Does Your Code Measure Up?

By:
Adam Culp
Twitter: @adamculp

[https://joind.in/**13300**](https://joind.in/13300)

Does Your Code Measure Up?

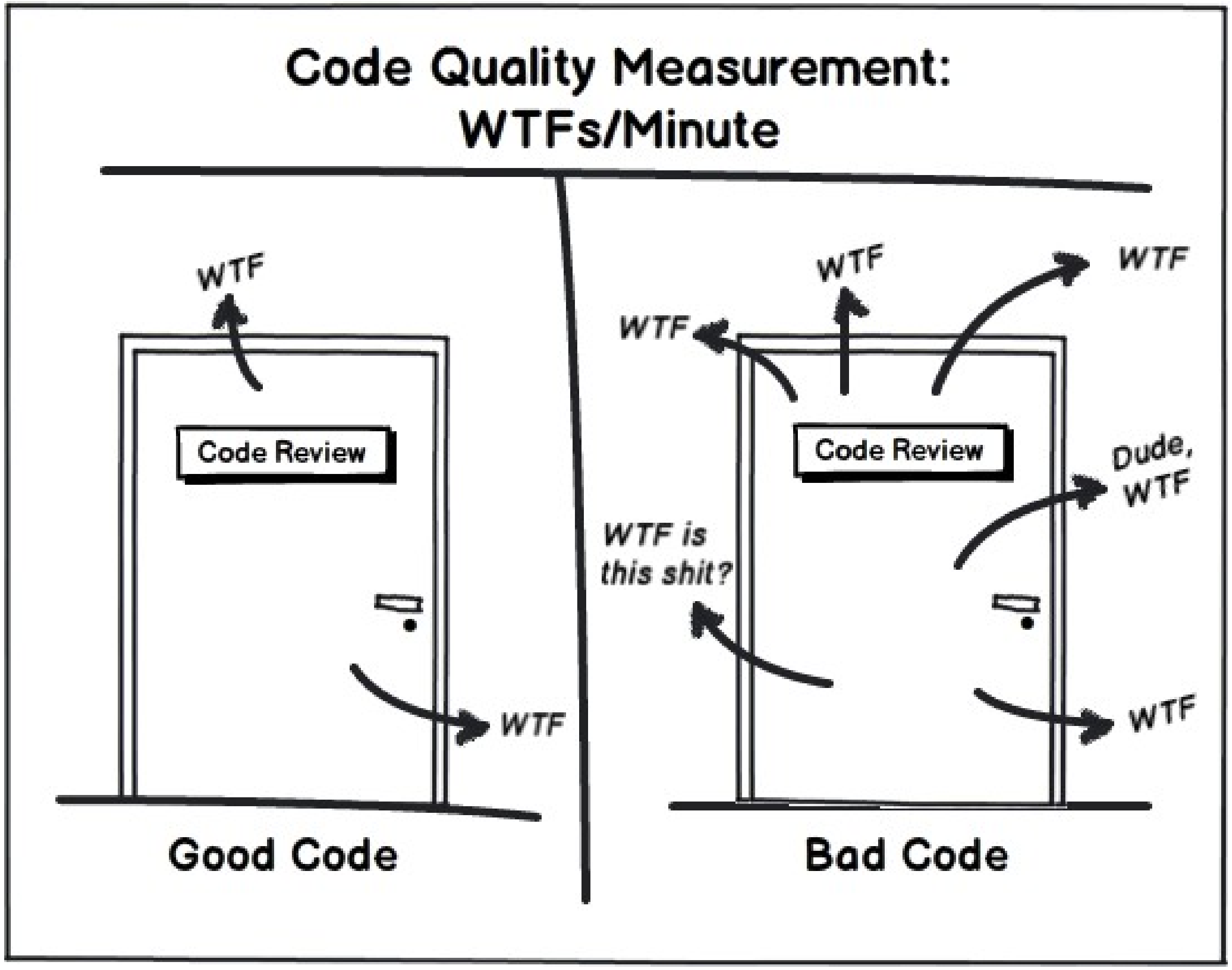
- About me
  - PHP 5.3 Certified
  - Consultant at Zend Technologies
  - Organizer SoFloPHP (South Florida)
  - Organized SunshinePHP (Miami)
  - Long distance (ultra) runner
  - Judo Black Belt Instructor

Does Your Code Measure Up?

