

静的コード解析ツールがカバーするSATD-issueに関する調査

2021/9/8

発表者: 松尾 和輝

Technical Debt

- Technical Debt(TD)

- 将来的なソフトウェアシステムの品質を犠牲にし，短期的な利益を優先した実装や設計のこと

- Self- Admitted TD(SATD)

- 開発者自身がコード上に技術的負債であることを明記した技術的負債のこと

- SATD-issue

- 課題追跡システムの技術的負債に関連するラベルを付与することにより明示されるSATDのこと

背景

- 技術的負債の蓄積は、新規機能の追加を妨げたり、ソフトウェア開発の遅滞や保守コストの増加を招く原因となるため、早期に取り除かれることが望まれる
- 技術的負債の発見を支援する方法として、SonarQubeのような静的コード解析ツールが用いられる
- 静的コード解析ツールにより検出される技術的負債が、どれだけ開発者が認識している技術的負債をカバーしているか不明である

目的とアプローチ

- 目的

- 静的コード解析ツールが検出する技術的負債と開発者が認識する技術的負債(SATD-issue)の関係を明らかにする

- アプローチ

- 解析対象はvscode
- SATD-issueに紐付くコミット(変更後のリビジョン)の特定
- 変更前のリビジョンの特定
- 変更前後のリビジョンを静的コード解析ツール(SonarQube)を用いて解析
- 解析した結果を用いて分析

SonarQube

- コード内の「バグ」, 「脆弱性」, 「Code Smell」などを予め設定されたルールに基づいて検出することができる静的コード解析ツール
- 自分でルールを追加して解析を行うことができる
- Java, JavaScript, TypeScript, C, C++, C#, HTML, PHP, CSS等のファイルを解析することができる

SATD-Issueに紐づくコミットの特定

- SATD-Issueを慣習的に扱っているプロジェクトである”vscode”を解析対象とする
- vscode内の”debt”というタグの付いた**1612**件のIssueを収集し、それに紐づくリビジョンを特定する
- 特定したリビジョンは**2219**件

リビジョンの特定-解析

- 変更前のリビジョンの特定
 - 変更後のリビジョンを”checkout”コマンドを用いることにより、一つ前のリビジョンに戻すことができ、変更前のリビジョンを特定することができる
- SonarQubeによる解析
 - 特定した変更前後のリビジョンをSonarQubeで解析
 - 変更前後それぞれで1398リビジョンが解析成功
- SATD-IssueはSonarQubeで検出できるようなTDを除去できているのかを調査

やったこと，わかったこと等

- 1612件のSATD-Issueに対して，SATD-Issueに紐づくリビジョンが一つ以上解析できた846件(52.5%)のSATD-Issueを分析対象とする
- 変更前後で一つ以上TDを除去できたSATD-Issueの数は77件(9.1%)だった

今後の予定

- 背景や目的, 問題等を明確にする
 - 文献調査等