



/ ANITA  
B.ORG

20<sup>th</sup>

GRACE HOPPER CELEBRATION  
V I R T U A L



/ANITA  
B.ORG

20<sup>th</sup>

# GRACE HOPPER CELEBRATION

V I R T U A L

Finding Your Voice  
as a Code Reviewer

# Finding Your Voice as a Code Reviewer



**ALICE BONHOMME-BIAIS**

@bonhommebiais



**KENDRA CURTIS**

# Outline

## Learn:

**what a code review is**

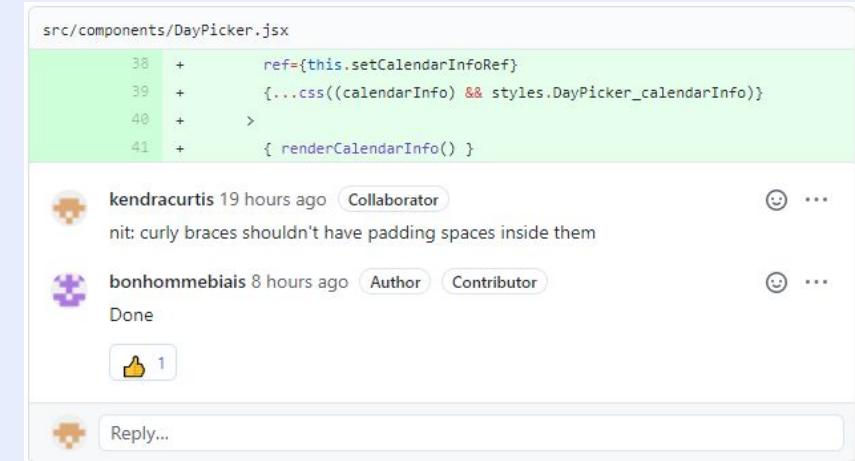
**how to be a good reviewer**

**benefits of a code review**

**rules of thumb** for a healthy review process

**how to adjust** your style based on the type of review (Feature, Bug, Other)

**contribute** to a review regardless of your experience level or the power dynamic



A screenshot of a GitHub pull request interface. The top part shows a diff of a file named 'src/components/DayPicker.jsx'. The diff shows four lines of code being added (lines 38-41). The bottom part shows a comment from a user named 'kendracurtis' 19 hours ago, which is a nit about curly braces. Another user, 'bonhommebiais', has responded with 'Done' and given a thumbs up. There is also a 'Reply...' button.

```
src/components/DayPicker.jsx
38 +     ref={this.setCalendarInfoRef}
39 +     {...css((calendarInfo) && styles.DayPicker_calendarInfo)}
40 +     >
41 +     { renderCalendarInfo() }
```

