OSS 開発者の離脱要因理解のための Politeness の質的調査

宮崎 智己
和歌山大学 システム工学部
s181052@sys.wakayama-u.ac.jp

大平 雅雄
和歌山大学 システム工学部
masao@sys.wakayama-u.ac.jp

要旨

近年，オープンソースソフトウェア *(OSS)* は企業のソフトウェア開発でも積極的に再利用されている．*OSS* 開発者の多くはボランティアとして *OSS* プロジェクトに参加するが短期間でプロジェクトを離脱することが知られている．そのため，多くの *OSS* プロジェクトでは，プロジェクトに長期間貢献する開発者が慢性的に不足しており，プロジェクトの安定運営およびプロダクトの品質維持に関する重大な懸念を抱えている．*OSS* プロジェクトに参加する開発者の *Politeness*（コミュニケーション上の配慮）に着目して我々が実施した先行研究 *[18, 19]* では，プロジェクト離脱直前に *Politeness* が急激に変動するコア開発者の存在を明らかにした．

Politeness の急激な変動は開発者の離脱の予兆を検知する手法の構築につながる可能性がある．離脱の予兆検知手法を構築できれば，長期間プロジェクトに貢献する重要な開発者の離脱をプロジェクト管理者等が早期に察知し，離脱を防ぐための方策の立案に役立つことが期待される．しかしながら *[18, 19]* では，開発者の *Politeness* が急激に変動した理由や，開発者の離脱と *Politeness* の急激な変動とにどのような関係があるのかについては詳細な分析をおこなっていない．

本稿では，*Politeness* が急激に変動する理由と離脱原因との関係を明らかにするために，コア開発者が離脱する直前 *3*ヶ月間の開発者メーリングリストへの投稿の内容を目視する定性的な調査を実施する．本調査の結果，離脱直前に *Politeness* が大きく低下するコア開発者はコミュニティ内で他の開発者と意見が対立していたことを確認した．一方，離脱直前に *Politeness* が大きく上昇したコア開発者の離脱理由については明示的な手がかりを見つけることができなかった．

1. はじめに

近年，オープンソースソフトウェア（以降，OSS と呼ぶ）は様々な用途で幅広く普及しており，個人ユーザによるエンドユースのみならず，企業におけるソフトウェア開発にも積極的に再利用されている．しかしながら，再利用する OSS がいつまで開発保守されるのか分からなかったり，不具合が発見された場合に確実に修正されるかどうかが不透明なため [7]，OSS の利用を躊躇するソフトウェア開発企業も少なくない．

OSS の多くはボランティア開発者によって開発保守されている．ボランティア開発者が OSS プロジェクトで中心的な役割を担うようになる（コア開発者になる）には，長期的な貢献を通じてコア開発者としての資質の有無を認められる必要がある [8, 15] が，ボランティア開発者の多くは約 1.5 年以内でプロジェクトから離脱する [2] ため，多くの OSS プロジェクトにおいてコア開発者が慢性的に不足している．特に大規模 OSS プロジェクトでは，ユーザから不具合報告や機能改善要求が多数寄せられるため，少数のコア開発者へ過度な負担がかかることが多い [9]．コア開発者の離脱はプロジェクトの安定運営に支障を来すだけではなくプロジェクトの消滅や品質の低下に繋がる恐れがあるため，ボランティア開発者の参入障壁を取り除く [13, 14]，コア開発者の負担を軽減する [1, 12]，コア開発者候補者を早期に発掘する [5, 16] など，様々なアプローチから OSS プロジェクトの安定運営のための支援方法が研究されている．

我々の研究グループでは，コア開発者の離脱の予兆を検知するための手法の構築に取り組んでいる．手法構築の予備調査として，OSS プロジェクトで開発者が情報交換するために利用する開発者向けメーリングリストを対象に，開発者の Politeness（コミュニケーション上の配

慮）[3, 17] を定量化するツール [4] を用いたケーススタディをおこなった [18, 19]．開発者の Politeness に着目した理由は，OSS 開発は他の開発者との協調作業を基本とするため，(1) 継続的に貢献するコア開発者は協調作業を円滑にするために他の開発者に対して配慮あるコミュニケーションをしているはずであるが，(2) コア開発者が離脱する際には，本業が急に忙しくなる，他の開発者と対立するなどによりコミュニケーションの質が変化する（Politeness が変化する）可能性がある，と考えたためである．

ケーススタディの結果，OSS プロジェクトに長期間参加しているコア開発者の Politeness は，短期間でプロジェクトを離脱した開発者より統計的に有意に低い値を示すことが明らかになった．さらに，OSS プロジェクトに長期間参加しているコア開発者であっても Politeness が急激に変動した場合，その直後にプロジェクトを離脱するコア開発者が存在することを明らかにした．しかしながら [18, 19] では，実際のコミュニケーションの内容については十分に分析をおこなっておらず，なぜ開発者の Politeness が急激に変動したのかや，なぜ Politeness の急激な変動がプロジェクトの離脱を招いたのかなどの理由については不明なままである．

