

EARLY IDENTIFICATION OF FUTURE COMMITTERS IN OPEN SOURCE SOFTWARE PROJECTS

オープンソースソフトウェアプロジェクトにおける早期に
コミッター候補者の発見

NUR ATHIRAH

60226192

TABLE OF CONTENTS

(目次)

- | | |
|----------------------------|------------|
| 1. WHAT IS A COMMITTER? | (コミッターとは) |
| 2. WHAT DOES COMMITTER DO? | (コミッターの仕事) |
| 3. WHAT DOES DEVELOPER DO? | (開発者の仕事) |
| 4. EXPERIMENT SETTINGS | (実験設定) |
| 5. RESULT | (結果) |
| 6. DISCUSSION | (考察) |
| 7. CONCLUSION | (まとめ) |

What is a committer?

(コミッターとは)



- Have permission to commit edited source code to the Version Control System(VCS)
- Such permission is given to a very limited number of developers
- Small number of committers
 - To maintain high quality of their source code

What does committer do?

(コミッターの仕事)

Triaging bugs

- **Receive** a large number of feature requests and bug reports

Assigning bugs

- **Assign** the requests to developers

Verifying bugs

- Have to **verify patches** that developers made to fix the bug
- After that committers commit the fixed source code to the VCS of the project
- In some cases, bugs have to be re-opened

What does developer do ?

(開発者の仕事)

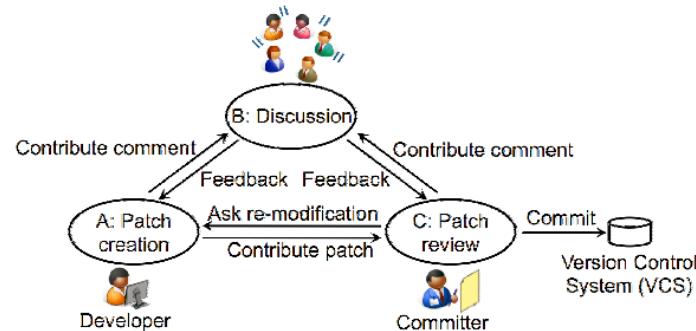


Fig. 1: Process for applying patches in a project.

A) Patch creation (パッチ投稿)

- Developers create patches to extend functionality and to fix bugs
- Then they contribute patches to a bug tracking system(BTS)
- In this study, developer who contributed a patch is regarded as the developer who created the patch

B) Discussion (ディスカッション)

- Developers discuss the creation of patches and review each other's patches
- They discuss their plans and designs

C) Patch verification (パッチ検証)

- Committers verify the contributed patches
- If they judge that a patch:
 - should be edited again: they will ask the developers to re-edit the source code
 - Doesn't need to be edited: the committers would commit the patch to the VCS on behalf of the developers

Popularity of an OSS project
increase
(OSSプロジェクトの人気の増加)

Number of contributed patches
increase
(パッチの数も増加)

Difficult for committers to
manage them
(管理するのが困難)

**Increase the number of
committers**
(コミッターの数を増える)

Most large-scale OSS projects have more than 10,000 developers
(ほぼOSSプロジェクトは10,000以上開発者が参加)



Difficult to identify a potential committers
(コミッター候補者を発見する事は困難)



A developer is promoted to be a new committer through the recommendation of
current committers
(一般的に既存コミッターの推薦からコミッター候補者になる)



To avoid the loss of capable developers, one need to identify future committers as
soon as possible
(有能な開発者の損失を避けるために、早期にコミッター候補者を発見する事は大事)



The committers do not know whether a developer recommended by committers
will continue to contribute to the project for a long time
(推薦された開発者がそのプロジェクトに関して長い間働くかどうか分からない)

In this study, to identify future committers:

- Developers activities were analyzed (number of contributed patches and comments)
- A committer-identification model was built using these metrics(to predict whether a particular developer will become a committer during our target period)

- Prior studies aim to provide techniques for the effective use of the limited human resources of a few committers
- On the other hand, the goal of this study is to increase the human resources by recommending additional capable committers

EXPERIMENT SETTINGS

(実験設定)

TABLE I: Summary of the studied data.

