

The Silent Helper: The Impact of Continuous Integration on Code Reviews

Nathan Cassee, Bogdan Vasilescu, Alexander Serebrenik
SANER 2020, London, ON, Canada Research Papers

背景

- コードレビューは品質を保証するために必要
- プロジェクトメンテナはプルリクエスト(PR)レビュープロセスを支援するために自動化ツールを使用
- CIの導入により, ソフトウェア品質の評価が困難な側面に焦点を当てることが可能になる [12-15]
 - プロジェクトの適合性
 - 非機能要求の遵守
 - 設計

研究の目的

CIは実際にメンテナのコードレビューの労力を
どれだけ軽減できるのか？

- 探索的な実証研究を行う。
 - CI導入後のプルリクエストレビューのコメント数の平均的な変化によって負担がどの程度軽減されるかを定量的に検証する

リサーチクエスト

RQ1

CIの導入によって、プルリクエストレビュー中のコミュニケーションの量はどのように変わるのか？

RQ2

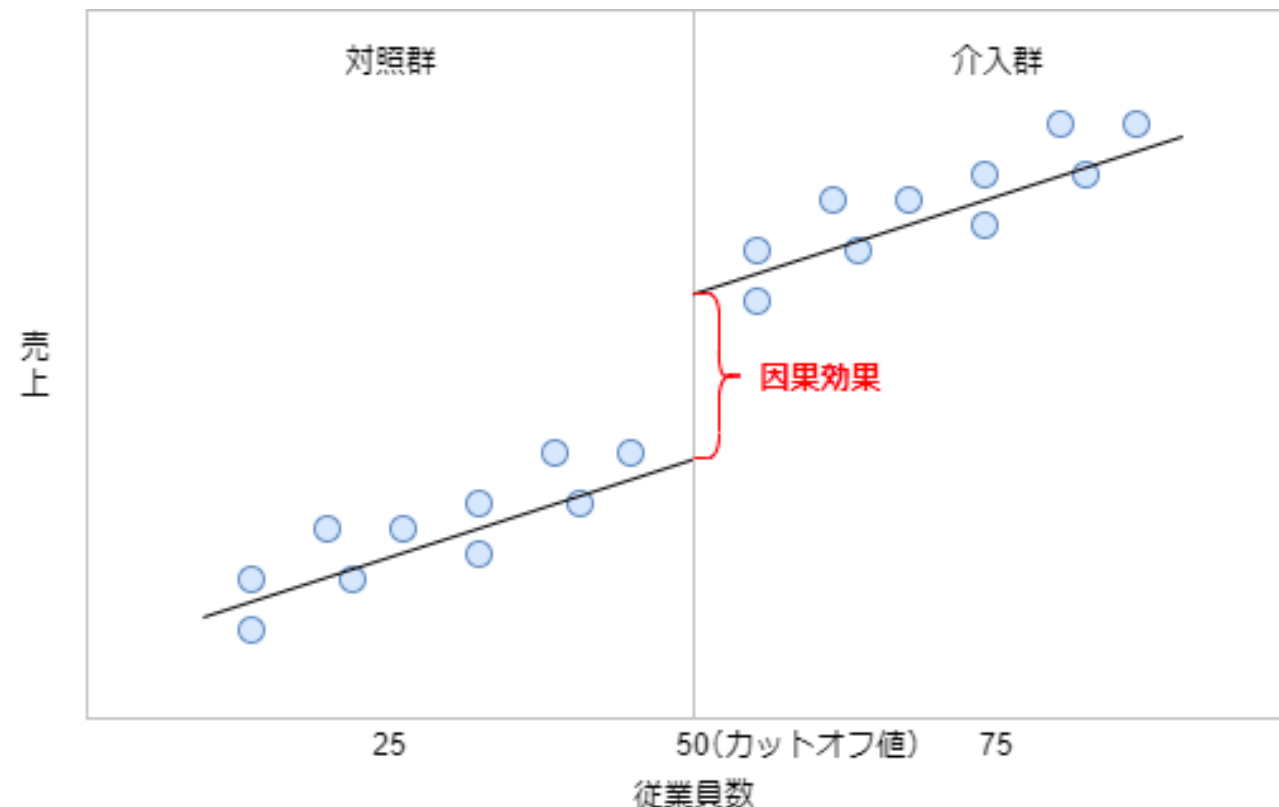
プルリクエストがオープンされた後の更新量は
CIの導入によってどのように変わるのか？

