

複数組織によるソフトウェア開発において 品質を低下させるソースコードの特徴の調査

石塚凌[†] 津田直彦[†] 鷺崎弘宜[†] 深澤良彰[†] 杉村俊輔[‡] 保田裕一朗[‡]
早稲田大学[†] 小松製作所[‡]

1. はじめに

ソフトウェア開発において、開発途中での開発組織の変更、外部への開発の委託等によって、複数の組織がソフトウェア開発に携わる事がある。しかし、複数の組織によって開発されたソフトウェアの品質は一般に低いと言われており、ソースコードの改修が難しくなる等の問題が生じている。また、開発サイクルの高速化が進み、ソースコードの改修やバグチェックに時間を掛けられなくなって来ている。そこで我々はそのようなソフトウェアの品質低下を防ぐため、複数の組織で開発されたソフトウェアにおいて、欠陥を含みやすいソースコードの特徴や、開発組織間のソースコードの編集傾向の違いを示す。

2. 既存研究

複数の組織が開発したソフトウェアについての品質への影響を研究する試みはこれまでも行われている。阿部ら[1]は委託先変更のあった社内用のソフトウェアにおいて、委託先変更される前の時点で Cyclomatic 複雑度(CC)の値が高かったメソッドが委託先変更後の開発によって更にCCが上昇する傾向にある事を発見した。次に、Satoら[2]はOpenOffice.orgとVirtual Boxの編集履歴と修正された欠陥の履歴を追跡し、複数の組織が開発しているソースコードに欠陥が生じる可能性が高い事を突き止めた。

また、Palombaら[3]は3つOSSにおいて開発者の設計・実装上のコードスメルを感じ方について実証研究を行い、スパゲッティコードや長いメソッドのようなコードスメルは他のコードスメルと比べ開発者に強く認知される事を確認した。

3. 研究概要

3.1. 既存研究からの発展

阿部ら[1]の研究では着目したメトリクスをCCのみに絞っているが、我々はより正確にソフトウェアの変化を捉えるため、ソフトウェア構造の複雑度を示すメトリクス[4]やコードクローンといった複数のメトリクスから品質評価を行う。また、

Investigating the characteristic of source code have worse quality at software development by multiple organizations

[†]Ryo Ishizuka, Naohiko Tsuda, Hironori Washizaki, Yoshiaki Fukazawa, Waseda University

[‡]Shunsuke Sugimura, Yuichiro Yasuda, Komatsu Ltd.

[2], [3]において解析したソフトウェアのリリース数はそれぞれ3, 1と少ないため、欠陥修正といったソフトウェアの変化を捉える事ができてないと考えられる。これらの研究を踏まえ、本研究では2つのリサーチクエストンに取り組む。

RQ1. ソフトウェアメトリクスや欠陥の発生件数の変化が開発組織・リリース間で差がみられるか？欠陥を含みやすいソースコードの特徴は？

RQ2. ソースコードの改修・リファクタリング等の編集傾向の違いはコードの品質に差は出るか？

RQ2 解決のために我々はコードレビューを行い、各開発組織のソフトウェアの改修やリファクタリングの追跡を行っている。

3.2. 解析対象のソフトウェア

解析対象は外部委託によって開発された社内用ツールと、OSSのOpenOffice.org(00o)とその派生ソフトウェアであるApache OpenOffice¹(A00)とLibreOffice²(Lib0)を選択した。

社内用ツールは最初2社によって開発されていたが、保守性向上のために開発途中から1社で開発を行ったソフトウェアである。

00oは10年間Sun Microsystem社によって開発されていたが、現在は00oの開発者が設立したThe Document FoundationがLib0を、Oracleからプロジェクトの寄贈を受けたApacheソフトウェア財団がA00をそれぞれ開発している。

表1. 解析したソフトウェア

派生元 ソフトウェア	派生先 ソフトウェア	開発言語	#R1s	
			元	先
社内用ツールA	社内用ツール A+B	C#	2	2
社内用ツールB			2	
00o	Lib0	C++/Java	2	13
	A00			9

3.3. 測定手法について

各ソフトウェアのソフトウェアメトリクスの測定には静的解析ツールUnderstand³とコードクローン検出ツールNicad Clone Detector⁴, PMD-CPD⁵を用いた。OpenOfficeの欠陥の発生件数の測定にはszzアルゴリズム[5]を用い、コミットログの中から修正に関するコミットを特定している。

