

OSS 開発における管理者と修正者の社会的関係を 考慮した不具合修正時間予測

吉行 勇人 大平 雅雄 戸田 航史

大規模なオープンソースソフトウェア (OSS) プロジェクトでは、日々大量の不具合が報告されている。すべての不具合を修正することは現実的ではないため、OSS プロジェクトの管理者は、次期バージョンのリリースまでに修正可能な不具合を決定するために、不具合修正に要する時間を見積もる必要がある。本研究では、不具合修正タスクを開発者に割り当てる管理者と不具合修正タスクを担当する修正者の関係に着目し、ランダムフォレスト法およびロジスティック回帰分析を用いて不具合修正時間予測モデルを構築する。評価実験の結果から、1 日以内に不具合修正が完了するか否かの予測において、先行研究の予測モデルに比べ、管理者と修正者の関係を考慮した予測モデルが修正時間予測の精度向上に寄与することを示す。

A large number of bugs are reported to a large-scale open source software (OSS) project. Since it is very difficult to fix all the bugs, a manager in the project needs to know time to fix bugs to decide which bugs should be fixed before the next release. In this paper we focus on social relationships between assignors and assignees and construct prediction models based on a random forest method and logistic regression, in order to predict the time to fix bugs. As a result of our experiment, we show that our model predicts more accurately whether a bug is fixed within a day or not, compared with the existing prediction model.

1 はじめに

大規模なオープンソースソフトウェア (OSS) プロジェクトには、日々大量の不具合が報告されている。すべての不具合を修正することは現実的ではないため、OSS プロジェクトの管理者は、次期バージョンのリリースまでに修正可能な不具合を決定するために、不具合修正に要する時間を見積もる必要がある。また、近年では、OSS を利用したソフトウェア開発が一般的になりつつあり、ベンダ企業はどの不具合がいつ頃までに OSS プロジェクト側で修正されそうかを知る必要がある。このような背景から、OSS 開発

における不具合修正時間の予測に関する研究が盛んに行われている [6][11]。特に近年では、OSS 開発者の社会的関係に着目した不具合修正時間予測モデル [2][8][12] が提案されているが、予測モデルの精度には改善の余地がある。

本研究では、不具合修正タスクを開発者に割り当てる管理者と不具合修正タスクを担当する修正者の社会的関係に着目する。具体的には、修正者はタスクを割り当てた管理者によってタスクの優先順位付けを行っていると考え、「修正者によるタスクの優先順位付けが割り当てから修正までにかかる時間に影響を与える」という仮説を立てる。仮説の検証のため、ランダムフォレスト法およびロジスティック回帰分析を用いて不具合修正時間予測モデルを構築する。評価実験の結果から、1 日以内に不具合修正が完了するか否かの予測において、先行研究の予測モデル [12] に比べ、修正者によるタスクの優先順位付けを考慮した予測モデルが不具合修正時間予測の精度向上に寄与することを示す。

Predicting Bug-Fix Time Using Social Relationships between Assignors and Assignees in OSS Development.

Hayato Yoshiyuki, Masao Ohira, 和歌山大学, Wakayama University.

Koji Toda, 福岡工業大学, Fukuoka Institute of Technology.

コンピュータソフトウェア, Vol.32, No.2 (2015), pp.128-134. [研究論文 (レター)] 2014 年 9 月 24 日受付.

2 関連研究

2.1 OSS 開発における不具合修正時間の予測

OSS プロジェクトに報告された不具合の修正が完了する期間を予測する先行研究には、[6][11]などが挙げられる。これらの研究では、不具合そのものに関連する基本的な情報（優先度や重要度など）に基づいて予測モデルを構築している。しかしながら、予測可能な期間の粒度には限界があり（例えば、6ヶ月以内に修正されるかどうか、など）OSS プロジェクト管理者や OSS を利用したシステム開発を行っているベンダ企業のニーズを十分に満たせておらず、より細かな粒度での予測が必要とされている。

2.2 不具合修正における開発者間の社会的関係

前述の研究の結果を受けて、近年では、不具合修正プロセスに関与する開発者らの属人性や社会性に着目し、予測精度の向上および予測期間の細粒度化が試みられている。例えば Anh ら [2] は、不具合修正プロセスに関与する開発者の経験や発言量（コメント数）などを予測モデルの説明変数として用いている。

