

Primers or Reminders? The Effects of Existing Review Comments on Code Review

Davide Spadini, Gül Çalikli, Alberto Bacchelli

2020 IEEE/ACM 42nd International
Conference on Software Engineering (ICSE)
2020,1171~1182 page

発表者：森岡勇介

背景

- コードレビューは，ソースコードの品質を維持，促進し，知識を共有するなど様々な効果がある
- 最新のコードレビューでは，レビューアのコメントが関係する他のレビューアに表示される
- この可視性が個人の判断に影響を与える可能性がある

目的

目的

以前のコードレビューのコメントによってレビューアが受ける影響を理解すること

RQ1

RQ1

通常は考慮しない珍しいタイプのバグをレビューアに表示した場合、 どのような影響があるか

仮説

H10:

珍しいタイプのバグをレビューアに表示しても、 同じタイプのバグを見つけるパフォーマンスに影響しない

H11:

珍しいタイプのバグをレビューアに表示しても、 通常探しているタイプのバグを見つけるパフォーマンスに影響しない

RQ2

RQ2

通常探索するタイプのバグをレビューアに表示した場合、
どういう影響があるか

仮説

H20:

通常探すタイプのバグをレビューアに表示しても、同じタイプのバグを見つけるパフォーマンスに影響しない

H21:

通常探すタイプのバグをレビューアに表示しても、珍しいタイプのバグを見つけるパフォーマンスに影響しない

アプローチ

4 種類のパッチ

1. 珍しいタイプのバグについてのコメントが表示されたもの
 2. 1.と同じだが、コメントが表示されていないもの
 3. 通常探すタイプのバグについてのコメントが表示されたもの
 4. 3.と同じだが、コメントが表示されていないもの
-
1. 先行研究に基づいて珍しいタイプのバグ(BugA)と通常探すタイプのバグ(BugB)を定義する
 2. BugAを2つ、BugBを1つ埋め込んだパッチを用意する(CodeChange1)
 3. Code Change1にBugAについてコメントを付与したものとしていないものを作成する
 4. BugBを2つ、BugAを1つ埋め込んだパッチを用意する(CodeChange2)
 5. Code Change2にBugBについてコメントを付与したものとしていないものを作成する

アプローチ

アプローチ

1. 目的を伝えずコードレビューを依頼する
2. 4種類のパッチをランダムに割り当て、レビューしてもらう
3. レビューにかかった時間を尋ねる
4. 追加の質問

分析対象

- 243人に参加してもらい,
 - ひとつ以上コメントしていること
 - レビューに五分以上かかった人を条件に85人に絞り込み, コードレビューをしてもらった

	Primed (Pr)	Not Primed (NPr)	Total
CodeChange1	21	17	38
CodeChange2	22	25	47
Total	43	42	

Pr: コメントの表示があるもの

NPr: コメントの表示がないもの

結果(RQ1)

RQ1:通常は考慮しない珍しいタイプのバグをレビューアに表示した場合, どのような影響があるか

Primed bug (NPE)	Primed (Pr)	Not Primed (NPr)	Total
found	13	2	15
not found	8	15	23

BugAに対するコメントがある場合, PrはNPrよりBugAを約12倍ほど見つけやすい追加の質問でPrの人にコメントの影響を聞くと, かなり影響されたと答える人が40%,とても影響されたと答える人が40%だった

Not primed bug	Primed (Pr)	Not Primed (NPr)	Total
found	14	14	28
not found	7	3	13

BugBを見つけるかどうかに影響は与えていない

■:コメントの表示あり (Pr)
■:コメントの表示なし (NPr)
BugA:珍しいタイプのバグ (Primed bug)

BugB:通常探すタイプのバグ (Not primed bug)

結果(RQ2)

RQ2:通常探索するタイプのバグをレビューアに表示した場合， どのような影響があるか

Primed bug (CC)	Primed (Pr)	Not Primed (NPr)	Total
found	10	8	18
not found	12	17	29

BugBに対するコメントがあっても， 見つけるかどうかに影響はない
追加の質問でも， 影響がないという回答をした人が50%いた

Not primed bug	Primed (Pr)	Not Primed (NPr)	Total
found	13	16	29
not found	9	9	18

コメントがない方が少し多く見つけている

■:コメントの表示あり (Pr)
■:コメントの表示なし (NPr)

BugB:通常探すタイプのバグ (Primed bug)

BugA:珍しいタイプのバグ (Not primed bug)

まとめ

- 珍しいタイプのバグのコメントをみたレビューアは同タイプのバグを見つけやすくなり，他の種類のバグを見つける妨げにならない
- 通常探すタイプのバグのコメントをみたレビューアは同タイプのバグのを見つけやすさは変わらないが，他の種類のバグを見つける妨げになる

考察

- コードレビューを他のレビューアに見せないレビュー方法が検討できる
 - デメリット
 - 知識の伝達の遅れ
 - 発見するバグの重複
- 現在のコードレビューの慣行を変更するほどの証拠にはならない

所感

- 考察にもあったが，四種類のグループに分けることも考慮すると被験者が少ない？
- 一つのバグについてコメントが表示されているので，埋め込まれているバグの数に違いが生じるのが少し気持ち悪く感じた
- 前の人を確認してるから大丈夫 という心理も調査してみたら面白そう