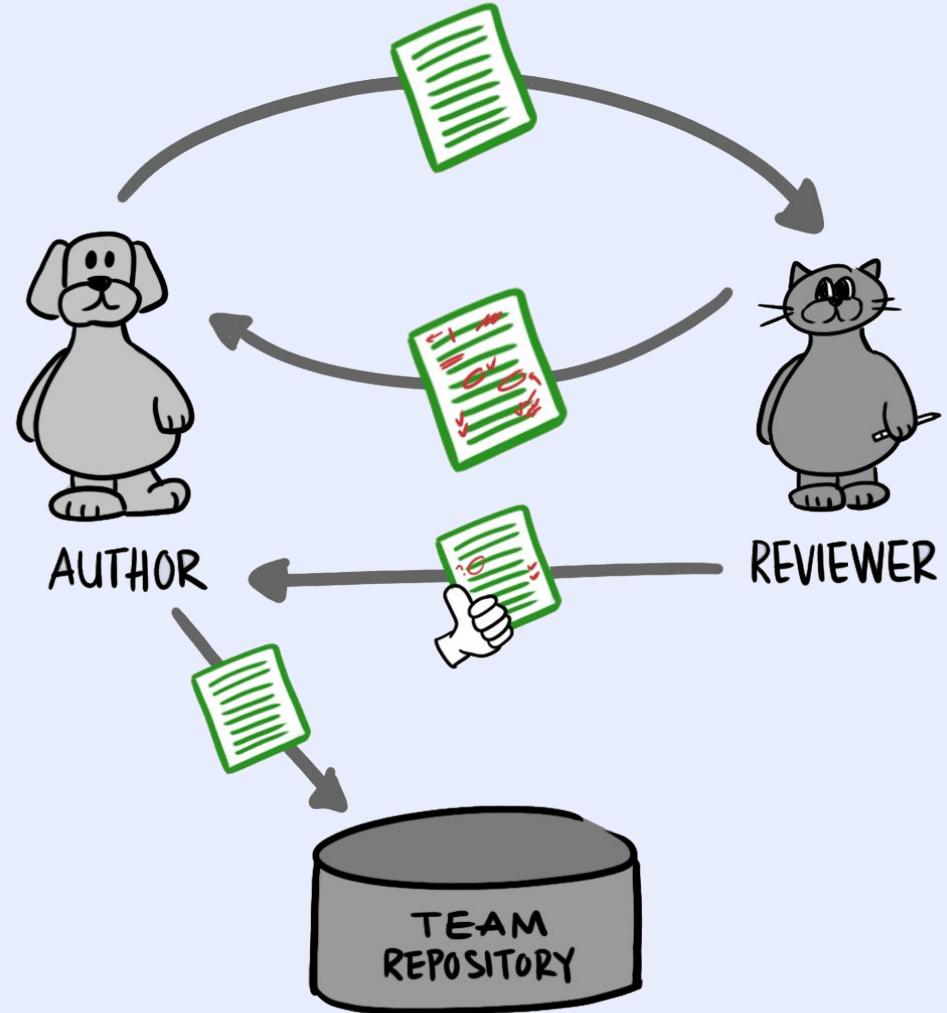
kendracurtis 19 hours ago Collaborator  
nit: curly braces shouldn't have padding spaces inside them

bonhommebiais 8 hours ago Author Contributor  
Done

Reply...

# What is a Code Review?

A code review is a practice/process where one or more people review the source code of another developer. The goal of the review can vary, but it is generally to increase the code quality and provide consistency to a code base.



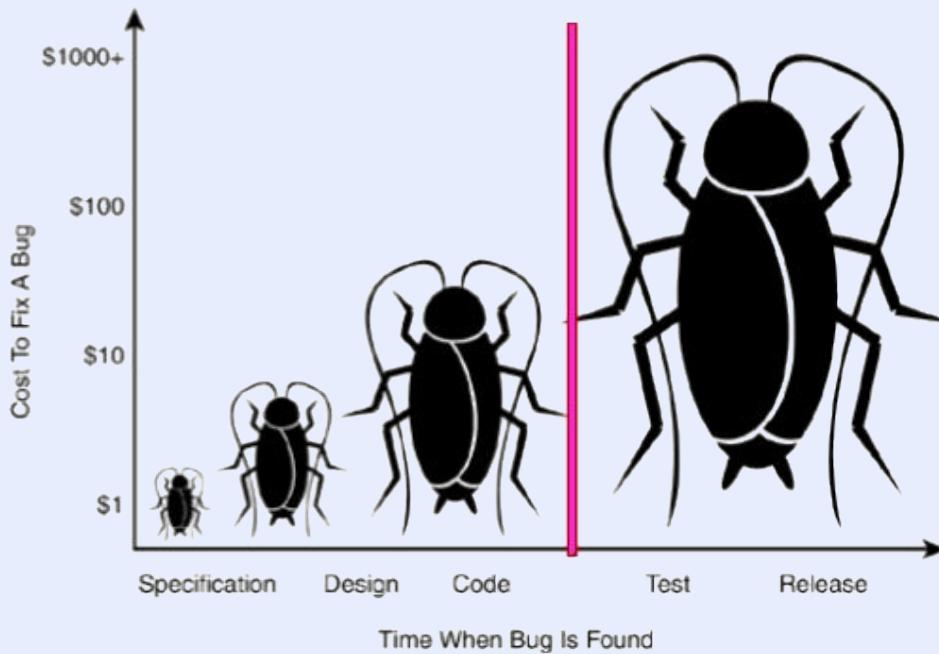
Source: <https://mtlynch.io/human-code-reviews-1/>

