



**AOE**

the open web company

# Agile Programming Practices

Tales from One Thousand and One Code Reviews

a little more than

# About me

I'm a

- software developing
- photography loving
- pipe organ playing
- dad of 2 wonderful kids

working at AOE  
in Wiesbaden (Germany)



# Agile Programming Practices

???



# Extreme Programming

Fine-Scale Feedback	Continuous Process
<ul style="list-style-type: none"><li>• Pair programming</li><li>• Test-driven development</li><li>• Planning game</li><li>• Whole team</li></ul>	<ul style="list-style-type: none"><li>• Continuous integration</li><li>• Refactoring</li><li>• Small releases</li></ul> <p>Pair Programming</p>
Shared Understanding	Programmer Welfare
<ul style="list-style-type: none"><li>• Coding standards</li><li>• Collective code ownership</li><li>• Simple design</li><li>• System metaphor</li></ul>	<ul style="list-style-type: none"><li>• Sustainable pace</li></ul> <p>Code Reviews</p>



# Facts

or Fiction?





Pair programming means  
double cost per developer





Pair programming is impossible  
for remote teams





Code Reviews require a special tool





Code Reviews and Pair Programming  
require big teams in large companies



# Pair Programming



# Pair Programming: Why

- Increased code quality
  - Significant reduction of defects
  - Higher confidence in the code
- Easier team-building and communication
  - Constant sharing of knowledge
  - Better transfer of skills
- Improved resiliency of a pair to interruptions



# Pair Programming: Economics

Potential Cost	Expected Benefits
<b>15%</b> Overhead for pairing	<b>15%</b> Fewer defects



