



AO 
the open web company



Code Reviews

T3Rookies Summer Camp 2016





Code reviews are opportunities to spot things that are dangerous, bad, mistaken or wrong in the code, and to offer quality improvement suggestions.



Code Reviews: Why

- Put a second set of eyes on a particular bit of code
- Think about consequences throughout the system
- Force a developer to explain what his/her code does
- Knowledge sharing
 - Learn new tricks
 - Correct bad habits



Code Reviews: How

- Agree on what kind of commits require a review
- Agree on review criteria
- Agree on when to review (pre-/post-commit)
- Take your time: faster is not better
- Eliminate as many defects as possible, regardless who “caused” the error
- Do not try to substitute the logic under review with your own implementation
- Do not abuse code reviews for educational purposes



Best Practices

- Code reviews are highly subjective, so don't take it personal
- Smaller commits lead to more manageable code reviews
- Outline objectives to understand the what and why
- Agree on comment tiers
 - “*You might...*” – Suggestion
 - “*You should...*” – Disagreement
 - “*You must...*” – Defect (blocking)
- Don't push reviews

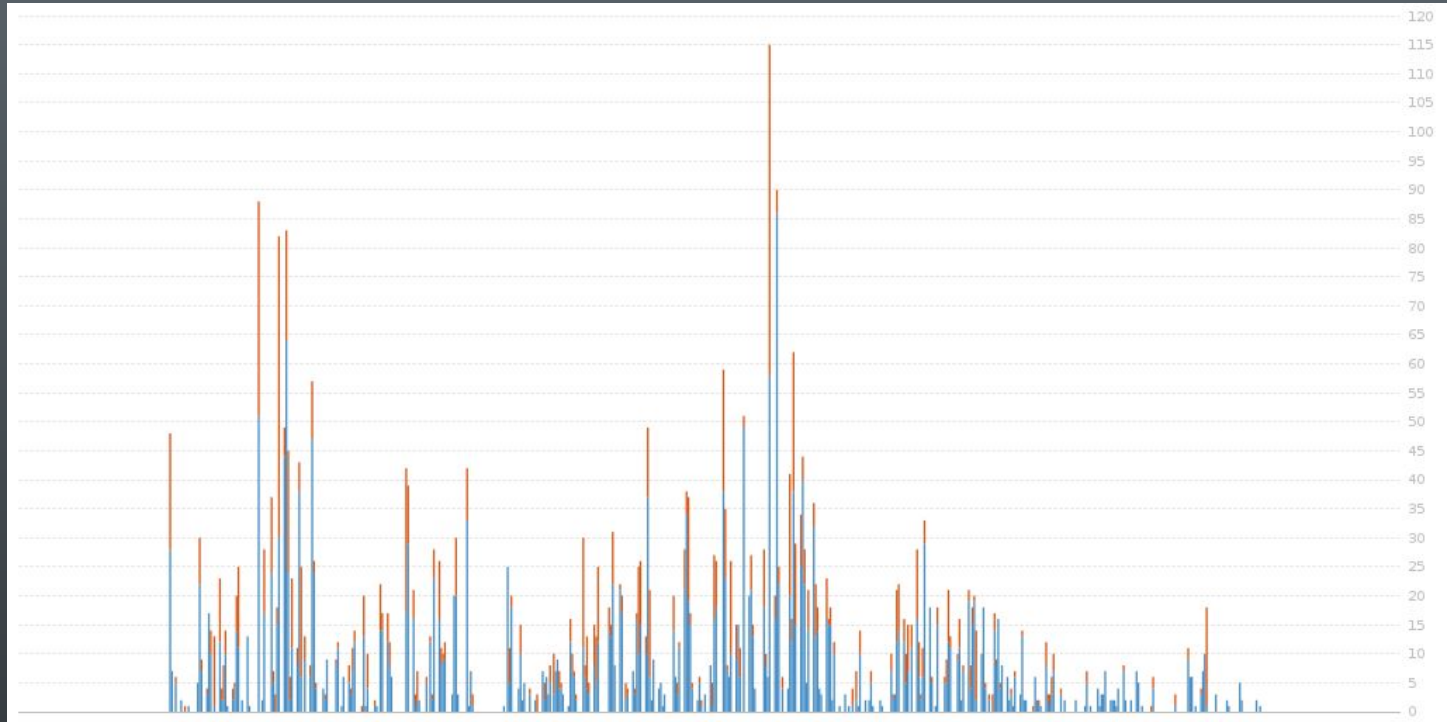


Example Code Review Checklist

- ❑ Code Style
- ❑ Code Smells
- ❑ DRY – Don't repeat yourself
- ❑ SRP – Single Responsibility Principle
- ❑ KISS – Keep it simple, stupid
- ❑ YAGNI – You aren't gonna need it



Project Sony: comment volume over time



Project Sony: open reviews over time