本稿では，Politeness が急激に変動する理由と離脱原因との関係を明らかにすることを目的として，Politeness の急激な変動を示したコア開発者のコミュニケーションの内容を目視により確認する．[18, 19] の知見に加え，本調査により，開発者の Politeness の変動とプロジェクト離脱との関係が明らかになれば，開発者の離脱の予兆を検知する手法を構築する上での重要な手がかりとなることが期待できる．

以降ではまず，2 章において本研究で用いる Politeness の概念について説明する．また，Politeness を定量化するためのツール Stanford Politeness について説明する．3 章では，開発者の Politeness と活動継続性との関係を調査した我々の先行研究 [18, 19] の概要と結果を示す．4 章では，開発者の Politeness の急激な変化とプロジェクト離脱との関係を明らかにするために行った調査結果について議論する．5 章で関連研究を示し，最後に 6 章において本研究の今後の研究方針について議論し本論文をまとめる．

2 Politeness と定量化ツール

本章ではまず，Politeness の概念について説明する．次に，[18, 19] で用いた Politeness を定量化するためのツール Stanford Politeness について説明する．

2.1 Politeness

Politeness[3, 17] とは，日本語の「丁寧さ」や「礼儀正しさ」とはニュアンスが異なる概念であり，円滑な人間関係を確立・維持するための相手への配慮を指す．人は，ポジティブフェイス（他者に認められたい・評価されたいという欲求）とネガティブフェイス（他者から批判されたくない，行動を妨げられたくないという欲求）の二つのフェイスを持っており，コミュニケーションをおこなう相手のポジティブフェイスとネガティブフェイスの両方を保つために Politeness を適切に表現する必要がある．

Politeness Strategy[3] とは，Politeness を示すための具体的な表現方法である．我々は，他者との会話の中で様々な種類の Politeness Strategy を使い分けながらポジティブ・ネガティブフェイスを考慮している．

表 1 に，Politeness Strategy を示す．Politeness Strategy の例をいくつか説明する．相手のポジティブフェイスを保つための Politeness Strategy の 1 つに Gratitude がある．Gratitude は，“I appreciate you.” や “Thank you for ～.” といったように，相手に対する感謝を示すための表現である．Gratitude を受け取った相手は，話し手から評価されていることを明示的に認識することができるためポジティブフェイスが保たれる．一方，Hedges は，相手のネガティブフェイスを保つための Politeness Strategy の 1 つである．Hedges は，“It appears that ～?” や “I suggest ～.” といったように，表現を曖昧にすることで相手のネガティブフェイスを侵害しないようするための表現である．

2.2 Stanford Politeness

本研究では，開発者の Politeness を分析するために，開発者メーリングリストから取得するコミュニケーションデータに対して Stanford Politeness[4][1]を適用する．Stanford Politeness とは，言語表現に込められた Politeness を定量化するツールである．

[1] https://github.com/sudhof/politeness

表 1. Politeness Strategy（影響の強さの順に上から並べたもの）

Politeness 値を上げる Strategy		Politeness 値を下げる Strategy	
Strategy	該当単語や文の例	Strategy	該当単語や文の例
Gratitude	I appreciate thank	Direct start	So 〜. But 〜.
Deference	great, nice, awesome	Factuality	really, actually
Indirect (btw)	By the way	Please start	Please 〜.
Please	〜 please	2nd person start	You 〜. Your 〜.
Counterfactual modal	Could you 〜? Would you 〜?	Direct question	What 〜? Why 〜?
Greeting	hi, hello, hey		
Apologizing	sorry, woops		
Hedges	think, appear, feel		
1st person start	I 〜. My 〜.		
Indicative modal	Can you 〜? Will you 〜?		
1st person pl.	we, our		
1st person	I, my, mine		
2nd person	you, your, yourself		

表 2. Stanford Politeness[4] を適用した例

No.	例文	Politeness 値
1	OK, thanks for your suggestion. I will change that. Regards, *first name*.	0.997
2	Oh dear. My question is already slipping down the list into obscurity... I guess I must be the only person using eclipse 3.4 who would like to be able to do this.	0.507
3	So, what was the solution for this problem then??	0.122

表 2 に，OSS 開発者のメーリングリスト上の議論に対して Stanford Politeness を適用した例を示す．表 2 中の Politeness 値とは，Stanford Politeness を用いて Politeness を数値化した値である．Politeness 値は 0 から 1 の間の値をとり，値が大きいほど相手を配慮する表現が取り入れられていることを示す．例えば，1 番目の文章の Politeness 値は 0.997 と高い値を示している．これは，文中に含まれる “thanks” が Politeness Strategy の Gratitude に，“suggestion” が Politeness Strategy の Hedges に該当しているためである．2 番目の文章は，単語数が多く情報量の多い文章ではあるが Politeness Strategy を含んでいないため，0.507 と中間の値が算出されている．3 番目の文章の Politeness 値は 0.122 と低い値となっている．これは，文中に含まれる “So, what...” の部分が Politeness 値を下げる Politeness Strategy である Direct start に該当しており，相手を不愉快にさせる可能性の高い表現であるためである．

