OSS開発プロジェクトにおける READMEファイルの進化に関する研究

佐渡島 悠樹[1]

概要：
GitHub 等の OSS（Open Source Software）開発プロジェクトにおいて README ファイルはプロジェクトに関わる際に一番初めに表示されるドキュメントである．しかし，README ファイルは不完全であったり古くなっており読まれないことが多い．本研究では，Prana らの README ファイルの自動ラベル分類の実験を参考に，README が初期バージョンからどのように，何回変更されているのかを調査する．そうすることによって，ドキュメント作成のための知見を得て，README に記述すべき項目を明らかにすることを目指す．

キーワード： OSS，GitHub，README，

1. はじめに

オープンソースソフトウェア（OSS）開発はソースコードをはじめとするファイルを GitHub 等のオンライン上のホスティングサービスに公開して開発を行い成果物（ソースコード）をリリースする．この時，一般的にはソフトウェアが正しく利用されるために，ソフトウェアについての情報を記述したドキュメントを一緒に公開されている．このドキュメントはソフトウェアの利用において重要な役割を担っている [1]．GitHub のストラテジー部門のバイスプレジデント Brian Doll 氏は「良い README ファイルを持つプロジェクトは最も使われる傾向にある．」と主張している [2].

その GitHub における README.md はプロジェクト利用者がプロジェクトを閲覧する際，一番初めのページに表示されるようになっている．GitHub ではプロジェクトを作成する際、最初に README ファイルを作成し，どのようなプロジェクトなのか，プロジェクトの有用性，プロジェクトの使い方や誰がプロジェクトを作成し維持しているのかなどを記述することを推奨している．[*1]

しかしながら，開発者の中には README ファイルの作成に多くの労力を費やす人がいる．2017 年に行われた調査[*2]では，GitHub の利用者は README ファイルは重要であると考えているものの，内容が不完全であったり，古いままになっているものが多いため，読まれないことも多いという課題が指摘されている．また，投稿者の 60 %はドキュメントに貢献することが滅多にない，あるいは全く貢献していないということも調査からわかっている．

本研究では，REAMDE ファイルに関する上記課題を解決することによって，README ファイルに記述すべき項目を明らかにし，ドキュメント作成方法の推奨を目標とする．そうすることによって，Igor ら [3] の OSS の新規参入者に対する障害に関する研究におけるドキュメンテーション不足などが改善され，新規参入者のプロジェクトの立ち上げの手助けにもなると考えられる．

目標を達成するための第一段階として本研究では README ファイルの変更過程を調査する．まず初めにどのようなことをドキュメントに記述し，どのような順番で書き足していくと良いのかを知るために，以下のような調査課題（以下 RQ：Research Question）を 4 つ設定した．

RQ1 README ファイルの初期バージョンには何が（どのカテゴリが）書かれているのか

RQ2 初期バージョンから最新バージョンまでの間に，どのカテゴリが追加されるのか

RQ3 複数のカテゴリを持つ README ファイルはどのような順番でカテゴリが追加されているのか

RQ4 コミット回数とカテゴリの増減にどのような関係があるのか

[1] 九州大学
[*1] https://help.github.com/articles/about-readmes/
[*2] http://opensourcesurvey.org/2017/

上記の調査課題を達成することによって README ファイルの作成の際，作成初期の段階でどれ程の内容が書かれており，どれぐらいの工数を割いて最新バージョンの README ファイルを作成しているのかを理解することができると考える．

次章以降の章構成は以下の通りである．2 章では関連研究についての説明をし，3 章では研究で利用したデータセットの説明をする．4 章では調査の内容と結果について説明する．5 章で妥当性への脅威について述べ，6 章で結論を述べる．

2. 関連研究

従来の研究では OSS 開発の導入支援やドキュメントに関する分析や作成支援システムなどの研究を行なっているものがある．

2.1 OSS 開発の導入支援

Igor らは OSS 開発プロジェクトへの新規参入者のためのポータルを作成，及び，評価を行なった．ポータルは新規参入者が障壁を乗り越えるためにどのようなことをすれば良いかの判断には役立ったが，技術的な障壁を解決するものとしては不十分という結果となった．彼らはこの研究以前に OSS 開発プロジェクトへの参入の障壁について調査を行なっている [4]．調査によって得られた 58 つの障壁を 6 つにわけている．その中にはドキュメントの問題も含まれおり，本研究が障壁の解決の手助けになると考える．