- **Fan of iteration**
  - Pretty much everything requires iteration to do well:
    - Long distance running
    - Judo
    - Development
    - Evading project managers
    - Quality!

Does Your Code Measure Up?

- **How To Measure?**
  - Not productive



Does Your Code Measure Up?

- **How To Measure?**
  - More accurate, quantifiable

Does Your Code Measure Up?

- Why Measure?
  - Highlight bugs

Does Your Code Measure Up?

- **Why Measure?**
  - Highlight bugs
  - Improve quality
    - Easier onboarding
    - Less reading, more writing
    - Testable

Does Your Code Measure Up?

- **Why Measure?**
  - Highlight bugs
  - Improve quality
    - Easier onboarding
    - Less reading, more writing
    - Testable
  - Satisfied customers
    - Faster development
    - Less broken

Does Your Code Measure Up?

- **Why Measure?**
  - Highlight bugs
  - Improve quality
    - Easier onboarding
    - Less reading, more writing
    - Testable
  - Satisfied customers
    - Faster development
    - Less broken
  - Personal pride

Does Your Code Measure Up?

- **Why Measure?**
  - Highlight bugs
  - Improve quality
    - Easier onboarding
    - Less reading, more writing
    - Testable
  - Satisfied customers
    - Faster development
    - Less broken
  - Personal pride
  - Higher salary

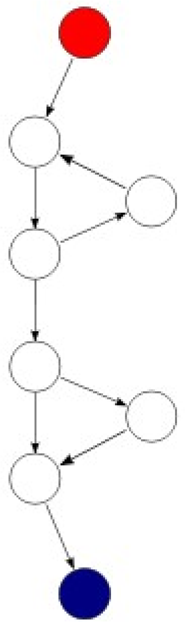
Does Your Code Measure Up?

- **What To Measure?**
  - Cyclomatic complexity
    - "The count of the number of linearly independent paths through the source code." - wikipedia
    - Decision points
    - Less than 10 (personally less than 6)

Does Your Code Measure Up?

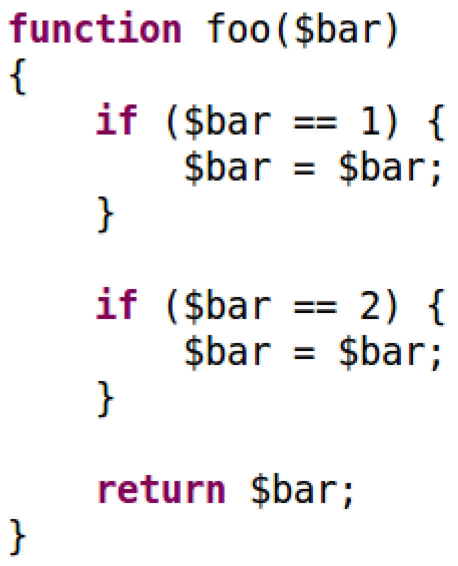
- Cyclomatic Complexity Example

```
function foo($bar)
{
    if ($bar == 1) {
        $bar = $bar;
    }

    if ($bar == 2) {
        $bar = $bar;
    }

    return $bar;
}
```

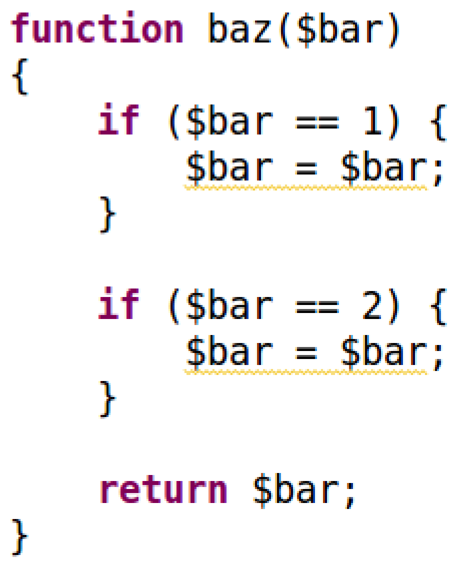


Does Your Code Measure Up?