Stanford Politeness は，大きく分けて 2 つの処理からなる．図 1 に処理方法の概要を示す．

まず，入力テキストの構文解析をおこない，テキストに含まれる単語と Politeness Strategy を検出し BOW (Bag of Words) ベクトルを作成する．なお，Stanford Politeness には構文解析機能がないため，本研究では構

図 1. Stanford Politeness における Politeness 値の算出方法

文解析ツールである Stanford Parser[2] を用いて入力テキストの構文解析をおこなう．

次に，作成した BOW ベクトルを機械学習アルゴリズムの 1 つであるサポートベクタマシン (SVM)[6] により学習させる．SVM の学習データには，Stanford Politeness Corpus[4] が用いられている．Stanford Politeness Corpus は，Wikipedia ユーザのトークページ[3]から抽出した 4,353 の質問と Stack Exchange[4]から抽出した 6,604 の質問に対して，人手により Politeness を評価した値からなるデータの集合である．最後に，入力テキストを SVM で判別し Politeness 値を算出する．

3. 先行研究の概要

本章では，[18, 19] における分析内容と結果について概説する．

3.1 対象プロジェクト

[18, 19] では，大規模 OSS である Apache HTTP Server プロジェクトにおける開発者メーリングリストから抽出したメールデータに対して Stanford Politeness を適用し，開発者の Politeness とプロジェクト離脱（あるいは継続性）との関係を理解するためのケーススタディを実施した．分析対象は，プロジェクトが設立された 1995 年 9 月から 2016 年 3 月時点の約 21 年間分のメールデータである．

Apache HTTP Server プロジェクトを選択した主な理由は以下の通りである．

- 20 年以上存続しているプロジェクトでありコア開発者の中にも離脱者が多数存在すること
- 企業とは異なり離脱の要因と考えられる業績や役職等による社会的ストレスの影響を受けにくいこと
- 同じプロジェクトを対象とした先行研究が数多くありケーススタディにより得られる知見の妥当性を確保しやすいこと
- 開発者メーリングリストを通じてやり取りされたメールデータがすべてアーカイブされており，かつ，容易に取得可能であること

3.2 分析内容

ケーススタディでは，以下の 2 点を主に分析した．

(1) プロジェクトに長期間参加した開発者と短期間参加した開発者の Politeness の違い

(2) プロジェクトに長期間参加したコア開発者の Politeness の経時的変化とプロジェクト離脱との関係

(1) の分析を実施するにあたり，まずプロジェクトに参加した開発者を定義する必要があった．開発者メーリングリストでは，投稿を 1 度だけおこなってそれ以降は 1 度も投稿しないような開発者も多数存在するため，そのような開発者を「プロジェクトに参加した」と見なすのは適切ではない．また，開発者の投稿数が 1 度のみといった様な少ない場合に関しても，プロジェクトに貢献していると見なすことは適切ではない．したがって，90 日以内に 1 度以上投稿した，かつ総投稿数 10 件以上の開発者を「プロジェクトに参加した」と見なし,「プロジェクトに参加した」開発者の投稿（他者の投稿に対する返信も含む）を分析の対象とした．次に，プロジェクトに長期間・短期間参加した開発者を区別するために，前述の条件を満たし，かつ，270 日以上開発者メーリングリストに投稿をおこなった開発者を長期開発者，それ以外を短期開発者と定義した．その結果，長期開発者 183 名，短期開発者 173 名を抽出した．

(2) の分析では，長期開発者 183 名の中から投稿数の多い順に上位 10 名をコア開発者と定義し，コア開発者の Politeness の経時的変化とプロジェクト離脱との関係を分析した．

[2] http://nlp.stanford.edu/software/lex-parser.shtml
[3] http://en.wikipedia.org/wiki/Wikipedia:User_pages
[4] http://stackexchange.com/about

3.3 分析結果

3.3.1 Politeness 値：長期開発者 vs. 短期開発者

結果の詳細については紙面の関係上割愛するが，検定の結果，長期開発者の Politeness は短期開発者よりも有意に低い値を示すことが分かった．プロジェクトに長期的に参加する開発者は，他の開発者との良好な関係が構築できているため，Politeness を過度に意識する必要がない，あるいは，他人行儀な改まった態度を示す必要がないということが示唆される．

Ortu らの研究 [11, 10] では，不具合修正プロセスにおける開発者の Politeness と不具合修正時間との関係を分析しているが，高い Politeness 値を示す議論を通じて解決される不具合は比較的長い時間を要することが明らかになっており，本ケーススタディの結果とも共通する現象を捉えていると思われる．

3.3.2 コア開発者の Politeness 値の経時的変化

図 2 に，コア開発者が投稿したメール（他の開発者の投稿に対する返信を含む）の Politeness 値を時系列にプロットしたグラフを示す．青の折線が Politeness 値，赤の折線が投稿件数を示している．なお，単純に月毎にデータを集計してプロットすると投稿件数の少ない期間の外れ値の影響を受けてグラフが読み取りづらくなるため，Politeness 値，メール件数ともに，3 か月間の移動平均をプロットしたグラフであることに注意されたい．