2.2 ドキュメント分析

Moreno ら [5] はソフトウェアの情報を記述したドキュメントの 1 つである，リリースノートの自動作成のシステムを設計，及び，評価している．Moreno らは 990 件のドキュメントを目視で確認し，ドキュメントに記述すべき項目を抽出することによってシステム開発を行なった．評価は高い有用性を示しているが，ソフトウェアの種類によってドキュメントに記述すべき内容は異なる．したがって，全てのソフトウェアに適用できるシステムではないのではないかと考えられる．ドキュメントの分析という点で似ている部分があるが対象とするドキュメントが異なっている．

池田ら [6] はソフトウェアを説明するドキュメントの作成支援のために，GitHub に登録されている 143,239 件のプロジェクトが公開する README ファイルに記述されている項目の分析を行なった．彼らの研究では README ファイルの見出しの単語を分析することによって，見出しとして使用している内容の上位 10 件について使用しているプロジェクト件数と全体のプロジェクトに対する記述率について調査を行った．見出しを抽出して分析している点では同じだが，作成途中の README ファイルについて調査している本研究とは異なる．

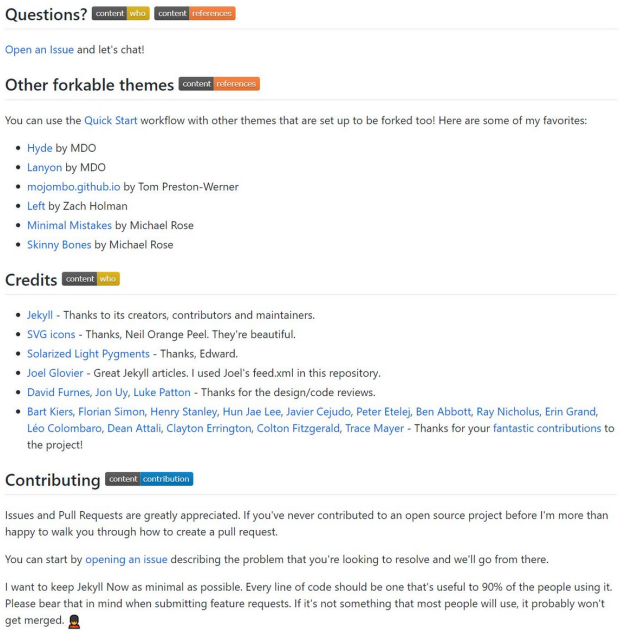

図 1 カテゴリ分類された README の例[*3]

Prana ら [7] は README ファイルの品質向上と関連情報の発見の効率化を図るために，README ファイルの内容をセクションごとに自動分類し，ラベルづけするツールを設計した．ツールを README ファイルに適用した例が図 1 である．彼らの作成したマルチラベル分類器は 0.746 の F1 値を達成し，有用性の評価に参加したソフトウェアの専門家の大部分は Prana らのツールが有用であると回答した．表 1 にカテゴリの分類内容を示す．彼らの研究では最新バージョンの README ファイルについてのみ分類を行なっているのでその点が本研究とは異なる．

3. データセット

本章では，調査のために用意した GitHub の OSS プロジェクトについての説明と，4 章で説明する調査のために先行研究から取得したデータについて記述する．

プロジェクト．本実験では Prana らの README ファイルの自動マルチラベル分類器の研究で使用されたプロジェクトをデータセットとして利用する．Prana らの研究では GitHub 内のプロジェクトをランダムに 1,193 件のクローンし，その中から README ファイルにタイトルしか書かれていないと思われるもの等の実験に不適切なプロジェクトを除外した 393 件のプロジェクトを研究に利用している．

