

Why is my code change abandoned?

Qingye Wanga, Xin Xiab, David Loc, Shanping Lia

Q. Wang, X. Xia, D. Lo, and S. Li, "Why is my code change abandoned?,"
Information and Software Technology, vol. 110, pp. 108–120, Jun. 2019.

背景

- ソースコードの変更を検査しソフトウェア品質の向上を狙う（以下コードレビューと呼ぶ）
- コードレビューで変更が破棄された場合、開発時間を無駄に費やしてしまう
- 破棄された理由を知ることによってこうした時間を無くしていきたい

- RQ1
 - なぜ変更が破棄されたか（理由）
- RQ2
 - 理由の頻度分布はどうなっているか
- RQ3
 - 変更が破棄されるまでにどれだけの時間を費やされるか

データセット

- 4プロジェクトのコードレビューシステムからランダムに1000件抽出し、コメントを読むのが困難なものを削除
- なぜ破棄したのか開発者にメールで聞いた

Table 1
Dataset.

Project	# Changes
Eclipse	309
LibreOffice	590
Opentack	346
Qt	214
In total	1459

分類の流れ カードソーティング

1. タイトルをもとにカード作成し, カテゴリ分類
2. 別プロジェクトも同様に(その時に新カテゴリが発覚)
3. 変更の大半がduplicateのため3つのサブカテゴリに分類
4. Contributor Operation をサブカテゴリに分類

ラベル付け

- 著者と院生1名でラベル付けし, Fleiss Kappa を比較
- 全体的なカッパ値は0.68になった

Table 5

Interpretation of Kappa values.

Kappa value	Interpretation
< 0	Poor agreement
[0.01, 0.20]	Slight agreement
[0.21, 0.40]	Fair agreement
[0.41, 0.60]	Moderate agreement
[0.61, 0.80]	Substantial agreement
[0.81, 1.00]	Almost perfect agreement

RQ1 変更が破棄された理由

- 12のカテゴリに分類
- “duplicate”による理由がもっとも多かった

Table 6

Reason categories.

Category	Count	Percentage
Duplicate	595	40.78%
<i>Already Done</i>	300	21.73%
<i>Suboptimal Solution</i>	181	12.40%
<i>Integrated</i>	97	6.65%
Lack of Feedback	213	14.60%
Contributor Operation	146	10.01%
<i>Wrong Branch</i>	47	3.22%
<i>Accidental Push</i>	37	2.54%
<i>Update Change</i>	20	1.37%
<i>Other Wrong Operation</i>	42	2.88%
New work	123	8.43%
Incomplete/Wrong Fix	110	7.54%
Superfluous	65	4.46%
Test	64	4.39%
Branch Transfer	43	2.95%
Merge Conflict	40	2.74%
Give Up	29	1.99%
Complicated Change	28	1.90%
Other	3	0.21%

duplicate (40.78%)

- “Already done” (21.73%)
 - すでに同様のパッチが投稿されている
- “Suboptimal solution” (12.40%)
 - 別のパッチの方がより良い変更だった
- “Integrated” (6.65%)
 - 別のパッチに統合された

Lack of Feedback (14.60%)

- レビュー者がコメントを追加したが応答がなかった
- 変更をアップロードしたが反応しなかった

Contributor operation (10.01%)

- "Wrong branch" (3.22%)
 - 間違えたブランチで変更してしまった
- "Accidental push" (2.54%)
 - 手違いでプッシュしてしまった
- "Update change" (1.37%)
 - アップデート前の変更を削除するために
- "Other wrong operations" (2.88%)
 - bug numberのミスなど

New work (8.43%)

- 新しくコミットを作成して作業を続けた場合
- 別のアプローチによる変更になった場合

Incomplete/wrong fix (7.54%)

- 不完全または間違っている変更
- 変更前よりバグが増えたり, そもそも変更のアプローチが間違っていることも

Superfluous (4.46%)

- 無効なパッチ
- 問題を修正する価値がない
- 変更の意味があると他のレビューアを説得できなかった

Test (4.39%)

- テストを目的として作成された
- マージされることを意図して作られていない

Branch transfer (2.95%)

- あるブランチから別のブランチに変更を移動したため
 - e.g.) cherry-picked
- クラッシュの心配をせずに作業できるようにするため

Merge conflict (2.74%)

- マージの競合を引き起こしたため
- コードの書き直しを必要とする

Give up (1.99%)

- 時間不足やコメントで指摘された問題を修正できなかったため
- 修正に大きな労力を必要としたため諦めた場合

Complicated change (1.90%)