図 2 から見て取れるように，Dev#1 を除いて，コア開発者であっても分析対象である約 21 年間のすべての期間において活動（メール投稿）している開発者は存在せず，どこかの時点でプロジェクトを離脱している．また，コア開発者間で Politeness 値に個人差があること，時期により Politeness 値が変動することが確認された．特に，プロジェクトを離脱する際に Politeness 値が急激に変動する開発者が確認された．表 3 に，プロジェクトから離脱したコア開発者のプロジェクト離脱 6 ヶ月前から離脱 3ヶ月前までの期間とプロジェクト離脱直前 3ヶ月間の期間のメール投稿数と Politeness 値の平均値を示す．表 3 から，Dev#2，Dev#3，Dev#4，Dev#7，Dev#8，Dev#10 の 6 名のコア開発者に関して，プロジェクト離脱直前に 0.11〜0.20 の Politeness 値の変動（上昇，もしくは低下）が確認された．

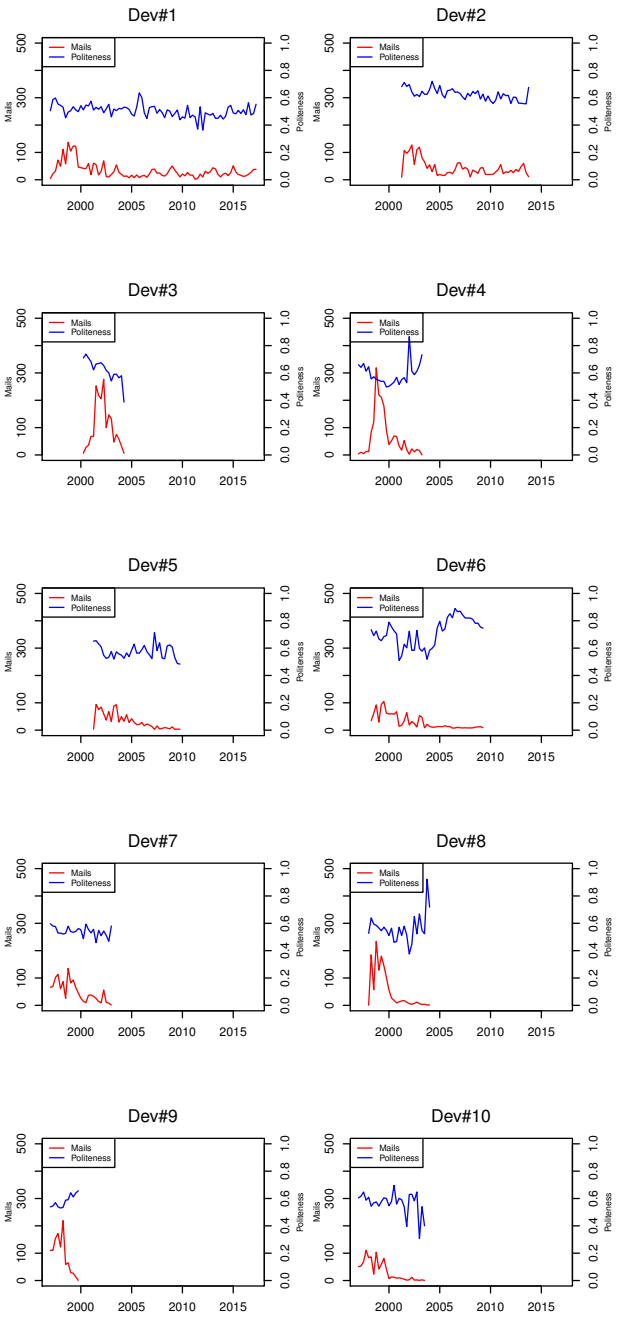

図 2. コア開発者が投稿したメールの Politeness 値の時系列変化（3ヶ月移動平均）

表 3. 離脱したコア開発者のメール投稿数と Politeness 値（月平均）

	(A) 離脱 6ヶ月前から離脱 3ヶ月前まで		(B) 離脱直前 3ヶ月間		Politeness 値の変動 (B-A)
	メール投稿数	Politeness 値	メール投稿数	Politeness 値	
Dev#2	25.7	0.56	11.0	0.68	0.12
Dev#3	32.3	0.58	6.0	0.39	-0.19
Dev#4	15.7	0.66	0.3	0.73	0.07
Dev#5	4.0	0.49	3.0	0.48	-0.01
Dev#6	13.0	0.75	9.0	0.75	0.00
Dev#7	7.7	0.47	1.3	0.58	0.11
Dev#8	1.0	0.92	1.7	0.72	-0.20
Dev#9	13.0	0.64	0.7	0.66	0.02
Dev#10	2.7	0.54	0.3	0.40	-0.14

Politeness の急激な変動が開発者がプロジェクトを離脱する予兆なのであれば，開発者の離脱の予兆を検知する手法を構築する上での重要な手がかりとすることができる．離脱予兆の検知手法を確立できれば，プロジェクト管理者等が開発者の離脱を未然に阻止するための対策を講じることが可能になる．

