Code Reviewing Like A Champion

Jonathan Maltz - @maltzj

Story Time

yelp®

I HAVE NO IDEA
WHAT I'M DOING

What Will We Talk About?

- Why Do Code Review
- Basic Code Review Outline
- How to Give Code Review Feedback

Why do Code Review?

Make Sure Code is Correct

@maltzj

Make Sure Code is Well-Designed

Share Knowledge

Basic Code Review Outline

1. Build Context

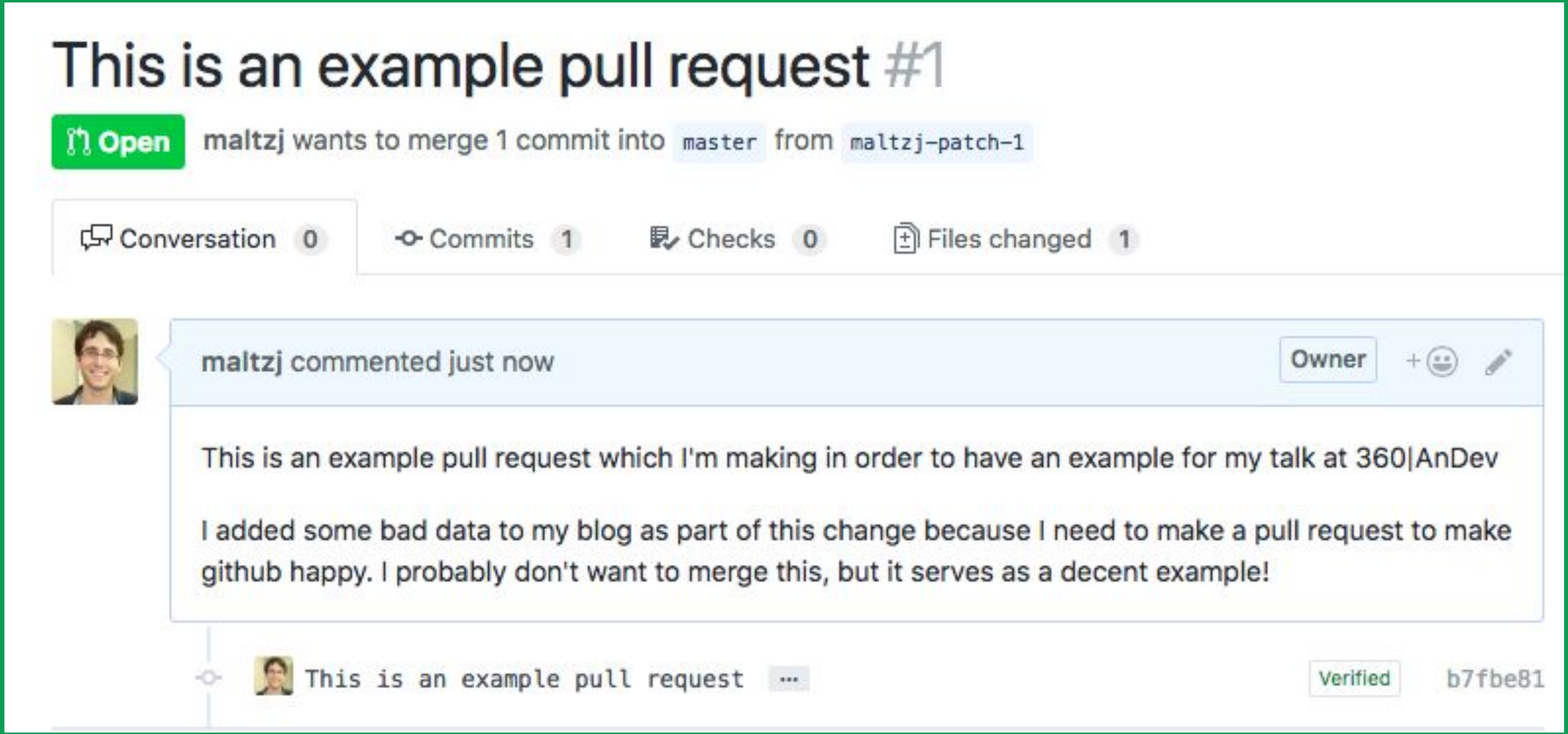
What’s Success?

