

Code Reviews

"Beware of bugs in the above code;
I have only proved it correct, not tried it."
- Donald E. Knuth.

Dev.Talk
June 2016

Thomas Mentzel

Everyone should do code reviews.

Code Review Nightmares

- “One day implementing a feature and now three days and counting on the code review.” - Nemanja Trifunovic
- “...they dilute the engineers sense of ownership of his work” - Fran Porretto
- “Every code review I’ve been in has been a complete waste of valuable development time. It usually turned into ‘Why did you do it this way? My way is better...’ “ - Kevin Marois

The Benefits of Code Review

- You write better code when you know it will be reviewed
- A second (or third, or fourth...) set of eyes will help spot defects
- More than one person understands your code
- It's a great way to learn more about the codebase
- It will make you a better developer

What to look (out) for

- Bad Design
- Lack of clarity
- Lack of conformity (→ Clean Code)
- Performance hazards

→ Create your weekly/monthly checklist

What's *not* important?

- Optimization
- Skill and experience gaps
- Personal style

How to sell code reviews to others

- Not as hard as selling unit tests
- Bottom-up approach
- Just do it

Code reviews are for everyone

- Small, medium and large companies
- Open and closed source
- Pros or amateurs
- Local or remote
- Solo projects are no exception

Summary

- Just do it
- Weekly/Monthly checklist
- Focus on small improvements

References

- <https://help.github.com/articles/using-pull-requests/>
- <https://github.com/GitTools/GitVersion/pull/661>
- <https://github.com/Humanizr/Humanizer/pull/386>
- <https://github.com/jgrassler/mkdocs-pandoc/issues/5>

Thank You

Time for Questions!