのリリース後の品質要求も高い一方でコスト低減や工期短縮も求められている開発現場からは，「品質を維持しながら開発を効率的に進められるプロセスが欲しい」という意見が強い．

2.2. 組織における開発プロセス

筆者の所属企業における開発プロセスは図 1 に示す V 字モデルを採用している．ソフトウェア要件分析（以下，SWE.1），ソフトウェアアーキテクチャ設計（以下，SWE.2），ソフトウェア詳細設計およびユニット構築（以下，SWE.3），ソフトウェアユニット検証（以下，SWE.4），ソフトウェア統合および統合テスト（以下，SWE.5），ソフトウェア適格性確認テスト（以下，SWE.6）の 6 プロセスから構成される[4]．SWE.1 から SWE.3 までを上流工程とし，SWE.4 から SWE.6 までをテスト工程として定義している．本論文における「リリース後」とは，SWE.6 終了後からの期間としている．

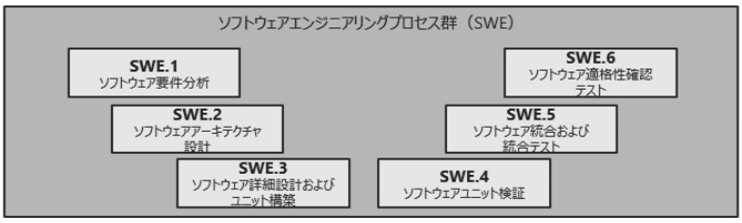

図 1 V 字モデル

2.3 自組織の現状

IPA/SEC のデータを用いたリリース判断基準はあるものの，自組織の実情に必ずしも適合しておらず，効果的な品質確保策にはなっていない．現実には，開発現場が，成果物を形式的に揃え，テストによる品質確保を行い，定性的な監査結果によりリリース可否が判断されている．品質が担保されているという明確な根拠は無く，リリース後に不具合が発生してしまうこともある．欠陥を発見したプロセス毎に欠陥件数を集計し，リリース後に不具合が発生したプロジェクトと発生しなかったプロジェクトで比較する分析を実施した．その結果，図 2 に示すように，リリース後に不具合が発生したプロジェクトは，上流工程の欠陥を摘出する割合が低いことが分かった(実データは公開できないためイメージ図で示す)．しかし，上流工程での品質作り込みの効果を定量的に提示することはできず，開発現場に対する十分な動機付けとはならなかった．

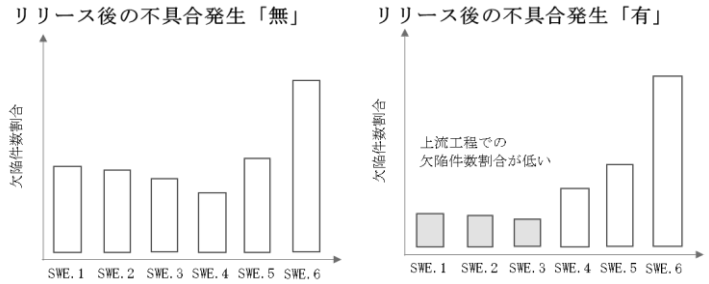

図 2 欠陥件数割合の分析結果（イメージ図）

3. 課題設定

3.1. 設定した課題

リリース後の不具合発生の防止を目的に，上流での品質作り込みを開発現場に動機付ける仕組みを作ることを課題とする．目指す姿を図 3 に示す．

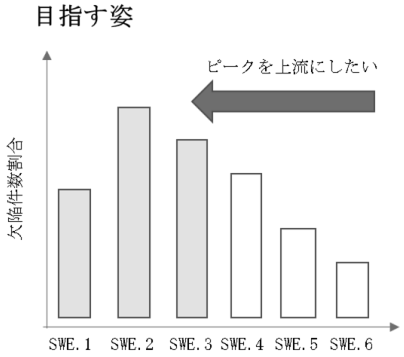

図 3 目指す姿

3.2. 課題の解決手段

開発現場のデータをもとに，リリース後の不具合発生要因の特定とその予測モデルを構築する．開発現場の動機付けの仕掛けとして，予測モデルから，改善活動の具体的な目標値や定量的な改善効果を示す．