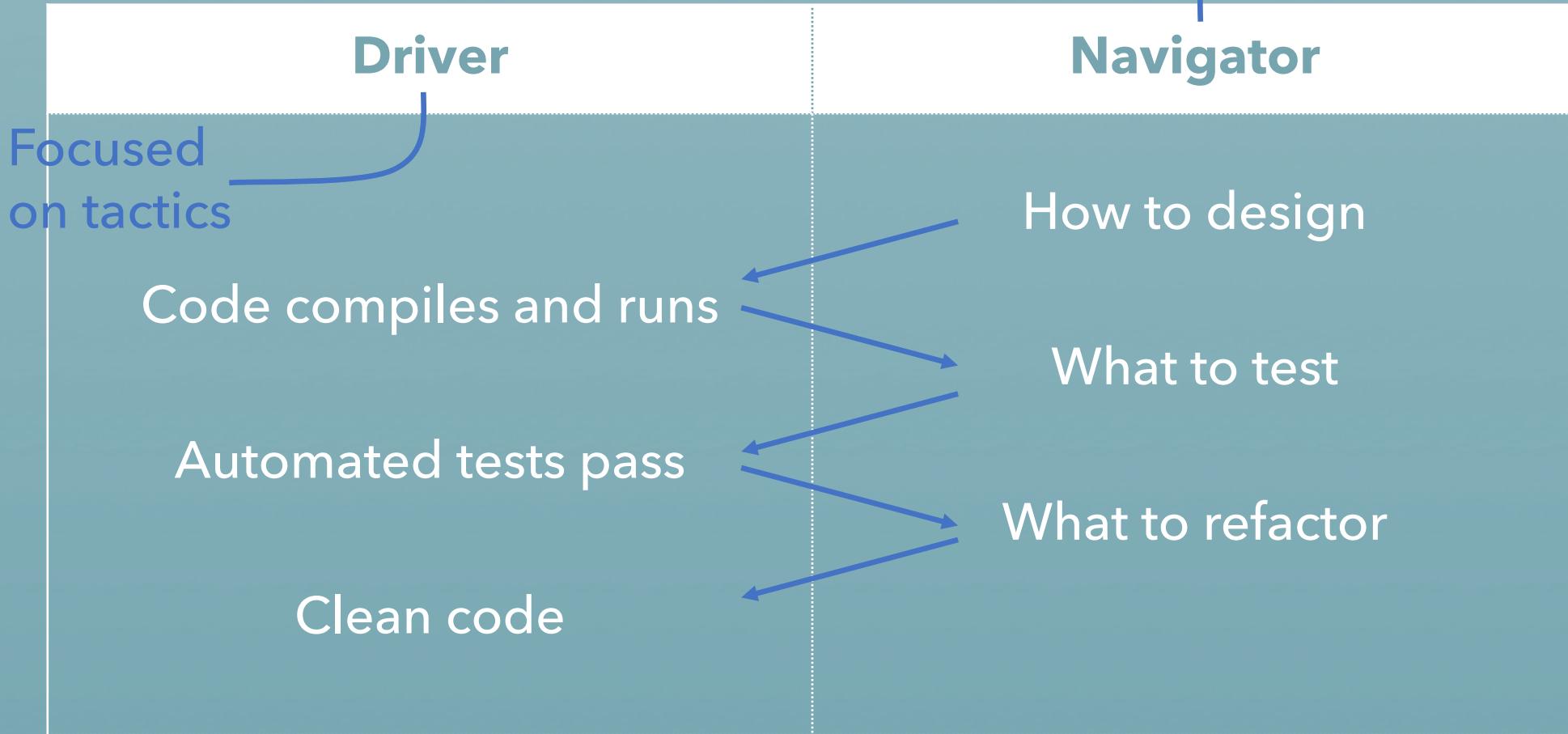


Pair programming means  
double cost per developer

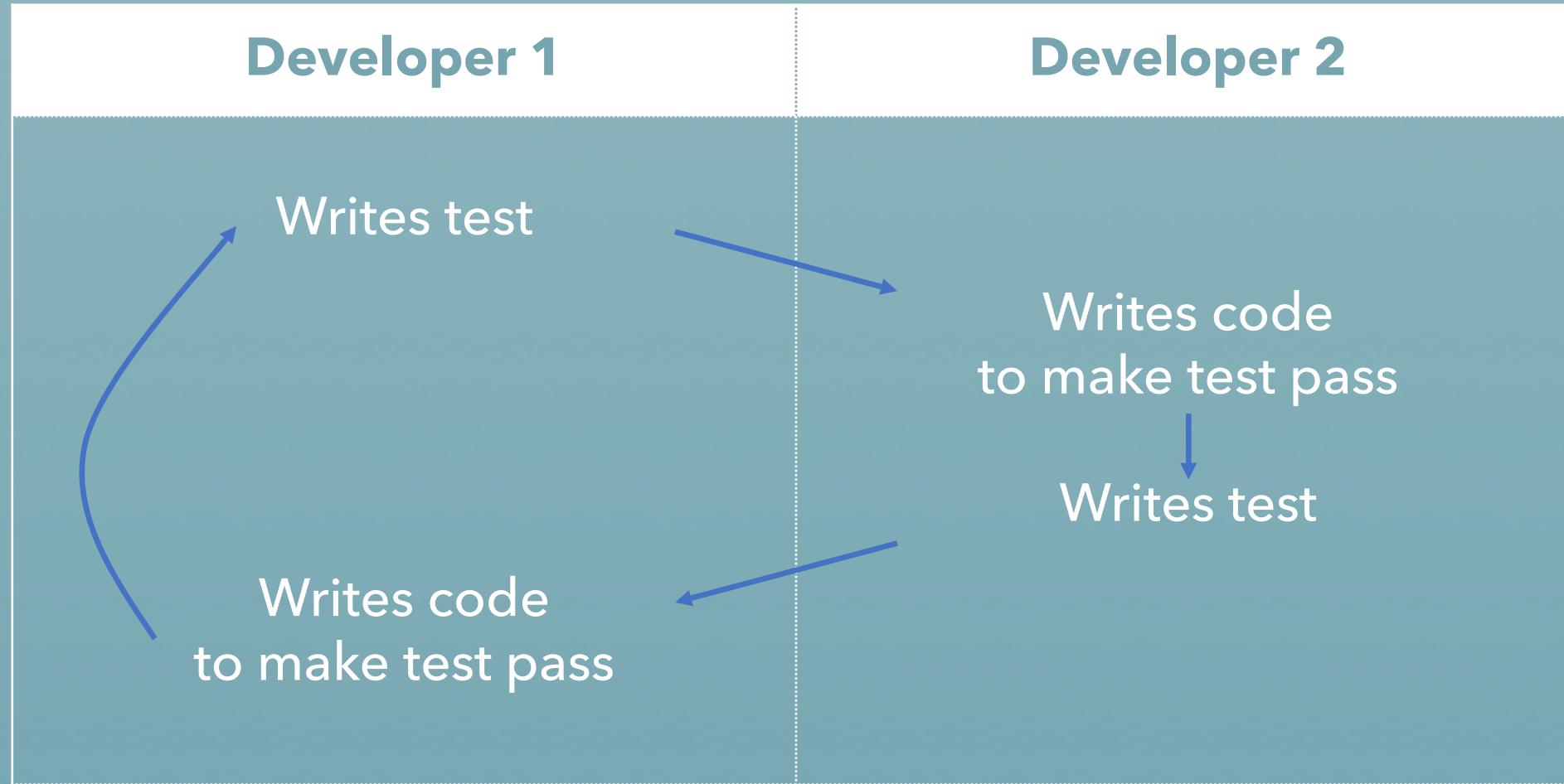
DOES NOT



# Patterns: Driver/Navigator



# Patterns: Ping/Pong

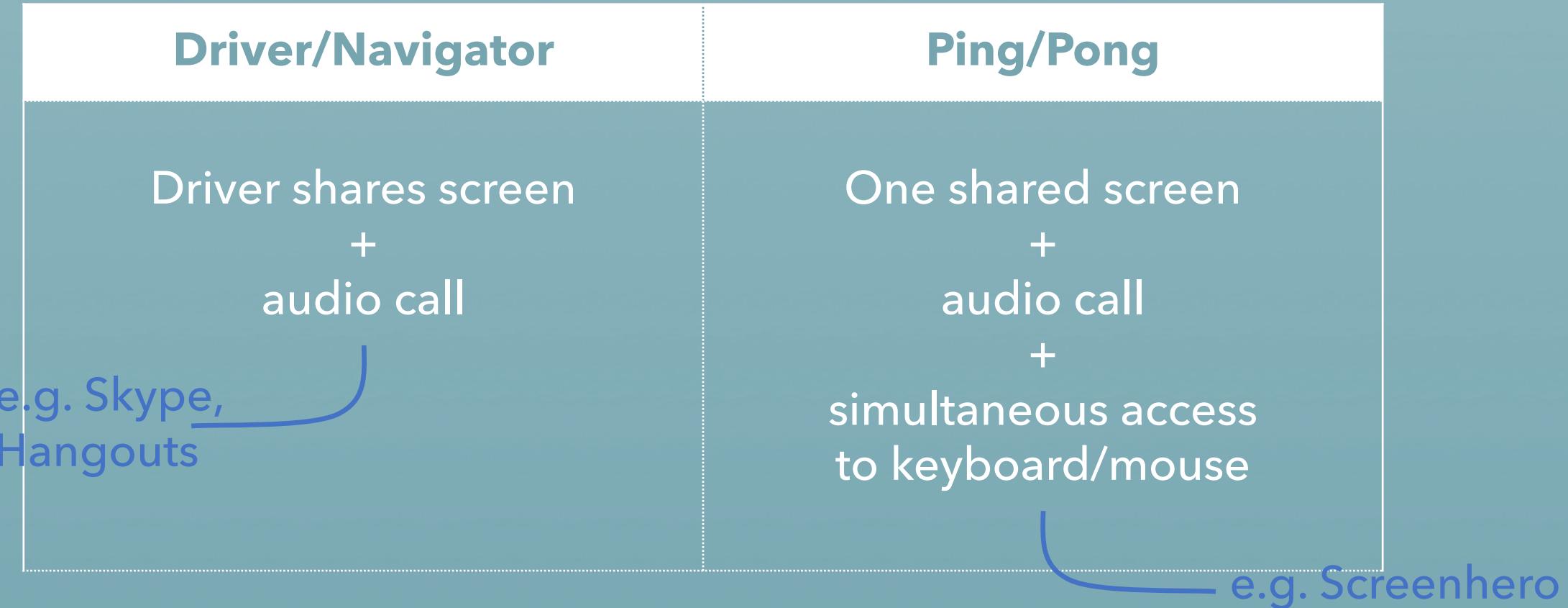


# Pair Programming: Best Practices

- ✓ Switch roles frequently (use a timer)
- ✓ Avoid the “watch the master” phenomenon
- ✓ Be actively engaged
- ✓ Keep talking
- ✓ Create a pairing friendly environment



