# Code Reviews: The "Secret" to Building Quality Software

Patrick Schwisow php[tek] 2013 May 15, 2013

#### What is a Code Review?

"Code review is systematic examination...of computer source code.

It is intended to find and fix mistakes overlooked in the initial development phase,

improving both the overall quality of software and the developers' skills."

(http://en.wikipedia.org/w/index.php?title=Code\_review&oldid=530204872)

#### Who does Code Reviews?

- Peers (other developers)
- Managers / Team Leads / Senior Developers
- Dedicated QA / Testing Personnel
- All of the above

## What kind of CRs are you talking about?

- Code review by a fellow team member
- Required part of development process
- Typically a step in an Agile workflow (Scrum, Kan-ban, etc.) but can be applied to other methodologies
- Reviews should happen as soon as possible after the task is complete
- Reviews are considered part of the task and should be included in all time / workload estimates
- Other types of code review (by management or QA) may also be used

# The Flip-Side: Functional Testing

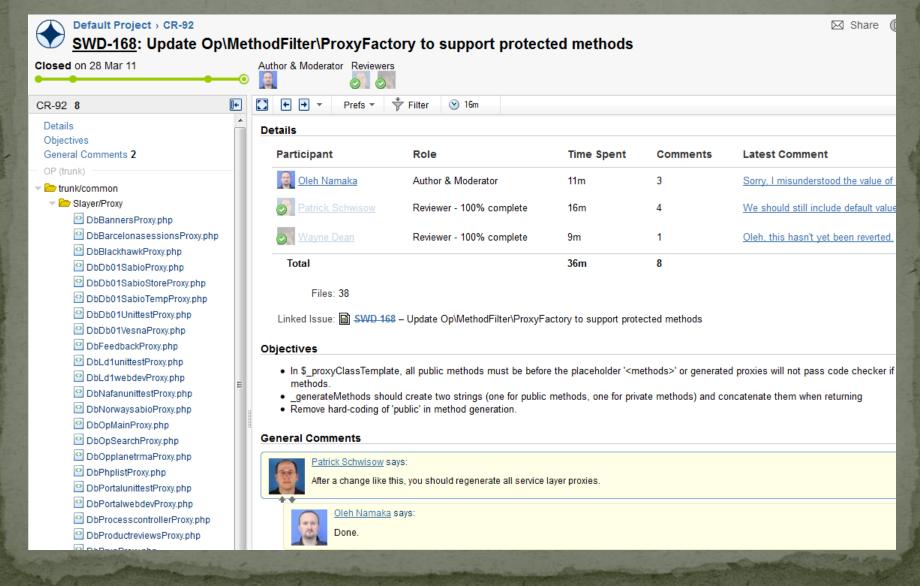
- "Functional testing" == "checking that all the stuff works"
- Test the updated code to make sure it does what is specified
- Check all the edge cases (Try to break stuff!)
- Run all the unit tests\*
- Test things that may be affected by this change
- Functional testing may happen together with CR or as a separate workflow step

<sup>\*</sup>Unless you've got them running automatically for you

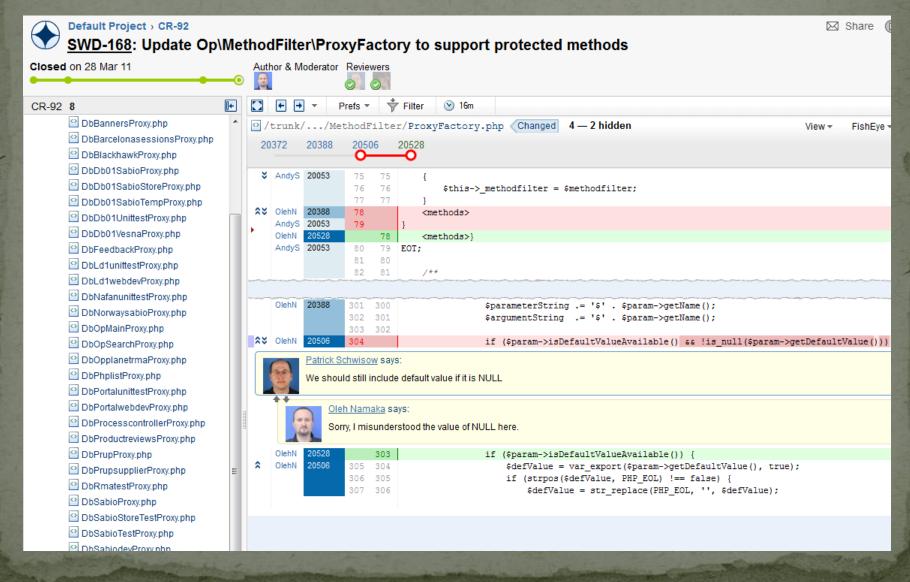
### Code Review Tools

- Prerequisites:
  - Issue Tracking (JIRA, Trac, GitHub Issues)
  - Version Control System (git, SVN, etc.)
- Atlassian Crucible
   (http://www.atlassian.com/software/crucible/overview)
- Atlassian Stash (<a href="http://www.atlassian.com/software/stash/overview">http://www.atlassian.com/software/stash/overview</a>)
- GitHub Pull Requests
- Review Board (<a href="http://www.reviewboard.org/">http://www.reviewboard.org/</a>)