データセット

- Zhaoら[8]が使用したリポジトリ17,000のリスト
- 長い歴史を持つプロジェクト(明記されていなかった)
- Github上でTravisCIのみを扱っている
- 685件のGithubプロジェクト
 - PR : 1,214,826件
 - コミット数 : 5,150,112件

評価方法

- 回帰不連続設計(RDD)を用いて評価
 - 関心のある変数を横軸に，導入前後を比較
- lmerTest Rパッケージ[47]を用いてモデル係数を推定
 - CI導入が従属変数に影響を与えたかどうか



RQ1

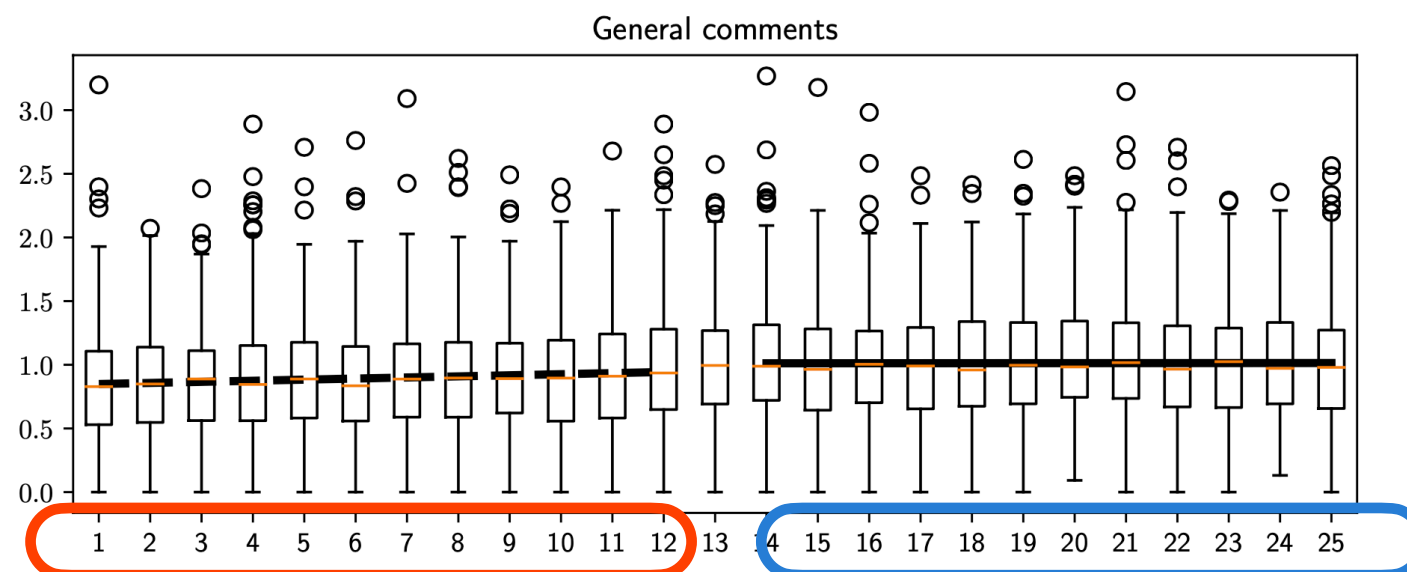
CIの導入によって、プルリクエストレビュー中のコミュニケーションの量はどのように変わるのか？

- PRあたりの一般的なコメントの総数
- PRあたりのレビューコメントの総数
- PRあたりの変更を誘発する
レビューコメント数 ([35]を参考にアルゴリズムを提案)

一般的なコメント

減少

一般的な
コメント数



時間(月)

Feature

モデル係数 二乗和

Intercept***	0.7698	
Additions***	0.0464	2.272
Deletions***	-0.0330	1.280
Commits***	-0.0726	1.225
Assignees***	0.3057	2.705
ChangedFiles*	-0.0419	0.390
CommitsAfterCreate***	0.4922	47.699
TotalPrs***	0.0006	1.155
time***	0.0077	2.600
intervention**	0.0588	0.756
time_after_intervention***	-0.0073	1.071