1. <http://www.openoffice.org/>

2. <https://www.libreoffice.org/>

3. <https://www.techmatrix.co.jp/product/understand/index.html>

4. <https://www.txl.ca/nicaddownload.html>

5. <http://pmd.sourceforge.net/pmd-4.3.0/cpd.html>

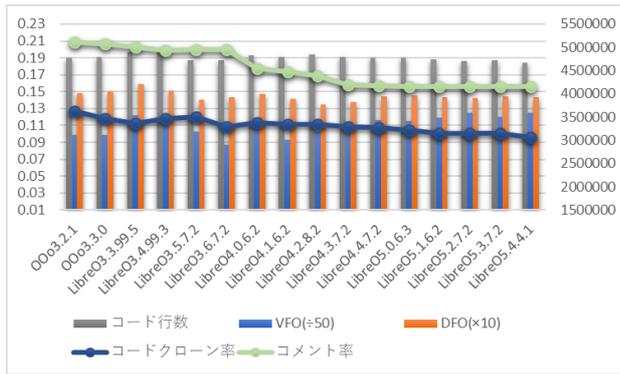


図 1. Lib0 のメトリクス測定値推移

また、ソースコードの編集傾向を把握するため、各ソフトウェアで不具合や修正の多かったファイル等を対象として、最初のリリースからソースコードがどう改修されているかソースコード、コミットログ等から追跡を行った。

4. 解析結果と評価

本研究で解析したソフトウェアに共通していた傾向として、他の多くのファイルに依存している([4]において Control, Core に分類される)ファイルに発生した欠陥のほとんどが含まれる事が判明した。また、Lib0, A00 の各最終リリースにおいても共通して約 80%のファイルが全体のうち約 40%のファイルと直接または間接的な依存関係がある事が分かった。今後は依存関係に起因する修正漏れ等が欠陥の原因にどの程度含まれていたかコミットログの分析を進める予定である。

次に、LibreOffice においてはリリースが進むにつれて図 1 のようにコード行数やコメント率、コードクローン率が減少している一方で、ソフトウェアの構造の複雑さを示す間接ファンアウト値(VFO)が増加傾向にある事が確認できた。コードレビューにおいて不要なコメント削除やメソッドの抽出、C++の新機能であった型推測やループ文を範囲ベースに書き直すといった修正が段階的になされている事が確認でき、その結果がソフトウェアメトリクスにも表れていると考えられるが、4.x 系までの改修でソフトウェア設計の見直しが進み、現在では機能追加による依存関係の複雑化が進んでいると考えられる。

対照的に、Apache OpenOffice では最初のリリースにおいて不要となったモジュールが削除されておらず、図 2 に示すように、OpenOffice の最終リリースと比べて多くのソフトウェアメトリクス値が増加していた。

また、社内用ツールにおいて、委託先変更後の開発において CC が増加したファイルは委託先変更後に多くの機能追加が行われ、その際に際にメ

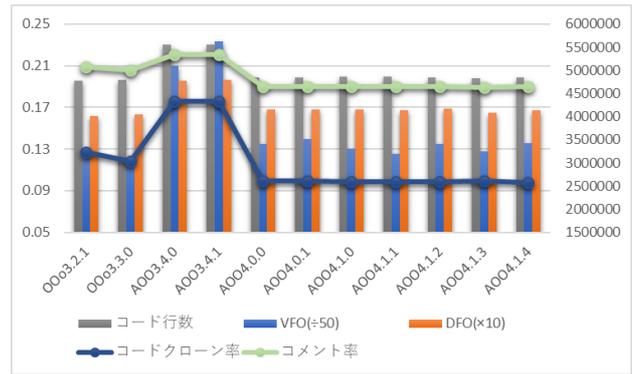


図 2. A00 のメトリクス測定値推移

ソッド抽出を行わず、元々コード行数の多かったメソッドに機能を追加したファイルがいくつかあった事が判明した。そのようなファイルは機能追加による不具合の修正回数も多かったため、複数組織によるソフトウェア開発で大きなメソッドを作るのは極力避けるべきだと考えられる。

5. おわりに

本研究はソフトウェアの品質を低下させる要因を特定するために、ソフトウェアメトリクスやコードレビューから分析を行った。今後としては、まず解析対象のソフトウェアを拡充してどのような傾向が得られるか確認する必要がある。そして今回の結果を踏まえ、実際の開発を支援するためのツール作りやメトリクスの考案をしたいと考えている。

6. 参考文献

- [1] 阿部晃佑, et al.: 開発委託先変更に対するソフトウェアの複雑さ・不具合修正回数の関係, 情報処理学会第 195 回ソフトウェア工学研究発表会, 2017.
- [2] Seji Sato, et al.: Effects of Organizational Changes on Product Metrics and Defects, Proceedings of the 20th Asia Pacific Software Engineering Conference (APSEC 2013), pp.132-139, 2013.
- [3] F. Palomba, et al.: Do they Really Smell Bad? A Study on Developers' Perception of Code Bad Smells, Proceedings of the IEEE international conference on Software maintenance and evolution (ICSME'14), pp.101-110, 2014.
- [4] Robert Lagerström, et al.: Exploring the Relationship Between Architecture Coupling and Software Vulnerabilities, Engineering Secure Software and Systems, 2017.
- [5] Jacek Sliwerski, et al.: When do changes induce fixes?, In Proc. the 2nd International Workshop on Mining Software Repositories (MSR 2005), pp.1-5, 2005.