

Code Review

Hyunchan, Park

<http://oslab.jbnu.ac.kr>

Division of Computer Science and Engineering

Jeonbuk National University



학습 내용

- 코드 리뷰 1
- 실습



코드 리뷰

- 코드 리뷰

- 협업 과정 중에 각자 수정한 코드를 서로 확인하는 과정
- 목적: 협력을 통한 품질 향상
 - 오류 검출 뿐 아니라 코드 가독성도 높이고,
팀원들의 전체 코드에 대한 이해를 높여,
보다 품질 좋은 코드를 작성하기 위함
- 다양한 방식으로 수행
 - 회의를 열고 line by line 으로 세밀하게 체크
 - Merge, release 이후에 필요한 시점에 필요한 코드만 서로 확인

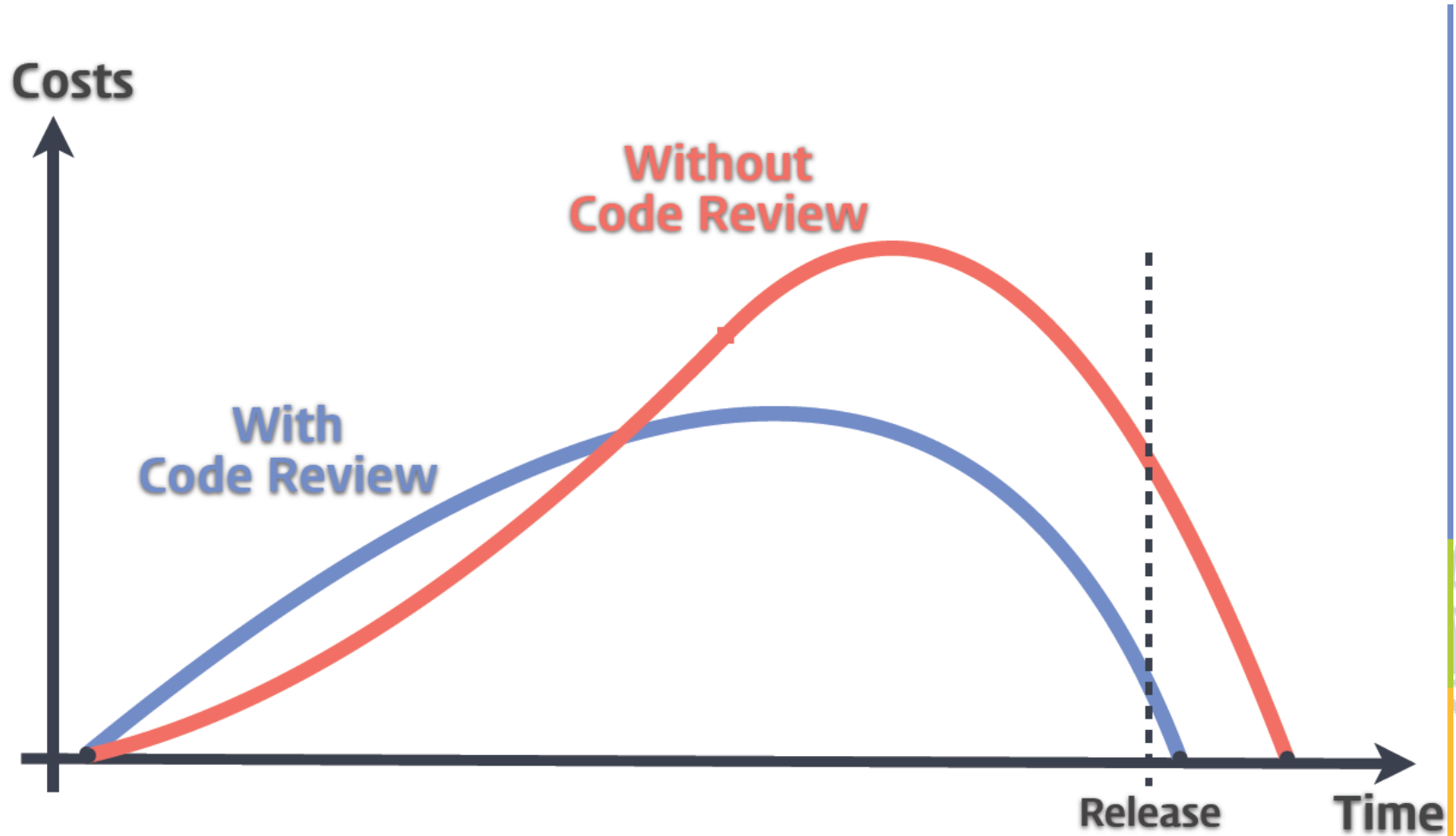
코드 리뷰 효과

- 버그 개수 감소
- 팀원의 트레이닝 도구로 활용
- 코드 가독성 증가 및 품질 상승
- 코드 세부 구현사항에 대한 기록 보존
- 팀 역량 상향평준화
- 디버깅 시간 및 프로젝트 수행 기간 단축



코드 리뷰 효과

- 장기적인 안목에서 도입

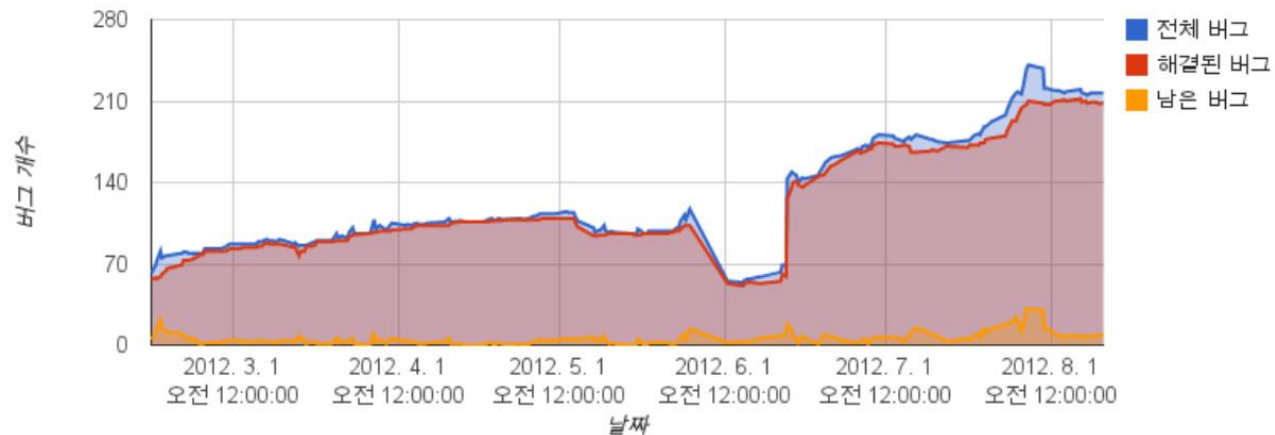


코드 리뷰 효과

차트 제목



C사 버그 관리 현황



코드 리뷰

- Experience from Naver Development team
 - 개발 전체 단계에서 코드 품질을 높이기 위한 노력
 - 표준 코드 스타일 제정
 - 단위 테스트, 코드 리뷰 회의
 - 자동화된 코드 품질 측정: Jenkins CI 도구의 plugin 사용
 - 기존 코드 리뷰 방법
 - 개발 이후, 브랜치 병합 이후에 정례/비정례 코드 리뷰 회의
 - 회의 시간이 길어짐: 작성자의 코드 설명
 - 배포 이후라, 코드 리뷰를 생략하게 되고, 하더라도 오류를 뒤늦게 발견
 - 코드 리뷰를 배포 이전에 강제할 필요가 있음
 - Gerrit을 코드 리뷰 시스템으로 선정하고, 개발 과정에 포함

코드 리뷰 예

```
83 +     isBlockNode: function (node) {  
84 +         return node && (node.nodeName === 'BR' || node.nodeName === 'IMG');
```



Owner



nodeName이 UpperCase가 아닌 경우도 있어요.

<http://ejohn.org/blog/nodename-case-sensitivity/>

버그 발견



Owner



아랫쪽도 같이 챙겨주세요.



Owner



HTML일때는 모두 upper case 아닌가요? 그렇게 되어 있는 코드가 좀 있을건데..



Owner



코드 리뷰 예코드 리뷰 예

```
299 +         if (offset >= this.getNodeLength(node)) {  
300 +             return node.nextSibling;
```



Owner

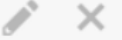


offset이 node.nextSibling.nextSibling 이 될 가능성은 없나요?
그리고 그 때 문제가 될 가능성은..?

잠재적 버그



Owner



point의 offset이 container의 크기를 넘는 경우는 사실 없을텐데... 방어코드 삼아 넣어줬어요. 넘는 경우 이미 깨진 상태라고 봐도 될거 같아요 ;;

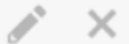


코드 리뷰 예

```
271 +     appendChildNodes: function (node, childNodes) {  
272 +         _.each(childNodes, function (childNode) {
```



Owner

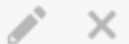


속도 문제가 있다면 글쓰기 쪽만 고전적인 방법으로 가도 좋을 것 같아요

성능



Owner



노드를 삽입하는 함수들은 자주 호출되는 편은 아니라 일단 이렇게 갔어요.



