

# README 修正内容に関連する ソースコード変更コミット追跡への試み

白崎 優奈<sup>†</sup>      伊原 彰紀<sup>‡</sup>      石岡 直樹<sup>§</sup>  
和歌山大学<sup>†</sup>      和歌山大学<sup>‡</sup>      和歌山大学<sup>§</sup>

## 1 はじめに

多くのソフトウェア開発プロジェクトでは、ソフトウェアの機能、使用方法などを開発者や利用者に発信するため、ソフトウェアドキュメントとして README を公開している。ソフトウェア開発の中でも GitHub を利用するオープンソースソフトウェア (OSS) 開発では、開発への参加方法、各機能の概要なども README に記述しており、README はリポジトリのトップページに提示される重要なドキュメントとなっている。

OSS プロジェクトは、実装中のソースコードを逐一公開しているため、ソースコードの更新に伴って README の更新が期待される。しかし従来研究ではソースコードと README が紐付けられていないため、ソースコードと README がどのように共進化しているか明らかでない。また、開発者はソースコードと README を同一コミットで更新していないこともあり、README の更新が必要となったソースコードの変更コミットを紐付けることは容易でない。

本研究は、README 更新の追従性を確認するため、ソースコードと README の変更内容に共通して含まれる文字列に基づいて関連付ける方法を提案する。特に README の変更

と、関連づけられたソースコードのコミットとの時間差を分析することで、プロジェクトにおける README の記述方針を明らかにする。**TODO:** [今後は、記述方針がソフトウェア品質や効率にどう影響するか明らかにしたいなあ.. ここまではこの論文で書けないけど.]

## 2 分析

### 2.1 分析手法

の両方に同じ文字列 README の変更に関連することが示唆されるソースコードの変更コミットを特定する。具体的には、README の更新**TODO:** [更新には追加, 変更, 削除を含むよね?] された文字列と、各コミットにおいて変更されたソースコード中の文字列をそれぞれ抽出し、同じ文字列を含む README とソースコードを関連づける。ソースコードファイル中から記号削除、NLTK を用いて形態素解析により名詞のみを抽出する。同様の手法で、README 変更コミットで追加された README から名詞のみを抽出する。**TODO:** [README は変更内容のみではないの? README 全体?]

### 2.2 ケーススタディ

本論文では、README が英語で記述され、他のリポジトリに比べて頻繁に保守している 2 つのプロジェクト (aliyun-openapi-nodejs-sdk, js-multiaddr) を対象とする。

図 1 と図 2 は、それぞれのプロジェクトを対象に、README の変更コミットの前後 10 コミットそれぞれにおいて、README において変更された文字列を含む頻度を計測した結果である。横軸は README を変更したコミット

---

Toward tracking source code change commits related to README revision

<sup>†</sup> Shirasaki Yuna, Wakayama University

<sup>‡</sup> Akinori Ihara, Wakayama University

<sup>§</sup> Naoki Ishioka, Wakayama University

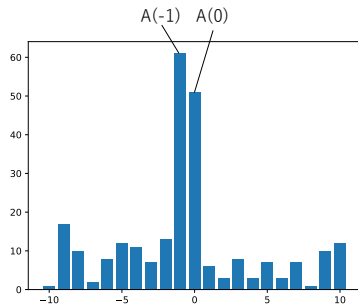


図 1: aliyun-openapi-nodejs-sdk における各コミット中の README の変更単語数

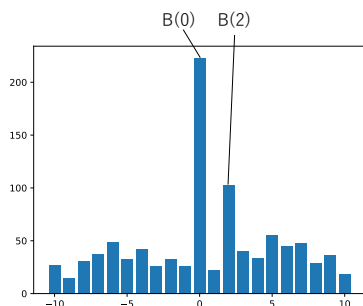


図 2: js-multiaddr における各コミット中の README の変更単語数

を 0 とする前後のコミット位置を示し、縦軸はソースコード中に含む README において変更された文字列の数を示す。紙面の都合上、XX を変更した README の変更箇所をケーススタディとして示す。 **TODO: [README が変更されたコミットは他にもあったと思うけど、なぜこのコミットを選んだ?]**

aliyun-openapi-nodejs-sdk: 図 1 の A(0) において README が更新され、1 コミット前の A(-1) と A(0) のソースコード変更が README の更新に関連していることが示唆される。A(-1) と A(0) のソースコードでは **TODO: [X の変更]** が行われ、README には **TODO: [XX の内容が追記? 削除? 変更?]** されている。このコミットでは、実装後に README が更新されたことが示唆される。

js-multiaddr: 図 2 の B(0) において README が更新され、2 コミット後の B(2) と B(0) のソースコード変更が README の更新に関連していることが示唆される。A(2) と A(0) のソースコードでは **TODO: [X の変更]** が行われ、README には **TODO: [XX の内**

**容が追記? 削除? 変更?]** されている。このコミットでは、実装前に README が更新されたことが示唆される。

### 3 おわりに

本論文では、README がソースコードの変更を追従しているかを確認するため、各ファイルの単語を比較した。分析の結果、README の更新は、ソースコード変更の前と後のいずれの時期でも変更されることがあり、README の内容に応じて追従するタイミングが異なると考える。今後は、README とソースコード変更内容による README の追従性違いを明らかにし、README の追従性によるソフトウェア品質への影響を示す。また、README の自動更新タイミングについても検討する。

### 謝辞

**TODO: [メモ: 少し削る, グラフ軸名記入]**