- **What To Measure?**
  - Cyclomatic complexity
  - Duplicate code
    - Rule of 3



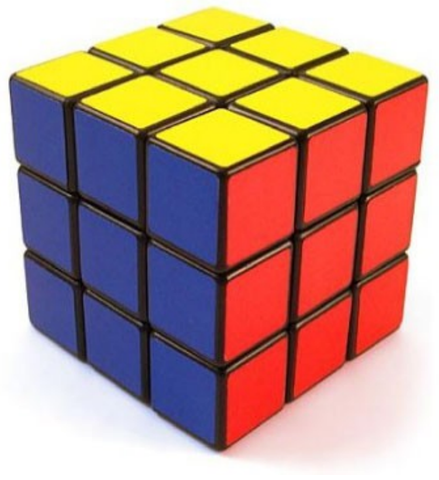
```
function foo($bar)
{
    if ($bar == 1) {
        $bar = $bar;
    }

    if ($bar == 2) {
        $bar = $bar;
    }

    return $bar;
}
```



```
function baz($bar)
{
    if ($bar == 1) {
        $bar = $bar;
    }

    if ($bar == 2) {
        $bar = $bar;
    }

    return $bar;
}
```

Does Your Code Measure Up?

- **What To Measure?**
  - Cyclomatic complexity
  - Duplicate code
  - **Long classes**
    - Less than 1,000 lines
    - Classes solve **a** ***(1)*** problem

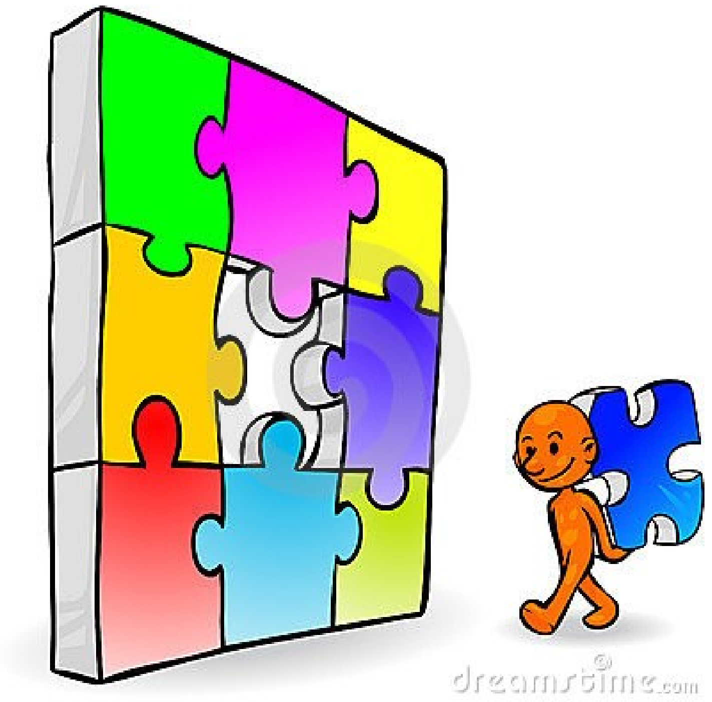


Does Your Code Measure Up?

- **What To Measure?**
  - Cyclomatic complexity
  - Duplicate code
  - Long classes
  - Class Complexity
    - Less than 50

Does Your Code Measure Up?

- **What To Measure?**
  - Cyclomatic complexity
  - Duplicate code
  - Long classes
  - Class complexity
  - Long methods
    - Less than 100 (personally less than 20)
    - Method should do one thing



Does Your Code Measure Up?