4. 分析内容

4.1. 分析対象のデータ

プロジェクトデータの内，開発傾向が類似している製品群から分析対象のデータを抽出した．外れ値の要因となり得る以下のプロジェクトのデータは，事前に除外している．

・変更規模が 100[LOC]未満のプロジェクト
・工数のデータが欠損しているプロジェクト

4.2. 結果を表すメトリクスの設定

筆者の所属企業におけるリリース後の不具合発生件数は限られており，多くの場合，不具合が有ったか無かったか，のみが問題になるため，リリース後の不具合発生「無」を 0，「有」を 1 とする二値を，結果を表わすメトリクスとして設定した．

4.3. 分析対象メトリクスの設定

上流工程での品質作り込みが，リリース後の不具合発生確率にどう影響するかを確認するため，レビューに関する指標を中心に分析対象メトリクスをカテゴリ毎に選択した．一覧を表 2 に示す．

表 2. 分析対象メトリクス一覧

カテゴリ	メトリクス	単位	説明
工数の比率	テスト工数比率	%	開発全体に占めるテスト工数の割合
	レビュー工数比率	%	開発全体に占めるレビュー工数の割合
	レビュー工数_テスト工数比率	%	テスト工数とレビュー工数の比率
	テスト及びレビュー工数比率	%	開発全体に占めるレビュー及びテスト工数の割合
	設計_コーディング工数比率	%	上流工程（設計工程）に占めるコーディング工数の割合
	設計レビュー工数_設計工数比率	%	上流工程（設計工程）に占めるレビュー工数の割合
	コードレビュー_コーディング工数比率	%	コーディング工数に占めるコードレビュー工数の割合
欠陥の比率	前倒し率	%	全欠陥数に占めるレビュー欠陥数の割合
レビュー欠陥	レビュー欠陥密度_SWE1	件/LOC	SWE.1における変更規模辺りのレビュー欠陥件数
	レビュー欠陥密度_SWE2	件/LOC	SWE.2における変更規模辺りのレビュー欠陥件数
	レビュー欠陥密度_SWE3	件/LOC	SWE.3における変更規模辺りのレビュー欠陥件数
テスト欠陥	テスト欠陥密度_SWE4	件/LOC	SWE.4における変更規模辺りのテスト欠陥件数
	テスト欠陥密度_SWE5	件/LOC	SWE.5における変更規模辺りのテスト欠陥件数
	テスト欠陥密度_SWE6	件/LOC	SWE.6における変更規模辺りのテスト欠陥件数

4.4. 分析結果

4.4.1. 二群の平均の差の t 検定

表 2 で定義したメトリクスに対して，不具合の発生しなかったプロジェクト(A 群)とリリース後に不具合の発生したプロジェクト(B 群)との平均の差を確認した結果，表 3 に示すように，4 つのメトリクスに対して有意差が見られた(有意水準 $p < 0.05$)．

表 3 結果一覧

メトリクス	p値	比較	5%有意差	結果
テスト工数比率	0.667	A<B	無	
レビュー工数比率	0.088	A>B	無	
レビュー工数_テスト工数比率	0.449	A>B	無	
テスト及びレビュー工数比率	－	－	－	(データ数不足により計測不能)
設計_コーディング工数比率	0.938	A>B	無	
設計レビュー工数_設計工数比率	**0.007**	**A>B**	**有**	**レビュー工数比率が高い方が良い**
コードレビュー_コーディング工数比率	0.552	A>B	無	
前倒し率	0.985	A<B	無	
レビュー欠陥密度_SWE1	0.236	A>B	無	
レビュー欠陥密度_SWE2	**0.038**	**A>B**	**有**	**レビュー欠陥密度が高い方が良い**
レビュー欠陥密度_SWE3	**0.043**	**A>B**	**有**	**レビュー欠陥密度が高い方が良い**
テスト欠陥密度_SWE4	0.194	A>B	無	
テスト欠陥密度_SWE5	**0.031**	**A>B**	**有**	**テスト欠陥密度が高い方が良い**
テスト欠陥密度_SWE6	0.601	A<B	無	

以上の結果から，リリース後に不具合が発生するプロジェクトは，レビューに十分な工数を割けず，上流工程で欠陥を十分に除去できていない傾向にあるということが言える．

4.4.2. ロジスティック回帰分析による予測モデルの構築

上流での品質作り込みとリリース後の品質を関連づけて，具体的な目標値や定量的な改善効果を提供するために，重回帰分析で予測モデルを構築する．目的変数はリリース後の不具合有無，説明変数は 4.4.1.で有意差のあったメトリクス(表 3 参照)を採用する．