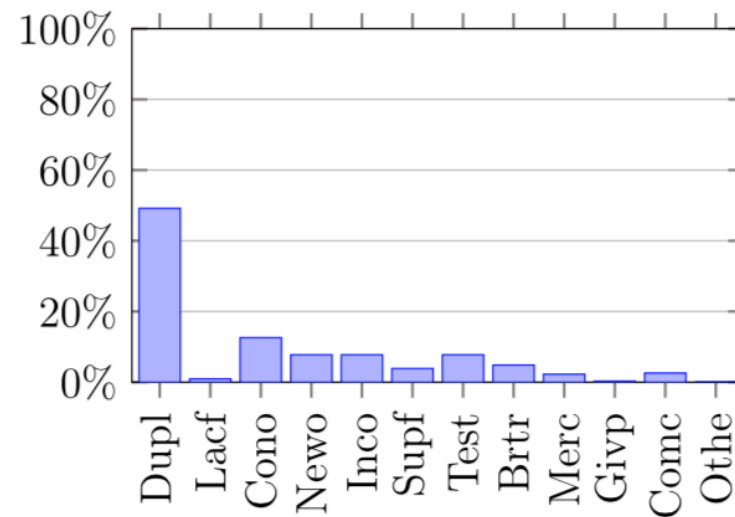
- より小さく独立した変更分割するため
- コードレビューでは小規模の変更の方がレビューしやすくなるとされている

Other (0.21%)

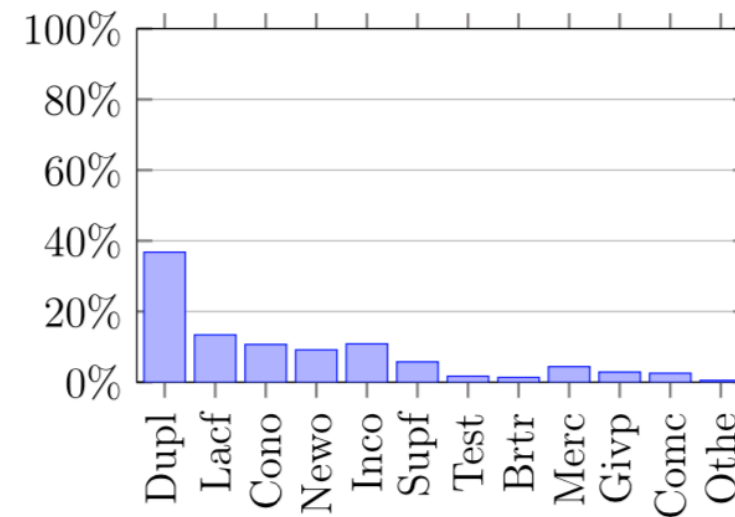
- 他カテゴリに含まれないもの
 - e.g.) プロトタイプを共有するため（テキストで説明するよりも楽だから）

