Stack Overflow のコード片へ加えられた変更に追従しない GitHub プロジェクトの変更パターン分類による考察

栗原　拓己[†]　　嶋利　一真[††]
神田　哲也[†]　　井上　克郎[†††]（正員：フェロー）

Classification of changes in GitHub projects not following changes made to Stack Overflow code snippets

Takumi KURIHARA[†], Kazumasa SHIMARI[††], Tetsuya KANDA[†], *Nonmembers*, and Katsuro INOUE[†††], *Fellow*

[†] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
[††] 奈良先端科学技術大学院大学先端科学技術研究科
Graduate School of Science and Technology, Nara Institute of Science and Technology
[†††] 南山大学理工学部
Faculty of Science and Technology, Nanzan University
DOI:10.14923/transinfj.???????????

あらまし　GitHub プロジェクトに再利用されている Stack Overflow のコード片に変更が加えられた場合に再利用側の追従状況や追従の必要性について，その変更内容に対してパターン分類を実施することで考察を行う．

キーワード　Stack Overflow，ソースコード再利用，OSS，ソフトウェア進化

1. ま え が き

ソフトウェア開発プラットフォームである GitHub（以下：GH）では，さまざまな Web サイトからコード片の再利用が行われている．その中でも，Stack Overflow（以下：SO）のようなプログラミングに特化した Q&A フォーラムからの，質問への回答に含まれるコード片の再利用が頻繁に行われている．SO のコード片は修正などの目的で変更が加えられることが多く，再利用先の GH プロジェクトのコード片が SO の最新バージョンのコード片と一致しなくなる可能性がある．

Manes らは参照関係にある GH プロジェクトと SO のコード片に対して調査を行い，再利用後に両者ともに変更が行われいるものは高々 10% 程度で，それ以外は片方だけが変更されていることを明らかにした [1]．つまり，バグ修正などの再利用時に影響を及ぼすようなコード片の変更を GH プロジェクトの開発者が見逃している可能性が存在するが，調査では両者の変更内容について詳細な分析が行われていないため，そのような状況がどの程度発生しているかは不明である．

そこで，本調査では GH プロジェクトに再利用されている SO の回答のコード片に変更が加えられた場合に，再利用先のプロジェクトがその変更に追従しているかの状況を調査し，また追従の必要性について変更内容をパターン分類して考察を行う．

2. 調 査 手 法

本調査では一度以上変更されている SO のコード片を含む回答の投稿とそのコード片を再利用している拡張子が “java” である GH プロジェクトを対象に，変更への追従状況や SO のコード片の変更パターンについて調査を行う．データセットとして，SO の全コード片の変更履歴にアクセス可能なデータセットである SOTorrent [2] (Version 2018-09-23) を用いた．本論文では，SO の回答のコード片について，明示的に GH プロジェクト側で再利用が示されている場合とそうでない場合の 2 種類を調査対象とする．

調査対象 1: 明示的に再利用が示された場合 では，コードコメント内に再利用元の SO の回答の投稿の URL を記しているプロジェクトと，その再利用元の SO の投稿を対象とする． このような明示的な再利用関係は SOTorrent では独立した項目として集計されており，調査対象となった SO の投稿は 7,481 件，それらに対応する GH プロジェクトの数は合計 4,621 件であった．これらの再利用関係から一度以上変更されている SO のコード片を再利用している GH プロジェクトを抽出したのち，調査における重複を避けるため，ある SO の投稿を再利用している GH プロジェクトが複数あった場合はそのうち 1 件のみを対象とした．最終的に，調査対象となった SO の投稿と GH プロジェクトの組は 232 件であった．

調査対象 2: 明示的に再利用が示されていない場合 では， Abdalkareem らによる SO のコード片の再利用に関する調査 [3] をもとに，22 個の Android アプリケーションと SO の回答のコード片の間でコードクローン検出ツール CCFinderX を用いて検出された構文的に一致したコード片の組を抽出した．これらの組においてタイムスタンプを比較することで，GH プロジェクトが SO のコード片を再利用している事例を対象とした．また，コードクローン検出における最低トークン数のパラメータは 60 とした．その結果，調査対象となった SO の回答の投稿と GH プロジェクトの組は 409 件であった．

なお，調査対象 1 と調査対象 2 で同一の SO の投稿を参照しているものはなかった．また，変更がコードコメントのみである場合は調査対象から除外した．

これらの調査対象について，以下の項目を調査した．

表 1　SO コード片の変更パターン
Table 1　Change Patterns of SO Code Snippets.