코드 리뷰 예

```
1023 + decoratorTypes = _.pluck(JSON.parse(requestData.params.containerId), 'decoratorTypes');
1024 + if (_.contains(decoratorTypes, decoratorUtil.DECORATOR_TYPE.PROFILE)) {
```



Owner



루프가 3번 도네요. 한 번에 끝낼 수 있을 것 같아요.

```
_.some(decoratorTypes, function (decoratorType) {
  switch (decoratorType) {
    case decoratorUtil.DECORATOR_TYPE.PROFILE:
      ...
      break;
  }
});
```

성능

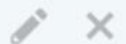


코드 리뷰 예

```
5 + /**
6 +  * W3C 레인지 모델을 제공하는 유틸리티
7 +  *
8 +  * 사전에 알아야할 자료형은 아래와 같다.
9 +  * - BoundaryPoint: DOM 트리 내부에 한 지점을 뜻함
10 +  * - BoundaryPoints: 두 지점으로 구성되어 트래 내부에 영역을 뜻함
```



Owner




트래 --> 트리 ㅎㅎ

오타



코드 리뷰 예




Owner

중간에 대시가 없는 것도 있어서 가독성이 좀 떨어지는데, 아래처럼 풀어두면 어때요~?

```
var exceptRange = [  
  '\u0000-\u002F',  
  '\u003A-\u0040',  
  '\u005B-\u005E',  
  '\u0060',  
  '\u007B-\u007F'  
].join('');
```

가독성



Owner

아스키 코드 범위에서 태그로 사용되지 않을 특수기호 및 문자들의 범위를 지정한 코드로, 저 코드를 직접적으로 읽을 일이 많이 없어서 보기 좋게 정리를 할 필요성은 적은 것 같아요^^

다른 분들 중에서도 저 코드의 가독성을 위해 변경이 필요한지에 대해 의견 부탁드립니다~



Owner

위에 주석에 잘 설명되어있으니 이러나 저러나 상관없다고 생각해요



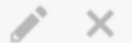
Owner

의견에 한 표. 더 읽기 좋은 방법이라고 생각해요. 그치만... 역시 저도 별 상관은 없다고 느껴지네요

코드 리뷰 예

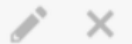
네이밍

Owner



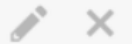
이게 `domUtil.split()` 인건데, 뭘 자르는건지 이름만으로는 모르겠어요. `String#split` 이 떠오르네요

Owner



텍스트 노드의 경우 `splitText` 실행하고 엘리먼트의 경우 엘리먼트를 클론해서 자식노드를 잘라요.

Owner



뭘 자르는건지 모르겠어서 찾아봤는데, 요건 특정 포지션을 기준으로 둘로 나누는 `TextNode`의 `splitText` 기능을 일반화한거군요. `splitNode` 하면 더 쉽게 이해할 것 같은데, 그럼 `domUtil` 보다는 `nodeUtil` 쪽에 `split` 으로 존재하면 더 좋을 것 같아요.



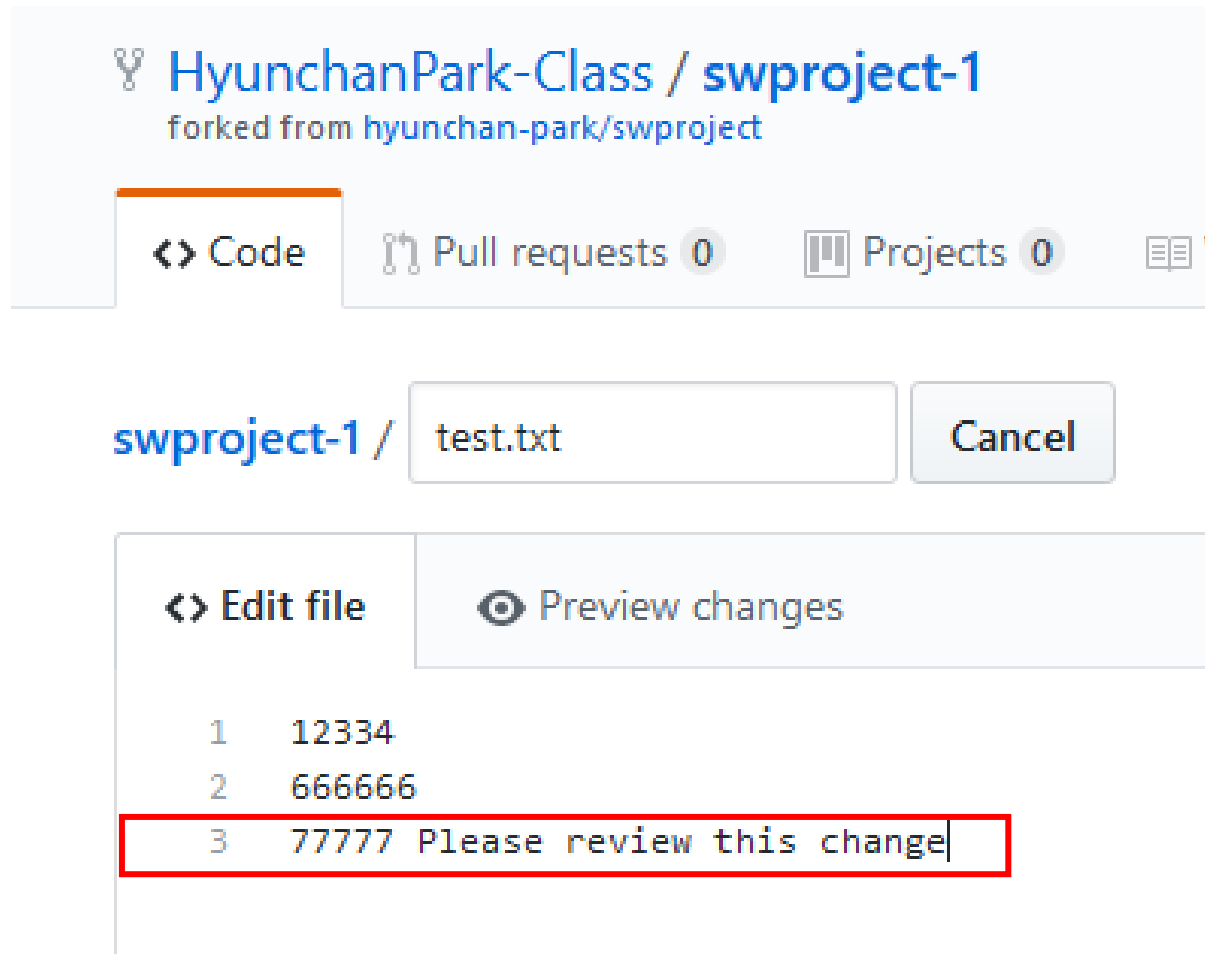
GitHub Code Review



GitHub 기반의 코드 리뷰

- Pull request 를 이용한 리뷰 운영 방법
 - 항상 Pull request를 이용해서 commit을 업로드함
 - 각 개발자는 fork해서, 혹은 개별 branch 에서 작업함
 - 리뷰 규칙을 정함
 - 예) Review master가 리뷰 여부 및 리뷰어 설정,
 - 리뷰 규칙: 1일 이내에 수정 요청, 코멘트 혹은 approve 여부 작성
 - 모든 response는 1일 이내에 할 것
 - 리뷰가 완료되면 review master가 merge (혹은 merge 담당에게 요청)
- 역할
 - Maintainer: repository 관리자 (merge 권한)
 - Reviewer: repository collaborator
 - Contributor: 외부 기여자

Contributor: Fork, 파일 수정 및 커밋



The screenshot displays the GitHub interface for a repository named 'HyunchanPark-Class / swproject-1', which is a fork of 'hyunchan-park/swproject'. The 'Code' tab is selected, showing the file 'test.txt' in the editor. The file content is as follows:


```
1 12334
2 666666
3 77777 Please review this change|
```

The third line of the file is highlighted with a red box, indicating the current position of the cursor. The interface also shows options for 'Pull requests' (0) and 'Projects' (0).

Contributor: PR 작성

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

 base repository: **hyunchan-park/swproject** ▼ base: **master** ▼ ← head repository: **HyunchanPark-Class/swproject-1** ▼ compare: **master** ▼

✓ **Able to merge.** These branches can be automatically merged.


 **Create pull request** Discuss and review the changes in this comparison with others. 

 **1** commit  **1** file changed  **0** commit comments  **1** contributor

 Commits on Nov 05, 2019

  **HyunchanPark-Class** Update test.txt Verified ebcd421


 Showing **1** changed file with **1** addition and **0** deletions. Unified Split

| ▼ 1 test.txt  | | | ... |
|--|-----|-----------------------------------|-----|
| ... | ... | @@ -1,2 +1,3 @@ | |
| 1 | 1 | 12334 | |
| 2 | 2 | 666666 | |
| | 3 | + 77777 Please review this change | |


기존 Repo 에서 PR 확인. 오른쪽 Reviewer 항목


Update test.txt 리뷰 부탁드립니다. #5


Edit

 Open HyunchanPark-Cl... wants to merge 1 commit into [hyunchan-park:master](#) from [HyunchanPark-Class:master](#) 

 Conversation 0

 Commits 1


 Checks 0

 Files changed 1


+1 -0 



HyunchanPark-Class commented now

Contributor +  ...


No description provided.


 Update test.txt

Verified ebcd421

Add more commits by pushing to the **master** branch on [HyunchanPark-Class/swproject-1](#).



 Continuous integration has not been set up
[Several apps are available](#) to automatically catch bugs and enforce style.

 This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).




Write

Preview

AA B i “ <>    @  

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.


 Close pull request

Comment

Reviewers



Suggestions

 hyunchan-park

[Request](#)

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet


Milestone



No milestone

Notifications

[Customize](#)

 Unsubscribe

You're receiving notifications because you're watching this repository.