RQ2 理由の頻度分布

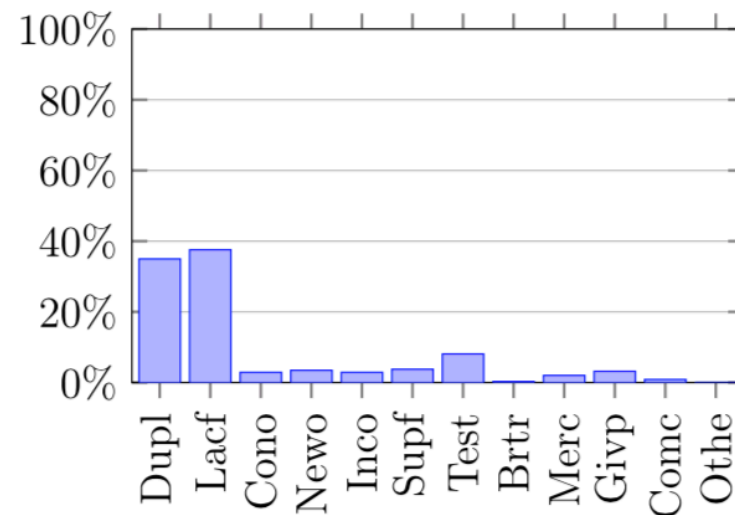
- 各プロジェクトでのカテゴリの分布
- 分布に統計的な有意差なし



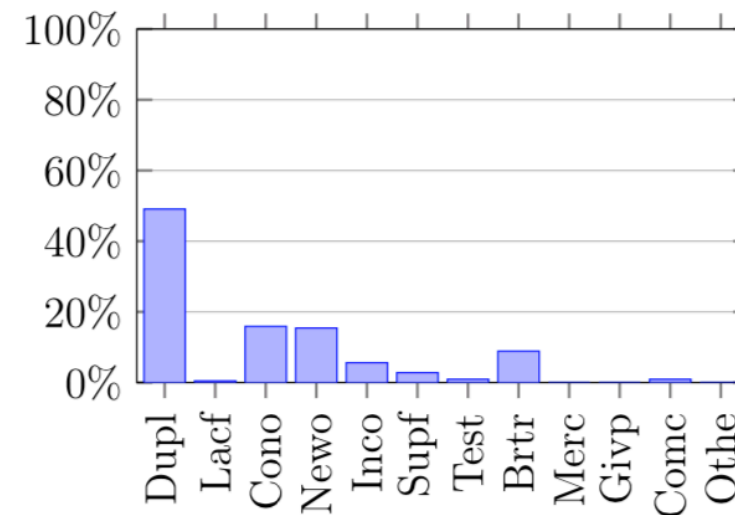
(a) Eclipse



(b) LibreOffice



(c) OpenStack



(d) Qt

RQ3 変更が破棄されるまでにどれだけの時間を費やされるか

- 各カテゴリの基本統計量

Table 7

Change abandoning durations in terms of days.

Reason categories	Min	Max	Mean	Median
Duplicate	0.0005	848.2421	34.5591	2.6662
Lack of Feedback	0.0028	744.8853	191.8712	161.7100
Contributor Operation	0.0003	111.7033	3.6336	0.0347
New Work	0.0049	408.3261	40.3557	4.2419
Incomplete/Wrong Fix	0.0032	590.1685	29.5627	2.3161
Superfluous	0.0043	939.9169	46.0039	1.8521
Test	0.0018	301.8160	13.6013	0.4729
Branch Transfer	0.0361	178.8673	43.7281	17.8710
Merge Conflict	0.0203	68.6853	11.9476	0.3403
Give Up	1.7536	423.1277	108.1894	97.6660
Complicated Change	0.0104	166.1683	24.0878	4.0059
Other	0.1419	27.6982	9.9646	2.0536

期間ごとの変更の破棄について

- 数年以上放置されていたものの大半が

Lack of Feedbackだった

Table 8

Numbers of change abandoned within various durations.

Reason category	Duration	Count	Proportion				
Duplicate	within a month	465	78.15%	Test	within a month	55	85.94%
	within a year	584	98.15%		within a year	64	100.00%
	more than a year	11	1.85%		more than a year	0	0.00%
Lack of Feedback	within a month	3	1.41%	Branch Transfer	within a month	23	53.49%
	within a year	192	90.14%		within a year	43	100.00%
	more than a year	21	9.86%		more than a year	0	0.00%
Contributor Operation	within a month	141	96.58%	Merge Conflict	within a month	35	87.50%
	within a year	146	100.00%		within a year	40	100.00%
	more than a year	0	0.00%		more than a year	0	0.00%
New work	within a month	91	73.98%	Give Up	within a month	7	24.14%
	within a year	121	98.37%		within a year	28	96.55%
	more than a year	2	1.63%		more than a year	1	3.45%
Incomplete/Wrong Fix	within a month	88	80.00%	Complicated Change	within a month	23	82.14%
	within a year	108	98.18%		within a year	28	100.00%
	more than a year	2	1.82%		more than a year	0	0.00%
Superfluous	within a month	56	86.15%	Other	within a month	3	100.00%
	within a year	61	93.85%		within a year	3	100.00%
	more than a year	4	6.15%		more than a year	0	0.00%

より良いパッチ作成のために

1. パッチを送る前に修正価値があるか, 機能が有用か確認する[1]
2. 重複した変更がないか調査するコードの重複はソフトウェア品質に影響を与える
3. 変更が1つの問題のみに対処しているか確認する

[1] Y. Fan, X. Xia, D. Lo, S. Li, Early prediction of merged code changes to prioritize reviewing tasks, Empir. Softw. Eng. (2018) 1-48.

より良いパッチ作成のために

4. プッシュ時にブランチ, コミットIDが間違えていないか確認

5. プッシュ後, 開発者は投稿者のコメントに時間内に返信する

類似研究との差分

- Tao[2]らもコメントから破棄された理由を調査している

本研究との差分

	本研究	Tao
調査対象	コードレビューシステム	バグレポート
データソース	Gerrit	Bugzilla
Duplicateの扱い	詳細に分類	そのまま
カテゴリ	12のカテゴリのうち、3カテゴリが同じ	

まとめと所感

- まとめ
 - 4つのプロジェクトで変更が破棄された理由を手動で調査した
 - 実際にパッチを投稿する時に気をつけるべきことをまとめた
- 所感
 - 関連研究だけでなく類似研究を述べ、類似研究との差分を述べていた点が分かりやすかった