

**科学研究費助成事業 研究成果報告書**

平成 28 年 5 月 17 日現在

機関番号：16301

研究種目：基盤研究(C) (一般)

研究期間：2013～2015

課題番号：25330083

研究課題名(和文) ソフトウェア品質に対してコメント文がもたらす功罪に関する実証的研究

研究課題名(英文) Empirical Study on Merits and Demerits of Comments Toward Software Quality

研究代表者

阿萬 裕久 (Aman, Hirohisa)

愛媛大学・総合情報メディアセンター・准教授

研究者番号：50333513

交付決定額(研究期間全体)：(直接経費) 3,700,000円

研究成果の概要(和文)：本研究では、ソースファイル中に書かれるコメント文に着目したデータ収集と分析を行った。一般に、コメント文はプログラムの理解を助けるが、その半面、複雑で分かりにくいプログラムにおける可読性の低さをこまかすために書かれている可能性もある。オープンソースソフトウェアに対する分析を行ったところ、メソッド(関数)の中に書かれるコメントは、そこでのバグの潜在性と関係が深いことが定量的に確認できた。この知見は、重点的にレビューすべき対象の絞り込みに有用であると考えられる。

研究成果の概要(英文)：This study performed data collections and analyses focusing on the comments written in source files. Comments usually help in comprehending the programs. However, comments can also be added to mask poor readabilities of complicated programs. The results of this study quantitatively showed that comments written inside method (function) bodies are likely to be related to the fault-proneness in those methods (functions). This finding would be useful in enhancing the code reviewing by narrowing suspicious parts to be checked.

研究分野：ソフトウェア工学

キーワード：コメント文 ソフトウェア品質 リファクタリング

## 1. 研究開始当初の背景

(1) 一般に、プログラムは分かりやすく読みやすいものであることが望まれる。実際、ソフトウェア開発を行う企業や組織は、「良いプログラム」を作り出すために一定のコーディング規則やガイドラインに従ってプログラムを書くことを標準としていることが多い。それに関連してさまざまな CASE ツールや静的解析技術も従来から研究されてきている。また、プログラムの「書き方」だけでなく、プログラムの「特徴」を定量的尺度(メトリクス)によって数値化し、そのデータ解析を通じて得られた知見を品質管理に役立てる研究も盛んに行われてきている。

(2) これまで研究・開発されてきたツールやメトリクスは、コーディングスタイルやコードの構造に着目し、どういったプログラムにバグが潜みやすいかといった観点で多くの研究成果を挙げてきた。その過程において、「コメント文」は削除されたり無視されたりすることがほとんどであった。なぜならば、コメント文の内容はプログラムの実行に一切影響しないためである。実際、コメント文を主対象とした研究報告は少なく、「可読性」という観点で実験・考察された論文がいくつか見られる程度であった。

(3) 一方、プログラミング作法やリファクタリングの分野では、定性的ではあるがコメント文についていくつかの提言がなされてきている。例えば、Kernighan らは悪いコードにコメントを付けるくらいならば書き直すべきであると主張している。また Fowler は、丁寧に書かれたコメントが実は分かりにくいコードを補うためである場合があり、一種の不吉な匂い (code smell; リファクタリングの対象) に関係すると述べている。しかしながら、データに基づいた定量的な研究は十分に行われておらず、新たな研究材料として注目に値するものとなっていた。

### <引用文献>

B.W. Kernighan and R. Pike: The practice of programming, Addison-Wesley Longman (1999).

M. Fowler, M: Refactoring: Improving the Design of Existing Code, Addison-Wesley Longman (1999).

## 2. 研究の目的

(1) 本研究の目的は、ソフトウェア品質に対してコメント文が及ぼす影響(功罪)について、データ解析を通じて明らかにしていくことである。主なポイントは、研究結果として得られる傾向の一般性を確認すること、さらには原因の解明とインプロセスでの有効性確認を行うことである。

(2) 応募時点では少数のプロジェクトについ

てのみのデータ解析にとどまっていた。本研究では、データ収集・分析ツールの充実を図り、より多くのデータをより多くのプロジェクトから収集して分析することで、それまでに得られた研究結果(傾向)の一般性を確認することが第一の目的である。