変更パターン	変更内容の例
是正保守	バグ修正・仕様に合わせた変更
適応保守	OS・言語のバージョンによる変更
完全化保守	計算アルゴリズムの改善・リファクタリング
機能追加	テストの追加，呼出し先メソッドの実装
非機能追加	import 文の追加・クラス宣言文の付与

2.1　調査項目 1: 再利用コード片のバージョンの特定と追従状況

再利用先の GH プロジェクトに対して，SO のコード片の最新バージョンと旧バージョンのどちらを再利用しているかを特定する．調査対象 1 では再利用先と再利用元の両コード片を目視で比較し手作業で再利用元を特定した．調査対象 2 では，再利用元のいずれかのバージョンのみがコードクローンと判定されたコード片において，再利用元のバージョンが最新バージョンかどうかで判断した．

GH プロジェクトが SO の最新バージョンのコード片を再利用していると判定されたものに対して，それが過去に SO の旧バージョンを再利用し，その後 SO のコード片の変更にあわせて GH プロジェクト側で同様の変更を加え追従したものであるのかを調査する．具体的には，まず SO のコード片における最初の投稿から最新バージョンへの変更までの期間を求め，その期間内において GH プロジェクト側で SO の旧バージョンのコード片を再利用した履歴が存在していれば，GH プロジェクトは追従によって現在は SO の最新バージョンを再利用している状態になったと判定する．

2.2　調査項目 2: SO 側の変更パターンの分類

GH プロジェクトが SO の旧バージョンのコード片を再利用しているとされた SO のコード片が，SO 側でどのように変更されているかを分類し，変更に追従する必要性を調査する．本調査では Hindle ら [4] の研究で用いられた変更パターンをもとに，SO 特有の変更パターンを考慮し，表 1 に示す 5 種類の変更パターンへ分類を行った．複数の分類に当てはまる変更が行われていた場合は，複数種類のラベル付けを行った．分類作業は第一著者と第二著者の 2 名が行い，結果が一致しない場合は協議のうえ分類を修正した．

3.　調 査 結 果

3.1　調査項目 1: 再利用コード片のバージョンの特定と追従状況

再利用先のプロジェクトにおける追従状況に関する調査結果を表 2 に示す．GH プロジェクトに再利用されている SO のコード片は 72.4% が最新バージョンであった．また，GH プロジェクトが SO のコード片の変更に追従した結果として最新バージョンを利用している事例は調査対象 1，調査対象 2 ともに見られなかった．ここから，再利用元の SO のコード片に変更が加えられた場合でも，再利用先の GH プロジェクトはその変更に追従しないことが一般的であると分かる．

表 2　GH プロジェクトで再利用されている SO コード片のバージョンごとの件数
Table 2　The number of SO code snippets for each version reused in GH projects.

利用バージョン	調査対象 1	調査対象 2	合計
最新（うち追従あり）	186 (0)	278 (0)	464 (0)
旧	46	131	177

表 3　SO のコード片の変更パターンごとの件数
Table 3　The number of code snippets for each change pattern.

変更パターン	調査対象 1	調査対象 2	合計
是正保守	11	53	64
適応保守	6	4	10
完全化保守	15	40	55
機能追加	2	22	24
非機能追加	12	39	51

3.2　調査項目 2: SO 側の変更パターンの分類

GH プロジェクトで再利用されている SO の旧バージョンのコード片に対して SO 側で加えられた変更の分類結果を表 3 に示す．この中から，件数が多い是正保守，完全化保守，非機能追加について考察を行う．

是正保守はバグ修正や不十分な機能に対する変更を含んでいるため，旧バージョンのコード片を利用した場合に不具合が発生する可能性がある．そのため，このような変更が行われた場合には，開発者は積極的に追従を検討すべきである．完全化保守は GH プロジェクトが SO のコード片の変更に追従しなくてもバグが発生するということはなく，是正保守に比べて追従の必要性は低い．しかし，パフォーマンスの向上やリファクタリングのようなソースコードの品質に影響を与える変更であるため，追従の必要性について議論する価値がある．非機能追加は，import 文の追加などの SO のコード片の再利用時における説明補足による変更などを示すため，再利用先の GH プロジェクトにおいてこれらの変更内容に追従する必要性は低い．

4.　今後の展望

GH プロジェクトにおける追従状況と SO での有用な変更の実態から，変更時におけるメッセージやその

変更内容の解析を行うことで，SO のコード片の変更の発生とその概要を GH プロジェクトの開発者に通知するツールの開発が挙げられる．

謝辞 本研究は JSPS 科研費 JP18H04094, JP19K20239 の助成を受けたものです．

文　献

[1] S.S. Manes and O. Baysal, “Studying the Change Histories of Stack Overflow and GitHub Snippets,” Proc. MSR, pp.283–294, 2021.

[2] S. Baltes, L. Dumani, C. Treude, and S. Diehl, “SOTorrent: reconstructing and analyzing the evolution of stack overflow posts,” Proc. MSR, pp.319–330, 2018.

[3] R. Abdalkareem, E. Shihab, and J. Rilling, “On code reuse from StackOverflow: An exploratory study on Android apps,” Information and Software Technology, vol.88, pp.148–158, 2017.

[4] A. Hindle, D.M. German, M.W. Godfrey, and R.C. Holt, “Automatic classication of large changes into maintenance categories,” Proc. ICPC, pp.30–39, 2009.

（xxxx 年 xx 月 xx 日受付）

Abstract This paper investigates and discusses the status and necessity of change tracking in GitHub projects for Stack Overflow code snippets that are reused in GitHub projects by performing pattern classification on code snippet changes.

Key words Stack Overflow, Code Reuse, OSS, Software Evolution