*implementation*  
*Finding* collective Learning/Knowledge  
generate base ideas maintaining  
gain solidarity transfer  
defects while quality found  
*innovative*  
*discussions* design fix  
*better* solutions maintainable Complying  
level software bug more members  
code learn guidelines coding standards  
understanding early cheaper team

*"peer code reviews are the single biggest thing you can do to improve your code"*

- Jeff Atwood

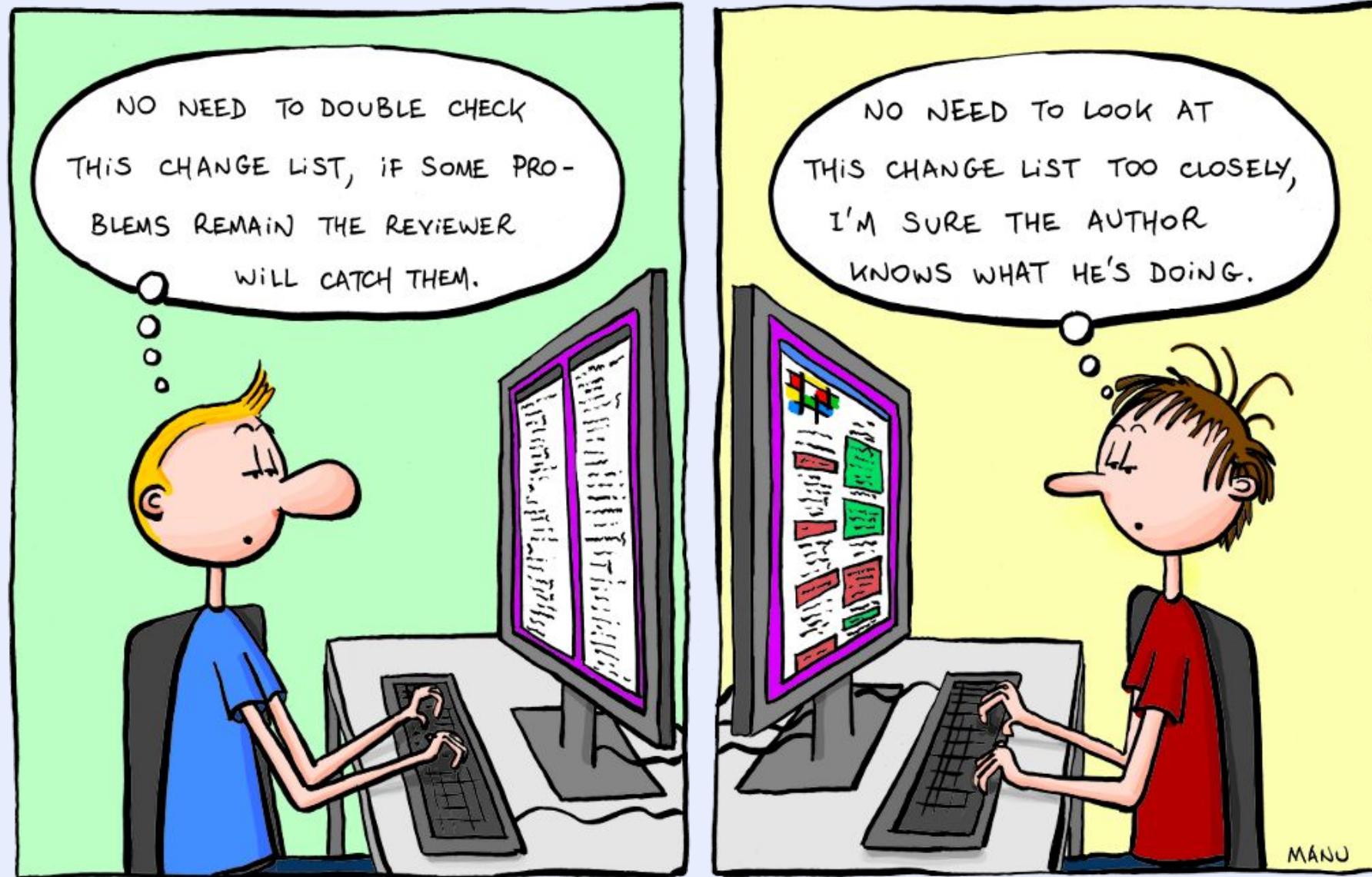
# Benefits of Code Reviews



- ▶ Better code quality
- ▶ Find defects early
- ▶ Learning/Knowledge transfer
- ▶ Increase sense of team ownership and solidarity
- ▶ Find better solutions