# Remote Pair Programming

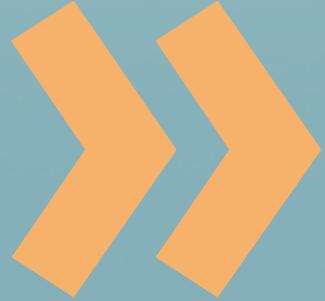




Pair programming is impossible  
for remote teams

NOT





The adjustment period from solo programming to collaborative programming was like eating a hot pepper. The first time you try it, you may not like it because you are not used to it. However the more you eat it, the more you like it.

*– Anonymous*



# Code Reviews



# Code Reviews: Why

- Put a second set of eyes on a particular bit of code
  - Find risky or incorrect code
  - Offer quality improvement suggestions
- Force a developer to explain what his/her code does
- Think about consequences of a change
- Knowledge sharing
  - Learn new tricks
  - Correct bad habits



# Code Reviews: When

Community Projects

Mature Teams

## Pre Commit

- + No unreviewed code will be merged
- + Code changes can be squashed
- △ Review blocking for moving on
- △ Risk of “pushing” code reviews

## Post Commit

- + Asynchronous
- △ Potentially incorrect code goes into the project
- △ Additional commits to fix findings
- △ Risk of ignoring the review outcome



# Code Reviews: Tools

Commit  
Diff

e.g. GitHub  
Pull Requests

Built in  
Tools

Complexity

e.g. Gerrit,  
Crucible

External  
Tools





Code Reviews require a special tool

DO NOT



# Code Reviews: How

- Agree on
  - what kind of commits require a review
  - review criteria
  - how many reviewers are required
- Eliminate as many defects as possible, regardless who “caused” the error
- Not all suggested changes have to be incorporated in the code



# Code Reviews: Dos

- Smaller commits => more manageable code reviews
- Outline objectives to understand the what and why
- Use comment tiers, e.g.
  - Suggestion: “*You might...*”
  - Disagreement: “*You should...*”
  - Defect (blocking): “*You must...*”
- Take your time: faster is not better



# Code Reviews: Don'ts

- Don't take it personal
  - Code reviews are highly subjective
- Don't push reviews
  - Address open reviews e.g. in the team's daily stand up
- Don't use metrics to single out developers
  - Number of defects is related to complexity
- Don't try to substitute code with the reviewer's logic

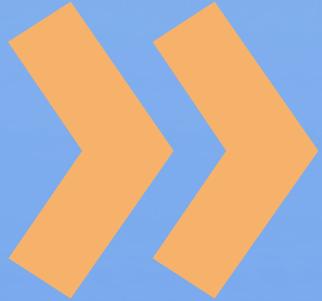


# Code Reviews: Checklist

- Obvious errors, risks, incompatibilities
- Compliance with team/community code style
- Following best practices of team/community
- Test coverage — CI
- Code smells
- DRY - Don't repeat yourself
- SRP - Single Responsibility Principle
- KISS - Keep it simple, stupid
- YAGNI - You aren't gonna need it

Code Sniffer





Developers should pair program often with other developers, learning their coding habits and styles. This provides context to comments received on proposed changes.

*– Brandon Savage*



# Agile Programming Practices

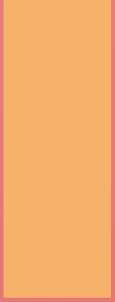
in an Open Source community



# Agile Programming Practices + Open Source

- Code Reviews
  - Gerrit for contributions to the TYPO3 Core
  - Pull Requests for TYPO3 Extensions
- Pair Programming
  - Code Sprints as an opportunity to try new patterns
- External Tools
  - GitHub ❤️ Travis CI