本研究では，目的変数が二値（リリース後の不具合発生有無）の回帰分析を行うため，ロジスティック回帰分析[5]を適用する．

(ア) 本研究で用いる回帰分析手法

ロジスティック回帰分析は，一つの目的変数（二値変数）の確率を，複数の説明変数によって説明，予測する多変量解析の一つである．以下で定義されるロジスティック関数を介して最適な β_i を求める回帰分析である．

$$\boldsymbol{f(X) = \frac{1}{1 + exp\left(-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \beta_n X_n)\right)} \left(i = 0,\ 1,\ 2 \cdots,\ n\right)}$$

本論文で算出されるのはリリース後の不具合発生確率となり，偏回帰係数について $\beta_i > 0$ の場合は説明変数の値が大きくなるほど，発生確率が高くなる．反対に，$\beta_i < 0$ の場合，説明変数の値が大きくなるほど発生確率が低くなる．

(イ) 多重共線性と VIF

説明変数間に強い相関（多重共線性）がある場合，分析が困難になる．たとえ結果が算出されたとしてもその信頼性は低い．チェックの手段としては，説明変数間の相関の調査や VIF(分散拡大係数)を求め，VIF<10 であることを確認する方法[6]がある．

(ウ) AIC とステップワイズ法

モデルの適合度を表す指標として，AIC(赤池の情報量規準)がある．AIC は，いくつかのモデルが現実のデータにどの程度適合するかを比較する尺度で，値が小さいほど良いモデルとみなされる．本研究では AIC を，回帰モデルを検討する際の変数選択法であるステップワイズ法[7]と共に用い，探索した範囲で AIC の値が最小となったモデルを選択する．

4.4.3. 予備分析によるメトリクスの追加

（ア） 予備分析結果

t 検定で有意差のあったメトリクスによるロジスティック回帰分析を行った結果，ステップワイズ法により，表 4 に示すように，偏回帰係数が正となる説明変数が選択された．品質向上効果からレビューやテスト実施の欠陥密度に対して偏回帰係数が負になると考えていたが，この期待に反しており，「上流での品質作り込みを開発現場に動機付けたい」という本研究の意図からすると満足できるモデルとは言えない．

表 4 予備分析結果

説明変数	係数	p値
定数項	-12.645	0.0337
レビュー欠陥密度_SWE2	-2.785	0.0136
レビュー欠陥密度_SWE3	2.134	0.0573
テスト欠陥密度_SWE5	-1.796	0.1357

AIC: 33.258

（イ） 予備分析結果に対する考察と改善方策

他の変数の影響を検討したところ，説明変数である「欠陥密度」と目的変数である「リリース後不具合」の両方に影響する複数の要因があるため，偏回帰係数が正になっていることが推察された．そこで，(1)説明変数のカテゴリを増やす．(2)品質向上効果が現れやすくなるように変数を取り替える．という２つの対処を実施した．

(1) カテゴリを増やす

レビューおよびテストの欠陥密度のみの，限定されたカテゴリの説明変数であったため，説明のつかないモデルになったと想定される．そこで，表 2 に示したカテゴリからバランス良く説明変数を再選択することとした．

(2) 比率変数の使用・導入

例えば，開発の難易度が高いとすべての工程に渡り欠陥が出やすくなり，リリース後不具合も増加することが予想される．このような要因の影響が強く現れると欠陥密度とリリース後不具合の間の偏回帰係数も正になる．こういった要因の影響を除くやり方として比率変数を導入するという手法がある[8]．比率をとることで開発工程全般に働く要因の影響を軽減できる．例えば，レビュー欠陥比_SWE2 は SWE.2 のレビューにどれだけ力を入れたかを表している．ただし，多重共線性を避けるため，上流工程の品質向上効果を見るためにレビューやテストを実施する工程の中でより上流に位置するものだけを残した．

新たに定義したメトリクスを表 5 に示す．

表 5 分析に使用するメトリクス

メトリクス	説明
テスト工数比率	開発全体に占めるテスト工数の割合
レビュー工数比率	開発全体に占めるレビュー工数の割合
レビュー欠陥比_SWE1	レビュー総欠陥数に占めるSWE.1のレビュー欠陥数の割合
レビュー欠陥比_SWE2	レビュー総欠陥数に占めるSWE.2のレビュー欠陥数の割合
テスト欠陥比_SWE4	テスト総欠陥数に占めるSWE.4のテスト欠陥数の割合
テスト欠陥比_SWE5	テスト総欠陥数に占めるSWE.5のテスト欠陥数の割合

