

開発者が管理する技術的負債 に関する課題票の内容理解

オープンソースソフトウェア工学研究室

学籍番号：60219001

氏名：木村 祐太

技術的負債：Technical Debt^[1]

時間的制約によって導入される低品質な設計や実装

- 例) コードスメル：深刻な問題を抱えているコードの特徴

保守コスト増加を引き起こす^{[3][4][5][6]}

- 開発の約23～35%が余分な作業
- 既存のTDによる新たなTDの導入（TDの蓄積）

開発資源（人的，金銭的，時間的）は有限
TDの蓄積量を管理する必要がある

[1] Ward Cunningham. The wycash portfolio management system. SIGPLAN OOPS Mess., 4(2):29–30, 1992.

[3] Carolyn Seaman, Yuepu Guo, Nico Zazworka, Forrest Shull, Clemente Izurieta, Yuanfang Cai, and Antonio Vetr`o. Using technical debt data in decision making: Potential decision approaches. In Proceedings of the 3rd International Workshop on Managing Technical Debt, MTD, pages 45–48, 2012.

[4] Edith Tom, Ayb uKe Aurum, and Richard Vidgen. An exploration of technical debt. 86(6):1498–1516, 2013.

[5] Terese Besker, Antonio Martini, and Jan Bosch. The pricey bill of technical debt: When and by whom will it be paid? In Proceedings of the 33rd International Conference on Software Maintenance and Evolution, ICSME, pages 13–23, 2017.

[6] Terese Besker, Antonio Martini, and Jan Bosch. Technical debt cripples software developer productivity: A longitudinal study on developers 'daily software development work. In Proceedings of the 1st International Conference on Technical Debt, TechDebt, pages 105–114, 2018.

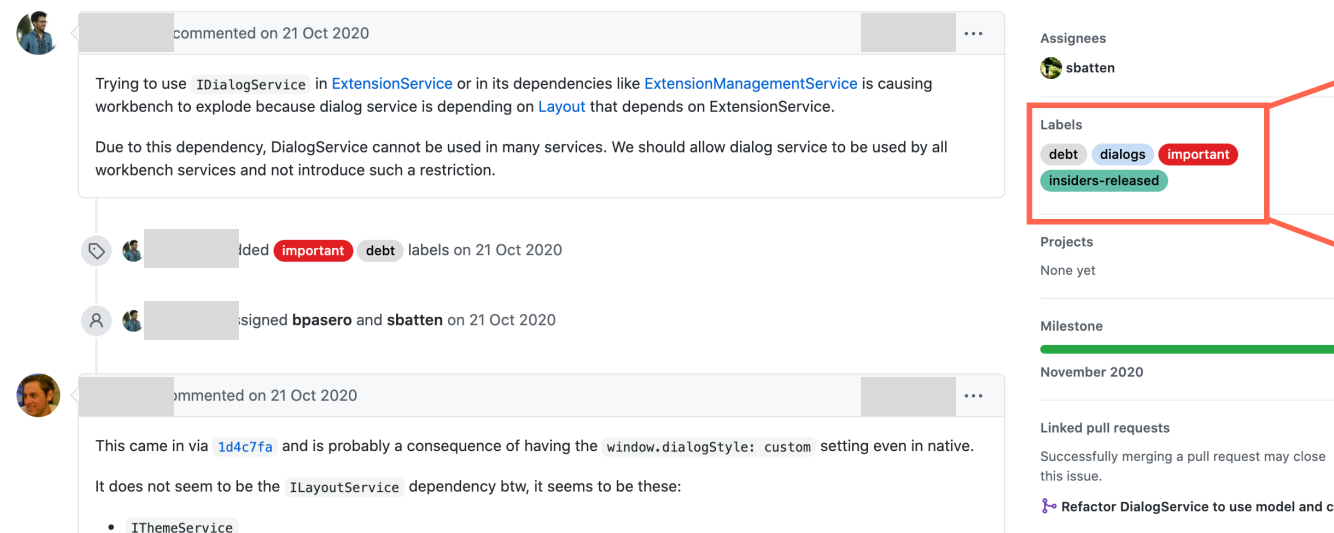
Self-Admitted Technical Debt

開発者によってTDが明示的に管理される事象

- SATD-Issue ^[7]
 - 課題票 (Issue) にTDラベルを割り当てて明示されるSATD
- ※ Issue : 不具合, 機能改善, 機能要求など様々な課題が管理される

Cannot use IDialogService in some services due to custom dialog service dependencies
#109056

 Closed sandy081 opened this issue on 21 Oct 2020 · 13 comments



“debt”というTDラベルが
割り当てられている

引用 : <https://github.com/microsoft/vscode/issues/109056>

[7] Laerte Xavier, Fabio Ferreira, Rodrigo Brito, and Marco Tulio Valente. Beyond the code: Mining self-admitted technical debt in issue tracker systems. In Proceedings of the 17th International Conference on Mining Software Repositories, MSR, pages 137–146, 2020.