1 participant



 Lock conversation



Maintainer: Suggestion 에서 선택하거나, 리뷰어 선택

The screenshot displays a GitHub pull request interface. At the top, a progress bar shows '+5 -0' with five green squares. Below this, a tabbed interface has 'Contributor' and 'Reviewers' tabs. A red box highlights the 'Request a review' dialog, which includes a 'Filter people' search bar, a 'Suggestions' section with a highlighted entry for 'HyunchanPark-Class' (noted as 'Recently edited these files'), and an 'Everyone else' section with another entry for 'HyunchanPark-Class'. To the right, the sidebar contains sections for 'Reviewers' (listing 'HyunchanPark-Class'), 'Assignees' (showing 'No one—assign yourself'), 'Labels' (showing 'None yet'), 'Projects' (showing 'None yet'), 'Milestone' (showing 'v0.1 구현'), and 'Notifications' (with an 'Unsubscribe' button and a message: 'You're receiving notifications because your review was requested.'). At the bottom of the sidebar, it indicates '2 participants'.

PR Review 요청 메일

★ [hyunchan-park/swproject] Update test.txt 리뷰 부탁드립니다. (#5)

▼ 보낸사람 HyunchanPark-Class <notifications@github.com> 연관 메일 보기 | 주소 등록 | 수신차단

You can view, comment on, or merge this pull request online at:

<https://github.com/hyunchan-park/swproject/pull/5>

Commit Summary

- Update test.txt

File Changes

- M [test.txt](#) (1)

Patch Links:

- <https://github.com/hyunchan-park/swproject/pull/5.patch>
- <https://github.com/hyunchan-park/swproject/pull/5.diff>

You are receiving this because you are subscribed to this thread.

Reply to this email directly, [view it on GitHub](#), or [unsubscribe](#).

<> Code ⓘ Issues 0 🔗 Pull requests 1 📁 Projects 0 📖 Wiki 🛡 Security 📊 Insights ⚙ Settings

This pull request is waiting on your review.

Add your review




Reviewer: Review 를 파일 별로 작성

Update test.txt 리뷰 부탁드립니다. #5

[Edit](#)

 Open HyunchanPark-Cl... wants to merge 1 commit into hyunchan-park:master from HyunchanPark-Class:master 

 Conversation 0

 Commits 1

 Checks 0

 Files changed 1

+1 -0 

Changes from all commits ▾ File filter... ▾ Jump to... ▾ ⚙ ▾

0 / 1 files viewed ⓘ

[Review changes ▾](#)

▼ 1  test.txt 

... @@ -1,2 +1,3 @@

1 1 12334

2 2 666666

3 + 77777 Please review this cl

Write

Preview

AA B i “ <> 🔗 ⋮ ⋮ ⋮ @ 📌 ↶

Leave a comment

Attach files by dragging & dropping, selecting or pasting them. 

☒ Comment

Submit general feedback without explicit approval.

☐ Approve

Submit feedback and approve merging these changes.

☐ Request changes

Submit feedback that must be addressed before merging.

[Submit review](#)

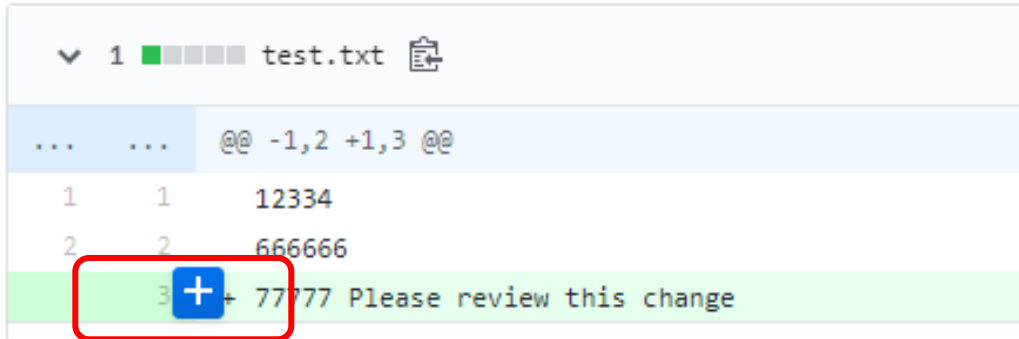
© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#)



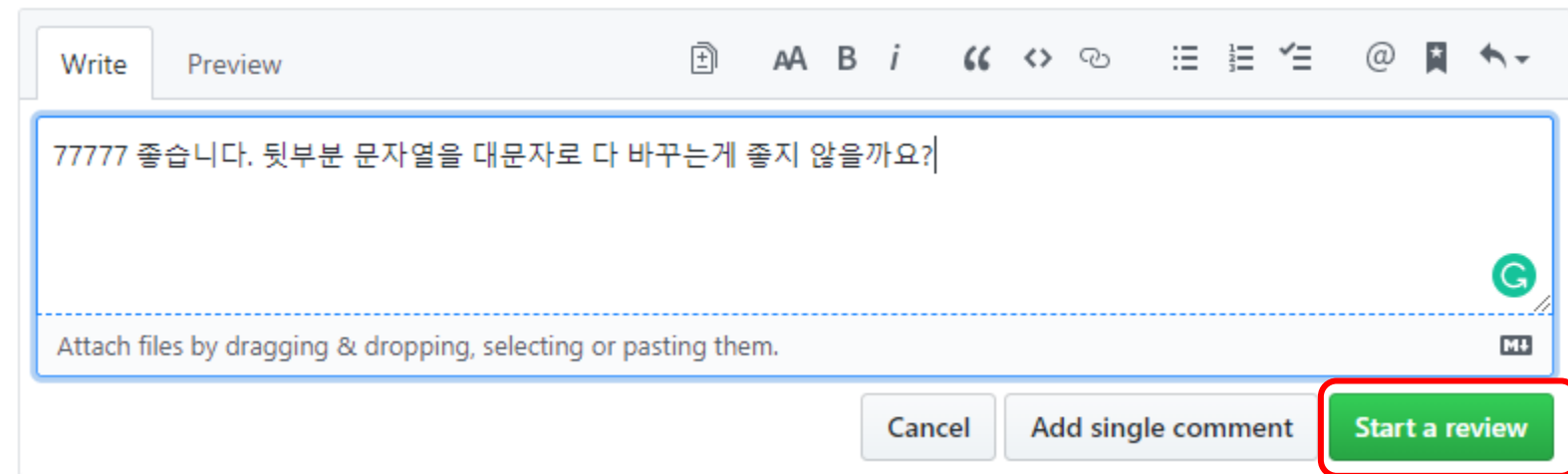
전북대학교 컴퓨터공학부

Division of Computer Science and Engineering
Chonbuk National University

Reviewer: 파일 변경 내용에서 코멘트



```
1 12334
2 666666
3 + 77777 Please review this change
```



Write Preview

77777 좋습니다. 뒷부분 문자열을 대문자로 다 바꾸는게 좋지 않을까요?

Attach files by dragging & dropping, selecting or pasting them.

Cancel Add single comment **Start a review**

Reviewer: Finish the review

Update test.txt 리뷰 부탁드립니다. #5


Edit

 Open HyunchanPark-Cl... wants to merge 1 commit into `hyunchan-park:master` from `HyunchanPark-Class:master` 

 Conversation 0

 Commits 1

 Checks 0

 Files changed 1

+1 -0 

Changes from all commits ▾ File filter... ▾ Jump to... ▾ ⚙ ▾

0 / 1 files viewed



Finish your review 1 ▾

▼ 1  test.txt 

☐ Viewed ...

... @@ -1,2 +1,3 @@

1 1 12334

2 2 666666

3 + 77777 Please review this change



hyunchan-park

Owner

Pending

+ 😊 ...

77777 좋습니다. 뒷부분 문자열을 대문자로 다 바꾸는게 좋지 않을까요?



Reply...



전북대학교 컴퓨터공학부

Division of Computer Science and Engineering
Chonbuk National University

Reviewer: Finish the review

Update test.txt 리뷰 부탁드립니다. #5

[Edit](#)

Open HyunchanPark-Cl... wants to merge 1 commit into `hyunchan-park:master` from `HyunchanPark-Class:master`

Conversation 0

Commits 1

Checks 0

Files changed 1

+1 -0

Changes from all commits ▾ File filter... ▾ Jump to... ▾

0 / 1 files viewed

Finish your review 1 ▾

1 test.txt

| | | |
|-----|-----|-----------------------------|
| ... | ... | @@ -1,2 +1,3 @@ |
| 1 | 1 | 12334 |
| 2 | 2 | 666666 |
| 3 | + | 77777 Please review this cl |

hyunchan-park **Owner** Pending
77777 좋습니다. 뒷부분 문자열을

Reply...

Write

Preview

코드에 남긴 코멘트를 확인해주세요


Attach files by dragging & dropping, selecting or pasting them.

- ☐ **Comment**
Submit general feedback without explicit approval.
- ☐ **Approve**
Submit feedback and approve merging these changes.
- ☒ **Request changes**
Submit feedback that must be addressed before merging.

Submit review 1 pending comment



Pull request 를 전송한 사람에게 메일 전송됨

Re: [hyunchan-park/swproject] Update test.txt 리뷰 부탁드립니다. (#5)  받은편지함 x

Hyunchan, Park <notifications@github.com>

hyunchan-park/swproject, 나, Author에게 ▾

@hyunchan-park requested changes on this pull request.

코드에 남긴 코멘트를 확인해주세요

In [test.txt](#):

```
> @@ -1,2 +1,3 @@
```

```
12334
```

```
666666
```

```
+77777 Please review this change
```

77777 좋습니다. 뒷부분 문자열을 대문자로 다 바꾸는게 좋지 않을까요?

—

You are receiving this because you authored the thread.

