

Testing and Documentation Framework for the Pharo Quality Assistant.

Lars Vögtlin

Universität Freiburg

lars.voegtlin@unifr.ch

Supervisor: Yuriy Tymchuk

23.05.2017

What is the Pharo Quality Assistant?

A live feedback code quality tool for Pharo.

- Critiques for methods and classes based on lint rules
- Makes you write better code
- Guideline for new programmers
- Used during continuous integration \Rightarrow whole ecosystem

Example

renraku

- Work
- Renraku
- Renraku-Help

ExampleClassNoComm

- ExampleClassNoComm
- ExampleEquivalentSup
- ExampleEquivalentSi

Hier. Class Com.

-- all --
as yet unclassified

aMethod

```
aMethod  
  ^true.
```

renraku

- Work
- Renraku
- Renraku-Help

ExampleClassNoComm

- ExampleEquivalentSup
- ExampleEquivalentSi
- ExampleEquivalentSi

Hier. Class Com.

-- all --
as yet unclassified

aMethod

```
aMethod  
  ^true.
```

1/2 [1] Format as you read W +L

⚠ Methods equivalently defined in superclass ? ✖ Helpful? 🍏 🍎

⚠ Unclassified methods ? ✖ Helpful? 🍏 🍎

- How many critiques do exist?
- What are they checking?
⇒ PharoStyle

What is the PharoStyle?

A tool that generates a document containing the style guidelines for writing code for the Pharo programming language.

- Takes all the lint rules in the system
- Creates a markdown file

Doesn't use the result of a collect:/select:

Checks for senders of typical collection enumeration methods that return an unused result.

Uses detect:ifNone: instead of contains:

Checks for the common code fragment: "(aCollection detect: [:each | 'some condition'] ifNone: [nil]) ~= nil".

Unnecessary "= true"

Check for a =, ==, ~=, or ~~ message being sent to true/false or with true/false as the argument.

Block immediately evaluated

Check for blocks that are immediately evaluated.

ifTrue:/ifFalse: returns instead of and:/or:'s

Checks for common ifTrue: returns that could be simplified using a boolean expression.

Literal array contains only characters

Literal arrays containing only characters can more efficiently be represented as strings.

Uses or's instead of a searching literal

Checks for repeated literal equality tests that should rather be implemented as a search in a literal collection.

Uses detect:ifNone: instead of contains:

Checks for the common code fragment: "(aCollection detect: [:each | 'some condition'] ifNone: [nil]) ~= nil".

Unnecessary "= true"

Check for a =, ==, ~=, or ~~ message being sent to true/false or with true/false as the argument.

Good Exmaples

```
anExampleGood
<reExampleGood>
| t |

t := true.

^ t ifTrue: [ 'do something' ] iffFalse: [ 'somethign else' ].
```

Bad Exmaples

```
anExampleBad
<reExampleBad>
| t |

t := true.

^ t == true ifTrue: [ 'do something' ] iffFalse: [ 'somethign else' ].
```

Problems

- Can't test the rules
 - No additional information for the rules
-
- There is already a basic testing framework but hard to use
 - Testing the rules by giving them examples
 - Works as known from normal unit tests
 - Documentation enrichment with examples

Renraku-TestDoc-Example

renraku|Refactoring-Co...
WORK
▶ Refactoring-Core
▶ Refactoring-Critics
▶ Renraku
▶ Renraku-Help
▶ Renraku-Test
▶ Renraku-Test-Classe
▶ Renraku-TestDoc

- RBDefinesEqualNotHashRule
- RBEqualsTrueRule
- RBEquivalentSuperclassMet
- RBInconsistentMethodClassi
- RBInstVarInSubclassesRule
- RBJustSendsSuperRule
- RBNoClassCommentRule
- RBRRefersToClassRule

-- all --
accessing
testing
*renraku-TestDoc

anExampleBad
- checksMethod
- uniqueIdentifierName

Hier. < Class ? Com.

renraku|Refactoring-Co...
WORK
▶ Refactoring-Core
▶ Refactoring-Critics
▶ Renraku
▶ Renraku-Help
▶ Renraku-Test
▶ Renraku-Test-Classe
▶ Renraku-TestDoc

- RBDefinesEqualNotHashRule
- RBEqualsTrueRule
- RBEquivalentSuperclassMet
- RBInconsistentMethodClassi
- RBInstVarInSubclassesRule
- RBJustSendsSuperRule
- RBNoClassCommentRule
- RBRRefersToClassRule

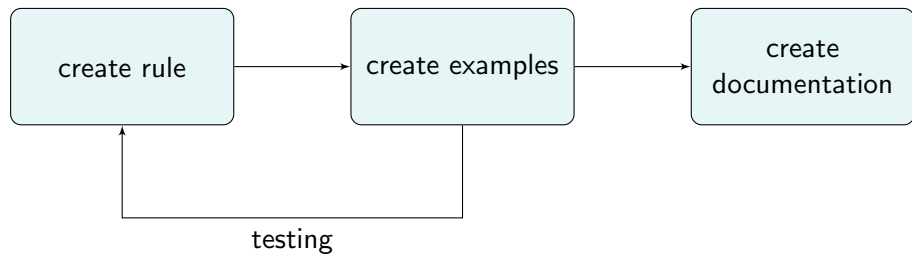
-- all --
accessing
testing
*renraku-TestDoc

anExampleBad
- checksMethod
- uniqueIdentifierName

Hier. < Class ? Com.

- Two types of tests (Compiled Methods or Classes)

Renraku-TestDoc workflow



DEMO

- Does not print hierarchy-related data

Methods equivalently defined in superclass

Check for methods that are equivalent to their superclass methods.

Good Exmaples

```
aMethod  
  ^false.
```

Bad Exmaples

```
aMethod  
  ^true.
```

- Reference rules are not testable (RBClassNotReferenceRule)

- Helps to test and document the rules
- Runs on class side methods and dedicated classes
- Created examples for 10 rules