Ohira ら [8] は不具合修正プロセスにおける開発者らの社会的関係（不具合報告者、不具合修正管理者、不具合修正者の組み合わせ）のパターン化を実施しており、正木ら [12] は、Anh らの属人的な変数を用いた予測モデルに Ohira らがパターン化した社会的関係を導入することで、不具合修正時間の予測精度と予測期間の粒度を改善している。

2.3 不具合修正における修正者の割り当て問題

OSS 開発における過半数の不具合は、有能な少数の修正者によって修正されていることが知られている [7]。そのような状況では、修正者は様々な管理者から割り当てられた不具合修正タスクを同時に処理しなければならない場合がある。実際、Ohira ら [9] の Eclipse プロジェクトを対象とした分析では、どの管理者にタスクを割り当てられたかによって修正者がタスクに要する時間に違いがあることを明らかにしており、管理者と修正者の社会的関係が、不具合修正の優先順位および不具合修正時間に大きな影響を与え

得ることを示唆している。

そこで本研究は、OSS 開発者の属人性および社会的関係を考慮した正木ら [12] の予測モデルをベースとして、予測精度のさらなる向上を目指す。本研究では特に、特定の管理者と修正者の組み合わせを考慮した予測モデルを構築し、管理者と修正者の社会的関係が予測精度の向上に寄与することを示す。

3 不具合修正時間予測モデルの構築

3.1 概要

本研究では正木ら [12] と同様に、指定期間を 1 日以内、1 週間以内、1ヶ月以内に設定し、それぞれを目的変数として不具合修正時間予測モデルを構築する。説明変数には、「特定の管理者と修正者のペアが優先して完了させたタスクの割合」を新たに追加し、不具合票から得られる合計 20 種類のメトリクスを用いる。

3.2 利用するメトリクス

3.2.1 不具合票に関連するメトリクス

不具合票に関連するメトリクスとは、不具合の報告者によって入力された不具合に関する情報（不具合が発生したコンポーネント (Component)、優先度 (Priority)、重要度 (Severity) 等) や、利用者または開発者らによる議論に関する情報（コメント数 (Comments)、コメントの平均文字数 (ComWords)) 等である。例えば、Mockus ら [7] は基幹となるモジュールや影響力の高いモジュールに対する不具合修正時間は短いことを明らかにしている。予測モデルで用いる不具合票に関連するメトリクスの概要を表 1 に示す。

3.2.2 時間に関連するメトリクス

時間に関連するメトリクスとは、不具合の報告から割りが行われるまでの時間や、不具合の修正者が決定した日時に関する情報である。例えば、不具合報告が行われた日の曜日が修正時間に関係するという報告などがある [1]。予測モデルで用いる時間に関連するメトリクスの概要を表 2 に示す。

3.2.3 人に関連するメトリクス

人に関連するメトリクスとは、不具合修正プロセスに関わる参加者の情報や、先行研究で用いられている不具合割り当てパターン (Patterns)、本研究で新たに用

表 1 不具合票に関連するメトリクスの概要

変数	尺度	説明
Component	名義	対象コンポーネント
Milestone	名義	マイルストーンの記載有無
Priority	名義	不具合の優先度
Severity	名義	不具合の重要度
DependsOn	比例	依存関係にある不具合の数
Blocks	比例	修正を妨げている不具合の数
CC	比例	不具合の同報者数
Discriptions	比例	不具合に関する概要の文字数
Comments	比例	コメントの数
ComWords	比例	コメントの総文字数
Attachments	比例	添付ファイルの数

表 2 時間に関連するメトリクスの概要

変数	尺度	説明
AssignTime	比例	修正者の決定までに要した時間
AssignedMonth	比例	修正者が決定した月
AssignedDay	比例	修正者が決定した日
AssignedWE	名義	修正者が決定した日が週末か否か

表 3 人に関連するメトリクスの概要

変数	尺度	説明
Reporter	名義	報告者のメールアドレス
Assignor	名義	管理者のメールアドレス
Fixer	名義	修正者のメールアドレス
TaskPref	比例	優先して完了させたタスクの割合
Patterns	名義	不具合割合パターン [12]