3.4 先行研究の問題点

図 2 に示したように，数年間プロジェクトに参加しているコア開発者であってもプロジェクトを離脱することが分かった．また，プロジェクトを離脱する直前には Politeness 値が大きく変動する，すなわち，コア開発者のコミュニケーション内容の質が変化していることが分かった．

しかしながら，[18, 19] ではコア開発者の投稿内容までは詳細に分析できておらず，なぜ開発者の Politeness 値が急激に変動したのかについては不明なままである．そのため，Politeness の急激な変動（コミュニケーションの質的変化）がプロジェクトの離脱の予兆と見なすことができるか，すなわち，Politeness の急激な変動と開発者の離脱との間に明確な関係が存在するのかが不透明である．

そのため本稿では，プロジェクト離脱の直前に Politeness の急激な変動を示したコア開発者の中から，Politeness 値が低下した開発者と Politeness 値が上昇した開発者をそれぞれ 1 名選定し，プロジェクト離脱の直前の開発者の投稿内容を目視により確認・分析する．

4 目視による調査

コア開発者がプロジェクトを離脱する直前に Politeness が大きく変動している理由を明らかにするために，開発者メーリングリストに投稿された離脱直前 3ヶ月間分のメールを目視により調査する．

4.1 調査対象

対象開発者の離脱直前 3ヶ月間の以下の項目を調査対象とし，Politeness の急激な変動とプロジェクト離脱理由を明らかにする．

- 開発者が投稿した内容（メール本文）
- 投稿の Politeness 値
- 本文に含まれている Politeness Strategy

次に，3.3.2 頁で述べた Politeness 値が急激に変動した開発者の中から調査対象者を選出する．表 3 より，Dev#4 や Dev#10 は離脱直前 3ヶ月間に投稿したメール数が著しく少ないことがわかる．同じ開発者が投稿したメールであっても，Politeness 値はメールによって大きく異なるため，Dev#4 や Dev#10 は外れ値によって Politeness 値の平均値が変動した可能性がある．そのため調査対象者は，離脱したコア開発者のうち，投稿数の多い Dev#2 および Dev#3 とする．Dev#2 は，離脱直前に Politeness が上昇しているパターンの開発者，Dev#3 は，離脱直前に Politeness が低下しているパターンの開発者である．

次節ではこれら 2 名の開発者を対象とした調査結果について詳しく述べる．

4.2 調査結果

本節では，投稿内容を目視した結果について述べる．まず，離脱直前に Politeness が低下しているパターンの開発者である Dev#3 の調査結果を示す．Politeness が急激に低下するということは，他の開発者のネガティブフェイスを侵害する表現が増えていることを意味しており，Dev#3 とその周辺の開発者の間に何らかの対立や緊張状態が生まれている可能性がある．

次に，離脱直前に Politeness が上昇しているパターンの開発者である Dev#2 の調査結果を示す．Politeness の上昇は，コミュニティに新たなメンバを迎える際，新人に友好的であることを古参メンバが示すための態度として我々の社会生活でも経験する事象であるが，Dev#2 の場合は直後に離脱している．これまで友好関係を築いていた他の古参メンバとの対立が生じたり，何らかの理由でコミュニティに一定の距離を取り始めた結果,「よそよそしく」なっている可能性がある．

4.2.1 Dev#3

表 4 に，Dev#3 の投稿内容の例，本文に含まれる Politeness Strategy，および，本文の Politeness 値を示す．

プロジェクト離脱直近 3ヶ月分の Dev#3 の投稿をすべて目視した結果，この時期の Dev#3 は他の開発者と意見が折り合えず，口論にまで発展していることが分かった．口論の原因は，他の開発者が変更したコードに不具合が含まれていることを Dev#3 が指摘しているが，他の開発者からは理解が得られないといった状況によるものである．

例えば，表 4 の 1 つ目の投稿は，コードの変更によりメモリリークを生み出す可能性をその理由とともに指摘している．本文中に含まれる “actually” や “In fact” は，Politeness 値を大きく下げる Politeness Strategy である Factuality（表 1 参照）に該当する．Politeness 値を上げる Politeness Strategy である 1st person や 1st person start など（I, my, we など）も数多く含むが，Factuality の効果を相殺するほどではない．Politeness Strategy 以外にも，“VERY big mistakes” や “too dangerous” といった相手のネガティブフェイスを損害する直接的な表現が目立つ．Stanford Politeness Corpus には，Politeness Strategy 以外にも単語単位で人手による評価が加えられているため，ネガティブな文脈で使われる単語等にも Politeness 値を大きく下げる効果があると思われる．

2 つ目の投稿は，Politeness 値を大きく下げる Politeness Strategy が含まれておらず，1 つ目の投稿と比較すると高い Politeness 値を示している．しかし，1 つ目の投稿と同様に，メモリリークを引き起こす変更に対して強く反対していることが読み取れる．また，1 つ目の投稿と同様に，相手のネガティブフェイスを明示的に侵害する表現がいくつか見られる．