- You understand the goal of the change
- You understand what is out of scope
- You understand any high-level decisions

What’s Failure?

- Jumping into a code review right away
- Not calling for backup as necessary
    - Especially important at boundaries

Authors are just as important here



This is an example pull request #1

maltzj wants to merge 1 commit into `master` from `maltzj-patch-1`

maltzj commented just now

This is an example pull request which I'm making in order to have an example for my talk at 360|AnDev

I added some bad data to my blog as part of this change because I need to make a pull request to make github happy. I probably don't want to merge this, but it serves as a decent example!

`This is an example pull request` … Verified b7fbe81

Protip: pre-filled templates

2. Evaluate Closer

What’s Success?

- High-level design feedback
- Bug-finding + edge case coverage
- Test coverage
- Naming
- Sharing new patterns

What’s Failure?

- Commenting on anything a tool can catch
  - PMD
  - Checkstyle
  - Findbugs
  - Pre-commit hooks
- Discussing the same things repeatedly

3. Take a Step Back

Could this be done easier/better/faster?

How to Give Feedback

What Do You Want To Encourage?

- Discussion and collaboration
- Authors owning their changes throughout the process
- People being excited about receiving feedback

FAIL

@maltzj

Feedback that makes people feel worse

How Can That Happen?

- “Your style here is inconsistent. Align your braces with our code style.”
- “This has a bug when running on Lollipop which will cause a crash. Fix it.”
- “Why did you do this refactor? The code was better in its previous form.”

@maltzj

“This code is good. Let’s make it even better.”

1. Start With Facts

“The style guidelines say you should put braces on the same line and this code puts braces on a new line. Align your braces with the styleguide.”

@maltzj

@maltzj

Opinion == Facts

Static methods cannot be mocked in normal JUnit tests.

✓ *Static methods cannot be mocked in normal JUnit tests.*

Static methods are bad. They cannot be mocked in normal JUnit tests.

Link to External Resources

2. Invite a Discussion

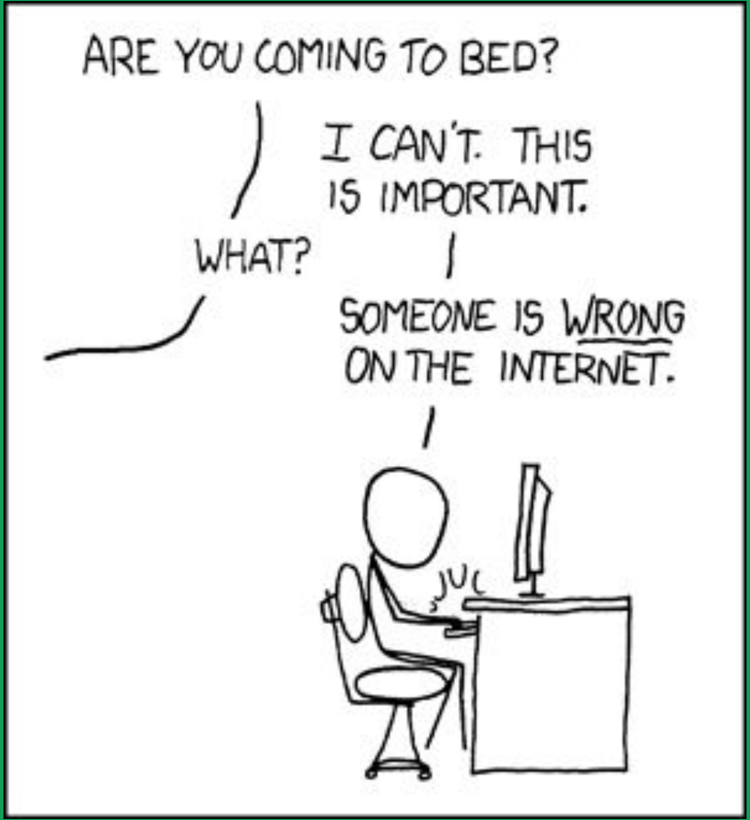
“The style guidelines say you should put braces on the same line and this code puts braces on a new line. Can you align your braces with the styleguide?”