研究課題

開発者が管理する技術的負債とは何か？

- 開発者がどのような内容をTDとして管理しているのか調査されていないため研究での支援が困難

課題に取り組むことで得られる貢献

- 既存のTD研究で扱われる内容との違い
 - 既存研究で網羅できている範囲と網羅できていない範囲の明確化
- 開発を支援するために必要なツールの理解
 - 開発者が扱うTDの管理（検出, 返済）を容易にするための支援

研究目的と事前調査

目的

- SATD-Issueで扱われている内容を理解する

事前調査

- コメントとTDラベルが付けられるタイミングの調査

動機

- 開発者はあらかじめTDを理解した状態で報告するのか，議論のなかでTDを理解するのかを明らかにしたい
- TDと判断するための情報の所在を理解する

方法

- コメントを含むSATD-Issueの割合を計算
- TDラベルが付けられる前にコメントを含むSATD-Issueの割合を計算

事前調査：データセット

リポジトリ	ラベル	SATD-Issue [件]
microsoft/vscode	debt	2,095
influxdata/influxdb	kind/tech det	120
saleor/saleor	technical debt	109
nextcloud/server	technical debt	91
合計		2,415

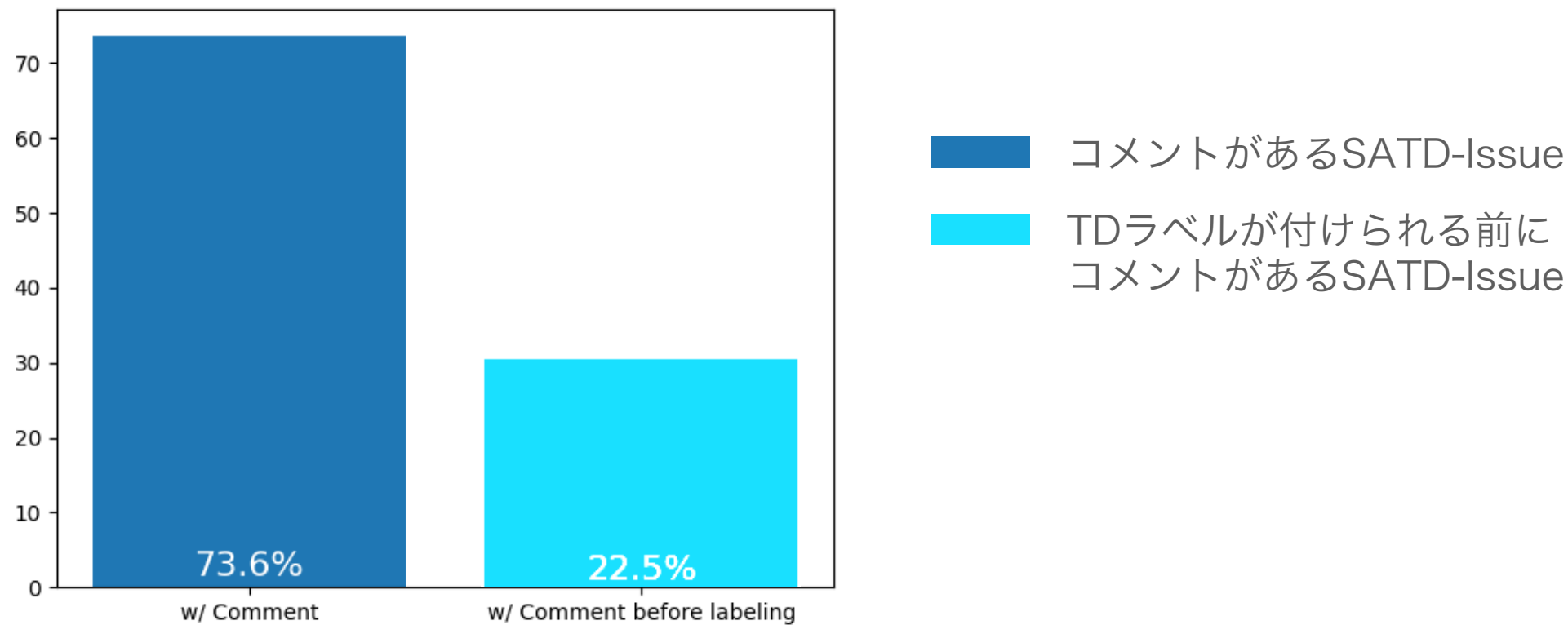
4つのOSSプロジェクトのデータを利用

- GitHub（開発プラットフォーム）にホスティングされている
- SATD-Issueを慣習的に扱っている

Closed Issue（解決済みIssue）のみを対象

- 報告されてからしばらく時間が経過して、Issue内容を表すラベルが割り当てられることがある

事前調査：結果



コメントがあるSATD-Issueの割合

コメントがあるSATD-Issueは73.6%

- コメントがないSATD-Issueは26.4%

ラベルが付けられる前にコメントがあるSATD-Issueは22.5%