	Eclipse project	Mozilla project
subproject	platform	Firefox
target period	2001/10-2010/12	2004/01-2008/12
existing committers	36	96
developers	8,964	12,287
future committers	53	51
studied developers (developers + future committers)	9,017	12,338

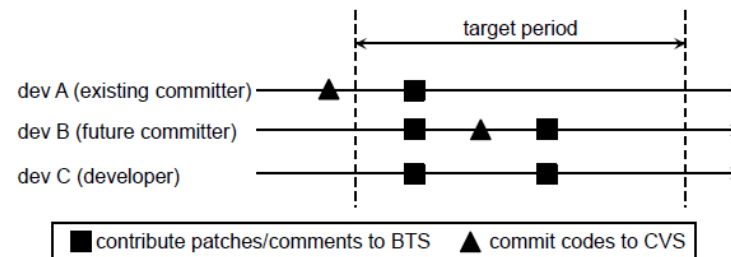


Fig. 2: An example of classifying developers.

A) Target data (対象データ)

- Data from the Eclipse and Mozilla projects are used
- Some developers are already committers
- Existing committers (既存開発者)
 - Have committed source code to VCS before the studied period
- Future committer (コミッター候補者)
 - Commits patches for the first time during the studied period
- Developer (一般開発者)
 - Never committed anything to VCS since the project's start and throughout the studied period

EXPERIMENT SETTINGS

(実験設定)

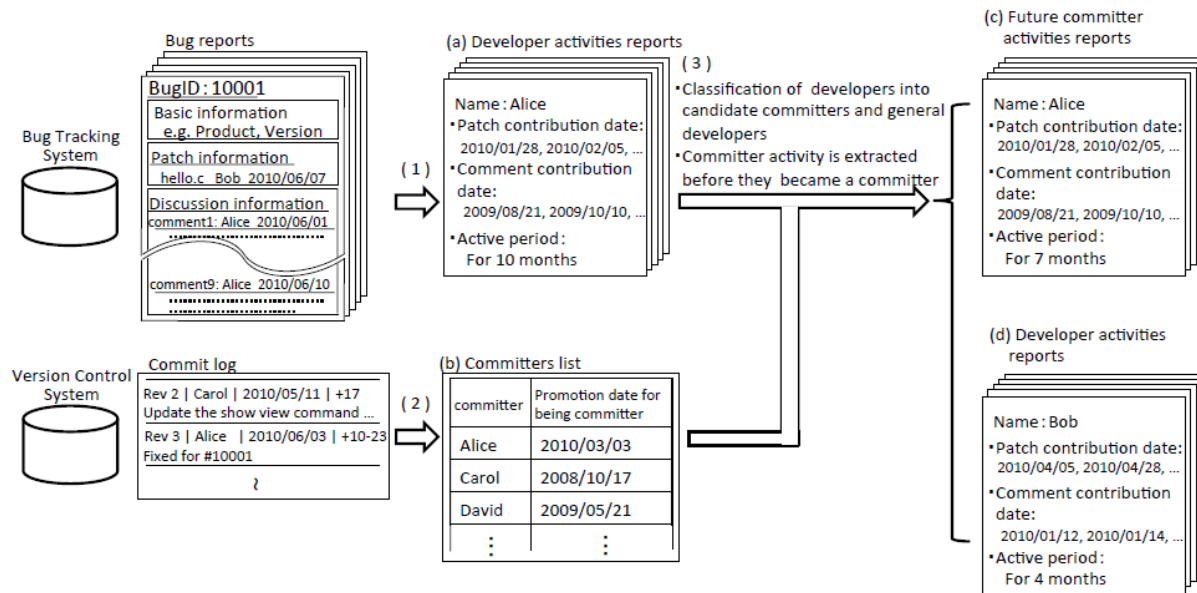


Fig. 3: Method to extract developer activities.

B) Extracting Metrics (データを抽出)

- The metrics were extracted from the Bugzilla data
- Steps to extract the metrics:

(1) Extracting developer activities from BTS data

(2) Extracting the promotion date for becoming a committer from VCS data

(3) Classifying the developer activities reports

RESULTS

(結果)

RQ1:Are there any difference in the activities of future committers and developers?
(コミッター候補者と一般開発者の活動に違いがある？)

RQ2:Which developer activities lead to early promotion to a committer role?
(早期にコミッター昇格ために、有用な活動量は何か)

RQ3:How accurate is a committer-identification model built using developer activities?
(開発者の活動量から構築した予測モデルはどの程度で正確なのか？)

RQ1: Are there any differences in the activities of future committers and developers?

(コミッター候補者と一般開発者の活動に違いがある?)

