

Self-admitted technical debt の混入・除去に関する追実験

西川 諒真[†] 西中 隆志郎[†] 亀井 靖高[†] 佐藤 亮介[†] 鷗林 尚靖[†][†]九州大学

1 はじめに

技術的負債とは開発上における設計状態の理想状態からの乖離を示す現象である。技術的負債を検出するための方法として、コードの臭いやコーディング規約違反といった指標が用いられてきた。その中でもコメントを用いて技術的負債を探る方法が示されており、SATD (Self-admitted technical debt) はコメントを用いてソースコード上へ開発者自身によって言及が行われた技術的負債のことを指す。

SATD について知ることは、その負債への扱い方を知ることに繋がり、プロジェクトの開発を円滑に進めることに貢献する。SATD を用いてプログラムの状態の不備を示す開発者が多いことから、コメントに基づく SATD 研究は技術的負債の検出に有用であると考えられる。

SATD に関する実証的研究は十分でなく、特に除去に着目した研究には課題がある [4]。Maldonado ら [4] は、一部の SATD がプロジェクト内へ混入後に除去されず悪影響を与え続けていることから、新たに SATD の除去に着目した研究を行った。Camel, Gerrit, Hadoop, Log4j, Tomcat の 5 つのプロジェクトに対してデータセットを取得し、誰が SATD を除去することが多いのか、SATD の除去までにかかる期間を示した。しかし、この調査で得られた結果にはプロジェクトによる差が大きく、調査範囲についての課題点が存在する。

本研究では、新たなデータセットを使用し Maldonado らの調査の追実験を行うことで、SATD に関する実証的な観察結果の拡大を試みる。

2 関連研究

Bavota ら [1] は、混入した SATD の時間経過による増減についての研究が少なかったことから、ソフトウェアプロジェクトから約 300 のコメントを抽出して SATD

の混入についての分類を行い、頻出する SATD の種類の傾向や、プロジェクトごとに平均 50 程度の SATD が存在すること、SATD の修正が行われなければ SATD が増え、修正後も残りやすいことを示した。これに対し、本研究は SATD の除去にかかる期間と、除去を誰が行うかについてを調査する。

亀井ら [2] は SATD の除去におけるコストについての研究が広く研究されていなかったことから、混入した SATD の大きさが混入後にどう変わるかについての研究を行い、SATD の内 40 % 超が混入後に増加するということを示した。この研究では利子の計算の際、プロジェクト内に SATD が混入したときと除去されたときの二つのリビジョンを利用し、その大きさの差から除去の重要性を示している。これに対し、本研究では SATD の混入と除去のリビジョン間にある時間差の観点から、除去についての調査を行う。

3 データセット

本研究では Maldonado らの使用したプロジェクトに加え、新規にデータセットを取得する。新しく取得するデータセットに使用するプロジェクトについては SATD の検出に利用する SATD Detector [3] のサポート対象である Java 言語を用いていること、オープンソースであること、バージョン管理システムが使われていて過去のリビジョンを取ることが可能であること、一定以上の規模があることから、geogig, ccw の二つを使用している。詳細を表 1 に示す。

4 実験方法と初期結果

本研究の目標は Maldonado らの研究の追実験を行い、結果の一般性を向上することである。そのため、本研究では以下の調査を行う。

4.1 (RQ1) SATD が混入してから除去されるまでどれほどの期間があるか

動機 : 一つ目の RQ は SATD の存在期間についてである。本 RQ に対する答えは、SATD が一般的にプロジェクトにどれほどの期間存在するかを知ることに役立つ。

A Replication Study on the Introduction and Removal of Self-Admitted Technical Debt

Nishikawa Ryoma[†], Nishinaka Ryujiro[†], Kamei Yasutaka[†], Sato Ryosuke[†], and Ubayashi Naoyasu[†][†]Kyushu University

{nishikawa, nishinaka}@posl.ait.kyushu-u.ac.jp

{kamei,sato,ubayashi}@ait.kyushu-u.ac.jp

表 1: 現時点で使用中的数据セット

Project	# Java files	# commits	# cont.	# SATD comments	# of removed
geogig	996	2,291	20	69	7
ccw	96	1,193	21	57	15

cont. = contributors

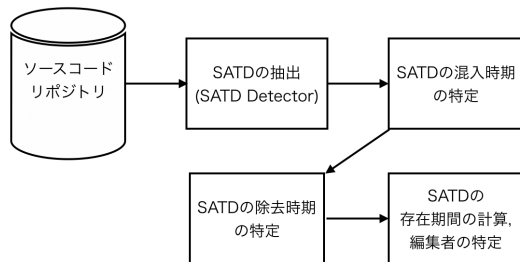


図 1: アプローチの概略

アプローチ: 図 1 にアプローチの概略を示す. SATD の検出には SATD Detector を利用した. SATD Detector はソースコード中に存在する SATD を自動検出し, 出力に SATD と, その SATD が存在するパス, ファイル名, コード内での行数を表示するツールである.