いる「管理者と修正者のペアが優先して完了させたタスクの割合 (TaskPref)」などである。例えば、多くの不具合を報告している利用者や開発者が報告を行う場合、詳細な不具合の再現方法等の修正に必要な情報を不具合票に登録することができるため、修正者は迅速に不具合修正に取りかかることができる [3]。予測モデルで用いる人に関連するメトリクスの概要を表 3 に示す。

TaskPref は、本研究で注目する修正者のタスク優先順位付けに関わるメトリクスであり、不具合管理者 A と修正者 B の組み合わせにおける TaskPref は以下の式で算出される。

$$TaskPref_{A-B} = \frac{Task_{A-BPref}}{Task_{A-BAU}} \quad (1)$$

$Task_{A-BAU}$ は A が B に割り当てたタスクの総数

であり、 $Task_{A-BPref}$ は A 以外の管理者から割り当てられたタスクよりも優先して完了させたタスク数である。例えばあるタスク (タスク 1) よりも後に割り当てられ、先に完了させたタスク (タスク 2) が存在するとき、タスク 2 はタスク 1 よりも優先されたとして扱う。式 1 を全ての管理者と修正者のペア毎に適用し、TaskPref としてモデル構築に用いる。本研究では、タスク優先順位付けに関わるメトリクスである TaskPref が予測精度の向上に寄与するかどうかを評価実験により確かめる。

3.3 不具合修正時間予測のためのモデル

本研究では指定期間内に不具合修正が完了するか否かの予測を行うため、目的変数を離散値として予測可能なランダムフォレスト法およびロジスティック回帰分析を用いる。ただし紙面の都合上、手法の説明と実験結果は予測精度向上率の高かったランダムフォレスト法のみを記述する。

ランダムフォレスト法は、多数の分類木を用いて集団学習を行う手法であり、Breiman によって提案された [4]。ランダムフォレスト法は各説明変数の重要度を算出でき、本研究で用いるメトリクスの有用性を確かめることができるため、本研究で採用した。ランダムフォレスト法は先行研究の不具合修正時間の予測 [12] [6] [2] においても用いられている。

4 評価実験

4.1 概要

評価実験は、本研究で用いる「優先して完了させたタスクの割合 (TaskPref)」が、不具合修正時間予測モデルの予測精度向上に寄与するか否かを確かめることを目的としている。前節で説明したメトリクスおよび予測モデルを用いて、指定期間内に不具合修正が完了するか否かを判別する予測モデルを構築し、まず、TaskPref を説明変数として用いる場合と用いない場合で予測精度の差があるか否かを確かめる。さらに先行研究のモデル [12]、先行研究に TaskPref を追加した併用モデル、機械学習を用いないモデル (非機械学習モデル) の 3 種との比較を行う。

4.2 データセット

本実験では、Eclipse Platform プロジェクトで利用されている不具合管理システム Bugzilla に 2001 年 10 月から 2012 年 10 月の間に報告された不具合票をデータセットとして用いる。本実験では、取得した不具合票に対してデータクリーニングを行い、正木ら [12] と同様の条件に加え、以下に当てはまる不具合を除いた計 18,464 件をデータセットとして用いた。

- OSS 開発における不具合報告の中には、長期間にわたって放置されるものが数多く存在する [5] [10] ため、予測モデルの構築に大きな影響を与える。そこで、一定期間以上放置されていると考えられる不具合票を外れ値として除外する。今回は第三四分位+ $IQR \times 1.5$ 以上離れている値 (箱ひげ図における外れ値の定義の 1 つ) を外れ値とした。その結果、不具合修正に 81.94 日以上かかっている不具合票が除外された。

4.3 実験手順と評価指標

実験の手順については先行研究 [12] に従う。メトリクスには表 1~表 3 で示した計 20 種類を用いる。提案手法として、このうち Patterns を除外したモデル (TaskPref 有モデル) と、TaskPref の影響調査のために、さらに TaskPref も除外したモデル (TaskPref 無モデル) を構築する。また先行研究モデルでは 20 種類のメトリクスから TaskPref だけを除外し、併用モデルでは全てを採用してモデルを構築する。非機械学習モデルは、表 4 に示した、実際に指定期間内に完了している不具合の割合を用いて、正木ら [12] と同様の方法で予測を行う。