データセットの取得．Prana らの研究によって最新バージョンの README ファイルについてカテゴリ分類がされており，本研究ではその分類した結果を利用している．Prana らの研究では README ファイルを見出しごとにセクションとして分けて分類を行い分類結果をまとめている．その分類結果に記述されているプロジェクト名の部分

[*3] https://pbs.twimg.com/media/DnsgUZJXsAAo7gg.jpg

表 1　カテゴリ一覧 [7] から引用

category	example section heading
what	introduction, project, background
why	advantages of the project, comparison with related work
how	getting started, how to run, installation, requirements, platforms, downloads, setup
when	project status, versions, project plans, roadmap
who	project team, community, mailing list, contact, acknowledgement, license,
references	API documentation, getting support, feedback, more information, translations, related projects
contribution	contributing guidelines
other	—

表 2　初期と最新バージョンでのカテゴリ

category	初期時の件数	追加件数	最新時の件数
what	122	139	261
why	26	45	71
how	111	172	283
when	19	40	59
who	50	102	152
references	49	134	183
contribution	25	58	83
other	6	18	24
no category	211	-	-

を利用して 393 件のプロジェクトをクローンした．Prana らの研究では利用されたプロジェクトの 393 件のうち，README ファイルが英語以外の言語で書かれている等でカテゴリ分類されていないドキュメントがある．それらのプロジェクトを除いた 370 件のプロジェクトをデータセットとした．

4. 調査と結果

README ファイルの進化の過程を調査するために 4 つの調査課題を設定し，調査を行なった．この章では，その方法と結果について記述する．

4.1 RQ1：README ファイルの初期バージョンには何が（どのカテゴリが）書かれているのか

動機．README ファイルの進化についての知見を得る第一歩として，README ファイルの初期バージョンに何が書かれているのか調査する．最新バージョンで書かれているカテゴリの全てが初期バージョンから記述されていることは少ないのではないかと考え，どのような内容が記述されているのかの知見を得ることで初期バージョンに記述すべき項目を議論する．

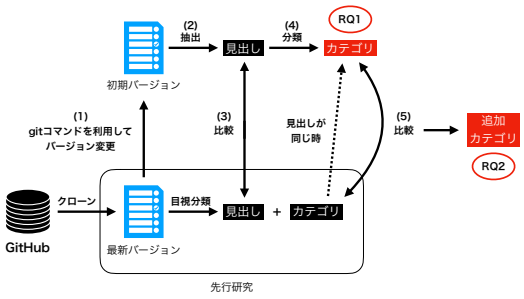

図 2　初期バージョンのカテゴリ分類と追加カテゴリの調査

アプローチ．初期バージョンのカテゴリ分類は Prana らの研究で分類された最新バージョンの見出しと比較することによって行なった．図 2 は行なったことを示している．クローンしてきたプロジェクトのバージョンを最新バージョンとする．最新バージョンの README ファイルを git コマンドを利用して初期バージョンにした (図中の (1))．git log コマンドを利用して初期バージョンのコミット ID を取得し，git checkout コマンドに取得したコミット ID を利用することで README ファイルのバージョンを変更した．

初期バージョンになった README ファイルから見出しを抽出した (図中の (2))．GitHub の README ファイルは markdown 形式を利用して記述される．markdown 形式では＃をつけることによって見出しを作成するため，README ファイルの＃がついている行を抽出した．

抽出した見出しと最新バージョンに含まれる見出しを比較し (図中の (3))，初期バージョンにどのような内容が書かれているのか分類を行なった (図中の (4))．3 章で記述した Prana らの研究で作成された分類結果の見出しと抽出した見出しが同じであればその見出しに分類されたカテゴリが初期バージョンのその見出し内にも同じカテゴリが含まれると考え，初期バージョンの見出しにカテゴリを分類する．これを初期バージョンに含まれる全ての見出しについて行い，初期バージョンの README ファイルのカテゴリ分類を行なった．