本ツールによりプロジェクトの特定のリリースに存在する SATD を検出した後, バージョン管理システムのアノテーション機能を利用することで, SATD の存在する行に対していつ最終変更があったかを見ることにより SATD の混入時期を特定する. その後, 同様にアノテーション機能を使い, SATD の存在する行がいつまで存在したかを見て, その直後の変更を見ることで SATD の除去時期を特定した後, その時間差を取ることで SATD の存在期間を計算する.

結果: 結果を表 2 に示す. 中央値は順に 760.18, 1312.74 日となっている. Maldonado らの研究結果はプロジェクトごとの中央値で 18.2~172.8 日であるため, 比較すると極めて大きな値となった.

4.2 (RQ2) 誰が SATD の除去を行っているか

動機: 二つ目の RQ は SATD の除去を行う人物についてである. 本 RQ の結果から, SATD をどのように扱うべきかを知ることができる. 例えば, 混入者以外によって SATD が除去されることが多いならば SATD はプロジェクト全体で共有, 把握される必要があるものといえ, そうでないならば混入者各自によって処理されるものであるといえる.

アプローチ: アプローチは RQ1 において SATD の混入・除去の時期を特定する際, それを行った人物の名前を取得することで調査を行う.

結果: 結果を表 3 に示す. 混入者自身による SATD の除去の割合は平均 66.2%となった. Maldonado らの研究結果は平均 54.4%であるため, 比較すると高めの値となった.

表 2: RQ1 初期結果

Project	Min.	1st quartile	Median	3rd quartile	Max.
geogig	92.12	760.18	760.18	921.84	1023.4
ccw	362.77	766.8	1312.74	1515.54	2015.68

表 3: RQ2 初期結果

Project	# Removed	# Self-removed	% Self-removed
geogig	7	6	85.7
ccw	15	7	46.7
Average	-	-	66.2
Median	-	-	66.2

5 現状と今後の予定

現時点ではプロジェクト中の 1 つのリリースから SATD を検出し, その混入時期, 除去時期を特定することで調査を行なっているが, そのリリース内には SATD は検出できない. そのため, 混入から除去までの時間が短い SATD は検出しづらく, 逆に混入から除去までの時間が長い SATD は検出しやすい, という結果の偏りが生じている.

この問題を解決するためには別のリリースを利用して SATD の除去時期を特定する必要があるが, 全てのリリースを網羅するのは現実的な手段ではないため, アプローチの改善を試みている. また, 現時点ではプロジェクトを一定の条件下で選択しているが, 無作為な選択ができておらず恣意的な要素が含まれるため, リポジトリの取得手法を改善することも考えている.

今後の研究では追加データセットの取得や, Maldonado らの調査に沿った, SATD の除去までの期間や誰がその除去を行ったかの調査に限らず, それらとは異なる調査から SATD の研究の幅を広げようと考えている.

謝辞

本研究は, JP18H03222 による助成を受けた.

参考文献

- [1] Bavota, G. and Russo, B.: A large-scale empirical study on self-admitted technical debt, *Mining Software Repositories (MSR), 2016 IEEE/ACM 13th Working Conference on*, IEEE, pp. 315–326 (2016).
- [2] Kamei, Y., Maldonado, E. d. S., Shihab, E. and Ubayashi, N.: Using Analytics to Quantify Interest of Self-Admitted Technical Debt., *QuASoQ/TDA@ APSEC*, pp. 68–71 (2016).
- [3] Liu, Z., Huang, Q., Xia, X., Shihab, E., Lo, D. and Li, S.: SATD detector: a text-mining-based self-admitted technical debt detection tool, *Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings*, ACM, pp. 9–12 (2018).
- [4] Maldonado, E. d. S., Abdalkareem, R., Shihab, E. and Serebrenik, A.: An empirical study on the removal of self-admitted technical debt, *Software Maintenance and Evolution (IC-SME), 2017 IEEE International Conference on*, IEEE, pp. 238–248 (2017).