(3) 次に、コメント文以外の特徴についてもデータ収集と分析を行い、なぜコメントが多いとバグを生みやすいのかを検討していく。具体的には、プログラムの規模や複雑さといった特徴量との関係分析を行い、どういった特徴のプログラムではコメントが書かれやすく、それゆえにバグを生みやすかったのかを検討していく。

(4) さらには、プログラムの開発・変更履歴にも着目し、開発プロセスの中(インプロセス)で本研究成果を有効利用できるかどうかを確認していく。具体的には、開発初期の時点でバグの潜在が疑われるプログラムをコメント文に着目することで絞り込むことができるかどうかを分析していく。

## 3. 研究の方法

(1) オープンソースソフトウェア開発プロジェクトが公開しているリポジトリについて、そのコピー(クローン)を作成する。そして、最新版のソースコードに含まれているメソッド(関数)の一覧を作成し、データベースへ登録する。

(2) 各ソースファイル及びメソッドについて、その改変履歴をリポジトリから調べ、データベースに登録する。リポジトリが標準的に提供している機能では、ソースファイル単位での改変履歴情報しか得られないため、メソッド単位での情報収集に関しては、ソースファイルに対して変更が施されるたびにその変更箇所と注目メソッドの位置情報を突き合わせ、当該メソッドが変更されたかどうかを判定する。そのような判定を繰り返して、メソッドごとの変更履歴を得ていく。

(3) 各メソッドの各バージョンについて、そのソースコードをリポジトリから抽出し、メトリクス値の測定を行ってデータベースへ格納する。測定に用いるメトリクスは代表的な規模メトリクスである LOC(コード行数)、代表的な複雑さメトリクスであるサイクロマティック数と平均ネストレベル、メソッド宣言の直前に書かれたドキュメンテーションコメントの行数、及びメソッドの中に書かれたインナーコメントの行数である。これらの測定にはそれぞれ専用のツールを開発して使用する。

(4) 各メソッドの変更時にリポジトリへ登録されているコミットメッセージ(開発者による変更内容の説明やメモ)を抽出し、そこに

バグ修正に関するキーワードが含まれているかどうかを判定する。それによって、そのメソッドにバグが含まれていたかどうかを判断し、バグ潜在の有無をデータベースへ記録する。

(5) データベースに記録された測定データを元に統計解析を行い、メソッドにおけるコメント記述の量や有無とバグ潜在性の関係を分析する。使用する統計手法としては、比率の差の検定や相関分析、ロジスティック回帰分析が挙げられる。

#### 4. 研究成果

(1) 四つの著名なオープンソースソフトウェア (Eclipse Checkstyle Plug-in, Hibernate ORM, PMD, SQuirreL SQL Client) について、メソッドごとにドキュメンテーションコメントとインナーコメントの行数を調べた結果、コメントを持たないメソッドが多数派であることがわかった (図1, 2)。

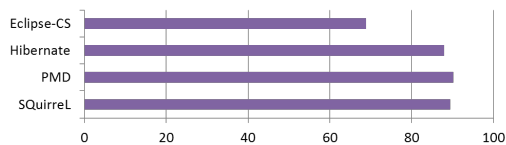


図1 インナーコメントを持たないメソッドの割合

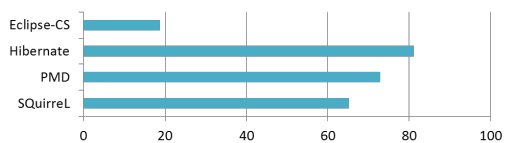


図2 ドキュメンテーションコメントを持たないメソッドの割合

この結果から、実際に広く運用されているオープンソースソフトウェアでは、必ずしもコメントが書かれているわけではないという傾向があることが確認された。それだけに、コメントが書いてあるかどうかはソースコードを分析する上で一種の特徴になることが分かった。その上で、バグ潜在の有無について調べると、インナーコメントの有無はバグ潜在の有無に関係しており、インナーコメントが書かれているメソッドはそうでないメソッドに比べてバグ潜在の可能性が 1.8 ~ 3.0 倍も高いことが確認された (図3)。