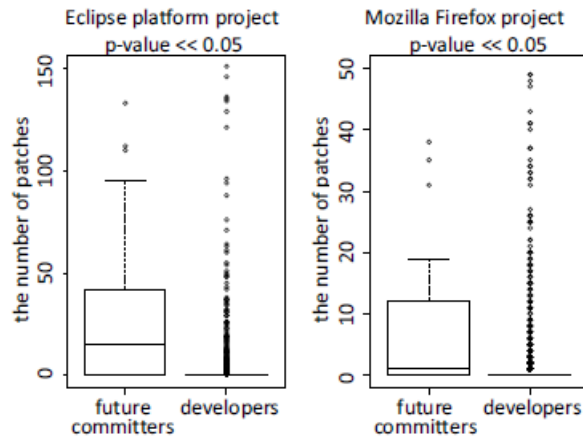


Fig. 4: Differences of the number of patches between committers and developers

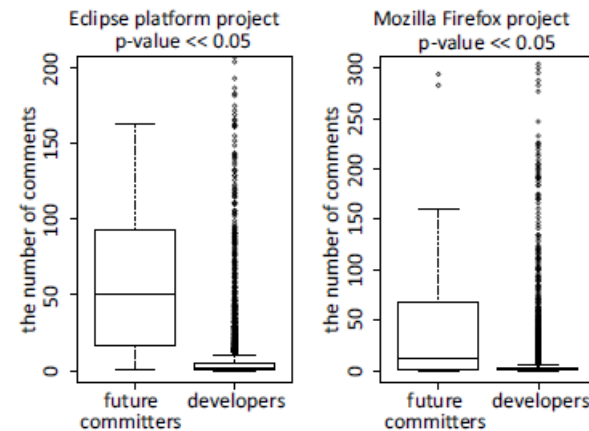


Fig. 5: Differences of the number of comments between future committers and developers

- The differences in the amount of activities between future committers and developers are presented in Fig. 4 and 5

The amount of activities by future committers is higher than the amount of activities by developers

RQ2: Which developer activities lead to early promotion to a committer role?

(早期にコミッター昇格ために、有用な活動量は何か)

- RQ2-1: Is there a difference in activities of **rapidly-promoted** committers and the activities of **regularly-promoted** committers?
- RQ2-2: What do **regularly-promoted** committers do more than the **rapidly-promoted** committers?

RQ2-1: Is there a difference in activities of **rapidly-promoted** committers and the activities of **regularly-promoted** committers?

- The number of activities performed by rapidly-promoted committers is more than regularly-promoted committers.

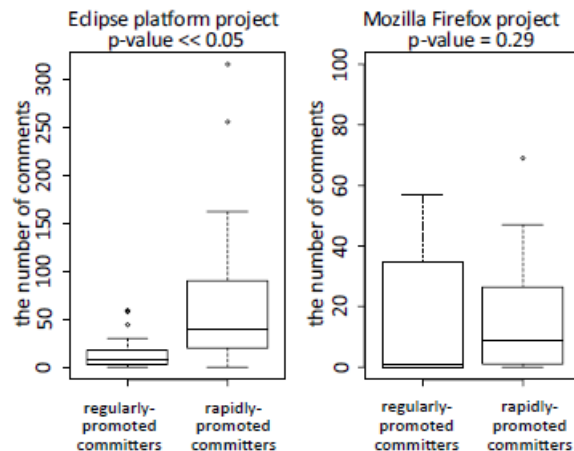


Fig. 7: Differences of the number of patches of regularly-promoted committers and rapidly-promoted committers

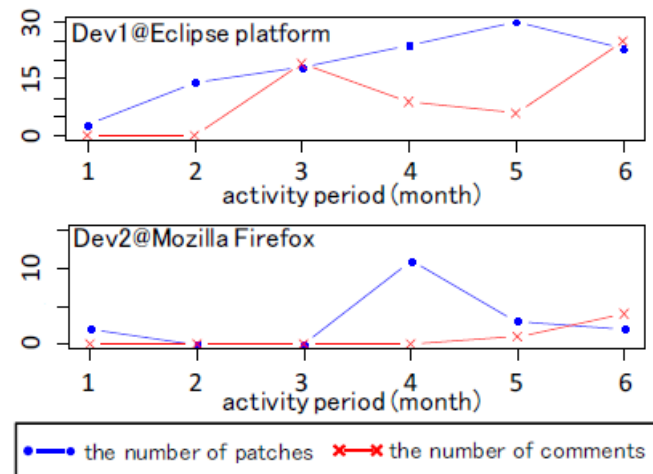


Fig. 9: Activity change of rapidly-promoted committers.

RQ2-1: Is there a difference in activities of **rapidly-promoted** committers and the activities of **regularly-promoted** committers?

(A) Median number of contributed patches

	REGULARLY-PROMOTED COMMITTERS	RAPIDLY-PROMOTED COMMITTERS
ECLIPSE	0	15
MOZILLA FIREFOX	0	2

- Median number of contributed patches by **regularly-promoted** committers is **less** than the number of contributed patches by the **rapidly-promoted** committers

RQ2-1: Is there a difference in activities of **rapidly-promoted** committers and the activities of **regularly-promoted** committers?