Reply to this email directly, [view it on GitHub](#), or [unsubscribe](#).



Contributor: 해당 PR 페이지에서 리뷰 내용 확인



hyunchan-park requested changes in 10 seconds

[View changes](#)

hyunchan-park left a comment

Owner



코드에 남긴 코멘트를 확인해주세요

test.txt

... ... @@ -1,2 +1,3 @@

1 1 12334

2 2 666666

3 + 77777 Please review this change



hyunchan-park in 10 seconds

Owner



77777 좋습니다. 뒷부분 문자열을 대문자로 다 바꾸는게 좋지 않을까요?



Reply...

Resolve conversation



Contributor: 바로 editing

Update test.txt 리뷰 부탁드립니다. #5

[Edit](#)

Open HyunchanPark-Cl... wants to merge 2 commits into `hyunchan-park:master` from `HyunchanPark-Class:master`

Conversation 2

Commits 2

Checks 0

Files changed 1

+1 -0

Changes from 1 commit ▾ File filter... ▾ Jump to... ▾ ⚙ ▾

0 / 1 files viewed

[Review changes ▾](#)

▼ 1 test.txt

☐ Viewed ...

... @@ -1,2 +1,3 @@

1 1 12334

2 2 666666

3 + 77777 Please review this change

☒ Show comments

View file

Edit file

Delete file

Open in desktop



hyunchan-park 5 minutes ago Owner

+ ...

77777 좋습니다. 뒷부분 문자열을 대문자로 다 바꾸는게 좋지 않을까요?




Reply...





[Resolve conversation](#)






Contributor: 코드 수정 및 커밋

- 작성자가 리뷰에 따라 다시 수정
 - 이때 작성자의 작업 브랜치에 commit이 되고, 기존 pull request에 병합됨


 HyunchanPark-Class / swproject-1
forked from hyunchan-park/swproject

 Code  Pull requests 0  Projects 0  Wiki

swproject-1 / test.txt 

 Edit file  Preview changes

```
1 12334
2 666666
3 77777 PLEASE REVIEW THIS CHANGE
4
```





Commit changes

리뷰에 따라 문자열을 대문자로 수정하였습니다.

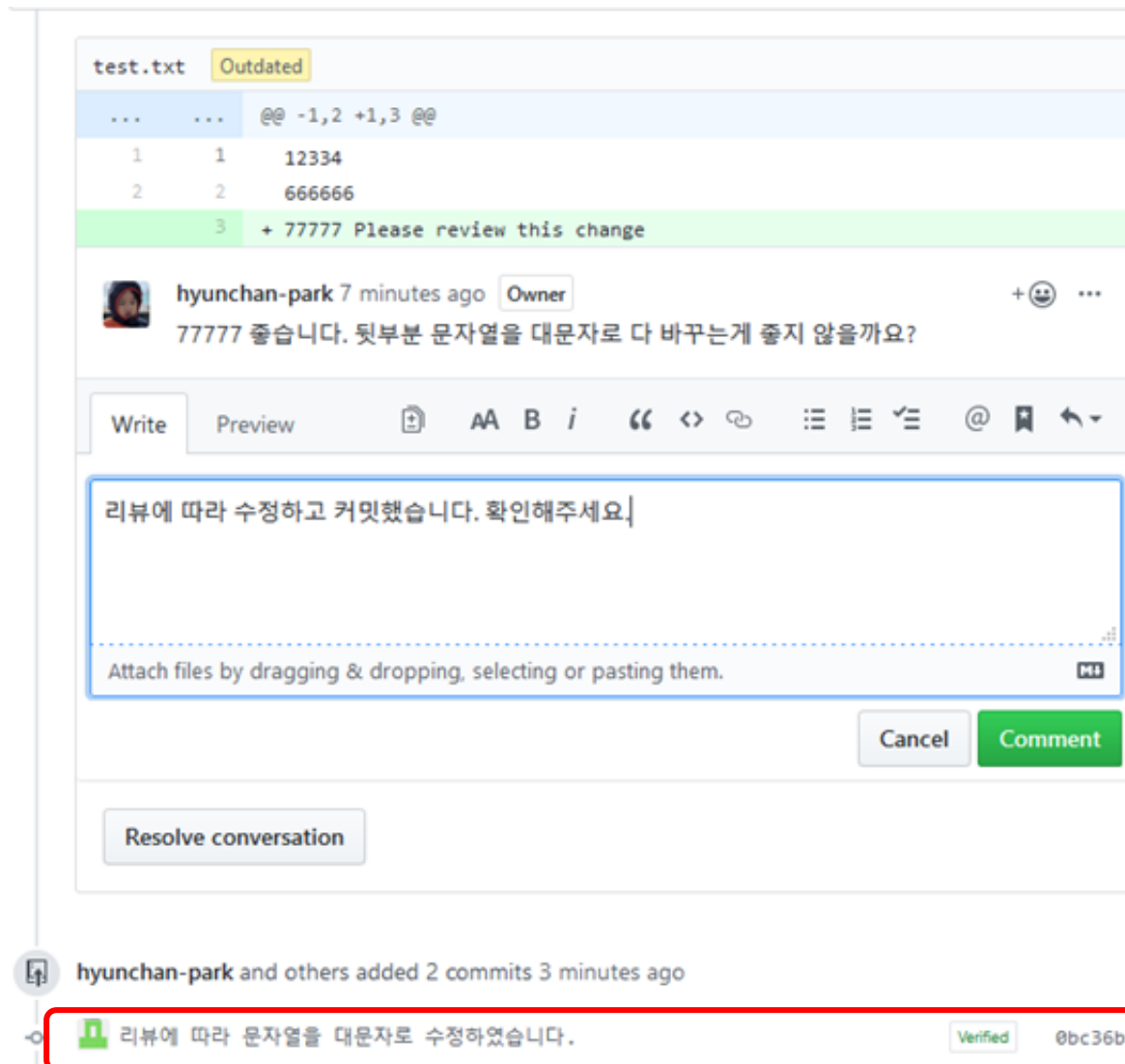
Add an optional extended description...

☒ Commit directly to the master branch.

☐ Create a new branch for this commit and sta

Contributor: 리뷰에 따른 수정 완료









The screenshot displays a GitHub pull request interface. At the top, a file named `test.txt` is shown with a yellow `Outdated` label. Below this, a diff view shows changes to the file. Line 3 has been added, containing the text `+ 77777 Please review this change`. A comment from `hyunchan-park`, the owner, is posted 7 minutes ago, stating: "77777 좋습니다. 뒷부분 문자열을 대문자로 다 바꾸는게 좋지 않을까요?". The comment input area contains the text "리뷰에 따라 수정하고 커밋했습니다. 확인해주세요|". Below the input area, there are buttons for `Cancel` and `Comment`, and a `Resolve conversation` button. At the bottom, a commit message is shown: "리뷰에 따라 문자열을 대문자로 수정하였습니다." (I modified the string to uppercase according to the review). The commit is marked as `Verified` with a green checkmark and the hash `0bc36b9`.

test.txt Outdated


```
...    ...    @@ -1,2 +1,3 @@
1      1      12334
2      2      666666
3      + 77777 Please review this change
```

hyunchan-park 7 minutes ago Owner + 😊 ...

77777 좋습니다. 뒷부분 문자열을 대문자로 다 바꾸는게 좋지 않을까요?

Write Preview  AA B i “ < >     @  ↶

리뷰에 따라 수정하고 커밋했습니다. 확인해주세요|

Attach files by dragging & dropping, selecting or pasting them. 

Cancel Comment

Resolve conversation

hyunchan-park and others added 2 commits 3 minutes ago


리뷰에 따라 문자열을 대문자로 수정하였습니다. Verified 0bc36b9



Reviewer: 리뷰에 대한 응답 확인

3

+ 77777 Please review this change



hyunchan-park


9 minutes ago

Owner

+ 😊

...

77777 좋습니다. 뒷부분 문자열을 대문자로 다 바꾸는게 좋지 않을까요?



HyunchanPark-Class

14 seconds ago


Author

Contributor

+ 😊

...

리뷰에 따라 수정하고 커밋했습니다. 확인해주세요.



Resolve conversation

+

New changes since you last viewed

View changes



Reviewer: 수정 확인 및 Approve (리뷰 완료)

Update test.txt 리뷰 부탁드립니다. #5

[Edit](#)

Open HyunchanPark-Cl... wants to merge 3 commits into `hyunchan-park:master` from `HyunchanPark-Class:master`

Conversation **3** Commits **3** Checks **0** Files changed **1**

+1 -0

Changes from 2 commits File filter... Clear filters Jump to...

0 / 1 files viewed

[Review changes](#)

| 2 test.txt | | |
|-------------|-----|----------------------------|
| ... | ... | @@ -1,3 +1,3 @@ |
| 1 | 1 | 12334 |
| 2 | 2 | 666666 |
| 3 | - | 77777 Please review this c |
| 3 | + | 77777 PLEASE REVIEW THIS C |

Write

Preview

A A **B** *i* “ ” < > @

잘 수정되었습니다|

Attach files by dragging & dropping, selecting or pasting them.

- ☐ **Comment**
Submit general feedback without explicit approval.
- ☒ **Approve**
Submit feedback and approve merging these changes.
- ☐ **Request changes**
Submit feedback that must be addressed before merging.

[Submit review](#)

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#)



Maintainer: 리뷰 확인 및 Merge 수행



hyunchan-park approved these changes now

[View changes](#)

[hyunchan-park](#) left a comment

Owner + 😊 ...

잘 수정되었습니다!

Add more commits by pushing to the **master** branch on **HyunchanPark-Class/swproject-1**.