4.4.4. 結果

表 5 のメトリクスを説明変数として，ステップワイズ法を適用した．表 6 に示す説明変数が AIC 最小の組合せとして選択された．VIF により，説明変数間に多重共線性の問題は無いことを確認している．

表 6 ロジスティック回帰分析で選択された説明変数

説明変数	係数	p値
定数項	-2.862	0.3301
テスト工数比率	11.528	0.1846
レビュー欠陥比_SWE2	-11.517	0.0621
テスト欠陥比_SWE5	-5.328	0.1911

AIC=27.432

4.4.5. 分析結果の考察

レビュー欠陥比_SWE2 の係数が負になったことで，欠陥摘出のピークをより上流にすることが重要であることを示すことができた．また，最終工程ではなく，SWE.5 の欠陥比が選択されたことから，テスト工程においても，より上流で欠陥を検出することが重要であることを示すことができた．

(1) 選択された説明変数の妥当性の考察

(ア) テスト工数比率

係数が正になっている．これは，テスト工数をかけすぎてテスト偏重とならないようにすることが有効と解釈され妥当である．

(イ) レビュー欠陥比_SWE2

係数が負になっており，この工程のレビューに力を入れることでリリース後不具合を防止できるものと解釈され妥当である．

(ウ) テスト欠陥比_SWE5

係数が負になっており，SWE.6 から SWE.5 にテストの力点をシフトすべきものと解釈され妥当である．

(2) 予測結果の確認

予測結果として，回帰曲線を図に示す．

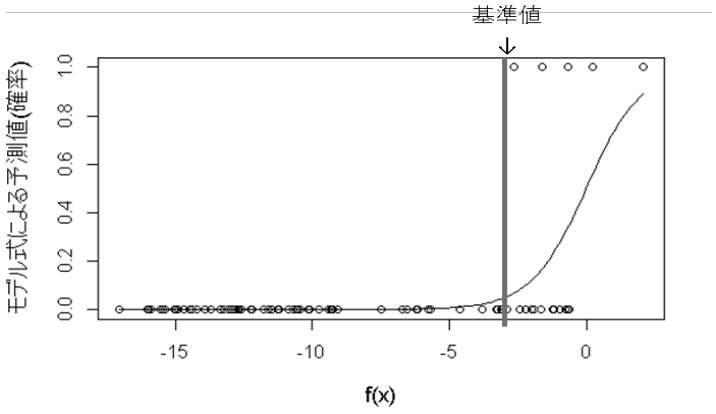

図 4 ロジスティック回帰曲線

本研究は，リリース後の不具合発生確率を低減するためのプロセス改善が目的であることから，図に示すように，リリース後の不具合「有」と予測されるプロジェクトを全て抽出できる基準線を定義する．分割表を表 7 に示す．

表 7 分割表

基準値：予測値=0.05

	リリース後の不具合「無」（予測）	リリース後の不具合「有」（予測）
リリース後の不具合「無」（実績）	64	13
リリース後の不具合「有」（実績）	0	5

表 7 から，分析対象 82 件のプロジェクトの中で，予測が外れたものに着目する．リリース後の不具合「無」と予測した内，実績が「有」となったものは 0 件．つまり 0%となり，リリース後の不具合が発生するリスクのあるプロジェクトを取りこぼすことは無い．逆に 82 件のプロジェクトの中で，リリース後不具合「有」と予測した内，実績が「無」となったものは 13 件，つまり 13/82=16[%]となる．この 13 件のプロジェクトに何らかの対応を施すならば無駄なコストとなり得るが，取りこぼしを防ぐためにはやむを得ない．割合も 16%と少ないため問題ではないと考えられる．

5. 分析から得られた知見

以上の分析から得られた知見をまとめると次のようになる．

(1) リリース後不具合発生に関係する複数のメトリクスを特定した．
(2) 既存のメトリクスを用いた分析結果にカテゴリの再設定，比率変数の導入などの技法を適用し，より適切な説明変数の組合せを見出すことができた．
(3) 予測モデルの構築により，上流での品質作り込みの重要性を自組織のデータを用いて，定量的かつ客観的に示すことができた．

6. 想定できる分析結果の活用方法

6.1. 組織的なプロセス改善への活用

プロセス毎の欠陥件数割合の分析により，開発現場の抱える問題点を可視化した．図 2 のようにプロセス毎の欠陥件数を集計し，プロジェクト毎に集計結果を比較することで組織の状況を監視し，改善につなげることができる．可視化により課題が明確になり，動機付けにもつながる．また，t 検定と予測モデルの構築により，品質向上に寄与する複数の指標を特定できた．これらの指標のベースラインを確立し，活用することで組織的なプロセス改善を推進することができる．