(B) Median number of contributed comments

	REGULARLY-PROMOTED COMMITTERS	RAPIDLY-PROMOTED COMMITTERS
ECLIPSE	9	41
MOZILLA FIREFOX	1	11

- The median number of contributed comments by **regularly-promoted** committers is **less** than the number of contributed comments by the **rapidly-promoted** committers

#In the Mozilla Firefox project, there is no significant difference in the number of activities between the **regularly-promoted** committers and the **rapidly-promoted** committers

RQ2-2: What do regularly-promoted committers do more than the rapidly-promoted committers?

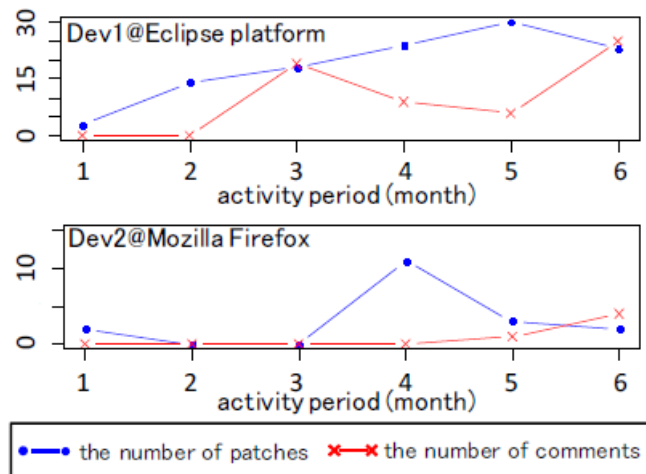


Fig. 9: Activity change of rapidly-promoted committers

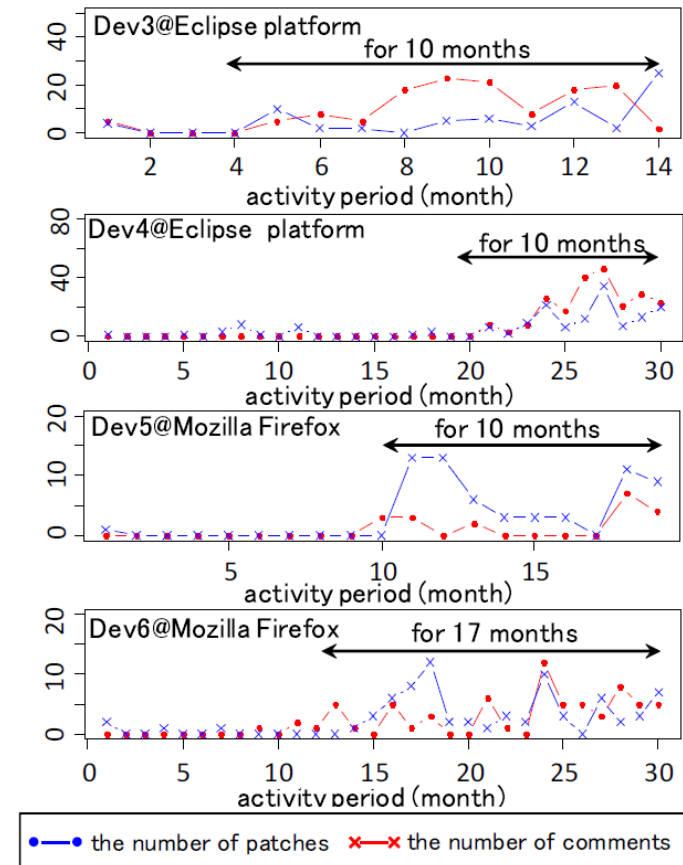


Fig. 10: Activity change of regularly-promoted committers

RQ2-2: What do **regularly-promoted** committers do more than the **rapidly-promoted** committers?

- The amount of activities by **regularly-promoted** committers is fewer than the **rapidly-promoted** committers at the beginning of their time
- However, the **regularly-promoted** committers gradually increase the amount of their activities, and they became committers after they have worked actively for 1-1.5 years
- Few developers continue actively work for about a year
- Most developers who have actively worked for about 1-1.5 years are promoted to committer role

The amount of contribution in the time period just before the promotion plays a big part in the committer promotion.

RQ3:How accurate is a committer-identification model built using developer activities?

(開発者の活動量から構築した予測モデルはどの程度で正確なのか?)