Changes approved

[Show all reviewers](#)

1 approving review [Learn more](#).



1 approval



Continuous integration has not been set up

[Several apps are available](#) to automatically catch bugs and enforce style.



This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request



You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



전북대학교 컴퓨터공학부

Division of Computer Science and Engineering
Chonbuk National University

개인 과제 #10: GitHub-based Code Review

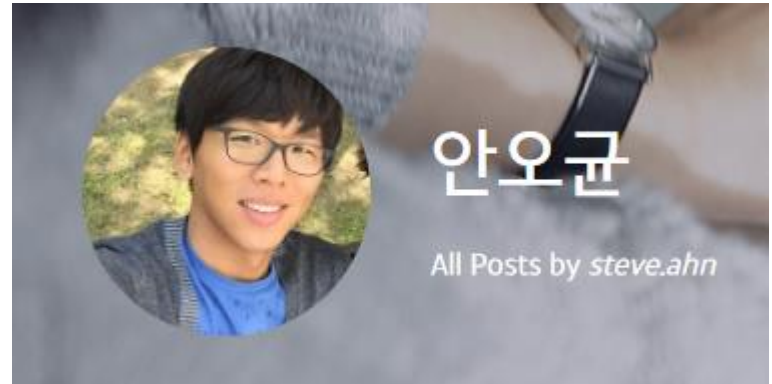
- 두 사람이 짝을 이뤄 진행
 - 각자 Repository 하나를 생성 (혹은 기존 예제 사용)
 - 각자 서로의 Repo 를 Fork
 - 각자 서로 두 개의 파일에 대한 수정 사항을 발생시킨 후, PR 요청
 - 받은 PR에 대해 본인 스스로를 리뷰어로 할당하고, 리뷰 수행
 - 보낸 PR에 대한 리뷰를 확인하고, 수정 작업 진행 (review.jpg)
 - 리뷰 마무리 후, Merge 수행
 - PR merge 완료 화면을 pull request (closed) 창에서 캡처 (prs.jpg)
- 제출 기한:
 - 11/17 (일) 23:59
 - 지각 감점: 5%p / day (3주 내 제출해야 함)

학습 내용

- 코드 리뷰 2: 카카오토티의 코드 리뷰 경험
- 코드 리뷰 3: 도구들
 - Gerrit 소개
 - Crucible (+ Fisheye)
 - 설치 및 구성
 - 써보자!



코드 리뷰 2:



코드 리뷰는 문화다

카카오톡에서의 코드 리뷰 경험



참고자료: 코드 리뷰 소개 및 경험

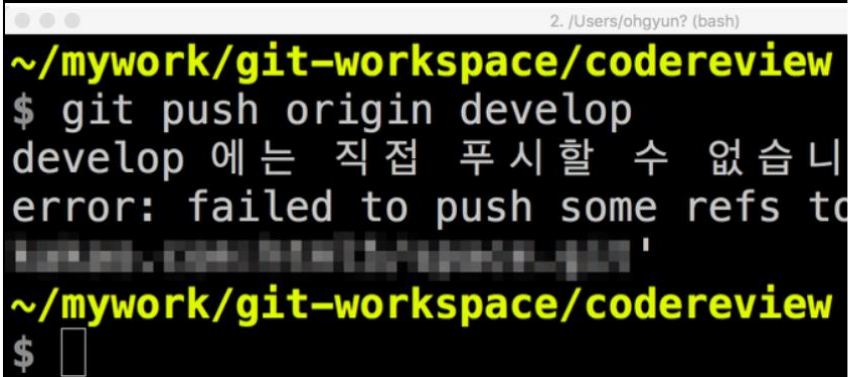
- 카카오스토리 웹팀의 코드리뷰 경험, 안오균 (2016)
 - <https://www.slideshare.net/OhgyunAhn/ss-61189141>
- JavaScript 코드 리뷰 - 코드 리뷰 문화 (2016)
 - <https://cimfalab.github.io/deepscan/2016/08/code-review-1>
- 코드 리뷰: 다음커뮤니케이션 (2013 DevOn)
 - <https://www.slideshare.net/pioneerhjlee/code-review-devon2013>

카카오스토리

- 서비스: 카카오스토리 웹
 - 프론트엔드 + 웹서버
 - JavaScript 95%, nginx, Bash
 - Backbone 기반 자체 프레임워크
- 코드 리뷰 도입부터 안정화까지
 - 도입: 3명, GitHub PR 활용, 모든 코드 리뷰 (2013.10~)
 - 2년간 사용하며 꾸준히 개선
 - 강제 리뷰 후 머지
 - 리뷰 규칙 구체화: 2단계 리뷰, 리뷰 데이, PR 및 커밋 규칙
 - 오프라인+온라인 병행
 - 브랜치 미리보기 서버 구성
 - 코드 리뷰 팀 구분, 리뷰 마스터 설정

경험 1: PR 이용 시 실수 발생

- 예외 상황
 - Develop 브랜치에 바로 푸시
 - 실수로, 혹은 간단해서
- 해결 방법
 - Pre-push Git Hook 활용
 - Push 이벤트에 대해 자동 스크립트를 추가해 바로 머지가 되지 않도록
 - <https://git-scm.com/book/ko/v1/Git%EB%A7%9E%EC%B6%A4-Git-%ED%9B%85>
- Pre-push 깃훅의 효과
 - 모든 피처가 Pull Request를 통해서 머지됨
 - 초기 리뷰 문화 정착에 도움
 - 이 단계부터 '리뷰는 필수'라는 인식이 자리잡음



```
2. /Users/ohgyun? (bash)
~/mywork/git-workspace/codereview
$ git push origin develop
develop 에는 직접 푸시할 수 없습니다
error: failed to push some refs to
'...'
~/mywork/git-workspace/codereview
$
```

경험 2: 코드 컨벤션에 대한 리뷰

- 코드 스타일에 대한 리뷰가 대부분
 - 탭, 공백, 들여쓰기, 캐멀케이스, 언더스코어
 - 스타일 가이드 + 에디터 포맷터는 존재
 - 하지만 잘 지켜지지 않음
- 사실 코드 스타일 리뷰는 피곤함
 - 코드 스타일을 통일하면 참 좋겠지만
 - 남기는 사람도 보는 사람도 불편함
 - 도구로 해결할 수 있을까?

경험 2: 코드 컨벤션에 대한 리뷰

- 해결방법: pre-commit 깃훅에서 린트 수행
 - ✓ Code linting
 - ✓ 정적 코드 분석 및 테스트를 통해 잠재적 에러 및 성능 분석을 하는 것
 - ✓ 사용자 정의 방식으로, 코드 컨벤션 검사를 위해 활용할 수 있음
 - ✓ 도구: jshint, ESLint, Lint in Android studio 등
 - ✓ 참고: <https://subicura.com/2016/07/11/coding-convention.html>
 - 해당 커밋에서 변경한 파일 대상
 - 린트 실패 시 커밋할 수 없음
- 효과: pre-commit 훅의 린트는 성공적
 - 코드 스타일에 대한 리뷰가 크게 감소함
 - 수정하는 파일만 대상이므로 거부감이 적었음
 - 대상 파일의 나머지 코드도 수정해야 함
 - 수 개월 후 프로젝트 린트 오류 0%

경험 2: 코드 컨벤션에 대한 리뷰

```
2. /Users/ohgyun? (bash)
$ git commit -m "invalid style"
Running "jshint-changed" task

Running "jshint:prepush" (jshint) task

src/xxxxx.js
  21 |      callFunctionWithoutSemicolon()
                                     ^ Missing
g semicolon.

>> 1 error in 1 file
Warning: Task "jshint:prepush" failed. Use --force
to continue.

Aborted due to warnings.
린트에 통과하지 못한 파일이 있습니다. 확인 후 다시
시도해 주세요.
~/mywork/git-workspace/codereview on develop*
$
```

경험 2: 코드 컨벤션에 대한 회고

- 코드 스타일은 꼭 맞춰야할까?
 - 한 사람이 짠 것 같은 코드: 읽고 수정하기 편하고 리뷰 속도도 빨라짐
 - 결국 팀의 속도가 빨라짐
- 코드 스타일에 대한 리뷰는 필요했을까?
 - 코드 스타일 리뷰는 모두의 스트레스였음
 - 특히 규칙이 정해져 있지 않았을 때 더했음
 - 시간이 아깝다고 느껴지기도 했음
- 코드 스타일은 도구로 해결하자
 - 상세한 단위까지 포맷터를 적용
 - 도구가 준비되지 않았다면 생략해도 좋을 듯
- 그래도 해야 하는 부분이 있더라
 - 주석 처리된 코드, 쓰이지 않는데 나중에 위해 아껴둔 코드

경험 3: 초기 단계에서 PR 규모 문제

- PR 규모가 커서 리뷰하기 어려움
 - 프로젝트 초기, 기초 구조를 잡던 때
 - 코드량이 많고 커밋의 단위도 커 리뷰하기 어려움
 - 전체 흐름을 파악하기도 어려움
- 해결방법: 온/오프라인 리뷰 병행, PR/커밋 단위 합의
 - 오프라인 리뷰에서 전반적인 의도를 설명, 회의 종료 후 온라인 리뷰
 - 개발 정기 미팅 때 단위 합의
 - PR은 피쳐 단위로
 - 커밋은 의미있는 작업 단위로