6.2. 予測モデルの活用方法

予測モデルの構築により，上流での品質作り込みの重要性を自組織のデータを用いて，定量的かつ客観的に示すことができた．今回構築した予測モデルは下記のようなシーンで適用できる．

(1) プロジェクト計画時

各工程での欠陥除去の目標設定に使用する．開発現場のデータから構築した予測モデルを用いているため，プロジェクトの実態にあった目標設定が可能になるものと期待される．説明変数の値を変化させ見積り直すことで具体的にどのプロセスをどのように改善すれば品質向上につながるかが分かり，現場の動機付けにつながる．

(2) プロジェクト進行中

開発現場のプロジェクトデータ変動を監視し，予測モデルから設定した目標値の達成状況を確認する．リアルタイムにデータを得られれば，より迅速にプロジェクトが破綻するリスクを特定することができる．

プロジェクト計画時や進行中においては，プロジェクト完了直前で手遅れになる前に，定量的な根拠を以て，具体的な是正勧告及び処置を行えることから，より効果的な改善活動が進められることが期待できる．

7. 関連研究

プロジェクトのデータから品質の状況を予測する研究結果として次のようなものがある．文献[9]では，出荷後バグ基準値達成度合いを目的変数としたロジスティック回帰分析により予測モデルを構築している．そして，その予測モデルから更に選択されたパラメータについて深掘り分析を実施している．これに対し本研究では，選択されたパラメータからプロセス改善ポイントを抽出することに主眼を置いている点に独自性を持たせている．文献[10]では，上流工程のレビュー欠陥数から下流工程でのテスト欠陥数を予測する品質予測モデルが提案されている．ある工程でのレビューの徹底度合いとして，その工程での欠陥摘出数をそれ以降の工程での欠陥摘出数との比で表した前倒し率という指標を定義している．本研究では，下流工程のテスト欠陥数ではなく，リリース後の不具合発生有無という，より開発現場へのインパクトが強い指標を目的変数として採用している点が異なっている．

8. まとめ

自組織の実績データを分析し，リリース後不具合に影響する既存メトリクスを洗い出した．さらに，上流での品質作り込みの効果が明確になるようにメトリクスを整理し直し，リリース後の不具合「有無」の判定を可能にするモデルを構築した．

これらの分析結果，特にリリース後不具合発生予測モデルにより上流での品質作り込みに対する重要性を客観的，定量的に示すことができるようになった．

このモデルを実際にプロジェクトに適用し，開発現場の動機付けをすることを今後の課題とする．

参考文献

[1] 村山 浩之，松本 隆明，「【所長対談】車載ソフトウェア開発の今後の方向性」，SEC journal 第 42 号，pp. 2-7， 2015

[2] SQuBOK 策定部会，「ソフトウェア品質知識体系ガイド（第 2 版）」，オーム社，2014

[3] 坂本 啓司，田中 敏文，楠本 真二，松本 健一，菊野 亨，「利益予測に基づくソフトウェアプロセス改善の試み」，電子情報通信学会論文誌 D． Vol． J83-D-I， No． 7，2000

[4] IATF 16949 セミナー， 「国際規格プロセスモデルを活用したプロセスアプローチ」，ビジネスキューブ・アンド・パートナーズ株式会社， 2019

[5] 大林 準，「ロジスティック回帰分析と傾向スコア(propensity score)解析」， 天理医学紀要 第 19 巻 第 2 号， pp. 71-79， 2016

[6] 木原 雅子，木原 正博， 「医学的研究のための多変量解析」， メディカル・サイエンス・インターナショナル，p. 77，2008

[7] 丹後 俊郎，山岡 和枝，高木 晴良：「ロジスティック回帰分析-SAS を利用した統計解析の実際」， 朝倉書店，pp. 198-200， 1996

[8] 野中 誠，小池 利和，小室 睦， 「データ指向のソフトウェア品質マネジメント」， 日科技連出版社， 2012

[9] 柳田 礼子，「効率的な品質改善に向けた CMMI 成熟度レベル別の要因分析」，ソフトウェア品質シンポジウム 2015，2015

[10] 小室 睦，薦田 憲久， 「ピアレビューデータに基づく品質予測モデル」， 電子情報通信学会論文誌 D． Vol． J94-D， No． 2， pp. 439-449， 2011