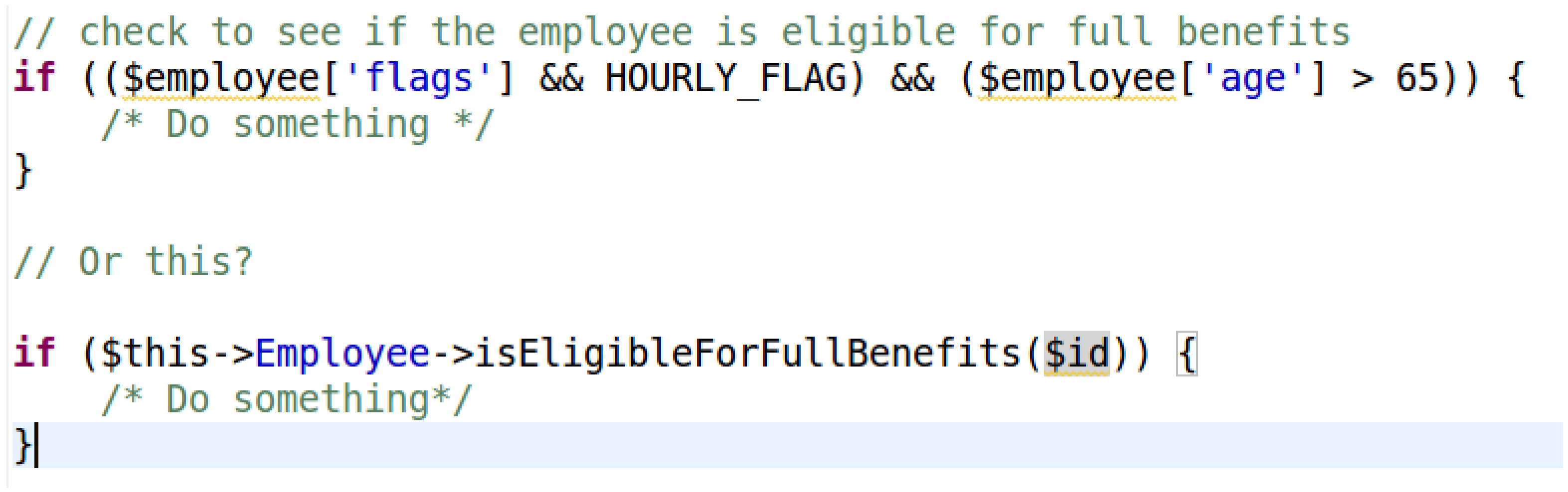
- **What To Measure?**
  - Cyclomatic complexity
  - Duplicate code
  - Long classes
  - Class complexity
  - Long methods
  - Unused variables

Does Your Code Measure Up?

- **What To Measure?**
  - Cyclomatic complexity
  - Duplicate code
  - Long classes
  - Class complexity
  - Long methods
  - Unused variables
  - **Lack or overuse of comments**
    - Clear, concise, not explain bad code

Does Your Code Measure Up?

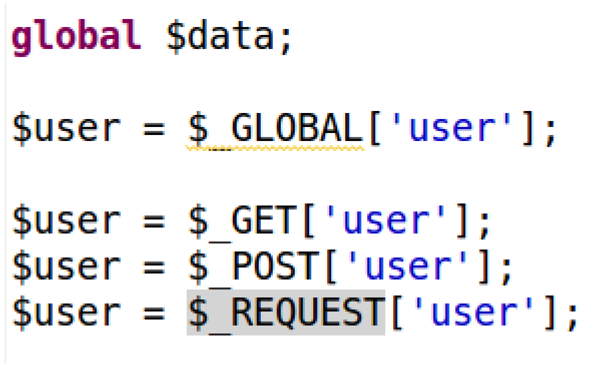
- **Comment Example**



```
// check to see if the employee is eligible for full benefits
if (($employee['flags'] && HOURLY_FLAG) && ($employee['age'] > 65)) {
    /* Do something */
}

// Or this?

if ($this->Employee->isEligibleForFullBenefits($id)) {
    /* Do something*/
}
```

Does Your Code Measure Up?