- TDの判断を議論でおこなうことは多くあるわけではない

考察：誰がラベルを付けるのか？

ラベルを付けた開発者が誰か調査し，コメントは参照されていそうか分析する

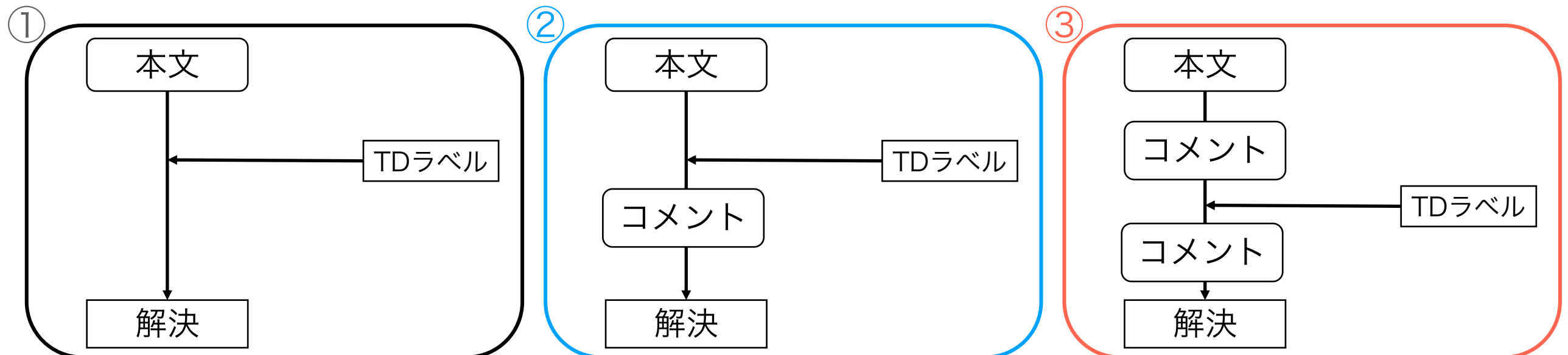
方法

- 3つのパターンで，ラベルを付けた開発者が報告者と同じであるかどうか割合を計算

① コメントがないSATD-Issue

② ラベルが付けられる前にコメントがないSATD-Issue

③ ラベルが付けられる前にコメントがあるSATD-Issue



考察：誰がラベルを付けるのか？

① コメントがないSATD-Issue

- 報告者：82.8%，その他の開発者：17.2%

② ラベルが付けられる前にコメントがないSATD-Issue

- 報告者：84.3%，その他の開発者：15.7%

③ ラベルが付けられる前にコメントをあるSATD-Issue

- 報告者：19.1%，その他の開発者：80.9%

①と②のパターンは，報告者自身がTDラベルを付ける傾向にある

③は，他の開発者によってラベルが付けられる傾向にある

コメントはTDの判断をするための情報として利用される

SATD-Issueの種類と性質の仮説

① コメントがないSATD-Issue

- 技術的負債を理解して報告している
- どのように返済するのか目処がついている

② ラベルが付けられる前にコメントがないSATD-Issue

- 技術的負債を理解して報告している
- どのように返済するのか目処がついていない

③ ラベルが付けられる前にコメントがあるSATD-Issue

- 報告の段階では技術的負債と理解していない
- どの問題なのか判断するために追加の情報を必要としている
- どのように返済するのか目処がついていない

Research Question

事前調査の結果

- SATD-Issueの運用方法が複数存在する
 - ① コメントがないSATD-Issue
 - ② ラベルが付けられる前にコメントがないSATD-Issue
 - ③ ラベルが付けられる前にコメントがあるSATD-Issue