CI導入前

CI導入後

導入前後で
有意差あり

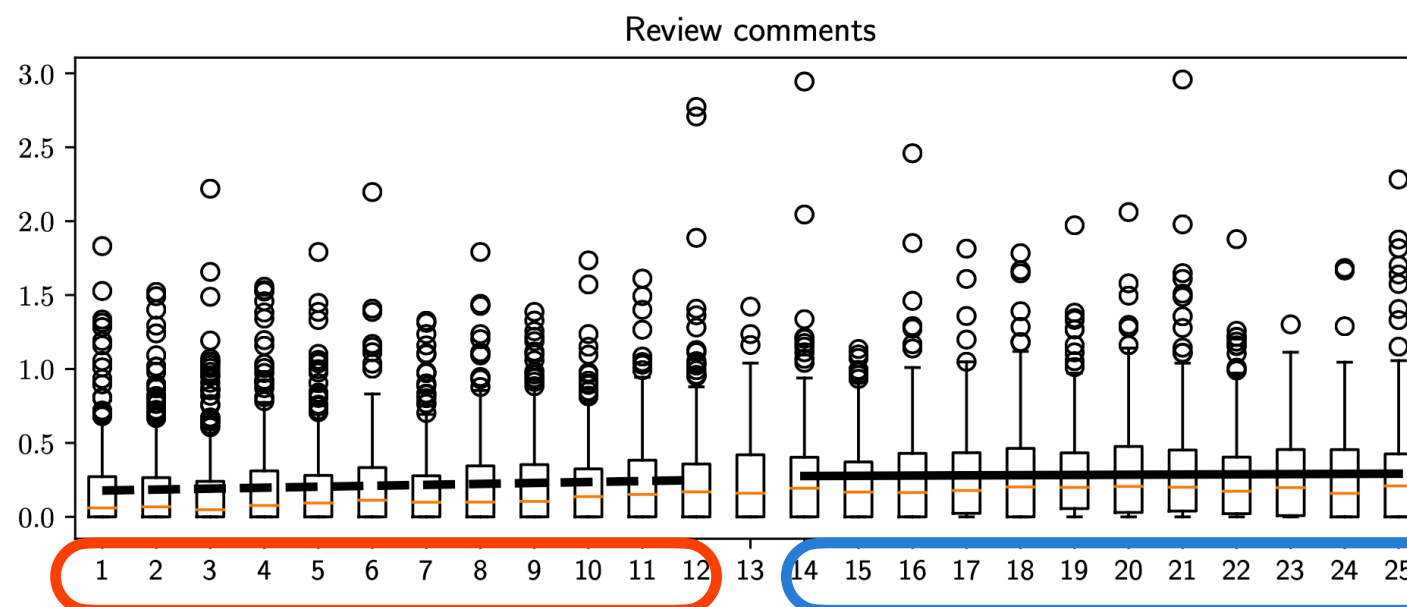
導入直後

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

レビューコメント

減少

レビュー
コメント数



時間(月)

Feature

モデル係数 二乗和

Intercept*	0.0421	
Additions***	0.0488	3.056
Deletions	-0.0050	0.030
Commits***	-0.0440	0.460
Assignees	0.0300	0.027
ChangedFiles***	-0.0424	0.407
CommitsAfterCreate***	0.3672	29.973
TotalPrs**	0.0003	0.302
time***	0.0057	1.437
intervention	0.0133	0.045
time_after_intervention**	-0.0040	0.350