# AUTHOR

# REVIEWER



**CODE REVIEW FAILURE**

image by [Manu Cornet](#),

# Rules of Thumb

To help establish a healthy code review process, do some preliminary check **as soon as you receive the PR!**

**Rule #1 Let people know right away, if you are the wrong code reviewer.**

# Rules of Thumb

**Rule #2: The PR is not more than X lines of code**



# Rules of Thumb

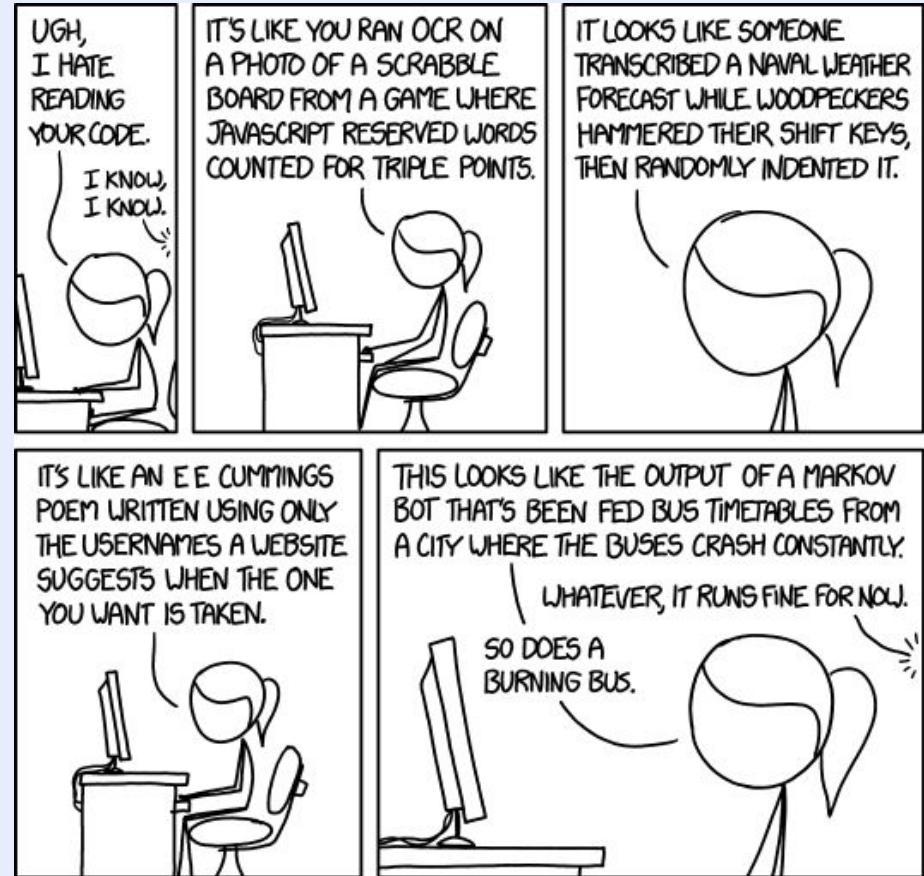
**Rule #3: The PR does not bundle more than one change.**

**Rule #4: The PR has a good description and/or context in the description**

=> First step of a code reviewer: if this breaks one of these rules, comment on PR right away.