Research Question

- RQ1：それぞれのSATD-Issueの特徴は何か？
 - 運用方法によってどんな違いが現れるかを明らかにする
- RQ2：それぞれのSATD-Issueで扱われている内容は何か？
 - 扱う内容によって運用の仕方に違いが出るのか明らかにする

RQ1：各SATD-Issueの特徴は？

内容の詳細さ， 解決されるまでの時間， コメントのラリー数について特徴の違いがあるか分析

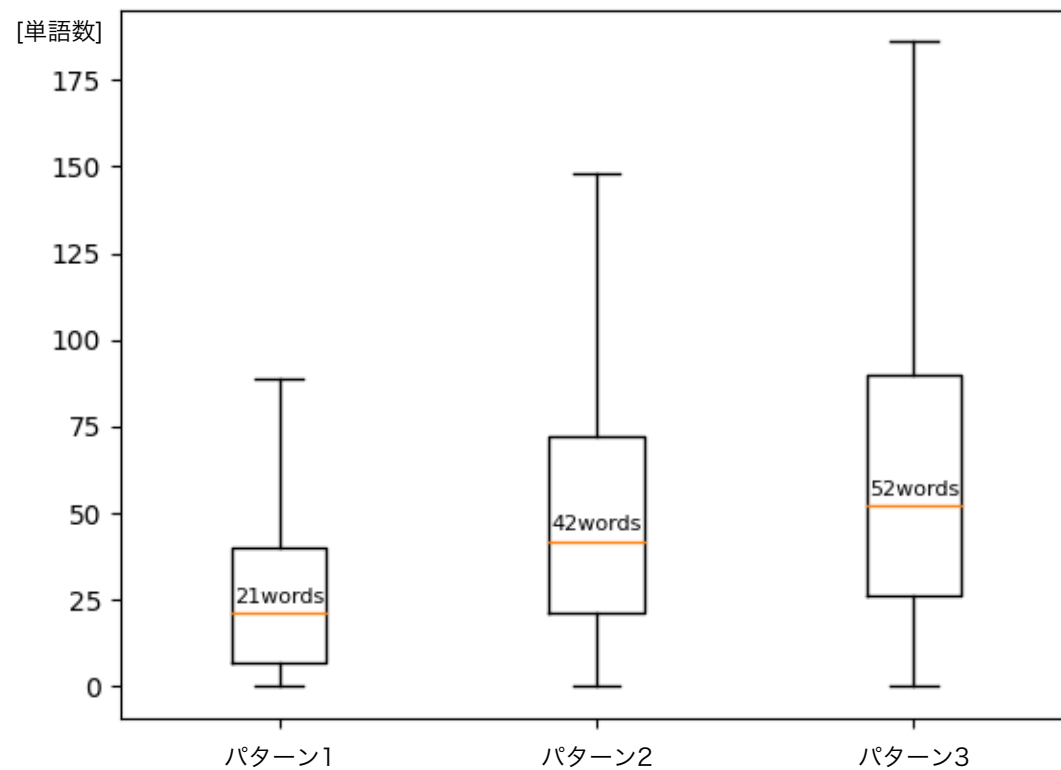
特徴	説明
内容の詳細さ	Issueの本文に記述されている単語数
解決されるまでの時間	Issueが報告されてからIssueのステータスがClosedになるまでの時間
コメントのラリー数	同じ開発者が連続しておこなったコメントを除いたコメント数

※ 単語数は，コード片とMarkdownタグを除去したのち半角スペースで分割し数えている

方法

- 箱ひげ図の描画と統計量の算出
- マンホイットニーのU検定とCliff's dを用いて分布に違いがあるか分析
 - 箱ひげ図を用いて外れ値除去

RQ1：内容の詳細さ（単語数）



- ① コメントがないSATD-Issue
 - ・ パターン1
- ② ラベルが付けられる前にコメントがないSATD-Issue
 - ・ パターン2
- ③ ラベルが付けられる前にコメントがあるSATD-Issue
 - ・ パターン3