結果と考察．調査の結果，表 2 に示した結果となった．結果から作成の際に記述すべき内容は what と how だと考えられる．初期バージョンでもっとも多かったのはどのカテゴリも含まれていないドキュメントで 211 件，約 57 %であり，記述されているカテゴリとしては，what と how がそれぞれ 122 件と 111 件であった．これは 1 つ以上のカテゴリを含む README ファイル 159 件の内記述される割合がそれぞれ 77 %と 70 %となっている．

本研究では最新バージョンの見出しに初期バージョンの

カテゴリが依存する形になっているので，見出しが変更されると初期バージョンがカテゴリを含まないことになるので，211 件全てが何も含まれていないと考えるのは適切ではないが，半数程度は README の作成の際に最初は無記入，あるいはプロジェクト名あるいは適当な単語のみの見出しで作成されると考えられる．

README ファイルの初期バージョンでは “what” と “how” が記述されていることが多い．

4.2 RQ2:初期バージョンから最新バージョンまでの間に，どのカテゴリが追加されるのか

動機． RQ1 で README ファイルの初期バージョンの内容についての調査を行なった．初期バージョンには含まれていない途中で追加されるカテゴリを調査することによって，README ファイルを最初に作成する時ではなく，プロジェクトが進むにつれて記述されるのはどのカテゴリなのか調査する．RQ1 に加え，追加されるカテゴリについて調査することで README ファイルに記述される内容の重要性における知見を得ることができるのではないかと考える．初期バージョンから記述されている内容は一番重要であり，途中で追加される内容はその次に重要なないようなのではないかと考えることができる．

アプローチ． 初期バージョンと最新バージョンに記述されているカテゴリを比較することによって調査を行なった (図 2 中の (5))．RQ1 で初期バージョンの README ファイルについてカテゴリ分類をおこなった．そのカテゴリと最新バージョンのカテゴリを比較することによって，最新バージョンのカテゴリから初期バージョンのカテゴリを除くことで途中で追加されたカテゴリを調査した．

結果と考察． 結果は表 2 に示した．結果から途中で追加するカテゴリは “who” と “reference” を優先すべきであると考えられる．追加件数は “how” が 172 件で他のカテゴリと比較すると多いが，このカテゴリは RQ1 の結果より初期バージョンから記述されることも多い．追加された数が多く，また追加された割合も多いのは who と reference である．追加件数はそれぞれ 102 件と 134 件であり，追加される割合は 67 %と 73 %であった．

初期バージョンに含まれず途中で README ファイルに追加されるカテゴリは “who” と “reference” が多い．

4.3 RQ3:複数のカテゴリを持つ README ファイルはどのような順番でカテゴリが追加されているのか

動機． README ファイルの作成においてどのような順序でカテゴリを追加していけばいいのか考察したい．RQ2 で追加されるカテゴリについての調査を行なった．この時カテゴリの追加はどのような順序で行われているのかを調べることで README ファイル作成における記述の順序の設定や記述される内容の重要性の優劣について考えられるのではないかと考える．

アプローチ． README ファイルの初期バージョンから最新バージョンまでをコミット数に基づいて等間隔で 4 つの区間にわけ，どのような順序で追加されているのか調査した．git log コマンドを利用して README ファイルに対するコミット数を抽出する．抽出したコミット数を除算し，その数のコミット ID を取得し README ファイルのバージョンを変更した．変更した README ファイルに対して RQ1 で行なった README ファイルの初期バージョンへのカテゴリ分類と同じ手法でそれぞれのバージョンに対してカテゴリ分類を行なった．分類した README ファイルのカテゴリについて比較することによって 4 区間の追加カテゴリを調査した．

結果と考察． 結果を表 3 に示す．追加件数が多いカテゴリである “what” と “how” と “who” と “reference” は初期段階での追加が多い．その中でも他の区間と比べて第 1 区間における追加件数が多くなっているのは “how” であった．最新バージョンでも多く記述されており，重要と考えられるカテゴリが先に追加されているのではないかと考えられる．今回の研究では 4 区間による調査となり，傾向が見えづらい部分も多く，さらなる詳細な調査が必要である．