評価指標としては、適合率、再現率、F1 値を用いる。本研究における適合率とは、指定期間よりも早く修正が完了すると予測した不具合のうち、正しく予測できた不具合の割合である。また、本研究における再現率とは、一定期間内に修正された不具合のうち、正しく予測できた不具合の割合を意味する。

4.4 実験結果

4.4.1 TaskPref が予測精度に与える影響

表 5 に、TaskPref の有無による予測精度の比較結

表 4 Eclipse Platform における不具合修正時間の分布

指定期間	件数	割合 (%)
1 日以内	9,093	49.25
1 週間以内	13,092	70.91
1ヶ月以内	16,399	88.82
1ヶ月以上	2,065	11.18
全期間	18,464	100

表 5 TaskPref の有無による予測精度比較

	指定期間	ランダムフォレスト法		
		適合率	再現率	F1 値
TaskPref 有	1 日	0.7063	0.5460	0.6159
	1 週間	0.7326	0.9608	0.8313
	1ヶ月	0.8929	0.9988	0.9429
TaskPref 無	1 日	0.6805	0.5148	0.5862
	1 週間	0.7258	0.9770	0.8329
	1ヶ月	0.8909	0.9990	0.9419
精度向上率	1 日	0.0153	0.041	0.0318
	1 週間	0.0106	-0.0135	0.0019
	1ヶ月	-0.0018	-0.0003	-0.0012

果を示す。表 5 の精度向上率は、TaskPref を用いない場合と比較して TaskPref を用いた場合の予測精度がどれだけ向上したかを表す。表 5 より、指定期間 1 日の予測では適合率、再現率、F1 値のすべての精度が向上した。指定期間 1 週間の予測では再現率以外の精度が向上したが、精度向上率は低かった。また、指定期間 1ヶ月では、精度が低下している。このことから、TaskPref を用いた予測モデルは、1 日以内に修正が完了するか否かの予測において特に有効であることが分かった。

4.4.2 先行研究との比較および精度向上率

表 6 に、提案手法の予測モデル (TaskPref 有モデル) と、先行研究モデル [12]、併用モデルを比較した結果を示す。表 6 の精度向上率は、非機械学習モデルの予測精度に比べてどれ位精度が向上したかを表している。また、3つのモデルで精度向上率の最も高い値を太字で表している。表 6 より、指定期間 1 週間および 1ヶ月の予測における再現率以外では、併用モデルの精度が最も高かった。このことから、本研究で提案したメトリクスである TaskPref は、先行研究と組み合わせると、さらに予測精度の向上が見込めることが分かった。次に、提案手法の予測モデルと先行研究モデルの予測精度を比較すると、指定期間 1 日で

表 6 予測モデルの精度比較

	指定期間	予測精度			精度向上率		
		適合率	再現率	F1 値	適合率	再現率	F1 値
提案手法 (TaskPref 有)	1 日	0.7063	0.5460	0.6159	0.2138	0.0535	0.1234
	1 週間	0.7326	0.9608	0.8313	0.0235	0.2517	0.1222
	1ヶ月	0.8929	0.9988	0.9429	0.0047	0.1106	0.0547
先行研究モデル [12]	1 日	0.6989	0.5302	0.6030	0.2064	0.0377	0.1105
	1 週間	0.7226	0.9701	0.8283	0.0135	0.2610	0.1192
	1ヶ月	0.8926	0.9993	0.9430	0.0044	0.1111	0.0548
併用モデル	1 日	0.7191	0.5673	0.6343	0.2266	0.0748	0.1418
	1 週間	0.7344	0.9631	0.8334	0.0253	0.2540	0.1243
	1ヶ月	0.8938	0.9988	0.9434	0.0056	0.1106	0.0552
非機械学習モデル	1 日	0.4925	0.4925	0.4925			
	1 週間	0.7091	0.7091	0.7091			
	1ヶ月	0.8882	0.8882	0.8882			