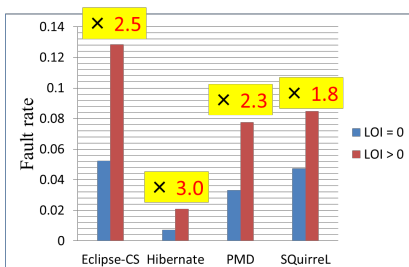


図3 インナーコメントの有無とバグ潜在率

一方、ドキュメンテーションコメントの有無に関してはそのような傾向は見られなかった (図4)。

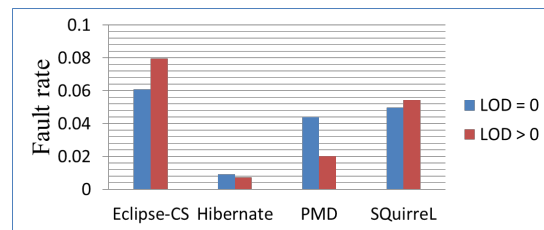


図4 ドキュメンテーションコメントの有無とバグ潜在率

このことから、インナーコメントはバグ潜在の疑われるメソッドの早期検出に役立つことが期待されること、ドキュメンテーションコメントにはそのような効果は期待できないことがそれぞれ分かった。なお、前者におけるバグ潜在率の差は統計的検定でも確認できている。

つまり、コメントの種類によっては、その存在が品質の低いものと何らかの関連性を持っていることが確認された。

(2) インナーコメントの存在がバグ潜在性と関係していることは(1)で報告した通りであるが、この傾向が、単に規模が大きかったり複雑過ぎたりしたために起こっていた可能性も否定できない。そこで、規模メトリクスや複雑さメトリクスとの相関関係を調べ、それをもとに「規模や複雑さの割にコメントが多い」メソッドとそうでないメソッドとに分けた比較も行った。具体的には、インナーコメント行数を目的変数、規模 (LOC) と複雑さ (サイクロマティック数) を説明変数とした回帰モデルを用意し、説明変数から想定されるコメント行数を計算して、実際のコメント行数が想定値よりも大きいかどうかで分類した。なお、複雑さメトリクスとしては平均ネストレベルも用意したが、サイクロマティック数との相関が強かったためにモデルには使用しなかった。分析の結果、やはりコメントの多いメソッドの方がバグ潜在性は高いことが確認された (図5)。

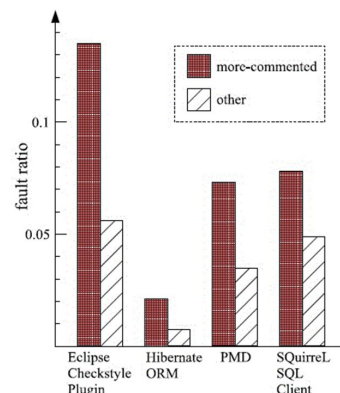
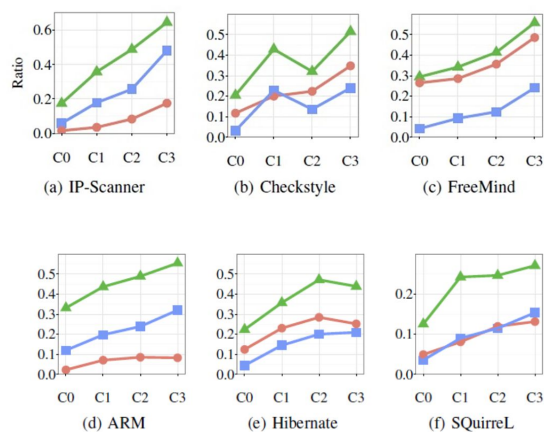


図5 コメント多寡とバグ潜在率

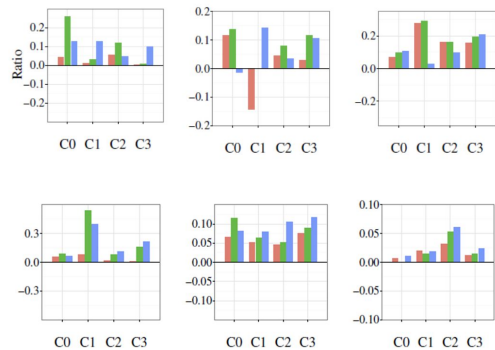
(3) インナーコメントの存在またはその行数がバグ潜在メソッドの予測にも使えるかどうかを分析するため、インナーコメント、LOC 及びサイクロマティック数の三つを説明変数としたロジスティック回帰モデルを考え、説明変数として可能なすべての組合せについてモデル評価を行った。具体的には、インナーコメントについては、(a)行数を用いた場合、(b)行数を対数変換して用いた場合、(c)インナーコメントの有無のみに着目した場合、(d)使用しない場合の4種類を考え、LOC とサイクロマティック数についてはそれぞれ(A)そのまま用いた場合、(B)対数変換した場合、(C)使用しない場合の3種類を考えた。そしてすべての組合せ35通り(「三つとも使用しない」を除く)について実データに対するモデル構築を行い、その当てはまりの良さを AIC (赤池情報量規準)でもって評価した。その結果、いずれのソフトウェアにおいてもインナーコメント行数を説明変数として使用したモデルが最良となり、代表的な規模メトリクスである LOC と代表的な複雑さメトリクスであるサイクロマティック数の二つと一緒に使用しても、存在意義のある説明変数であることが確認された。