導入前後で
有意差あり

導入直後

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

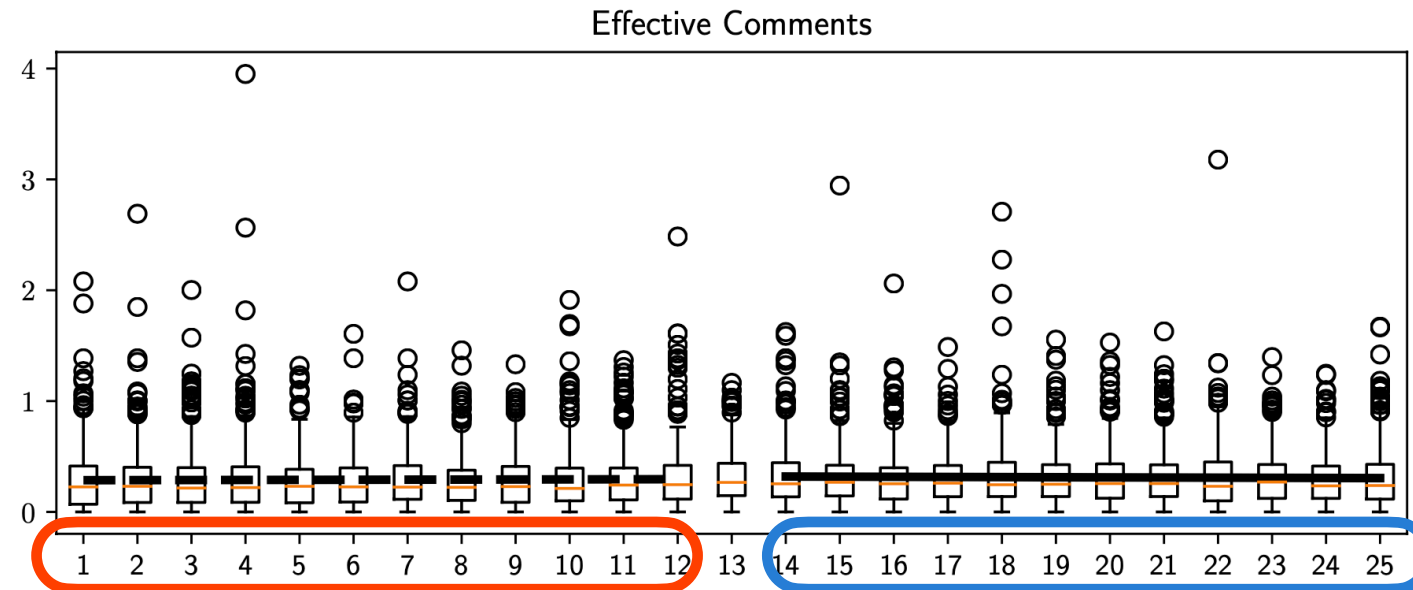
CI導入前

CI導入後

変更を誘発するコメント

減少

誘発する
コメント数



Feature

モデル係数 ; 二乗和

Intercept***	-0.1312	
Additions***	0.0273	0.986
Deletions***	-0.0153	0.287
Commits***	0.1073	4.324
Assignees	0.0024	<0.001
ChangedFiles***	-0.0267	0.167
GeneralComments***	0.0581	2.380
TotalPrs	< 0.0001	0.021
time***	0.0024	0.260
intervention	0.0009	<0.001
time_after_intervention***	-0.0030	0.196

CI導入前

CI導入後

導入前後で
有意差あり

導入直後

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

RQ2

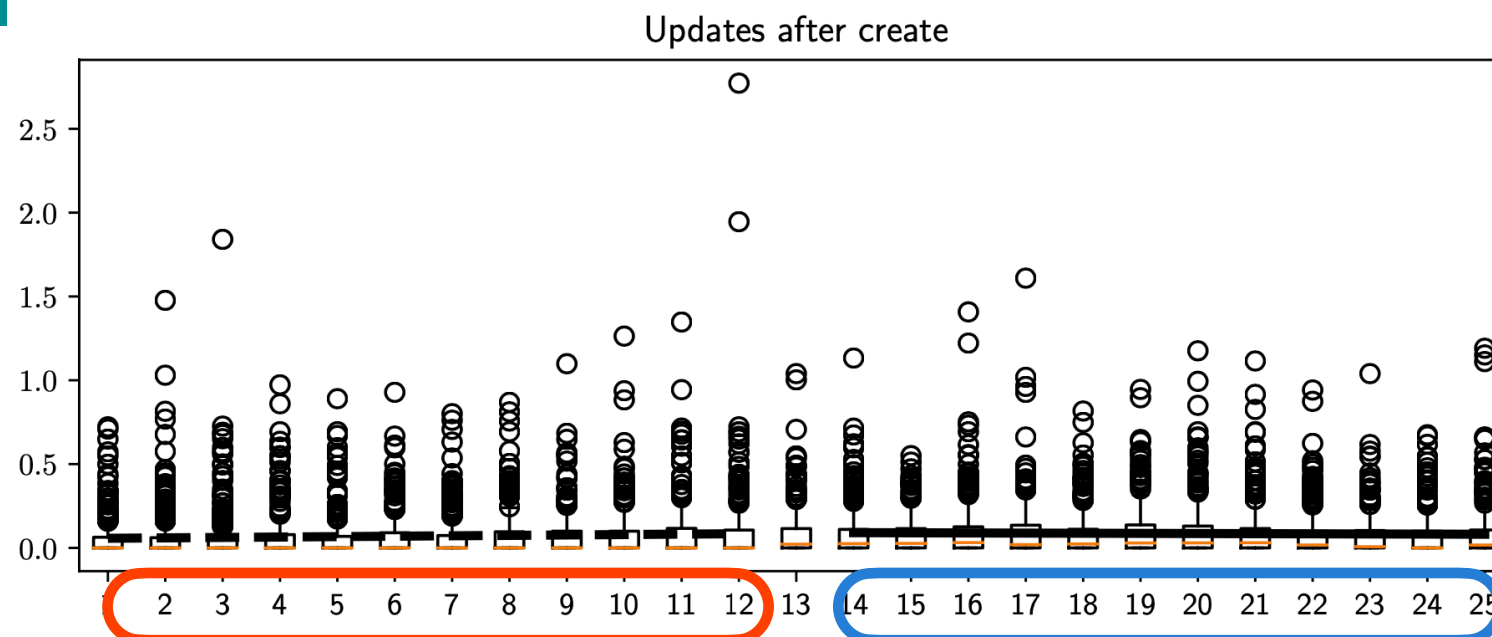
プルリクエストがオープンされた後の更新量は
CIの導入によってどのように変わるのか？