は提案手法の方が全ての予測精度が高かった。指定期間 1 週間では、適合率と F1 値は提案手法の方が予測精度が高かった。指定期間 1ヶ月では、適合率は提案手法の方が予測精度が高かった。このことから、提案手法は先行研究と比較しても、特に 1 日以内の予測において予測精度向上に寄与することが分かった。

4.4.3 各変数の重要度の比較

各変数が構築した予測モデルに与える影響を分析するために重要度を用いて分析を行った。重要度はランダムフォレスト法でのモデル構築過程で算出でき、その値は各変数がモデルに与える影響力を表す [4]。重要度の値が大きいほどその変数が予測モデルに強く寄与していることを表す。表 7 に指定期間 1 日における各予測モデルの説明変数の重要度を示す。表中の太字の値は各予測モデルで重要度の上位 5 つの変数を表している。表 7 から上位 5 つの変数のうち 4 つは Component, Reporter, Assignor, Fixer であり、この結果は先行研究 [12] と一致している。提案手法の予測モデルでは、TaskPref が 4 番目に重要度が高く、予測精度向上に寄与していることが分かった。併用モデルでは、TaskPref よりも Patterns の方が重要度が高いものの、TaskPref は 6 番目に重要度が高く、併用した場合でも予測精度向上に寄与することが分かった。

5 考察

5.1 管理者と修正者の社会的関係

実験の結果、本研究で新たに用いた「管理者と

表 7 各予測モデルにおける説明変数の重要度

	変数名	重要度		
		提案	先行	併用
不 具 合 票	Component	73.04	76.43	70.66
	Milestone	12.18	11.35	11.23
	Priority	21.44	19.84	19.63
	Severity	5.48	5.95	6.07
	DependsOn	2.35	5.53	4.91
	Blocks	2.11	2.89	3.07
	CC	8.89	7.13	8.53
	Descriptions	7.71	7.95	7.45
	Comments	12.50	12.55	11.83
	ComWords	13.04	12.72	12.65
	Attachments	7.42	7.89	7.41
時 間	AssignTime	17.09	15.34	16.95
	AssignedMonth	11.48	10.80	12.87
	AssignedDay	11.47	10.46	11.57
	AssignedWE	5.02	4.54	3.79
人	Reporter	23.83	25.67	24.99
	Assignor	60.50	52.73	54.04
	Fixer	58.91	52.94	50.47
	TaskPref	27.53		23.11
	Patterns		57.46	52.97

修正者のペアが優先してタスクを完了させた割合 (TaskPref)」が、予測精度向上に寄与することが分かった。ただし、TaskPref は、指定期間 1 週間以内および 1ヶ月以内の予測に関しては予測精度向上にほとんど寄与しなかった。これは、TaskPref が高いペアは不具合修正時間が短くなるタスクを多く担当しているため、指定期間 1 日以内の予測に関しては TaskPref が有用であるものの、不具合修正時間が広く分散している TaskPref の低いペアには有効ではなかったと考えられる。

5.2 本論文の制約

本研究では Eclipse Platform プロジェクトのみを対象に評価実験を行った。そのため、本研究で用いたメトリクスおよび予測モデルの一般性を確かめるために、今後は複数のプロジェクトを対象にプロジェクトに適した分析や予測モデルの構築を行う必要がある。

本研究ではデータセットの全期間を用いて cross validation により予測を行っている。本研究で新たに提案したメトリクスはデータセットの全期間を用いて計算されているため、未来のデータを用いて過去のデータを予測する場合が存在することが示唆される。今後はバージョン間での予測を行う必要がある。

6 おわりに

本研究では、管理者と修正者の社会的関係に着目した不具合修正時間予測モデルを構築した。評価実験の結果から、修正者によるタスクの優先順位付けが不具合修正時間予測の精度向上に寄与することが分かった。特に、1日以内に不具合修正が完了するか否かの予測では、先行研究の予測モデルに比べて予測精度が向上することが分かった。また、提案手法と先行手法を併用した場合は、より精度が向上することが確認できた。今後は複数のプロジェクトに本研究の予測モデルを適用し、より一般性の高い知見を得る必要がある。

謝辞 本研究の一部は、文部科学省科学研究補助金(基盤(C): 24500041)による助成を受けた。また、独立行政法人情報処理推進機構が実施した「2013年度ソフトウェア工学分野の先導的研究支援事業」の支援を受けた。