경험 4: 리뷰로 인한 병목 현상

- 리뷰를 포함한 개발 프로세스
 - 피쳐 작업 -> 리뷰 -> develop 머지 -> 알파 배포 -> 테스트
 - 알파 빌드&테스트 서버는 develop 브랜치를 사용
- 리뷰 병목 현상
 - 바쁠수록 리뷰를 미루게 됨
 - 피쳐 작업하기 바쁘니 리뷰를 미루게 됨 (> 10 PRs in review queue)
 - 리뷰시간이 예상보다 오래 걸림
 - 큰 피쳐인 경우 1시간 이상, 많으면 하루를 다 쓰는 경우도 있음
 - 태스크 관리 도구에 '개발 리뷰 중' 단계를 추가
 - 피쳐 작업은 완료해도 리뷰 대기 중
 - 리뷰가 되지 않아 develop 으로 머지하지 못함
 - 대상 피쳐가 알파 서버에 배포되지 못함

경험 4: 리뷰로 인한 병목 현상

- 리뷰 병목으로 피해
 - 기획/디자인 직군의 불만
 - 타직군은 알파 서버에서 피쳐 확인 가능
 - 개발은 완료됐다고 하는데, 리뷰가 안돼서 확인할 수 없다!
 - 결국, 통합 테스트 때 스펙 변경이 발생함
 - 타직군은 구현된 기능을 통합 테스트 때나 보고 피드백
 - 버그 뿐 아니라 스펙과 디자인 변경이 다수 발생
 - 서로에 대한 불만
 - 개발: 내일 모레가 배포인데 스펙 변경?
 - 기획/디자인: 개발 다 됐다면서 이제 참 보여줌?
- 이 시점이 피로도가 가장 컸음
- 야근이 많아지고 의욕도 떨어짐

경험 4: 리뷰로 인한 병목 현상

- 해결방안 1: 미리보기 서버
 - ✓ 우선 기획/디자인팀과의 협업부터 해결
- 브랜치 미리보기 서버를 구성함
 - 각 피처 브랜치의 스냅샷을 배포
- 미리보기 서버는 매우 성공적
 - develop 머지 전(리뷰 전) 피처 공유 가능
 - 기획/디자인의 피드백을 미리 받고, 통합 테스트 때 스펙 변경이 크게 감소
 - 아이디어 프로토타입 공유 용도로 활발히 사용



경험 4: 리뷰로 인한 병목 현상

- 해결방안 2: 리뷰팀과 리뷰데이, 리뷰 마스터
 - ✓ 리뷰 자체를 효율적으로 수행
 - 1차 시도: 리뷰 팀을 두 개로, 별도 리뷰어도 지정
 - 각 팀별로 리뷰 분담
 - 결과: 해결 안됨. 계속 업무는 바빠짐. 본인 피쳐 작업을 우선해야 하므로, 통합 테스트 날짜 직전이나 리뷰 수행
 - 2차 시도: 리뷰 데이 도입
 - 매주 정해진 요일에 최우선 PR 작업 리뷰
 - 결과: 여전히 비슷한 문제가 발생. 리뷰는 계속해서 우선순위가 떨어짐.
 - 3차 시도: 리뷰 마스터 도입
 - Merge 담당 역할, 개인 판단으로 merge 수행 및 주기적으로 리뷰 독려
 - 결과: 개인 책임 하에 수행되어 잘 동작됨. 애매한 경우, 결정권을 행사하여 의사 결정이 빠름

경험 4: 리뷰로 인한 병목에 대한 회고

- 리뷰가 왜 병목이 됐을까?
 - 동료의 리뷰와 동의가 있어야 머지 가능한데, 늦게 함
- 모든 동료의 동의, 효과 있었을까?
 - 서로 어떤 작업을 하는지 알게 되고, 어렵풋하지만 대부분의 코드를 알게 됨
 - 효과는 좋음. 인원이 많아지며 병목의 원인이 됨
- 몇 명의 팀일 때 리뷰가 가장 잘 될까?
 - 2명: 피드백 빠르지만 논의 상대 부족
 - 3~5명: 전체 동의 조건으로 효과적이었음
 - 6명~: 의견/논의도 많음. 결과 대비 비효율적
- 시간이 갈수록 리뷰속도가 빨라짐
 - 일관성 있는 코드 스타일이 도움이 됨
 - 각자 중요하다고 생각하는 포인트 위주로 리뷰
 - 배포 주기가 짧아 쉽게 수정 배포 가능한 환경



경험 5: 새로운 팀 멤버 영입

- 프로젝트 멤버가 늘어남
 - 최대 8명까지 늘어남
 - 대부분 리뷰 문화가 거의 없던 팀에서 온 멤버
- 해결방안: 문화로 정착시킴. “우리 팀은 코드 리뷰를 하는 팀!”
 - ‘리뷰는 당연하다’는 문화는 정착된 상태
 - 영입 전부터 코드 리뷰 문화에 대해 강하게 언급
 - 첫 PR부터 폭풍 리뷰
- 새 멤버들의 공통된 리뷰 후기
 - 초기의 리뷰는 스트레스였다 (특히 코드 스타일)
 - 코드 학습 효과가 좋았다
 - 시간이 지나니 코드 스타일에 익숙해지더라

경험 5: 새로운 팀 멤버 영입

Filters ▾ Labels Milestones [New pull request](#)

✕ Clear current search query, filters, and sorts

| <input type="checkbox"/> | 0 Open <input checked="" type="checkbox"/> 3 Closed | Author ▾ | Labels ▾ | Milestones ▾ | Assignee ▾ | Sort ▾ |
|--------------------------|---|----------|----------|--------------|------------|--------|
| <input type="checkbox"/> | | | | 233 | | 233 |
| <input type="checkbox"/> | | | | 288 | | 288 |
| <input type="checkbox"/> | | | | 216 | | 216 |

리뷰 댓글이 200개가 넘는 Pull Request



경험 6: 여러 사람이 담당하는 피처의 리뷰

- 여러 사람이 담당하는 피처의 리뷰
 - 한 피처를 여러 명이 함께 작업하는 경우
 - 작업 범위가 겹쳐 develop으로 PR 애매함
 - 피처 단위가 커서 한 번에 리뷰하기엔 부담스러움
- 해결방법: 메인 피처 브랜치로 PR하도록 함
 - 피처의 메인 브랜치인 feature/A를 따고
 - 하위 피처를 feature/A-1, feature/A-2로 작업
 - 작업 후, feature/A-1 > feature/A 로 PR

경험 6: 여러 사람이 담당하는 피처의 리뷰

- 2번에 걸쳐 리뷰함
 - 상위 피처 브랜치로의 1차 리뷰는 담당자끼리만
 - 1차 리뷰는 큼직한 구조나 로직에 대해 러프하게
 - develop 브랜치로의 2차 리뷰는 모두가 참여
- 2차 리뷰의 효과
 - 구조 변경에 대한 피드백이 1차 리뷰에서 가능
 - 테스트 직전에 큰 변경이 적어짐
 - 두 번째 리뷰부터 확실히 속도가 빠름

정리.

2년 동안의 개선

예외적인 상황이 발생됨
컨벤션에 대한 리뷰가 대부분
PR 규모가 커서 리뷰하기 어려움
바쁠수록 리뷰를 미루게 됨
리뷰시간이 예상보다 오래 걸림
리뷰가 병목이 됨
기획/디자인 직군의 불만
통합 테스트 때 스펙 변경이 발생함
서로에 대한 불만
리뷰 병목이 폐해
프로젝트 멤버가 늘어남
생산되는 코드만큼 리뷰 대상도 늘어남
여전히 리뷰는 지연됨
테스트 직전 리뷰가 몰림
사이즈가 큰 피처는 리뷰하기 어려움
짧아진 배포 주기
리뷰가 지연되는 PR이 발생
жат은 배포에 대한 피로감

리뷰 없이 머지할 수 없도록 제한
pre-commit 깃훅에서 린트 수행
오프라인과 온라인 리뷰를 병행
PR과 커밋 단위에 대한 합의
브랜치 미리보기 서버를 구성함
우리 팀은 코드 리뷰를 하는 팀
리뷰 팀을 구분
리뷰 데이 도입
리뷰 마스터 제도 도입
메인 피처 브랜치로 PR하도록 함
2번에 걸쳐 리뷰함
리뷰 규칙을 간소화
오래된 리뷰는 빨리 머지
주간 배포 방식으로 변경



회고 1. 리뷰는 서로에게 도움이 되었을까?

- 팀원들 후기
 - 새로운 스타일 또는 접근 방법을 알게 됨
 - “배울 게 많았다”
 - 다른 사람의 코드를 읽는 시간이 많아졌다
 - 논의 과정에서 서로 성장하는 느낌
 - ‘왜 이렇게 했을까’ 생각하고, ‘왜 이렇게 했는지’ 설명하는 시간이 많아짐
 - 이제는 안하면 뭔가 불안함
- 직접적 효과
 - 긴급 핫픽스 코드에서 버그 발견!
 - 코드 파악: 내가 짰 것 같은데 알고 보니 다른 사람이 짰음
 - 인수인계 할 게 거의 없음

회고 2. 리뷰의 유익함, 무익함

- 유익하다고 느꼈던 리뷰
 - 미리 발견하는 버그
 - 경험의 공유 (삽질 회피, 기존 코드 히스토리)
 - 더 나은 제안 (로직, 변수명)
- 조금은 불필요한 논쟁이라고 생각한 리뷰
 - 취향의 차이 (if vs switch)
 - 애매한 수준의 제안 (변수명, 미미한 성능 개선)
 - 너무 먼 미래에 대한 방어 코드

회고 2. 리뷰의 유익함, 무익함

- 리뷰에 대해 공통적으로 느꼈던 스트레스
 - 코드 스타일의 사소한 부분까지 강요당했을 때
 - 피쳐도 작업하라, 리뷰하라, 피드백하라
 - 내일이 마감인데, 전체 구조를 변경하는 리뷰
 - 내일이 테스트인데, 쌓인 리뷰가 수십 개
- 근본적으로 해결하기 어려운 문제들
 - 여전히 리뷰가 병목이 될 수 있다
 - 리뷰 문화 정착까지의 비용이 크다
 - 가끔은 리뷰가 생산 의욕을 꺾을 때도 있다

회고 3. 어떻게 리뷰를 유지할 수 있었을까?