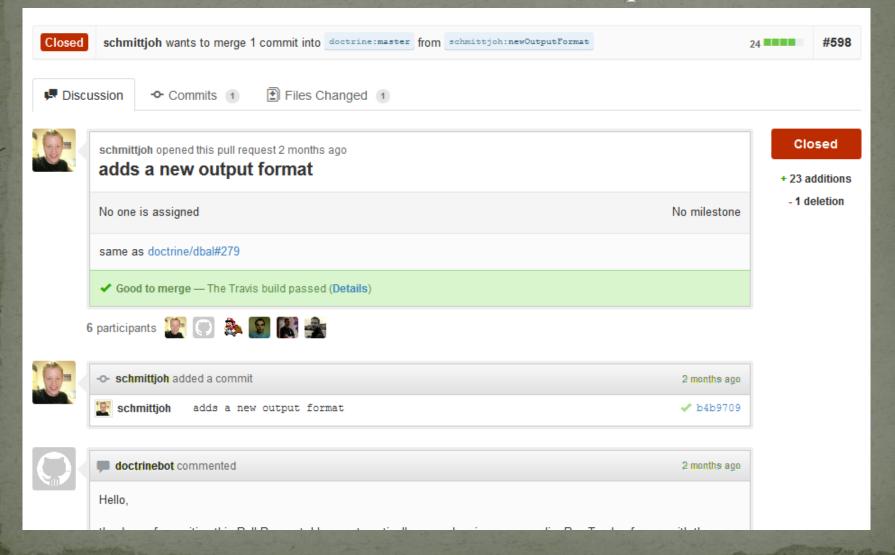
#### CR Tools: Atlassian Crucible



#### CR Tools: Atlassian Crucible



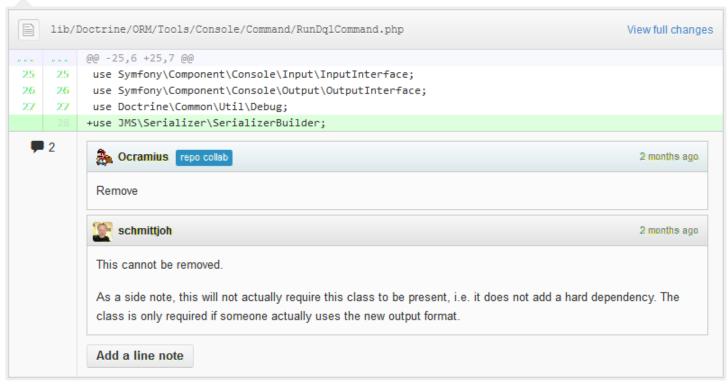
## CR Tools: GitHub Pull Requests



## CR Tools: GitHub Pull Requests



Ocramius started a discussion in the diff 2 months ago.





Ocramius started a discussion in the diff 2 months ago



lib/Doctrine/ORM/Tools/Console/Command/RunDqlCommand.php

View full changes

#### Rules of Code Reviews

- Review *ALL* the changes.
- Do functional testing as much as possible (within reason).
- Nothing goes to production until it's been reviewed and tested.
- If something looks wrong or you're just unsure of the reasoning behind it, ASK!!!
- If you don't see anything wrong, it's OK to not say anything.

## Rules of Code Reviews (cont.)

- Code must be *correct*, but it doesn't have to be exactly how you would write it.
- Don't just look at what they *did*; also think about what they *didn't do*. (What was missed?)
- CR / testing is a higher priority than development.
- If you see a problem that's outside the scope of the ticket, open a separate ticket for that issue.
- The goal is better code quality. It's not personal.

## A Suggested Workflow

- 1. Jack creates ticket and puts "in scrum"
- 2. Henry grabs ticket and writes code
- 3. Henry sets ticket to "Testing" and puts "in scrum"
- 4. Charlie grabs ticket and does CR / testing
  - If ticket fails, re-open and send back to Henry, repeat as necessary
  - If ticket passes, Charlie moves changes to production

# A Suggested Workflow

- 1. Jack creates ticket and puts "in scrum"
- 2. Henry grabs ticket and writes code
- 3. Henry sets ticket to "Testing" and puts "in scrum"
- 4. Charlie grabs ticket and does CR / testing
  - If ticket fails, re-open and send back to Henry, repeat

as necessary

If ticke



to production

# Benefits of Requiring Peer CRs

- Increased Quality Reduced chance of mistakes making it to production
- **Peer Pressure** If you know someone will see your code, you'll be more conscious of the quality
- **Distributed Knowledge** At least 2 people must know what a given line of code does
- **Distributed Ownership** Instead of each person having his / her own area of responsibility, everything is owned by the team. (We're all in this together!)

# Recommended Reading

- Mark Chu-Carroll: <u>Things Everyone Should Do:</u> <u>Code Review</u>
- Jeff Atwood: <u>Code Reviews: Just Do It</u>
- Steve McConnell: Code Complete: A Practical Handbook of Software Construction (http://amzn.com/0735619670)
- Karl Wiegers: Peer Reviews in Software: A Practical Guide (http://amzn.com/0201734850)

## Who am I? Feedback / Contact / Slides

- Team Leader, Oibas / CMS Development, at OpticsPlanet, Inc.
- Zend Certified Engineer PHP 5 & ZF
- Founder / Organizer of Lake / Kenosha PHP
- Joind.in: <a href="https://joind.in/8160">https://joind.in/8160</a>
- Email: patrick.schwisow@gmail.com
- Twitter: @PSchwisow
- Slides: <a href="https://github.com/PSchwisow/Miscellaneous/">https://github.com/PSchwisow/Miscellaneous/</a>