参考文献

- [1] Anbalagan, P. and Vouk, M.: “Days of the Week” Effect in Predicting the Time Taken to Fix Defects, in *Proceedings of the 2nd International Workshop on Defects in Large Software Systems (DEFECTS’09)*, 2009, pp. 29–30.
- [2] Anh, N. D., Cruzes, D., Conradi, R. and Ayala, C. P.: Empirical validation of human factors in predicting issue lead time in open source projects, in *Proceedings of the 7th International Conference on Predictive Models in Software Engineering (PROMISE’11)*, 2011, pp. 13:1–13:10.
- [3] Bettenburg, N., Just, S., Schröter, A., Weiss, C., Premraj, R. and Zimmermann, T.: What makes a good bug report?, in *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering (SIGSOFT ’08/FSE-16)*, 2008, pp. 308–318.
- [4] Breiman, L.: Random Forests, *Machine Learning*, Vol. 45(2001), pp. 5–32.
- [5] Canfora, G., Ceccarelli, M., Cerulo, L. and Di Penta, M.: How Long Does a Bug Survive? An Empirical Study, in *Proceedings of the 18th Working Conference on Reverse Engineering (WCRE ’11)*, 2011, pp. 191–200.
- [6] Marks, L., Zou, Y. and Hassan, A. E.: Studying the fix-time for bugs in large open source projects, in *Proceedings of the 7th International Conference on Predictive Models in Software Engineering (PROMISE’11)*, 2011, pp. 11:1–11:8.
- [7] Mockus, A., Fielding, R. T. and Herbsleb, J. D.: Two Case Studies of Open Source Software Development: Apache and Mozilla, *ACM Transactions on Software Engineering and Methodology*, Vol. 11, No. 3(2002), pp. 309–346.
- [8] Ohira, M., Hassan, A. E., Osawa, N. and Matsumoto, K.: The Impact of Bug Management Patterns on Bug Fixing: a Case Study of Eclipse Projects, in *Proceedings of 28th IEEE International Conference on Software Maintenance (ICSM2012)*, 2012, pp. 264–273.
- [9] Ohira, M. and Yoshiyuki, H.: A New Perspective on the Socialness in Bug Triaging: A Case Study of the Eclipse Platform Project, in *Proceedings of 5th International Workshop on Social Software Engineering (SSE ’13)*, 2013, pp. 29–32.
- [10] Saha, R. K., Khurshid, S. and Perry, D. E.: An empirical study of long lived bugs, in *Proceedings of IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE)*, 2014, pp. 144–153.
- [11] Weiss, C., Premraj, R., Zimmermann, T. and Zeller, A.: How Long Will It Take to Fix This Bug?, in *Proceedings of the 4th International Workshop on Mining Software Repositories (MSR ’07)*, 2007, p. 1.
- [12] 正木仁, 大平雅雄, 伊原彰紀, 松本健一: OSS 開発における不具合割当パターンに着目した不具合修正時間の予測, *情報処理学会論文誌*, Vol. 54, No. 2(2013), pp. 933–944.

**吉行 勇人**

2014年和歌山大学システム工学部情報通信システム学科卒業。現在同大学大学院システム工学研究科博士前期課程在学中。オープンソースソフト

ウェア工学、ソフトウェア保守開発に興味を持つ。学士(工学)。情報処理学会学生会員。

**大平 雅雄**

1998年京都工芸繊維大学工芸学部電子情報工学科卒業。2003年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。同大学情報科学研究科助教を経て、2012年和歌山大学システム工学

部講師、現准教授。博士(工学)。ソフトウェア工学、特にリポジトリマイニング、ソフトウェア開発における知的協調作業支援の研究に従事。情報処理学会、電子情報通信学会、ヒューマンインターフェース学会、ACM 各会員。

**戸田 航史**

2004年大阪大学基礎工学部卒業。2009年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。同年同大学同研究科博士研究員。2012

年より福岡工業大学情報工学科助教。博士(工学)。ソフトウェアメトリクスの研究に従事。情報処理学会、電子情報通信学会、IEEE 各会員。