

Code Reviews: The “Secret” to Building Quality Software

Patrick Schwisow
Midwest PHP
March 15, 2014

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

*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 (<http://www.atlassian.com/software/stash/overview>)
- GitHub Pull Requests
- Review Board (<http://www.reviewboard.org/>)

CR Tools: Atlassian Crucible

 **Default Project** > CR-92

SWD-168: Update Op\MethodFilter\ProxyFactory to support protected methods

Closed on 28 Mar 11

Author & Moderator:  Reviewers:  

CR-92 8

Details

Objectives

General Comments 2

OP (trunk)

trunk/common

Slayer/Proxy

DbBannersProxy.php

DbBarcelonasessionsProxy.php

DbBlackhawkProxy.php

DbDb01SabioProxy.php

DbDb01SabioStoreProxy.php

DbDb01SabioTempProxy.php

DbDb01UnittestProxy.php

DbDb01VesnaProxy.php

DbFeedbackProxy.php

DbLd1unittestProxy.php

DbLd1webdevProxy.php

DbNafanunittestProxy.php

DbNorwaysabioProxy.php

DbOpMainProxy.php

DbOpSearchProxy.php

DbOpplanetрмаProxy.php

DbPhplistProxy.php

DbPortalunittestProxy.php




DbPortalwebdevProxy.php

DbProcesscontrollerProxy.php


DbProductreviewsProxy.php

DbProxyFactory.php

Details

Participant	Role	Time Spent	Comments	Latest Comment
 Oleh Namaka	Author & Moderator	11m	3	Sorry, I misunderstood the value of
 Patrick Schwisow	Reviewer - 100% complete	16m	4	We should still include default value
 Wayne Dean	Reviewer - 100% complete	9m	1	Oleh, this hasn't yet been reverted.
Total		36m	8	


Files: 38


Linked Issue:  [SWD-168](#) – Update Op\MethodFilter\ProxyFactory to support protected methods

Objectives


- In `$_proxyClassTemplate`, all public methods must be before the placeholder '`<methods>`' or generated proxies will not pass code checker if methods.
- `_generateMethods` should create two strings (one for public methods, one for private methods) and concatenate them when returning
- Remove hard-coding of 'public' in method generation.

General Comments

 [Patrick Schwisow](#) says:
After a change like this, you should regenerate all service layer proxies.

 [Oleh Namaka](#) says:
Done.

CR Tools: Atlassian Crucible

 Default Project > CR-92

SWD-168: Update Op\MethodFilter\ProxyFactory to support protected methods

Closed on 28 Mar 11

Author & Moderator Reviewers

CR-92 8

- DbBannersProxy.php
- DbBarcelonasessionsProxy.php
- DbBlackhawkProxy.php
- DbDb01SabioProxy.php
- DbDb01SabioStoreProxy.php
- DbDb01SabioTempProxy.php
- DbDb01UnittestProxy.php
- DbDb01VesnaProxy.php
- DbFeedbackProxy.php
- DbLd1unittestProxy.php
- DbLd1webdevProxy.php
- DbNafanunittestProxy.php
- DbNorwaysabioProxy.php
- DbOpMainProxy.php
- DbOpSearchProxy.php
- DbOpplanetmaProxy.php
- DbPhplistProxy.php
- DbPortalunittestProxy.php
- DbPortalwebdevProxy.php
- DbProcesscontrollerProxy.php
- DbProductreviewsProxy.php
- DbPrupProxy.php
- DbPrupsupplierProxy.php
- DbRmatestProxy.php
- DbSabioProxy.php
- DbSabioStoreTestProxy.php
- DbSabioTestProxy.php
- DbSabiodevProxy.php

/trunk/.../MethodFilter/ProxyFactory.php **Changed** 4 — 2 hidden

20372 20388 20506 20528

▼	AndyS	20053	75 75	{	
			76 76	\$this->_methodfilter = \$methodfilter;	
			77 77	}	
⌵	OlehN	20388	78	<methods>	
	AndyS	20053	79	}	
	OlehN	20528	78	<methods>}	
	AndyS	20053	80 79	EOT;	
			81 80		
			82 81	/**	


OlehN 20388 301 300

\$parameterString .= '\$' . \$param->getName();


\$argumentString .= '\$' . \$param->getName();

⌵ OlehN 20506 304

if (\$param->isDefaultValueAvailable() && !is_null(\$param->getDefaultValue()))

 Patrick Schwisow says:

We should still include default value if it is NULL

 Oleh Namaka says:

Sorry, I misunderstood the value of NULL here.

⌵ OlehN 20528 303

if (\$param->isDefaultValueAvailable()) {

\$defValue = var_export(\$param->getDefaultValue(), true);

if (strpos(\$defValue, PHP_EOL) !== false) {

\$defValue = str_replace(PHP_EOL, '', \$defValue);


CR Tools: GitHub Pull Requests

Closed schmittjoh wants to merge 1 commit into `doctrine:master` from `schmittjoh:newOutputFormat` 24 #598

Discussion

Commits 1

Files Changed 1



schmittjoh opened this pull request 2 months ago

adds a new output format

No one is assigned No milestone







same as [doctrine/dbal#279](#)


✓ Good to merge — The Travis build passed ([Details](#))

Closed


+ 23 additions


- 1 deletion

6 participants      



schmittjoh added a commit 2 months ago


 schmittjoh adds a new output format ✓ b4b9709




doctrinebot commented 2 months ago

Hello,

CR Tools: GitHub Pull Requests

 **Ocramius** started a discussion in the diff 2 months ago

 lib/Doctrine/ORM/Tools/Console/Command/RunDqlCommand.php [View full changes](#)

@@ -25,6 +25,7 @@

25

25

use Symfony\Component\Console\Input\InputInterface;

26

26

use Symfony\Component\Console\Output\OutputInterface;

27


27

use Doctrine\Common\Util\Debug;


28

+use JMS\Serializer\SerializerBuilder;

2

 **Ocramius** [repo collab](#) 2 months ago


Remove

 **schmittjoh** 2 months ago

This cannot be removed.

As a side note, this will not actually require this class to be present, i.e. it does not add a hard dependency. The class is only required if someone actually uses the new output format.

Add a line note

 **Ocramius** started a discussion in the diff 2 months ago

 lib/Doctrine/ORM/Tools/Console/Command/RunDqlCommand.php [View full changes](#)

@@ -118,6 +124,22 @@

protected function execute(InputInterface \$input, OutputInterface \$output)

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 ticket passes, move to production



Benefits of Requiring Peer CRs

- Increased Quality of Work
- Decreased Production Bugs
- Peer Pressure to Improve Quality
- Distributed Knowledge
- Improved On-Boarding & Mentoring
- Distributed Ownership / Team Building

Recommended Reading

- Mark Chu-Carroll: Things Everyone Should Do: Code Review
- Jeff Atwood: Code Reviews: Just Do It
- Steve McConnell: *Code Complete: A Practical Handbook of Software Construction*
(<http://amzn.com/o735619670>)
- Karl Wieggers: *Peer Reviews in Software: A Practical Guide* (<http://amzn.com/o201734850>)

Who am I?

Feedback / Contact / Slides

- Software Development Team Leader at OpticsPlanet, Inc.
- Zend Certified Engineer – PHP 5 & ZF
- Founder / Organizer of Lake / Kenosha PHP
- Email: patrick.schwisow@gmail.com
- Twitter: @PSchwisow
- Slides:
<https://github.com/PSchwisow/Miscellaneous/>