(4) さらなる分析のため、メソッド内で使われているローカル変数の名前にも着目した分析を行った。具体的には、変数名の長さ(文字数)でもってメソッドを四つの群に分類し、変更の起こりやすさやバグ修正の起こりやすさを群間で比較した。さらにはインナーコメントの有無で各群を二分した比較も行った。六つのオープンソースソフトウェアを対象とした分析の結果、使用されるローカル変数の名前が長いほど変更やバグ潜在の可能性が高まることが確認でき(図6)、その際にインナーコメントの存在は変更やバグ潜在の可能性をより強める傾向にあることも分かった(図7)。



(緑：1回以上変更，青：2回以上変更，赤：バグ修正)

図6 ローカル変数名の長さで分類したバグ潜在率  
(C0:変数なし, C1:1-2文字, C2:3-9文字,  
C3:10文字以上)



(緑：1回以上変更，青：2回以上変更，赤：バグ修正)

図7 図6でのバグ潜在率をインナーコメントの有無で分け、有りの場合から無しの場合を引いた値

## 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文](計 1 件)

Hirohisa Aman, Sousuke Amasaki, Takashi Sasaki and Minoru Kawahara, Lines of Comments as a Noteworthy Metric for Analyzing Fault-Proneness in Methods, IEICE Trans. Inf. & Syst., 査読有, E98-D(12), 2015, 2218-2228 DOI: 10.1587/transinf.2015EDP7107

[学会発表](計 13 件)

Hirohisa Aman, Sousuke Amasaki, Takashi Sasaki and Minoru Kawahara, Empirical Analysis of Change-Proneness in Methods Having Local Variables with Long Names and Comments, The 9th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM2015), 2015年10月22日, 北京市(中国)。

Hirohisa Aman, Sousuke Amasaki, Takashi Sasaki and Minoru Kawahara, Empirical Analysis of Fault-proneness in Methods by Focusing on their Comment Lines, The 2nd International Workshop on Quantitative Approaches to Software Quality (QuASoQ2014), 2014年12月1日, 濟州特別自治道(韓国)。

Hirohisa Aman, Sousuke Amasaki, Takashi Sasaki and Minoru Kawahara, Empirical analysis of comments and fault-proneness in methods: can comments point to faulty methods?, The 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM2014), 2014年9月18日, トリノ市(イタリア)。

阿萬裕久，コメントの記述位置の違いに着目したフォールト潜在性の分析 メソッドの前に書かれるコメントとメソッドの中に書かれるコメント，ソフトウェアエンジニアリングシンポジウム 2013，2013 年 9 月 11 日，東洋大学（東京都文京区）。

阿萬裕久，小規模プログラムにおけるコメント行数とフォールト潜在性の関係に関する調査，電子情報通信学会ソフトウェアサイエンス研究会，2013 年 5 月 10 日，香川大学（香川県高松市）。

〔図書〕（計 1 件）

藤井里沙，天寄聡介，阿萬裕久，横川智教 著，近代科学社，ソフトウェア工学の基礎 2014 年，6 頁(69-74)。

〔その他〕

ホームページ（開発したツールを公開）

<http://se.cite.ehime-u.ac.jp/tool/>

## 6．研究組織

### (1)研究代表者

阿萬 裕久（AMAN, Hirohisa）

愛媛大学・総合情報メディアセンター・准教授

研究者番号：50333513

### (2)研究分担者

天寄 聡介（AMASAKI, Sousuke）

岡山県立大学・情報工学部・助教

研究者番号：00434978

平田 孝志（HIRATA, Kouji）

関西大学・工学部・准教授

研究者番号：10510472