欠陥とソースコードの変更回数の関係分析

本田 澄[†1] 坂口 英司[†2] 伊原 彰紀[†2]
鷲崎 弘宜[†1] 深澤 良彰[†1]

企業はそれぞれの目的のためにオープンソースソフトウェア (OSS) を利用するが，OSS の採用する基準は不明瞭で属人的である．我々は，OSS にバグが少なければ，採用しやすいという仮定にしたがい，ファイルやコードが編集されるときにバグが作りこまれるという考えをもとに，我々はバグの数と編集の回数の関係を分析する．

Analysis of Relation between Faults and Number of Source Code Changes

Kiyoshi Honda,[†1] Hideshi Sakaguchi,[†2] Akinori Ihara,[†2]
Hironori Washizaki[†1] and Yoshiaki Fukazawa [†1]

Companies employ Open Source Software (OSS) for their purpose, however evaluation criteria to employ OSS are unclear and depending on developers or managers. We assume that if an OSS has less bugs, it would be an employable OSS. Based on the idea that bugs are created when files or codes are edited, we analyze the relation between the number of bugs and the number of edited codes.

1. はじめに

複数の企業がオープンソースソフトウェア (OSS) を，その企業の目的に応じて利用されている．しかし，OSS を利用する判断基準は不明瞭であり，開発者やマネージャに依存している．特に OSS の多くは，企業に属するものではなくそれぞれの有志の集まりであるコミュニティーによって開発されるものである．それぞれのコミュニティーでは独自の判断でリリースを行ない，バグを含んだ状態でリリースされるものも多く，利用者は OSS の利用に当たり，その成熟度やバグの修正状況を確認する必要がある．

そのため，OSS を企業で利用する際には様々な判断基準が必要となるが，企業や開発者に依存しており，研究があまりされていない．Mileva らは，OSS におけるライブラリについて，それぞれのバージョンの利用者が多いものを利用することを推薦する研究を行っている.[1)] しかし，本研究では，利用者の数にフォーカスするのではなく，OSS そのものの開発履歴にフォーカスし，バグの発生状況とコードの変更回数との関係を分析する．これにより，多くの OSS が持つ開発履歴を用いて，バグが今後発生するかどうかを分析でき，共通の機能を持つ OSS においてどちらの OSS の方が安定した OSS かを判断できる基準を与える．

†1 早稲田大学
Waseda University
†2 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

2. 背景

有用な OSS が新たに開発されている一方，開発が滞るものや，利用者が離れていく OSS も少なくはない．そういった OSS はゾンビ OSS[2)] と呼ばれている．参考文献 2) では，ゾンビ化について，1) 新機能が他のソフトでも実現可能になることで、その OSS の存在意義が無くなる (Struts 1, Seasar2)，2) 新機能を実現する OSS が乱立した結果、競争に敗北 (Accumulo) 3) 新機能を追求するあまり旧バージョンのサポートがおろそか (OpenStack) といった種類があることを言及している．利用して良いと考えられる，バグが少ない OSS の判断基準として，我々はソフトウェア信頼性モデル (SRGM) を用いて十分にバグを発見できているかを考える．SRGM は様々なドメインで利用され，また様々な利用方法が提案されている車メーカーでの利用や[3)]，OSS のコード変更回数と時間の関係を用いた SRGM の提案や[4)]，web サービスの開発での利用[5)] などがあげられる．

3. 評価方法

本研究では，オープンソースソフトウェアの開発履歴を含んだリポジトリを提供している GitHub[6)] から

開発履歴を取得する．対象とした OSS は Xstream[7] といわれる，Java オブジェクトを XML へシリアライズする OSS ライブラリである．評価方法として，開発履歴を含むリポジトリから，コードの変更回数を時系列で取得する．また，バグ情報についてもリポジトリから時系列で取得する．それらの結果をそれぞれグラフに示したのちに，時系列の情報をもとに，コードの変更回数とバグ情報を関連付ける．