3 つ目の投稿では，1st person よりも 2nd person の Politeness Strategy が多用されている．また，Politeness 値を大きく下げる Politeness Strategy の Factuality (“in fact”) も使用している．本文からは，メモリリークを引き起こす問題について強く相手を批判している様子が読み取れる．

これらのことから，この時期に Dev#3 には納得できない変更があり強く反対意見を述べた結果，Politeness 値を大きく下げることに繋がっていることが分かった．したがって，この変更に関するプロジェクトの方針に賛同できなかったため，Dev#3 は直後にプロジェクトを離脱するという選択を取ったと思われる．

4.2.2 Dev#2

表 5 に，Dev#2 の投稿内容の例，本文に含まれる Politeness Strategy，および，本文の Politeness 値を示す．

プロジェクト離脱直近 3ヶ月分の Dev#2 の投稿をすべて目視した結果，Dev#2 がおこなった変更に関してフィードバックした他の開発者に対する返信や，他の開発者の投稿したパッチに対して賞賛する内容の投稿が確認できた．

例えば，表 5 の 1 つ目の投稿は，パッチの投稿に感謝の意を伝えるためのものである．本文中に含まれる Politeness Strategy は Deference の “great” と Gratitude の “thanks” のみであるが，それ以外の情報が少ないため高い Politeness 値を示している．

2 つ目の投稿は，他の開発者が作成したパッチに対して感謝の意を伝えるとともに，さらなるパッチの検証を他の開発者に依頼するものである．Hedges や Gratitude など Politeness 値を大きく上げる Politeness Strategy が複数含まれている．一方，Politeness 値を大きく下げる Factuality（“really”）も含まれているが，前向きな文脈の中で使用されている（“I’d really prefer we ∼.”）

表 4. Dev#3 の投稿内容の例（本文中の太字は Politeness Strategy に該当）

id	本文	Politeness Strategy	スコア
1	**I** am **actually** pretty **sure** that allocating brigades out of the bucket_allocator is a VERY big mistake. **In fact**, **I** am close to asking that change to be backed out because it is too dangerous. When **we** first designed buckets and bucket brigades, **we** made one VERY clear distinction. Bucket_brigades are allocated out of a pool, because that stops us from leaking memory. Buckets shouldn't be allocated out of a pool, because nobody knows which pool to allocate them from. Basically, **we** said that in Apache, it is safe to just drop a bucket brigade because the pool cleanups will free the memory. It is not safe to just drop a bucket, it must either be left on the brigade to be cleaned up by the brigade_cleanup function, or it must be destroyed when it is removed from the brigade. By allocating bucket_brigades out of the bucket_allocator, **we** have removed the protection that the pool cleanups used to give us. **I** may be wrong about how the bucket_allocator works, but **I** don't believe **I** am. Basically, this change will make Apache leak memory on some requests, and it goes against the original design of buckets and bucket brigades. **In fact**, allocating the brigades out of a pool was a requirement, because some people at the meeting were afraid of memory leaks with bucket brigades.	Factuality, 1st person pl., 1st person, 1st person start	0.08
2	**I** am vetoing this change until **I** get a clear description of the problem it is solving. **I** have explained multiple times why it is a bad change, but **I** can't offer any other solutions until **I** know why it is needed. It will not. If **you** _ever_ create a brigade with a NULL pool, **you** will have memory leaks. **I** dare say that every filter ever written will leak if the feature implemented with this patch is ever used. **I** have already posted one code segment in the core_output_filter that proves the memory leak exists. Trust me when **I** tell **you** that there are hundreds more, and some of them are in 3rd party modules that **you** don't control. ++1, but they should go into an external library so that projects like serf have access to them. Happily, **I** don't have to worry about that.	1st person, 1st person start, 2nd person	0.38
3	No, that is a solution, **in fact**, it was part of the original design. What **you**'ve done is said "The brigade may not live long enough", but **you** haven't explained how that is the case. **You** also haven't explained why moving the buckets to another brigade won't work. Which **I** have a very hard time believing, because that is how Apache works today. **You** also haven't answered the problem that even _WITH_ this change, **you** still need to move the buckets to a new brigade. At the end of the day, this change doesn't get **you** anything, because **you** _still_ have to do the brigade migration that **you** say **you** can't do.	Factuality, 1st person, 2nd person, 2nd person start	0.05

ため大きな影響はなかったと考えられる．結果として，Politeness 値が 0.99 と非常に高い値を示しているものと思われる．

3 つ目の投稿では，メールクライアント Thunderbird とプロバイダの相性から問題が生じ，メールの復旧作業に入るために活動をできないことをコミュニティに伝えている．また，他の開発者へのアドバイスも含まれており，Gratitude や Apologizing，Hedges といった Politeness を上げる Politeness Strategy を多く含んでいる．

このように，表 5 以外のものを含めても，Dev#3 とは異なり Dev#2 の投稿内容からはプロジェクト離脱の予兆を示す言動を読み取ることはできなかった．しかしながら，Politeness の上昇後に離脱するという現象は不可解である．何らかの理由によりすでに離脱を決めている，あるいは，離脱しようとしている過程での人間の意識の現れとも考えられるが推測の域を出ない．