表 3 各区間での各カテゴリの追加件数

category	第 1 区間	第 2 区間	第 3 区間	第 4 区間
what	50	37	21	31
why	9	16	6	14
how	73	32	40	27
when	12	10	9	9
who	31	25	18	28
references	47	34	25	28
contribution	12	10	18	18
other	1	8	3	6

他の区間に比べ第 1 区間で追加されることが多いのは “how” である．

4.4 RQ4:コミット回数とカテゴリの増減にどのような関係があるのか

動機． コミット回数とカテゴリ増減を調査することによってドキュメント作成における工数予測などのための知見が得られると考える．README ファイルに記述すべき項目が明らかになり，作成方法の提案ができるようになった場合，その作成にどれだけの工数がかかるのかという情報はプロジェクト開発者にとっては重要であると考える．

アプローチ． プロジェクト毎に追加カテゴリ数と README

表 4　コミット数と追加カテゴリ数の関係

追加カテゴリ数	件数	README のコミット数の平均	プロジェクト全体のコミット数の平均	README のコミットの割合
1	56	21.5	1521.4	1.4
2	59	27.5	799.5	3.4
3	60	36.9	705.4	5.2
4	33	59.7	1266.5	4.7
5	20	53.5	3092.5	1.7
6	17	154.9	2489.4	6.2
7	3	107.3	5550.3	1.9
8	0	-	-	-
平均	31	65.9	2203.6	3.5

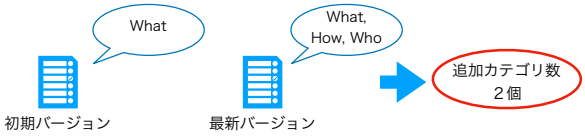

図 3　追加カテゴリ数の図解

ファイルに対するコミット回数を取得し，追加カテゴリ数毎に分けてコミット数を調査した．追加カテゴリ数についての一例として図 3 を示す．最新バージョンの README ファイルに含まれるカテゴリの数から初期バージョンでのカテゴリ数を引くことによって追加カテゴリ数を計算する．追加カテゴリ数毎にプロジェクトをまとめて，プロジェクトの件数とそのプロジェクトの README ファイルに対するコミット回数を'git log –oneline – README.md — wc -l' のコマンドより抽出し，追加カテゴリ数に対するコミット数の平均を求めた．

結果と考察．結果は表 4 と図 4 に示す．追加カテゴリ数は 1-3 個のドキュメントが多い．図 4 より追加カテゴリ数が多くなるにつれて README ファイルに対するコミット数が多くなることがわかる．追加カテゴリが増えるとプロジェクト全体のコミット数も増えているので，README ファイルのコミットの割合は増加の傾向は見られない．割合の平均をとると 3.5 %であった．これより README ファイル作成における工数の割合は 3.5 %程度であると考えられる．

追加カテゴリ数が多くなるほど README ファイルのコミット数が増える．README ファイル作成にかかる工数の割合は 3.5 %程度である．

4.5　結果に対するまとめ

調査の結果によって，README ファイルを作成する際，初期バージョン時には “what” と “how” が記述されていることが多い．最新バージョンに含まれており，途中で追加される割合が高いカテゴリは “who” と “reference” であった．README ファイルの作成における工数としては全体 3.5 %を要する程度と考えることができる．