@maltzj

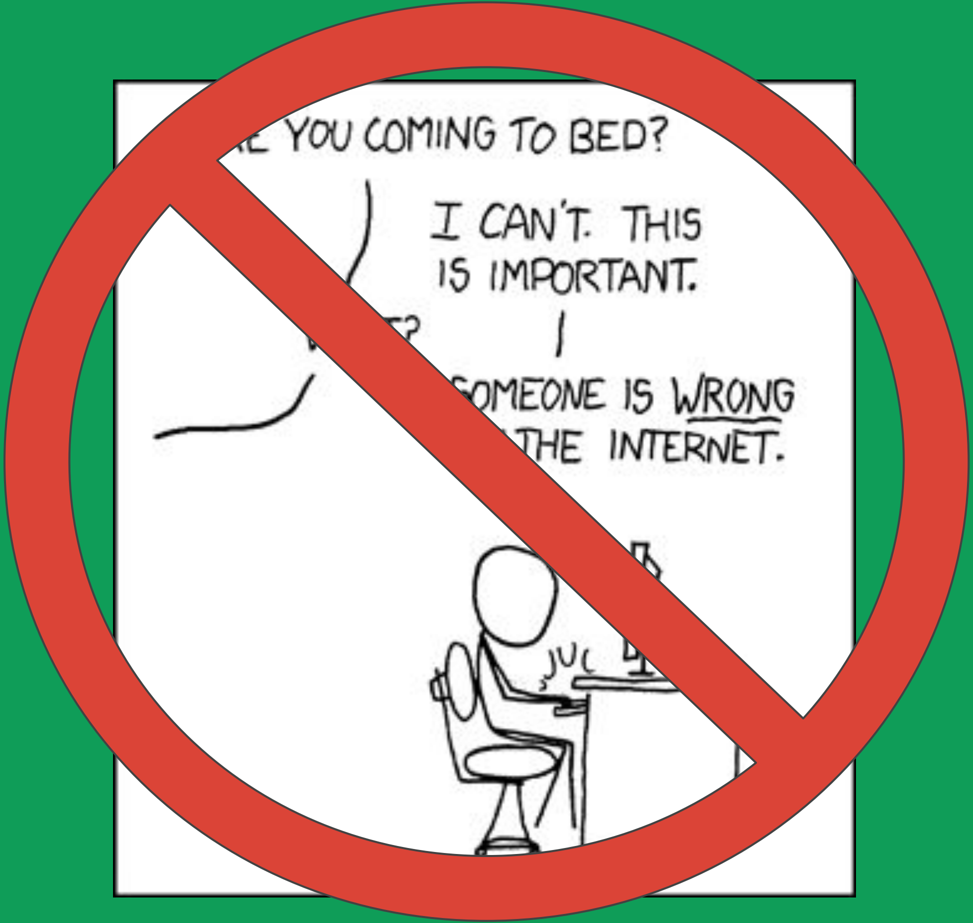
3. Replace “You” with “We” or “Us”

@maltzj

“The style guidelines say we should put braces on the same line and this code puts braces on a new line. Can we align the braces with our styleguide?”


ARE YOU COMING TO BED?
I CAN'T. THIS IS IMPORTANT.
WHAT?
SOMEONE IS WRONG ON THE INTERNET.

@maltzj


E YOU COMING TO BED?
I CAN'T. THIS IS IMPORTANT.
T?
SOMEONE IS WRONG
THE INTERNET.

Rule of Three

“On a scale of 1-10 I care about this at about a 5. What about you? What are your major concerns?”

What to do with this?

- **Different concerns?** Find a third way
- **Large Difference?** Principle of the person who cares the most in the code review
  - Protip: This shouldn’t always be you
- **Both high?** Maybe need a larger decision

Celebrate the Good Stuff

@maltzj

External Links

- Yelp’s Code Review Guidelines
- Writing Commit Messages
- How to Use Code Review To Execute Someone’s Soul
- Creating a Strong Code Review Culture
- Honesty Kindness, Inspiration: Pick Three
- Bettercode.reviews
- Giving and Getting Technical Help
- Crucial Conversations
- Pre-commit