Code Reviews and Pair Programming  
require big teams in large companies

DO NOT



rather big

# Numbers and Figures



17.045 Reviews

∅ 100 per week

in ~3 years



too many

**665** Files

---

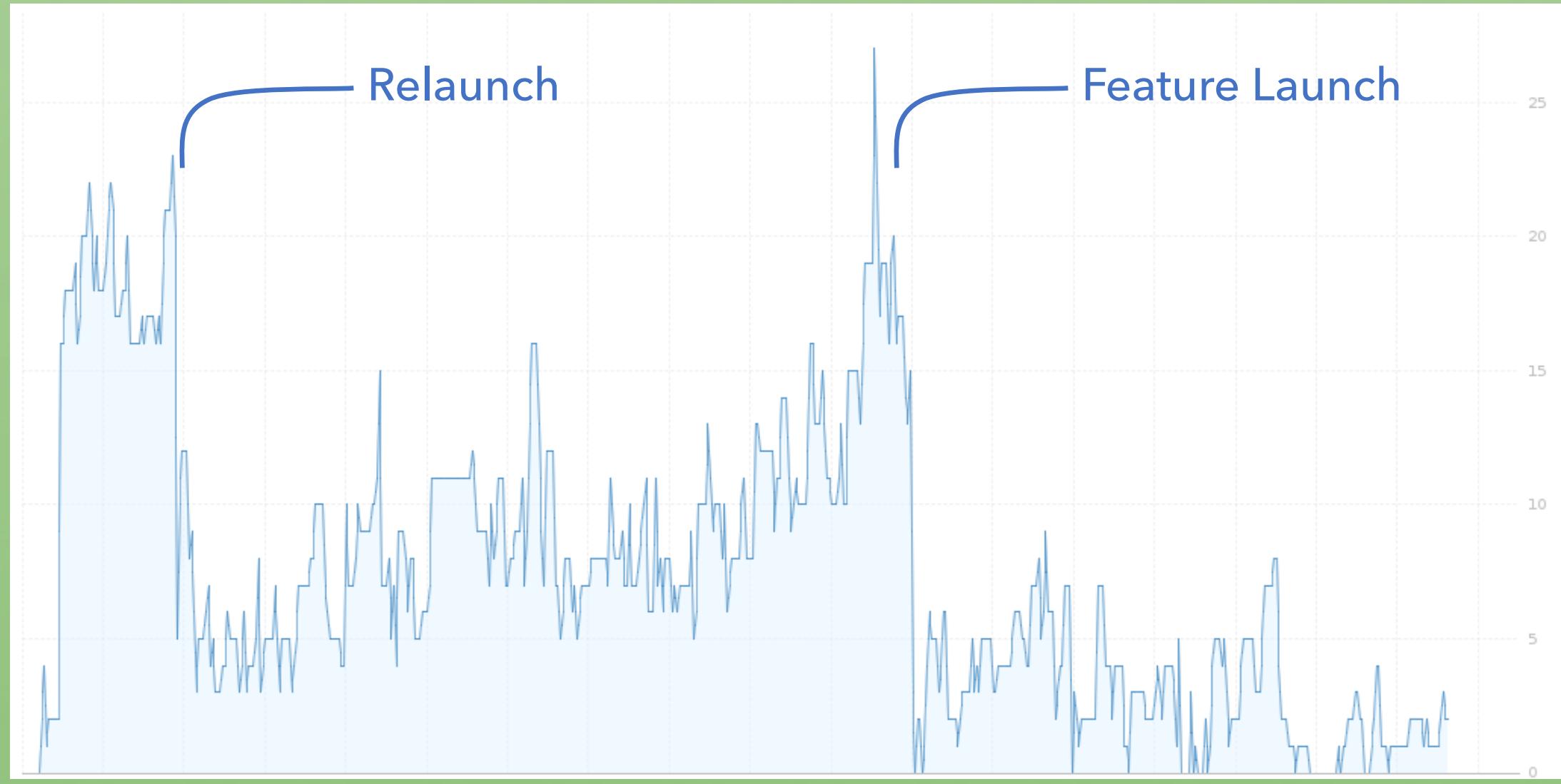


# 144 Comments

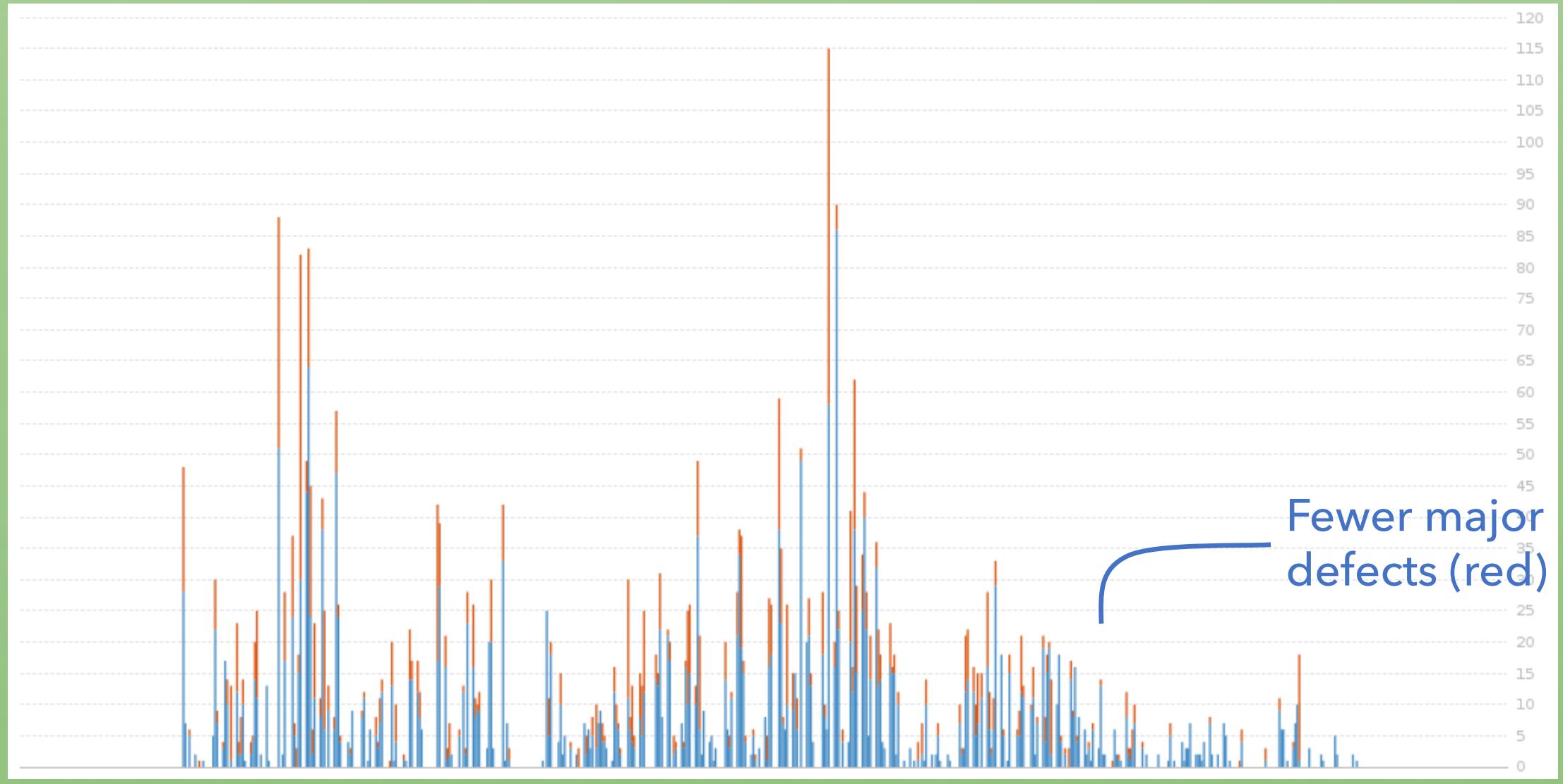
that should have  
been addressed  
otherwise



# Project Perspective: Open Review Count



# Project Perspective: Comment volume



# Conclusion



# Agree on guidelines and standards



# Keep commits and code reviews small



Keep calm: Git happens :-)





Any fool can write code that a computer  
can understand.  
Good programmers write code that humans  
can understand.  
– *Martin Fowler*





# Resources

- 11 proven practices for more effective, efficient peer code review

<http://www.ibm.com/developerworks/rational/library/11-proven-practices-for-peer-review/>

- 10 ways to be a faster code reviewer

<http://blog.codacy.com/top-10-faster-code-reviews/>

- The Pitfalls of Code Review (And How To Fix Them)

<http://www.brandonsavage.net/the-pitfalls-of-code-review-and-how-to-fix-them/>

- Screenhero

<https://screenhero.com/>

