

研究テーマ発表

コードレビュープロセス

60171033 築山文香
2016/3/22

先行研究

対象プロジェクト

Mozilla : Mozilla-all、Core、Firefox、Firefox for Android

期間 : 2013年1月1日～2014年1月1日

- 不具合データ(Bugzilla)とコミットデータ(Mercurial)を紐付け
- レビュー後再修正されている不具合を抽出
 - Buggy review とする
- Buggy reviewを出しやすい人と社会的地位に関係があることを明らかにした

Investigating Code Review Quality: Do People and Participation Matter?

気になった点

TABLE IV: Number of code reviews that missed bugs.

System	# Reviews	# Buggy Reviews	% Buggy Reviews
Mozilla-all	28,127	15,188	54.0 %
Core	18,759	10,184	54.3 %
Firefox	2,668	1,447	54.2 %
Firefox4Android	2,160	1,210	56.0 %

全体のコードレビューの54%がバグを見逃している

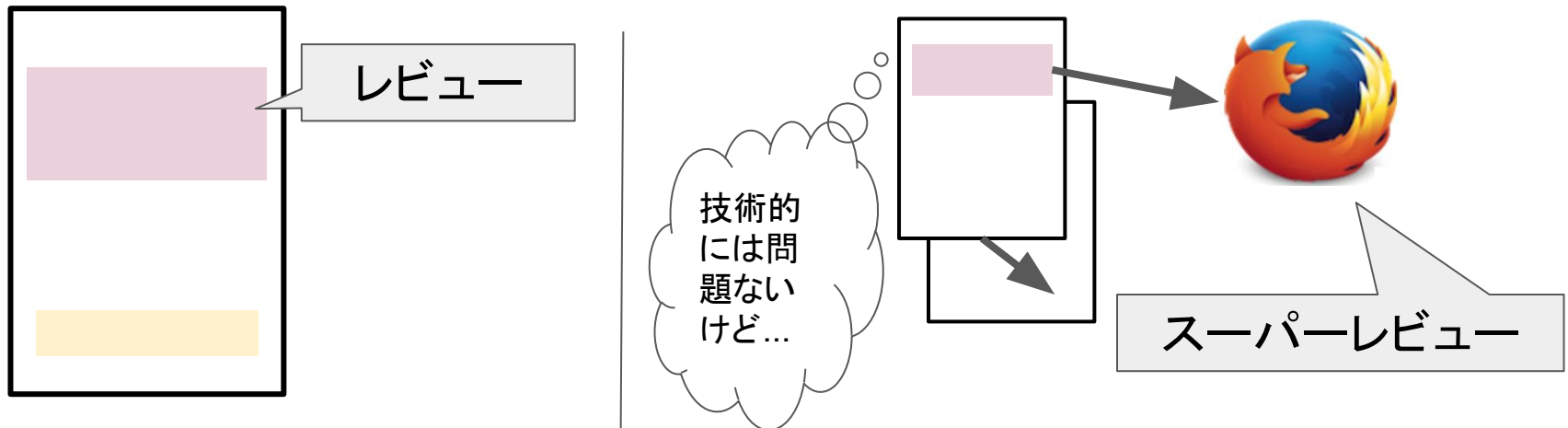
Investigating Code Review Quality: Do People and Participation Matter?

コードレビュープロセスに改善の余地があるのでは？

- 人に問題があるのか
 - バグを見落としやすい人がある
- プロセスやシステムに問題があるのか
 - レビュープロセスが時間不足などで形式化している
 - 簡単なバグは議論されやすい
- 不具合に特徴があるのか

目的

- レビュー後再修正される不具合について調べる
 - なぜ半数以上の不具合が見逃されるのか
 - 不具合の特徴など理由が分かれば、効率よくレビューできるのでは
 - そもそもBuggy reviewの定義が違うのかも
 - Mozillaプロジェクトは2段階レビューを行っている
 - レビュー: 修正箇所に技術的な問題がないか
 - スーパーレビュー: プロジェクトの品質を保つ(統合性が保たれるか、Mozillaのコーディング規約に違反していないか)



やったこと

- Bugzillaから‘REOPENED’されている不具合IDの抽出
 - 約2400件
 - 対象はFirefox
 - 2013年までのMozillaのdump内
 - ‘history’に‘REOPENED’の文字列が含まれる不具合を抽出
- FirefoxのコミットIDと紐付け
 - 不具合ID597件と紐付いた
 - MozillaのGecko-dev(Git)のbrowser(Firefoxに相当)以下のファイル
 - GitLogのCommitTextに特定のBugIDが含まれていたもの
- 紐付いた不具合のhistoryを目視(現状途中20件くらい)

‘REOPENED’されている不具合を目視して(1/2)

- 1回しかコミットされていない不具合
 - 294/597件
 - レビューと関係なく‘REOPENED’されている
 - レビュー前:(例)不具合だと承認されない→不具合だった
 - レビュー中:(例)他と重複している→重複していなかった
 - レビュー後:(例)修正できていない→修正できていた
 - 修正した不具合を改善している？
 - レビューと関係している
 - データがきちんととれていない？
 - レビューと関係しているのかわからない

‘REOPENED’されている不具合を目視して(1/2)

- 複数回コミットされている不具合
 - これからです

今後

- レビューデータの取り方を考える