コア開発者の離脱直前の 3ヶ月分のメールデータは 1 名で数十件を目視する必要があった．また，メールによって非常に長い本文を含む場合もあったため，本稿ではまず，Politeness が上昇したパターンを持つ Dev#2 のみを分析対象とした．同様のパターンを持つ開発者は Dev#2 以外にも，Dev#4 や Dev#6，Dev#8 が該当するため，今後はこれらの開発者の分析を進め離脱理由を明らかにしたい．ただし，いずれの開発者の離脱理由は投稿内容からは読み取れない可能性もある．その場合は，他のプロジェクトを対象に追加分析をしたり，可能であれば離脱した開発者へのインタビューを実施し，Politeness の大きな上昇とプロジェクト離脱との関係を明らかにしたい．

5 関連研究

Politeness 分析に関しては Ortu らの研究 [11, 10] が関連研究として挙げられる．OSS 開発で利用されている不具合管理システムのコメントデータに対し Politeness 分析を適用し，開発者の Politeness と不具合修正時間の関係を分析したものである．分析の結果，Politeness の高いコメントが寄せられた不具合は，Politeness の低いコメントが寄せられた不具合よりも修正時間が長く必要となることを明らかにしている．3 章において述べたよう

表 5. Dev#2 の投稿内容の例（本文中の太字は Politeness Strategy に該当）

id	本文	Politeness Strategy	スコア
1	Looks **great**! **Thanks** for the patch :)	Gratitude, Deference	0.90
2	**Thanks** for the feedback, Kaspar. Other than shifting the one #ifdef case to the correct line of code, **I believe** the patch is complete, **I**'ve tried to state **my** case for no further changes. If **you would like** to **propose** further changes, **I**'d **really prefer we** defer these to 2.2.24-dev so we can T&R already. **I**'m pretty **sure you** already agree with me that the flexibility to disable a particular cipher in light of exploit research in the very fresh openssl code base makes this patch pretty critical to release for legacy, as well as stable. Even if **you** can give the patch **your** +1, we still need one more individual to chime in before **we** can finish this backport, and as **I** mention, OpenSSL's CVE-2012-2333 suggests that this patch is a showstopper. Can one more person take a pass through the code, since Stefan didn't have a chance to re-review this week?	Gratitude, Hedges, Factuality, 1st person pl., 1st person, 2nd person	0.99
3	**Sorry**, **I**'m not ignoring the list (entirely). Seems Thunderbird and **my** ISP have decided not to dance anymore and it looks like **I**'m spending Thursday doing some fundamental email restructuring (sigh). Hopefully **I**'ll have the list traffic back sometime by Friday, **thanks** to Jeff, Reindl, and Steffen for **your** review, **I** didn't read Ruediger, Rainer, or Jim as voting one way or another - even with **my** vote there is still a missing PMC +1 for release. Rainer, are **your** observations a regression? Because the security issues are relatively minor, **I** took this opportunity to update to a slightly refreshed autoconf (not a radically new version, nor the last in that version minor which barfs on our configure. in and m4 files)... so it's possible **I** had introduced this. If not a regression, please confirm.	Gratitude, Hedge, Apologizing , 1st person pl., 1st person, 2nd person, Direct start	0.87

に，Politeness の高いコミュニケーションは一見望ましいように思われるが，問題を効率的に解決するという観点では必ずしも必要ではなく，チームワークの高いチームでしばしば見られるフランクな会話の方が良い場合があることを示唆するものである．我々の先行研究においても [18, 19]，プロジェクトに長期間参加する開発者の Politeness はその他の一般開発者に比べて統計的に有意に低いことを示しており，Ortu らの研究とは目的が異なるものの得られた知見は矛盾するものではない．

OSS プロジェクトへ参加する開発者の離脱阻止の支援という観点では，Steinmacher らの研究 [13, 14] が挙げられる．Steinmacher らの研究は，プロジェクトへこれから参加したいと考えている開発者の参入を妨げる要因について分析したものである．ドキュメントの不足や不親切な対応など，新人にとって障壁となる計 58 種類の要因を明らかにしている．本研究は古参メンバの離脱要因を理解し離脱の予兆を検知する手法を構築することを目的としており，アプローチが大きく異なるものの OSS プロジェクトの安定運営の支援という観点では互いに補完しあうものである．

6. まとめと今後の課題

本研究では，開発者のプロジェクト離脱と Politeness の変動の関係を明らかにするために，Politeness の急激な変動を示したコア開発者のメールを目視した．目視の結果，Politeness が大きく低下した開発者はコミュニティ内で意見衝突をしていることが確認された．開発者間の意見衝突は開発者のプロジェクト離脱を促す要因の 1 つとして考えられるため，Politeness の大きな低下は開発者のプロジェクト離脱の予兆とみなすことができる．プロジェクト管理者は Politeness が大きく低下した開発者に対してなんらかの処置をすることで，その開発者の離脱を未然に防止することができると考えられる．一方で，離脱直前に Politeness が大きく上昇した開発者は他の開発者に感謝をしていることが確認されたが，明確な離脱理由を見つけることはできなかった．そのため，プロジェクトの離脱と Politeness の上昇に普遍的関係が存在するかを明らかにするために，さらなる分析をおこなう必要がある．