- 修正が必要になったPRに対するコミット数

修正が必要であるPRのコミット数

変化なし

コミット数



Feature

モデル係数 : 二乗和

Intercept***	-0.4357	
Additions***	0.0294	1.166
Deletions	0.0060	0.045
Commits***	0.5237	105.467
Assignees**	0.0830	0.273
ChangedFiles***	-0.0749	1.348
TotalPrs	<0.0001	0.033
GeneralComments***	0.1505	16.903
time*	0.0019	0.155
intervention	0.0028	0.002
time after intervention	-0.0017	0.067

導入後では
有意差なし

導入直後

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

CI導入前

CI導入後

まとめ

- 以下の項目がCIの影響を受けるかを調査
 - コードレビュー中のコミュニケーション
 - コードレビュー中に行われた変更の数

結果として

- コードレビューあたりのコメント数が減少する
 - コードレビュー中に行われた変更の数は一定
- コードレビューの負担は軽減される

付録

一般的なコメント

Use CMD to allow running a shell inside the container
#269

Closed wants to merge 1 commit into `codercor:master` from patch-2

Conversation 7 Commits 1 Checks 1 Files changed 1

commented 9 days ago Contributor + 😊 ...

Describe in detail the problem you had and how this PR fixes it

I can't start the container with `/bin/sh` to debug it.

Is there an open issue you can link to?

No but I hope this <https://stackoverflow.com/a/21564990/4446318> helps explain the issue better than I can.

Use CMD to allow running a shell Verified ✓ a684ae9

requested a review from as a code owner 9 days ago

changed the title ~~Use CMD to allow running a shell~~ Use CMD to allow running a shell inside the container 9 days ago

suggested changes 8 days ago View changes

left a comment Contributor + 😊 ...

You can just use `docker exec -ti` and I agreed about having this on `ENTRYPOINT`, which is relatively allowing args to be passed onto `docker run image`.

So no, This isn't it chief.

Reviewers

Assignees

No one assigne

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

You're not recei from this threac

3 participants

Figure 1. Anonymized fragment of a GITHUB pull request review. Usernames and avatars have been masked in this screenshot to safeguard the privacy of the authors. Blue is the submitter, red and gray are the reviewers.