結果

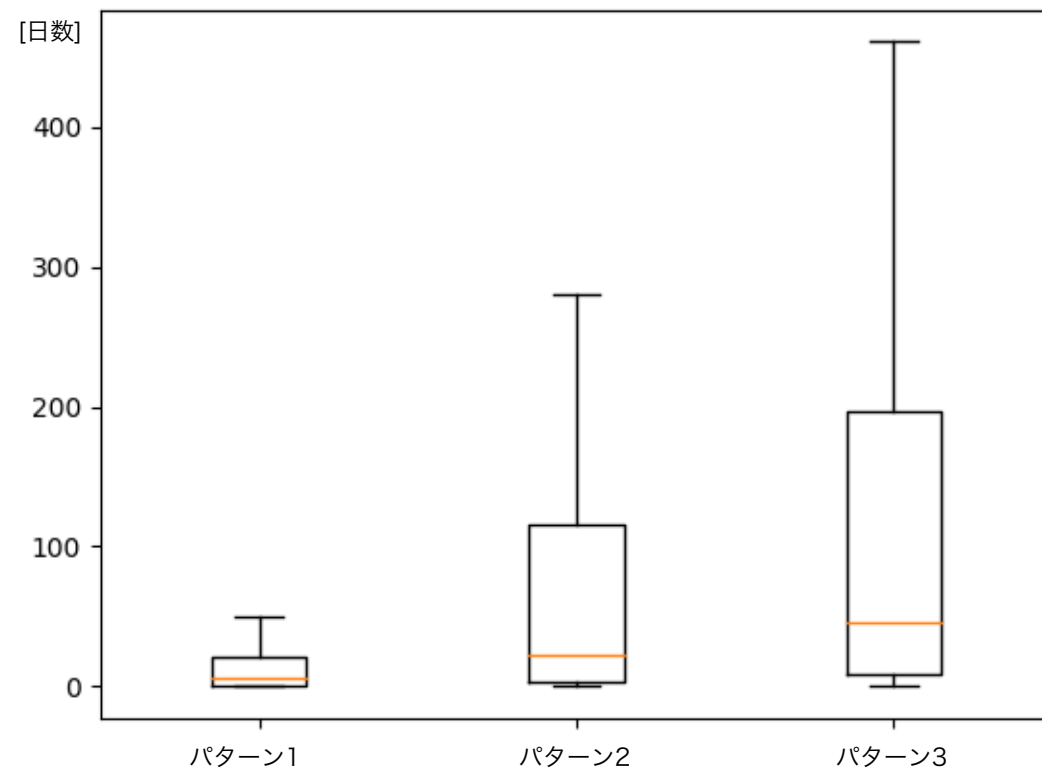
各SATD-Issueの統計量

	最小値	中央値	最大値
パターン1	0	21	89
パターン2	0	42	148
パターン3	0	52	186

ペアワイズ比較による検定結果

	有意差	効果量 d
パターン1 vs パターン2	* ($p < 0.01$)	0.385 (Medium)
パターン1 vs パターン3	* ($p < 0.01$)	0.528 (Large)
パターン2 vs パターン3	* ($p < 0.01$)	0.167 (Small)

RQ1：解決されるまでの時間



- ① コメントがないSATD-Issue
 - パターン1
- ② ラベルが付けられる前にコメントがないSATD-Issue
 - パターン2
- ③ ラベルが付けられる前にコメントがあるSATD-Issue
 - パターン3