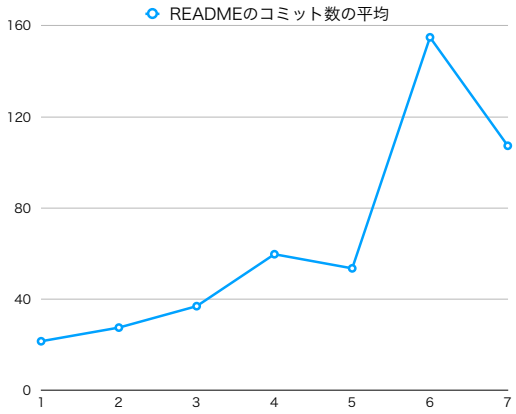

図 4　追加カテゴリ数と README へのコミットの割合との関係

5.　妥当性の脅威

本研究の調査結果には妥当性に対する脅威が存在している．外的妥当性として，本研究では 370 件の GitHub で公開されているプロジェクトについて調査を行なったが，調査する対象のプロジェクトを変更した場合結果が異なる可能性がある．また，Prana らの研究において英語以外の README ファイル等の実験に不適切と判断されたプロジェクトが除かれているが，このフィルタリングの基準が変われば結果が変わる可能性がある．本研究は README ファイルについてのみ調査を行なったので，他のドキュメントに対する一般化を主張することはできないと考えられる．本稿のカテゴリ分類は Prana らの研究で考えられたものを利用したが，この分類が有効なのかは考慮が必要である．またカテゴリ分類時に見出し毎にセクションとして分けて調査を行なったが，README ファイルの分け方を変更することによって結果が変わる可能性がある．

見出しは＃がついている部分で抽出したが，それ以外の記法で作成された見出しは今回抽出できていない．全ての見出しを抽出できると結果が変わる可能性がある．

6. おわりに

本稿では README ファイルの初期バージョンからの進化に着目して 370 件のプロジェクトに対して調査を行った．初期バージョンの README ファイルには what と how が記述されることが多く，途中で追加される割合が高いのは who と reference であった．今後は RQ4 で行なった調査の複数カテゴリを持つ README ファイルはどのような順番で追加されるのについてさらに詳しく調査したり，ソースコードやマニュアル内で頻出する単語と README ファイルの関連性などに着目して調査していき，ドキュメントの記述方法の提案を目指す．

謝辞．研究ならびに論文作成にあたり，熱心にご指導下さった九州大学大学院システム情報科学研究院の，鵜林尚靖教授，亀井靖高准教授，佐藤亮介助教に深く感謝します．研究室での活動を助けていただいた秘書の三浦亜矢氏にも深く感謝の意を表します．研究室でともに研究活動に従事してきた学生諸氏にも深く感謝します．

参考文献

[1] Mens, T. and Goeminne, M.: Analysing the evolution of social aspects of open source software ecosystems, *Proceedings of the Third International Workshop on Software Ecosystems, Brussels, Belgium, June 7th, 2011*, pp. 1–14 (online), available from ⟨http://ceur-ws.org/Vol-746/IWSECO2011-1-InvitedPaper-MensGoeminne.pdf⟩ (2011).

[2] Begel, A., Bosch, J. and Storey, M. D.: Social Networking Meets Software Development: Perspectives from GitHub, MSDN, Stack Exchange, and TopCoder, *IEEE Software*, Vol. 30, No. 1, pp. 52–66 (online), available from ⟨https://doi.org/10.1109/MS.2013.13⟩ (2013).

[3] Steinmacher, I., Conte, T. U., Treude, C. and Gerosa, M. A.: Overcoming open source project entry barriers with a portal for newcomers, *Proceedings of the 38th International Conference on Software Engineering, ICSE 2016, Austin, TX, USA, May 14-22, 2016*, pp. 273–284 (2016).

[4] Steinmacher, I., Chaves, A. P., Conte, T. U. and Gerosa, M. A.: Preliminary empirical identification of barriers faced by newcomers to Open Source Software projects, *Software Engineering (SBES), 2014 Brazilian Symposium on*, IEEE, pp. 51–60 (2014).

[5] Moreno, L., Bavota, G., Penta, M. D., Oliveto, R., Marcus, A. and Canfora, G.: ARENA: An Approach for the Automated Generation of Release Notes, *IEEE Trans. Software Eng.*, Vol. 43, No. 2, pp. 106–127 (online), available from ⟨https://doi.org/10.1109/TSE.2016.2591536⟩ (2017).

[6] 池田祥平，伊原彰紀，ラウラ ガイコビナクラ，松本健一：GitHub における README 記述項目の分析，第 24 回ソフトウェア工学の基礎ワークショップ，FOSE, pp. 135–140 (2017).

[7] Prana, G. A. A., Treude, C., Thung, F., Atapattu, T. and Lo, D.: Categorizing the Content of GitHub README Files, *arXiv preprint arXiv:1802.06997* (2018).