4. 評　　価

図 1 に Xstream の欠陥の発見数と修正数と時間の関係を示す．図 2 に Xstream のファイルの変更回数を示す．図 1 が示すように欠陥の発見数と修正は徐々に減少してる．しかし，図 2 が示すように，後半に多くの変更が行われていることがわかる．図 3 に Xstream の欠陥の発見数と変更回数の関係を示す．図 4 に Xstream の欠陥の修正と変更回数の関係を示す．図 3 と 4 が示すように欠陥の発見と修正は，変更回数が増えるにつれて少なくなっていることがわかる．

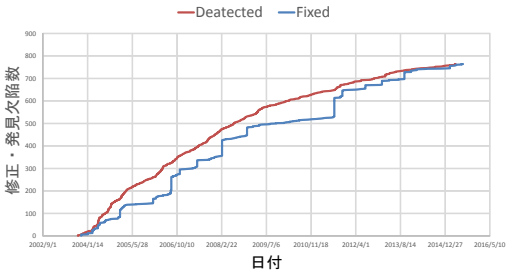

図 1　欠陥の修正数と発見数と時間の関係

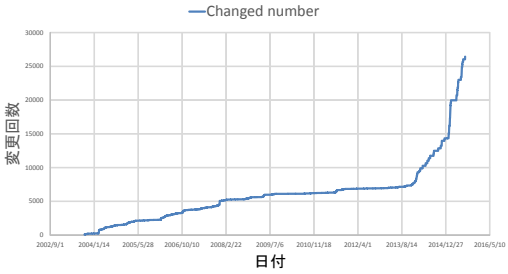

図 2　変更回数と時間の関係

5. 結　　論

2014 年はソースコード変更が頻繁に起こっているが，リリースノートを確認すると開発者が新規機能の追加やリファクタリングを行ったと考えられる．ソースコード変更回数と欠陥修正数の関係については，7500 回の変更から，欠陥修正数はあまり増加していないこ

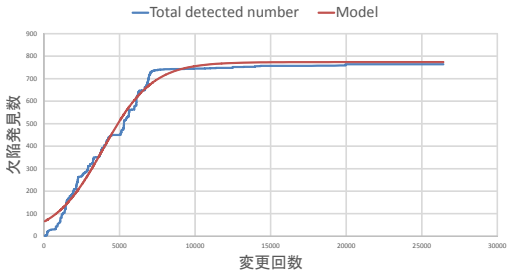

図 3　欠陥の発見数と変更回数の関係

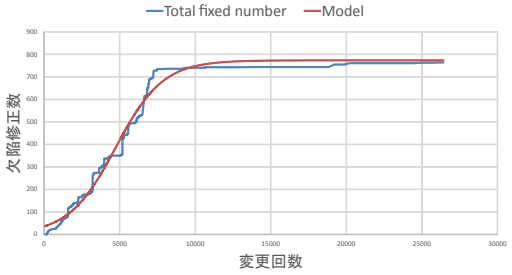

図 4　欠陥の修正数と変更回数の関係

とがわかった．また，リリースノートから主な開発は完了していると考えられる．2014 年 2 月 8 日にリリースされた XStream 1.4.7 ではセキュリティにおける対策が主であり，2015 年 2 月 18 日にリリースされた XStream 1.4.8 では Java 8 ラムダ式への対応が主であった．これにより，セキュリティ対策とラムダ式への対応が主に変更回数を増加させたと考えられる．

参　考　文　献

1) Y. Mileva, et al. “Mining Trends of Library Usage.” IWPSE-EVOL 2009.
2) ゾンビ OSS が危ない - ［2］オープンソースソフトウエアがゾンビ化する事情：IT-pro, http://itpro.nikkeibp.co.jp/atcl/column/15/031800050/031800002/
3) R. Rakesh, et al. “Evaluating long-term predictive power of standard reliability growth models on automotive systems.” ISSRE 2013
4) H. Aman, et al. “Multistage Growth Model for Code Change Events in Open Source Software Development: An Example Using Development of Nagios.” SEAA 2014
5) K. Honda, et al. “Predicting Time Range of Development Based on Generalized Software Reliability Model.” APSEC 2014
6) GitHub, https://github.com
7) Xstream, http://x-stream.github.io/