# Types of Codes Reviews

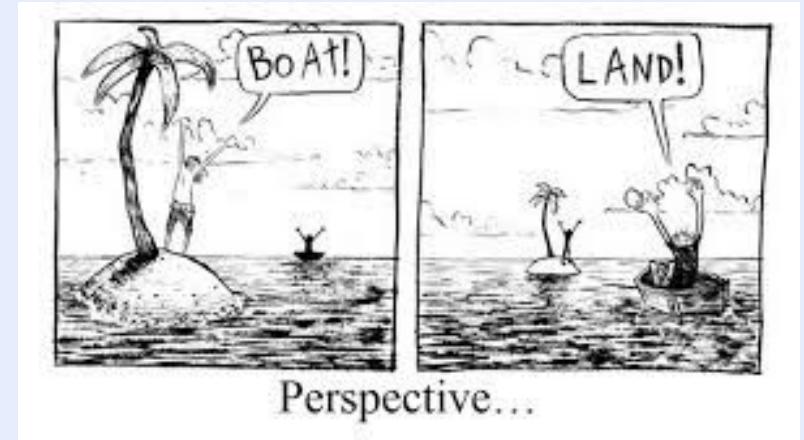
- ▶ Features/Components
- ▶ Bug Fixes
- ▶ Other (scripts, template, ...)



<https://xkcd.com/1695/>

# Reviewers Bring Fresh Perspective

- What are the edge cases?
- How could I break this code?
- Is there potential for concurrency problems?
- Is there prior knowledge or art that I know about this portion of the code base?
- Is this code “too complex”? Can I understand the code quickly or are there ways it can be simplified for the future.



Remember you can use a TODO or similar

# Feature

**Discuss and iterate on overall design before discussing details in the code.**

Review Questions:

- Do the interactions of various pieces of code make sense?
- Does this change belong in this codebase, or in a library?
- Does it integrate well with the rest of the system?

# Bugs

- Smaller and more focused.
- Usually on a timeline.
- Prior knowledge may be the most useful way you can contribute.
- Try to break the code, to ensure it really fixes the issue and does not introduce another bug.
- If needed, open another issue to track other changes rather than delaying a bug fix.

# Making your comments Actionable

A key to a successful code review is to keep it short and effective.

- Explain your reasoning
- Provide suggestion
- Ask specific question
- Specify importance of the comment

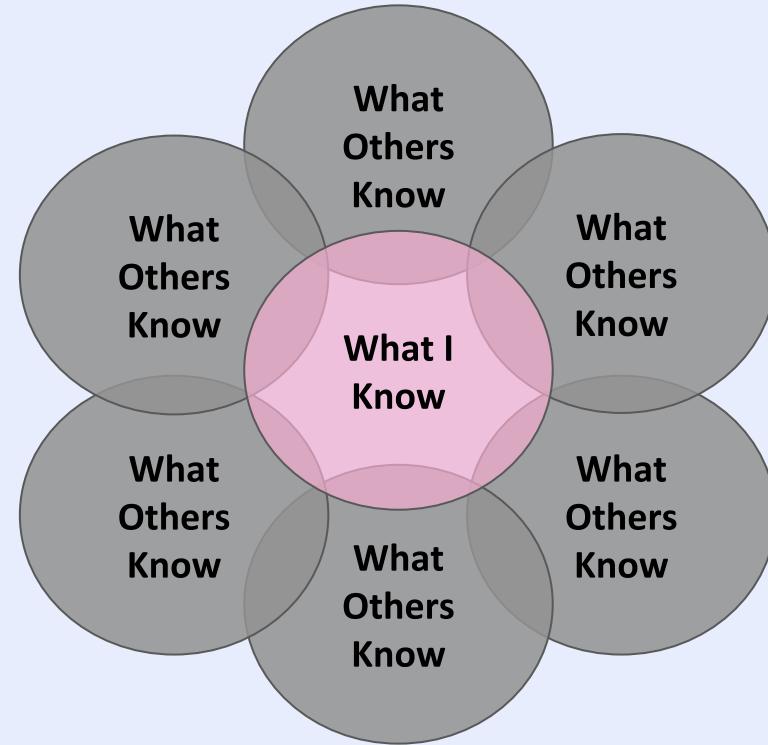
=> If too many back and forth on a PR , talk in person/video. Don't let it drag.