今後の課題としては，本研究で分析した対象開発者以外の開発者に対しても分析をおこない，開発者のプロジェクト離脱と Politeness の変動の関係についての知見を得たいと考えている．本研究では開発者の離脱直前の 3ヶ月間を分析対象としたが，開発者の離脱の予兆を察知する期間として適切であるかは確かではない．今後より詳細な分析をおこない，開発者の離脱の予兆を察知するための適切な期間を明らかにしていきたい．

7. 謝辞

本研究の一部は，文部科学省科学研究補助金（基盤(C): 15K00101）による助成を受けた．

参考文献

[1] John Anvik, Lyndon Hiew, and Gail. C. Murphy. Who should fix this bug? In *Proceedings of the 28th international conference on Software engineering (ICSE '06)*, pp. 361–370, 2006.

[2] Christian Bird, Alex Gourley, Prem Devanbu, Anand Swaminathan, and Greta Hsu. Open borders? immigration in open source projects. In *Proceedings of the 4th International Workshop on Mining Software Repositories (MSR '07)*, 2007.

[3] Penelope Brown and Stephen C Levinson. *Politeness: Some Universais in Language Usage*. Cambridge University Press, Cambridge, UK, 1987.

[4] Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A computational approach to politeness with application to social factors. In *Proceedings of 51st Annual Meeting of the Association for Computational Linguistics (ACL '13)*, 2013.

[5] Mohammad Gharehyazie, Daryl Posnett, Bogdan Vasilescu, and Vladimir Filkov. Developer initiation and social interactions in oss: A case study of the apache software foundation. *Empirical Software Engineering*, Vol. 20, No. 5, pp. 1318–1353, 2015.

[6] Steve Gunn. Support vector machines for classification and regression. Technical Report, School of Electronics and Computer Science, University of Southampton, 1998.

[7] IPA（独立行政法人情報処理推進機構）. 第 3 回オープンソースソフトウェア活用ビジネス実態調査（2009 年度調査）, 2009.

[8] Chris Jensen and Walt Scacchi. Role migration and advancement processes in ossd projects: A comparative case study. In *Proceedings of the 29th International Conference on Software Engineering (ICSE '07)*, ICSE '07, pp. 364–374, 2007.

[9] Audris Mockus, Roy T Fielding, and James D. Herbsleb. Two case studies of open source software development: Apache and mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, Vol. 11, No. 3, pp. 309–346, 2002.

[10] Marco Ortu, Bram Adams, Giuseppe Destefanis, Parastou Tourani, Michele Marchesi, and Roberto Tonelli. Are bullies more productive?: Empirical study of affectiveness vs. issue fixing time. In *Proceedings of the 12th Working Conference on Mining Software Repositories (MSR '15)*, pp. 303–313, 2015.

[11] Marco Ortu, Giuseppe Destefanis, Mohamad Kassab, Steve Counsell, Michele Marchesi, and Roberto Tonelli. Would you mind fixing this issue? an empirical analysis of politeness and attractiveness in software developed using agile boards. In *16th International Conference on Agile Processes in Software Engineering and Extreme Programming (XP 2015)*, pp. 129–140. Springer International Publishing, Cham, Switzerland, 2015.

[12] Jin-woo Park, Mu-Woong Lee, Jinhan Kim, Seungwon Hwang, and Sunghun Kim. Costriage: A cost-aware triage algorithm for bug reporting systems. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI-11)*, pp. 139–144, 2011.

[13] Igor Steinmacher, Tayana Uchoa Conte, Christoph Treude, and Marco Aurélio Gerosa. Overcoming open source project entry barriers with a portal for newcomers. In *Proceedings of the 38th International Conference on Software Engineering (ICSE '16)*, pp. 273–284, 2016.

[14] Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. Social barriers faced by newcomers placing their first contribution in open source software projects. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*, pp. 1379–1392, 2015.

[15] Yunwen Ye and Kouichi Kishida. Toward an understanding of the motivation open source software developers. In *Proceedings of the 25th International Conference on Software Engineering (ICSE '03)*, pp. 419–429, 2003.

[16] Minghui Zhou and Audris Mockus. What make long term contributors: Willingness and opportunity in oss community. In *Proceedings of the 34th International Conference on Software Engineering (ICSE '12)*, pp. 518–528, 2012.

[17] 宇佐美まゆみ. ポライトネスという概念. 月刊 言語, Vol. 31, No. 1, pp. 100–105, 2002.

[18] 宮崎智己. OSS 開発における活動継続性と Politeness の関係. Technical report, 和歌山大学システム工学部 2016 年度卒業論文, 2017.

[19] 宮崎智己, 山谷陽亮, 東裕之輔, 大平雅雄. OSS 開発コミュニティの進化の理解を目的としたコミュニケーション分析：Politeness 分析適用の試み. 情報処理学会 マルチメディア，分散，協調とモバイル (DICOMO2016) シンポジウム論文集, pp. 703–713, 2016.