結果

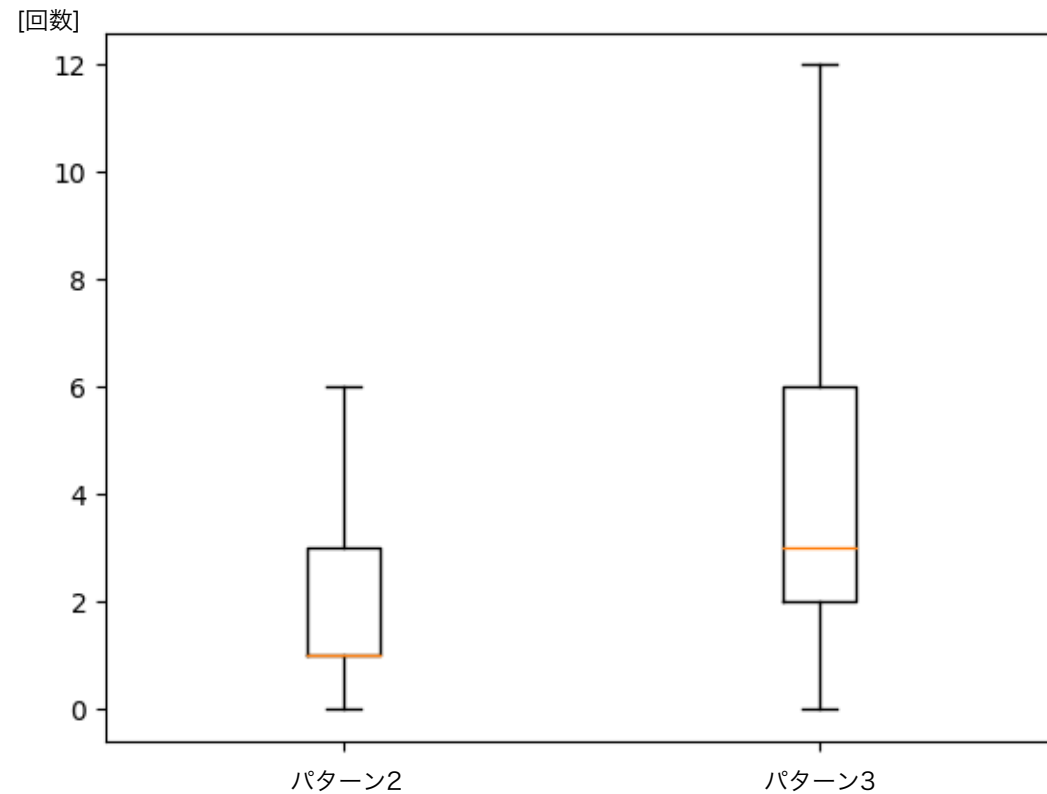
各SATD-Issueの統計量

	最小値	中央値	最大値
パターン1	0.0	5.9	50.1
パターン2	0.0	22.4	280.1
パターン3	0.0	45.4	460.9

ペアワイズ比較による検定結果

	有意差	効果量 d
パターン1 vs パターン2	* (p < 0.01)	0.365 (Medium)
パターン1 vs パターン3	* (p < 0.01)	0.566 (Large)
パターン2 vs パターン3	* (p < 0.01)	0.225 (Small)

RQ1：コメントのラリー数



- ① コメントがないSATD-Issue
 - パターン1
- ② ラベルが付けられる前にコメントがないSATD-Issue
 - パターン2
- ③ ラベルが付けられる前にコメントがあるSATD-Issue
 - パターン3

結果

各SATD-Issueの統計量

	最小値	中央値	最大値
パターン2	0	1	6
パターン3	0	3	12

検定結果

	有意差	効果量 d
パターン2 vs パターン3	* ($p < 0.01$)	0.481 (Large)

RQ1への回答

内容の詳細さ

- コメントがないSATD-Issue (①) は簡潔に書かれる傾向にあり, コメントがあるSATD-Issue (②, ③) は詳細に書かれる傾向にある

解決されるまでの時間

- TDラベルが付けられる前にコメントがあるSATD-Issue (③) が最も解決されるまでに時間が必要
 - ① < ② < ③

コメントのラリー数

- TDラベルが付けられる前にコメントがあるSATD-Issue (③) の方がコメントでのやりとりが多くなる

今後の予定

各パターンのSATD-Issueの内容の分析

- 現在、ラベルでの分類結果を目視中
 - ラベルの数が多いため抽象化およびカテゴリーを作成
 - ラベル名では判断しづらいものがあるためラベルの説明を確認中
- パターンごとにランダムサンプリングをおこない、実際にどのような内容が扱われているのか目視で分析する予定
 - パターン3のSATD-Issueは解決するまでに時間がかかる傾向にあるため、実際にどのような内容だと時間がかかるのか理解できると嬉しい

まとめ

目的

- SATD-Issueで扱われている内容の理解

事前調査

- SATD-Issueの運用方法が複数存在する
 - ① コメントがないSATD-Issue
 - ② ラベルが付けられる前にコメントがないSATD-Issue
 - ③ ラベルが付けられる前にコメントがあるSATD-Issue

Research Question

- RQ1：それぞれのSATD-Issueの特徴は何か？
- RQ2：それぞれのSATD-Issueで扱われている内容は何か？