**UNDERSTAND  
YOU DO NOT?**

**ASK QUESTIONS  
YOU SHOULD.**

# Imposter Syndrome

""*Impostor syndrome (also known as impostor phenomenon, impostorism, fraud syndrome or the impostor experience) is a psychological pattern in which one doubts one's accomplishments and has a persistent internalized fear of being exposed as a "fraud".*"" - [Wikipedia](#)



# Healthy Communication

You do not want your code review comments to  
create these reactions



# Healthy Communication Tips

## #1 Avoid ownership of the code

- Avoid terms “mine, yours...”,
  - *“You forgot to add a method comment”*
  - *“This method is missing a comment”*
- Favor the use of “we”
  - *“you should use a constant”*
  - *“Here it seems we need a constant because ...”*



# Healthy Communication Tips

## #2 Asking questions instead of making demand

*"rename to user\_id" -> "what do you think of naming this user\_id?"*

## #3 Be humble and assume the best of author

*"Why didn't you use X here?", use "Have you considered using X ? because ..."*

## #4 a comment can Simply be a question asking for clarification

*"I am not sure to understand why this has an impact on latency, could you add more details?"*

# Healthy Communication Tips

## #5 Positive Reinforcement

AhnpGit commented 7 days ago

Contributor



@thymikee @rickhanlonii thank you I understand now :)



majapw reviewed 27 days ago

[View changes](#)

This is awesome! Thank you so much. I have two small comments and then I think we're good to go.

Thank you so much for debugging and putting in the work @mariepw! :)

# Takeaways

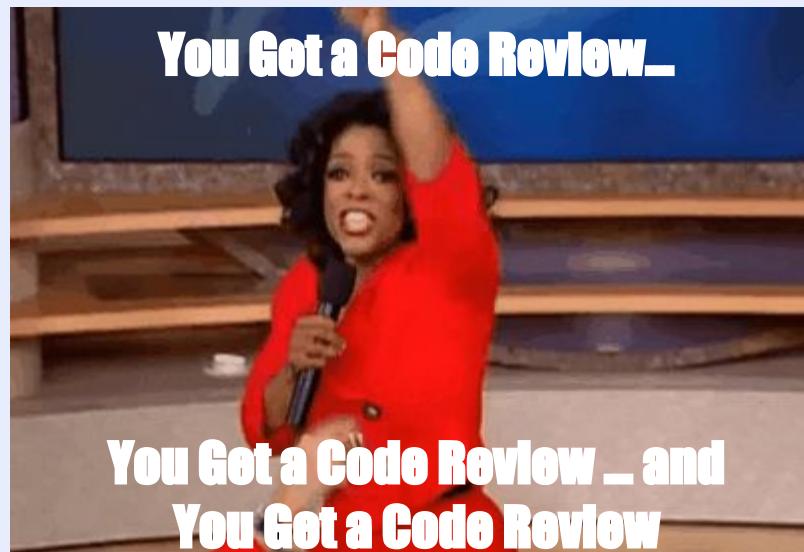
**Follow the rules of thumb** for a healthy review process

**Adjust your style** based on the type of review (Feature, Bug, Other)

Comments should be are **actionable** and follow **healthy** communication tips

Don't let **Imposter syndrome** prevent you from being a **Awesome Code Reviewer!!**

**You can always contribute** to a review



# Thank you!



@bonhommebiais



[www.linkedin.com/in/bonhommebiais](https://www.linkedin.com/in/bonhommebiais)  
[www.linkedin.com/in/curtiskendra/](https://www.linkedin.com/in/curtiskendra/)



[www.kendracurtis.com/ghc2020](http://www.kendracurtis.com/ghc2020)

# References

- [Code Reviews: Just Do It](#), Coding Horror Blog, Jeff Atwood, 2006
- [Expectations, outcomes, and challenges of modern code review](#), Bacchelli, A; Bird, C. Proceedings of the 35th IEEE/ACM International Conference On Software Engineering (ICSE 2013).
- [Modern code reviews in open-source projects: which problems do they fix?](#) Beller, M; Bacchelli, A; Zaidman, A; Juergens, E; Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014).
- [Google's Engineering Practices](#), Google, 2020
- [Thoughtbot Code Review Guidelines](#), Derek Prior, 2015
- [Wikipedia - Code Review](#), Wikipedia, 2020
- [Wikipedia - Imposter Syndrome](#), Wikipedia, 2020