TABLE III: Committer prediction results in Eclipse platform project

	Threshold	Precision	Recall	F1-value	AUC
patch	0.20	0.02	0.96	0.04	0.94
+comment	0.50	0.05	0.77	0.09	0.94
+period	0.80	0.18	0.58	0.28	0.94
patch	0.20	0.01	1.00	0.01	0.81
	0.50	0.09	0.65	0.15	0.81
	0.80	0.14	0.54	0.22	0.82
comment	0.20	0.02	0.85	0.04	0.88
	0.50	0.04	0.73	0.08	0.89
	0.80	0.07	0.54	0.12	0.88

TABLE IV: Committer prediction results in Mozilla Firefox project

	Threshold	Precision	Recall	F1-value	AUC
patch	0.20	0.01	1.00	0.02	0.95
+comment	0.50	0.02	0.96	0.04	0.95
+period	0.80	0.08	0.44	0.13	0.94
patch	0.20	0.00	1.00	0.01	0.76
	0.50	0.06	0.56	0.11	0.76
	0.80	0.06	0.56	0.11	0.76
comment	0.20	0.00	0.92	0.01	0.76
	0.50	0.01	0.72	0.03	0.76
	0.80	0.05	0.44	0.09	0.76

- A very strong AUC of the model using all activities
- The AUC:
 - Eclipse: 0.94
 - Mozilla Firefox: 0.95
- The model had higher accuracy than the random predictor->AUC is considerably higher than 0.5 in both projects
- Precision and recall using all metrics are:
 - about 18% and 58% in the Eclipse platform project (at 0.80 threshold)
 - about 8% and 44% in Mozilla Firefox project (at 0.8 threshold)

RQ3:How accurate is a committer-identification model built using developer activities?

(開発者の活動量から構築した予測モデルはどの程度で正確なのか?)

TABLE V: Variable importance measure

	Eclipse project	Mozilla project
SumNumPatch	1.93	1.08
MedNumPatch	1.43	0.78
SumNumComment	3.76	1.38
MedNumComment	1.55	0.76
ActPeriod	2.29	3.47

- SumNumComment (総コメント投稿数) and ActPeriod (活動期間) are the most important factors in identifying future committers in both projects
- Not just **technical** contributions(i.e., **patch** contributions), **social** contributions (i.e., **comment** contributions) also play a role in the promotion decision
- The role of **social** contributions (3.76/1.38) is even larger than **technical** contributions (1.93/1.08) in both projects.
- The role of **social** contributions (3.76) is strong compare to **activity** period (2.29) in the Eclipse project.

Our committer-identification model considerably outperforms a random predictor

DISCUSSION

(考察)

TABLE VI: NUMBER OF FUTURE COMMITTERS THAT HAVE BEEN PROMOTED AS A COMMITTER DURING TARGET PERIOD AND AFTER A YEAR

	FUTURE COMMITTER	AFTER A YEAR
ECLIPSE	53	40
MOZILLA FIREFOX	51	30

1. Activities after becoming a committer (コミッター昇格後の活動)
 - Does future committers actively work after they become committers ?
 - The future committer activities after became a committer were analyzed
 - Regularly-promoted committers may lose their motivation because they have already worked as developers for more than a year

DISCUSSION

(考察)

- Rapidly-promoted committer activities and the regularly-promoted committer activities after they became committers
 - The median number of comments and the median number of commits to VCS by the **rapidly-promoted** committers are higher than those by the **regularly-promoted** committers
- The results highlight the importance of early identification of strong developers and their rapid promotion to committer roles.
- In fact, rapidly-promoted committers actively contribute more than regularly-promoted committers.

DISCUSSION

(考察)

2. Threats to validity (本論文の制約)

- The date when a developer commits to VCS for the first time - the date of promotion to committer
- Some developer may get permission to commit patches before the date of promotion to become a committer
- Focused on developer activities in the patch contribution and integration process
- Many other types of activities (such as the number of edited lines of code, the source code quality, and the content of comments) that could be studied in future work

CONCLUSION

(まとめ)

- The goal is to be able identify future committers easier
- Developer activities was analyzed and a committer-identification model was built to identify future committers among the thousands of developers in an OSS project
- Using data from the Eclipse platform and Mozilla Firefox projects:

RQ1:Are there any difference in the activities of future committers and developers?

- The amount of activity by a future committer is higher than the activity of a developer

RQ2:Which developer activities lead to early promotion to a committer role?

- Regularly-promoted committers have actively worked for 1-1.5 years before they become committers

RQ3:How accurate is a committer-identification model built using developer activities?

- Our committer-identification model outperforms a random predictor

- After the promotion, rapidly-promoted committers were more active
regularly-promoted committers
- Enhance the committer-identification model