レビューコメント

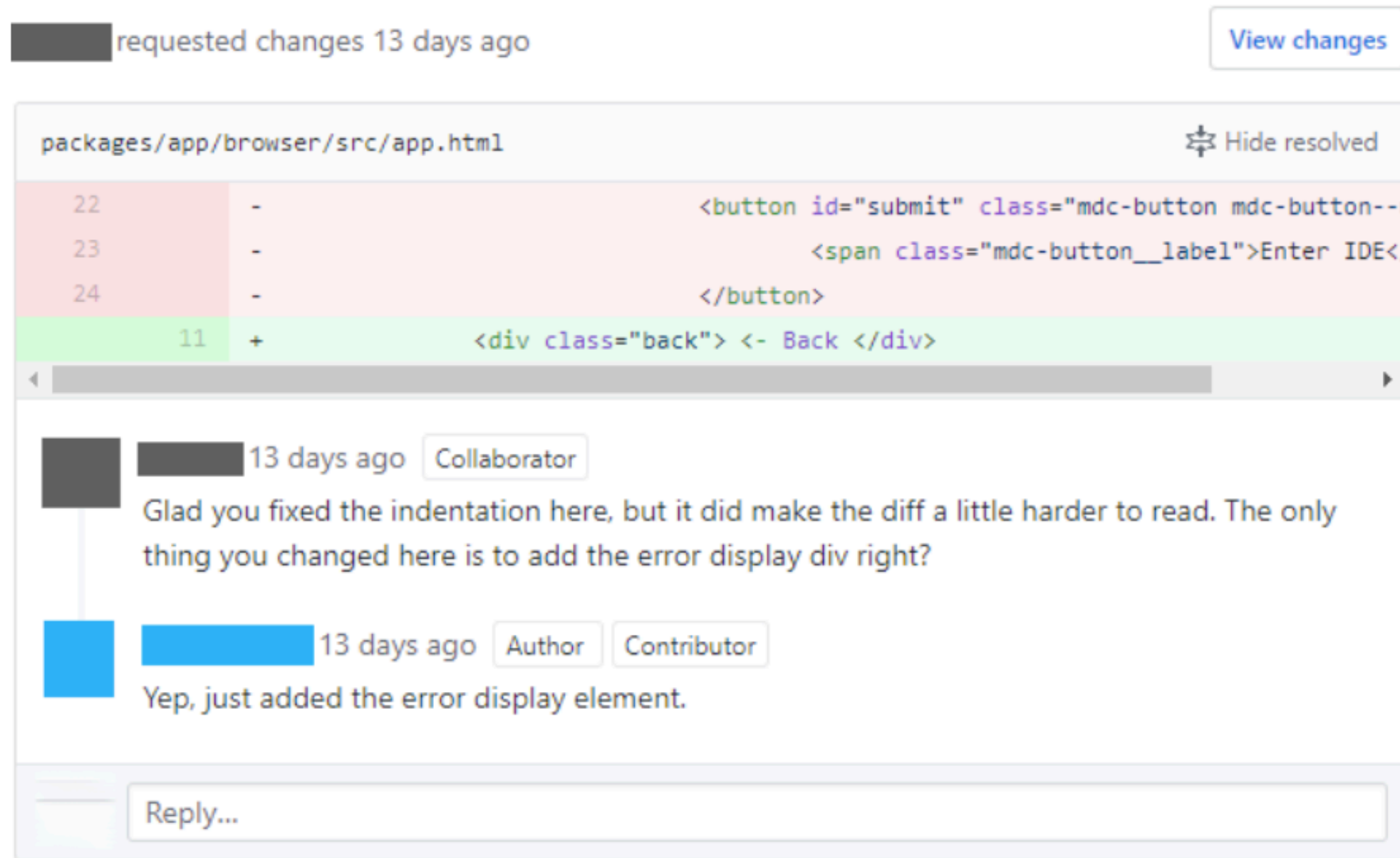


Figure 2. *Code review comments* on a pull request are attached to specific lines of code. As above blue is the author of the changes, while gray is the reviewer.

変更を誘発するコメントを判定するアルゴリズム

Given a pull-request with review comments and commits, we order the comments and commits such that the oldest comment or commit is first, then we iterate over them, from oldest to newest.

- If the item is a comment:
 - Record the file and line on which the comment has been placed.
- If the item is a commit:
 - If the lines and files modified by the commit intersect with any of the recorded comments:
 - * Record all of the intersecting comments as *effective comments*. Remove all of the intersecting comments, so they cannot be considered twice.
 - For every recorded comment we update its position based on the files modified by the commit.
 - * If lines have been added by the commit above a recorded comment the current position of the comment is moved down by the number of added lines. This is done for all recorded comments.
 - * If lines have been removed by the commits above a recorded comment the current position of the comment is moved up by the number of removed lines. This is done for all recorded comments.

Figure 4. Algorithm used to extract *effective-comments*.