- 우리가 리뷰를 유지할 수 있었던 이유
 - 초기부터 모두의 동의 하에 자율적으로 도입
 - 코드 리뷰는 당연하다라는 문화의 정착
 - 문제의 인식과 지속적인 개선 노력
 - 정기적인 개발 미팅: 특히 좋았음
 - 매주 정해진 시간에 자유 주제로 논의
 - 이터레이션 테스트 종료 후 회고
 - 리뷰 정책 / 개선에 대한 논의할 수 있는 기회
 - 도구: 엔터프라이즈 깃헙
 - 모두 같은 언어로 같은 서비스를 담당
 - 지속적으로 한 서비스를 담당
 - 수평적 문화 (영어 이름)

회고 3. 어떻게 리뷰를 유지할 수 있었을까?

- 반면, 예전 좋지 않았던 리뷰 경험
 - 팀 내에서 코드 리뷰 진행
 - 자바스크립트 개발자가 모여있는 기능 조직
 - 각자 다른 프로젝트에 투입
 - 자율적으로 만들어진 분위기가 아니었음



코드 리뷰가 좋지 않을 수 있는 이유 및 환경

- 오프라인 코드 리뷰만 수행
 - 리뷰 미팅에서 프로젝트로 공유
 - 리뷰 미팅은 분위기 영향을 많이 받음
 - 과열된 논쟁이나 귀차니즘의 전파
- 도구의 부실함
 - SVN + 마땅한 리뷰 도구 없었음
 - 메모장이나 에디터에 주석으로 달아 전달
 - 별도 리뷰 도구를 도입했지만 잘 연동되지 않음
- 서로 다른 업무
 - 개발하는 언어는 같았지만
 - 담당하는 서비스가 모두 다름
 - 리뷰 범위의 한계 (스타일이나 일반적인 로직)

코드 리뷰가 좋지 않을 수 있는 이유 및 환경

- 주니어-시니어 간의 코드 리뷰
 - 시니어가 주니어의 코드를 고쳐주는 일방향 리뷰
 - 시니어들에겐 큰 도움이 되지 않음
- 자리잡지 못한 문화
 - 코드 리뷰가 잘 워킹하지 않는 걸 모두 알고 있음
 - '뭘 코드리뷰를 해~'라는 분위기
 - '팀별로 코드리뷰를 하라'는 상위 조직의 강제성

리뷰 도입을 위한 제언

- 새로 시작하는 조직에 리뷰를 도입한다면
 - 모든 멤버의 자율적 동의로 시작하고,
 - 최대한 강제성을 적용한 규칙으로 시작
 - 규칙은 도구를 사용해 제한
- 시작할 때는 이렇게 해보면 어떨까?
 - develop, master push 제한 / 모두 PR로
 - 모든 멤버의 동의
 - 코드 스타일 체크는 자동화 (깃훅 등)
- 기존 조직에 리뷰를 도입한다면
 - 나와 마음이 맞는 동료로 찾아 소규모로 시작
 - 도구를 적극적으로 활용
 - 기존에 일하던 방식에 자연스럽게 적용될 수 있게
 - 다른 멤버가 거부감을 갖지 않도록 천천히 도입

리뷰 도입을 위한 제언

- 근데 좀 해보려고 하면,
 - 다른 멤버는 시큰둥하다. 나만 하고 싶나...
 - 자꾸 하자고 하려니 귀찮고 미안하다...
 - 그냥도 이미 바쁜데 오바 아닌가...
- 코드리뷰는 문화
 - 기존의 습관을 단번에 바꾸기 어려움
 - 억지로 바꾸려고 하면 더 어려움
 - 여유와 시간을 갖고 천천히
 - 정답은 없음. 우리 팀에 맞는 방식으로.

리뷰 도입을 위한 제언

- 작은 경험의 반복으로 익숙해지도록
 - 내 코드를 먼저 리뷰하도록 시도
 - 처음엔 리뷰하기 쉽도록 PR은 가능한 작게
 - Pull Request - 리뷰 - 머지의 경험 반복
 - (선택) 리뷰어를 지정해서 부탁
- 리뷰에 어떻게 반응하면 좋을까?
 - 피드백! 피드백!
 - 반영 여부는 본인이 결정하는 것이 좋은 듯
 - 코드는 내가 아니고, 그저 내가 작성한 코드일 뿐

리뷰 도입을 위한 제언

- 어떻게 리뷰하면 좋을까?
 - 부드럽고 쟁탈하게
 - 궁금한 건 의도를 물어보는 식으로 접근
 - 이견이 있다면 구체적인 방법을 제시
 - 마음이 불편하더라도 적극적으로 리뷰
- 리뷰 문화를 잘 유지하려면
 - 적극적으로 리뷰하고 잘 피드백하자
 - 코드 스타일 리뷰는 말 대신 도구로 처리하자
 - 서비스와 코드, 리뷰에 대해 자주 이야기하고
 - 리뷰가 병목이 되지 않게 개선하자

(기타)

비자아적 프로그래밍(Egoless Programming)

- 당신이 실수하리라는 것을 받아들여라.
- 당신이 만든 코드는 당신이 아니다.
- 당신이 얼마나 많이 알고 있다고 해도, 항상 누군가는 더 많은 것을 알고 있다.
- 권위는 지위가 아니라 지식으로부터 나온다.
- 자신보다 많이 알지 못하는 사람이라 해도 존경과 인내로 대하라.
- 사람이 아니라 코드 그 자체를 비판하라.



코드 리뷰 3: 도구들



코드 리뷰 도구

- 상용
 - Crucible, Collaborator
 - Pre-review, IDE 연동 등 많은 편의 기능 제공
- 공개
 - Gerrit, Phabricator, Review board
 - 오픈 소스 도구. 필수적 기능 및 기타 부가 기능도 상당수 제공
- GitHub PR 연동 서비스
 - reviewable.io, Review Ninja

Gerrit 소개



Gerrit code review system

- Gerrit
 - Git 기반의 코드 리뷰 시스템
 - Git 기반의 코드 개발 과정에 쉽게 적용 가능
 - 구글 안드로이드 프로젝트에서 파생
 - AOSP (Android Open Source Project)
 - 자체 서비스를 사용하다가, 오픈 소스 프로젝트로 발전시킴
 - 네덜란드 개발자가 네덜란드의 디자이너이자 건축가인 Gerrit Rietveld의 이름을 따서 Rietveld 시스템을 만들었고, 이를 토대로 Gerrit 으로 발전함
 - 이후 JAVA로 새롭게 작성됨
 - 웹을 통해 서비스
 - 자체 서버를 구축해야 함

Gerrit code review system

- <https://www.gerritcodereview.com/>



Gerrit Code Review

[About](#) [Releases](#) [Documentation](#) [Issues](#) [Wiki](#) [Source](#) [Reviews](#) [Builds](#)

Code Review for Git

Gerrit provides web based code review and repository management for the [Git](#) version control system.



Discuss code

and boost your team's code fu by talking about specifics.

Serve Git

as an integrated experience within the larger code review flow.

Manage workflows

with deeply integrated and delegatable access controls.

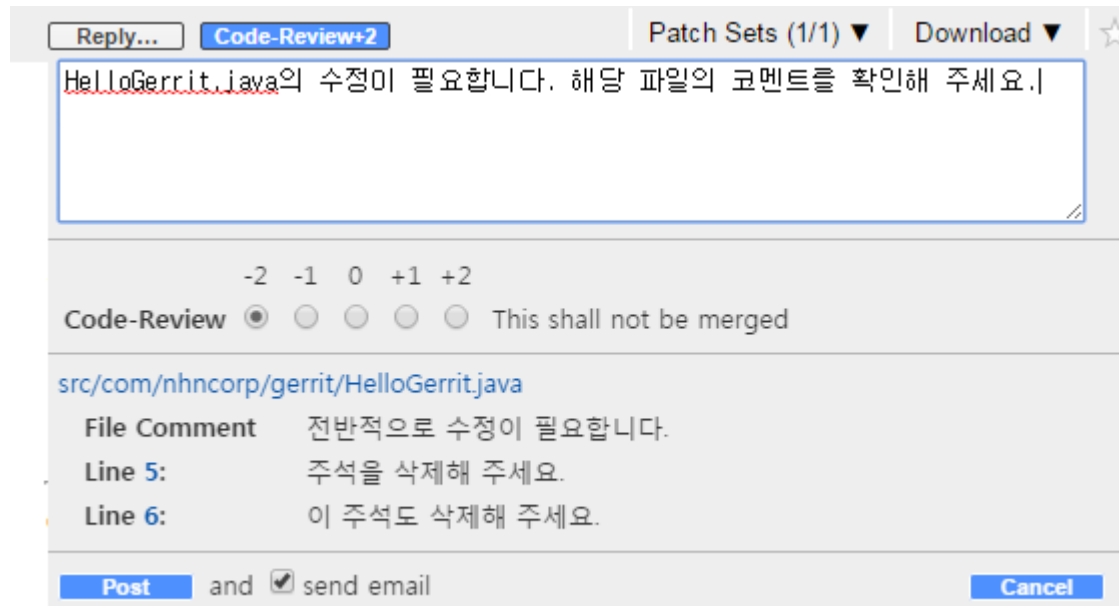
Download

[Gerrit 2.14](#)



Gerrit 특징

- 리뷰 통과 조건 설정
 - 다른 리뷰어로부터 일정 점수 이상을 얻어야 merge 가능
 - 점수와 의견을 통해 코드 품질 향상을 도움



The screenshot shows the Gerrit Code-Review interface. At the top, there are buttons for 'Reply...', 'Code-Review+2', 'Patch Sets (1/1)', 'Download', and a star icon. Below these is a text area containing the message: 'HelloGerrit.java의 수정이 필요합니다. 해당 파일의 코멘트를 확인해 주세요.' (HelloGerrit.java needs modification. Please check the comments for this file.). Below the text area is a row of buttons: '-2', '-1', '0', '+1', '+2'. Below these is a row of radio buttons: 'Code-Review' (selected), 'This shall not be merged'. Below this is a table with the following content:

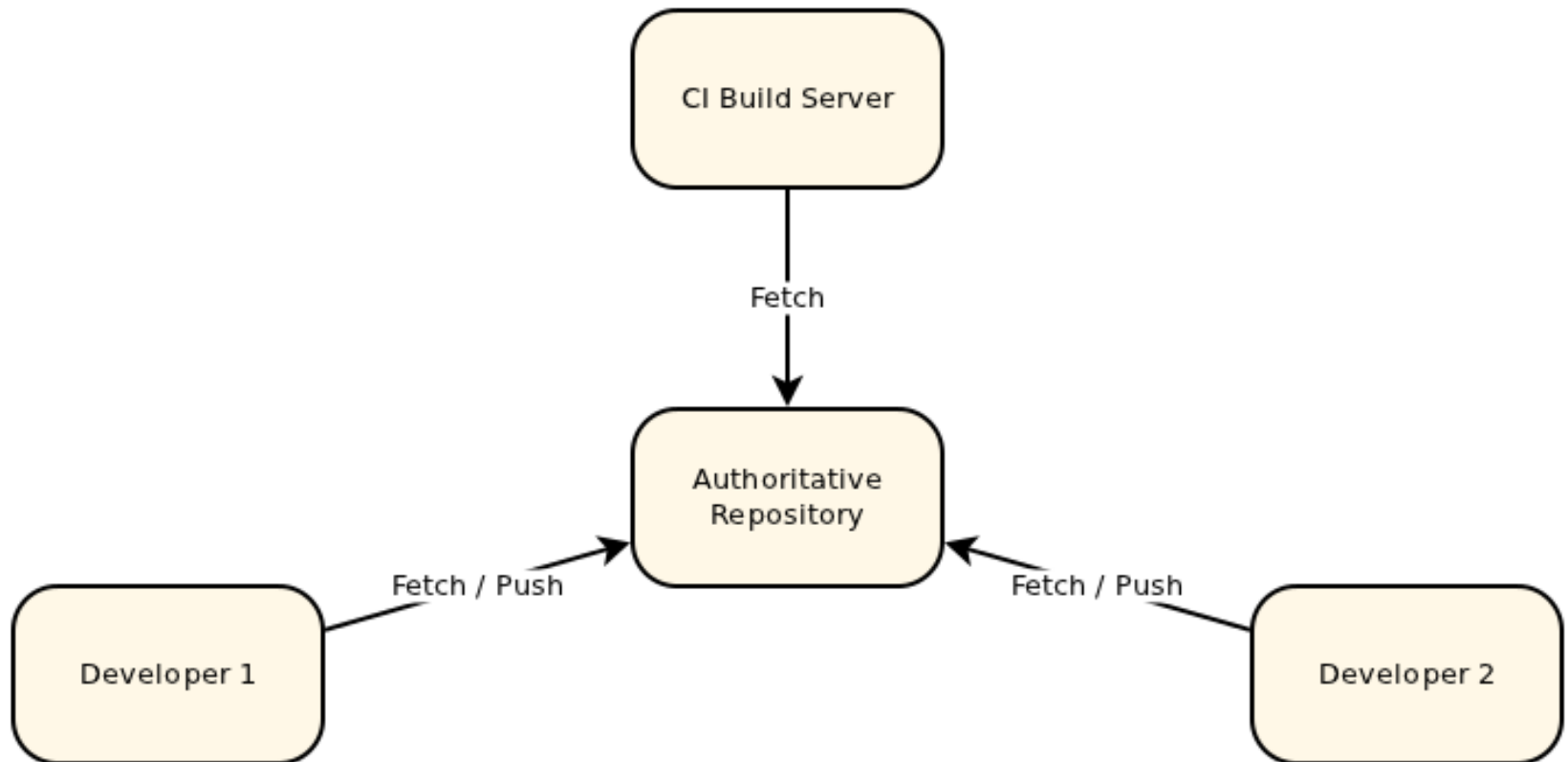
| File | Comment |
|---|------------------|
| src/com/nhncorp/gerrit/HelloGerrit.java | 전반적으로 수정이 필요합니다. |
| Line 5: | 주석을 삭제해 주세요. |
| Line 6: | 이 주석도 삭제해 주세요. |

At the bottom, there is a 'Post' button, a checkbox for 'send email' (checked), and a 'Cancel' button.

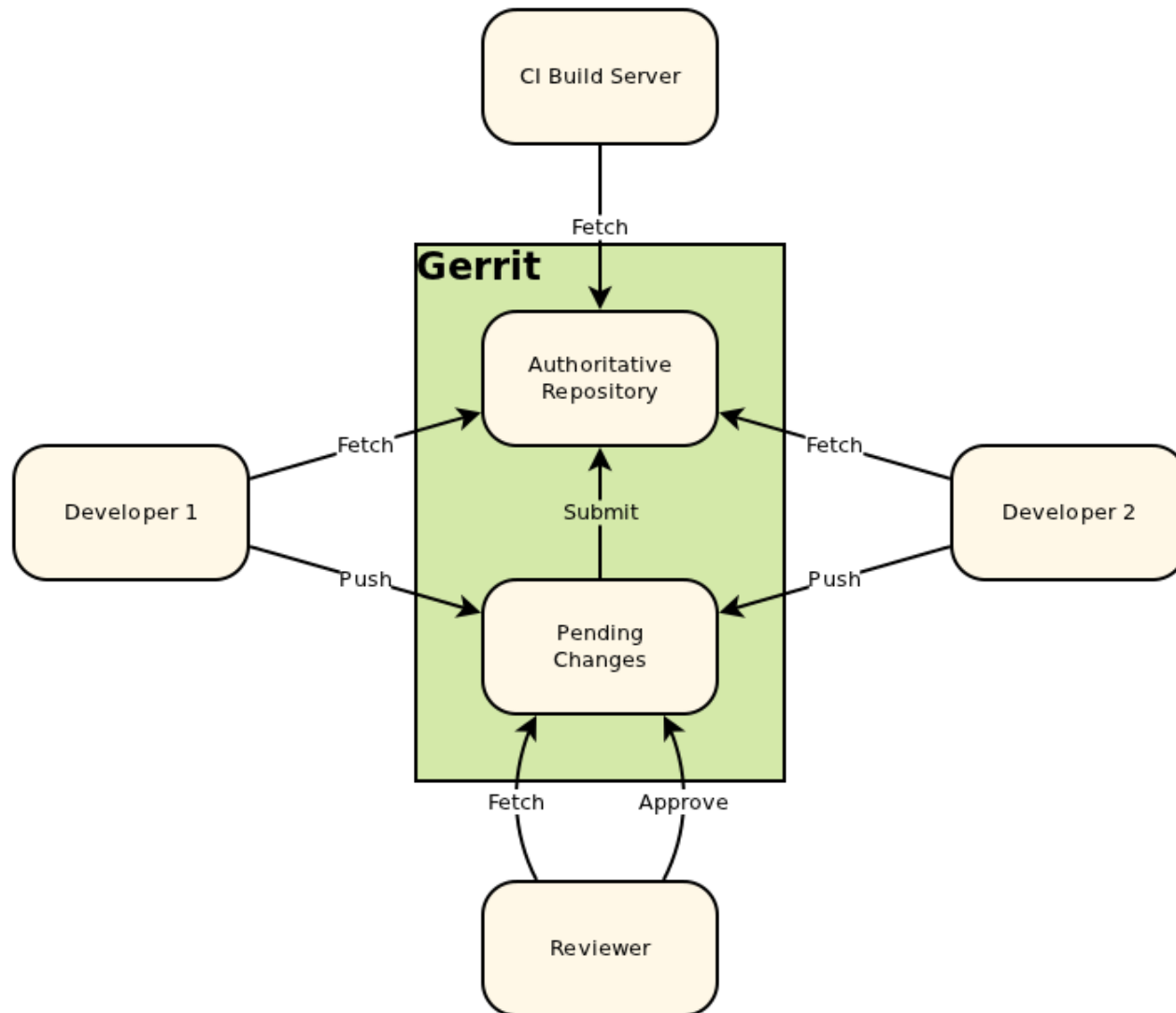
Gerrit 특징

- 사용자 권한 관리
 - 사용자 별로 접근 권한 관리
- 연동
 - Jenkins와 같은 외부 CI 도구와 쉽게 연동 가능
 - Eclipse 와 같은 IDE 와 연동 가능 (Mylyn 플러그인 사용)
- 다양한 인증 수단 지원
 - HTTP, LDAP, OpenID 와 같은 외부 인증 수단과 연계하여 사용 가능함

기존 Git repository 사용 방식



Gerrit 을 이용한 방법



Gerrit 을 사용한 개발 흐름