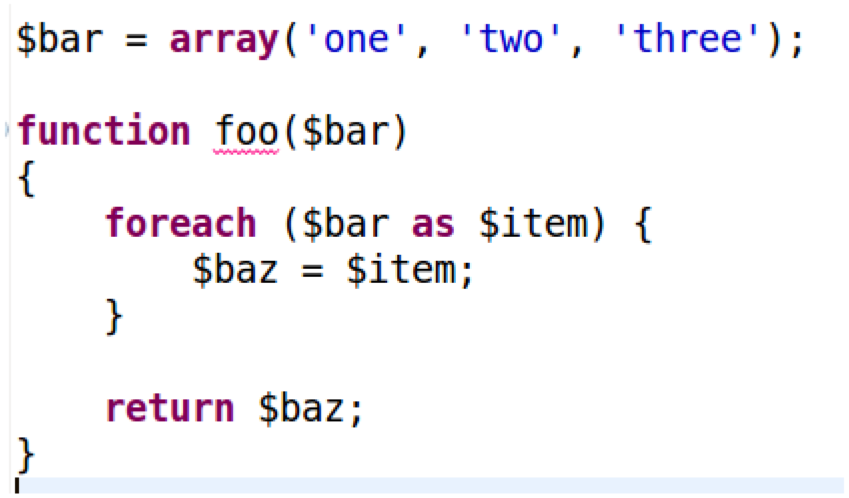
- **What To Measure?**
  - Cyclomatic complexity
  - Duplicate code
  - Long classes
  - Class complexity
  - Long methods
  - Unused variables
  - Lack or overuse of comments
  - **Heavy global usage**



```php
global $data;

$user = $_GLOBAL['user'];

$user = $_GET['user'];
$user = $_POST['user'];
$user = $_REQUEST['user'];
```

Does Your Code Measure Up?

- **What To Measure?**
  - Cyclomatic complexity
  - Duplicate code
  - Long classes
  - Class complexity
  - Long methods
  - Unused variables
  - Lack or overuse of comments
  - Heavy global usage
  - **Npath complexity**
    - Possible paths through code
    - Less than 200 paths



```
$bar = array('one', 'two', 'three');

function foo($bar)
{
    foreach ($bar as $item) {
        $baz = $item;
    }

    return $baz;
}
```

Does Your Code Measure Up?

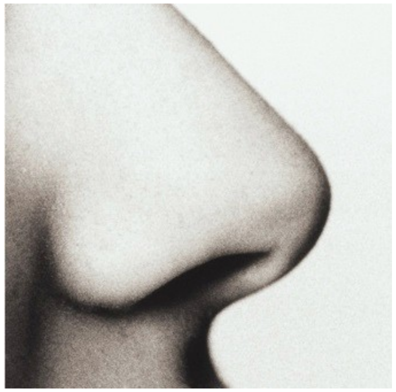
- **What To Measure?**
  - Cyclomatic complexity
  - Duplicate code
  - Long classes
  - Class complexity
  - Long methods
  - Unused variables
  - Lack or overuse of comments
  - Heavy global usage
  - Npath complexity
  - **Much, much, more**
    - Code smells

Does Your Code Measure Up?

- **Code “smells”**
  - What are “smells”?
    - Indications of spoiled code nearby
    - Not conclusive
    - The “smell” is not bad

Does Your Code Measure Up?

- Code “smells”
  - “Smells” hinting a refactor may be needed:
    - Duplicate Code (rule of 3)
    - Long Methods
    - Large Class
    - Long Parameter (argument) List
    - Divergent Change – cascade change to accommodate another
    - Shotgun Surgery – change ripples as bugs
    - Feature Envy – method uses parts from other class
    - Switch Statements – sacrifice polymorphism

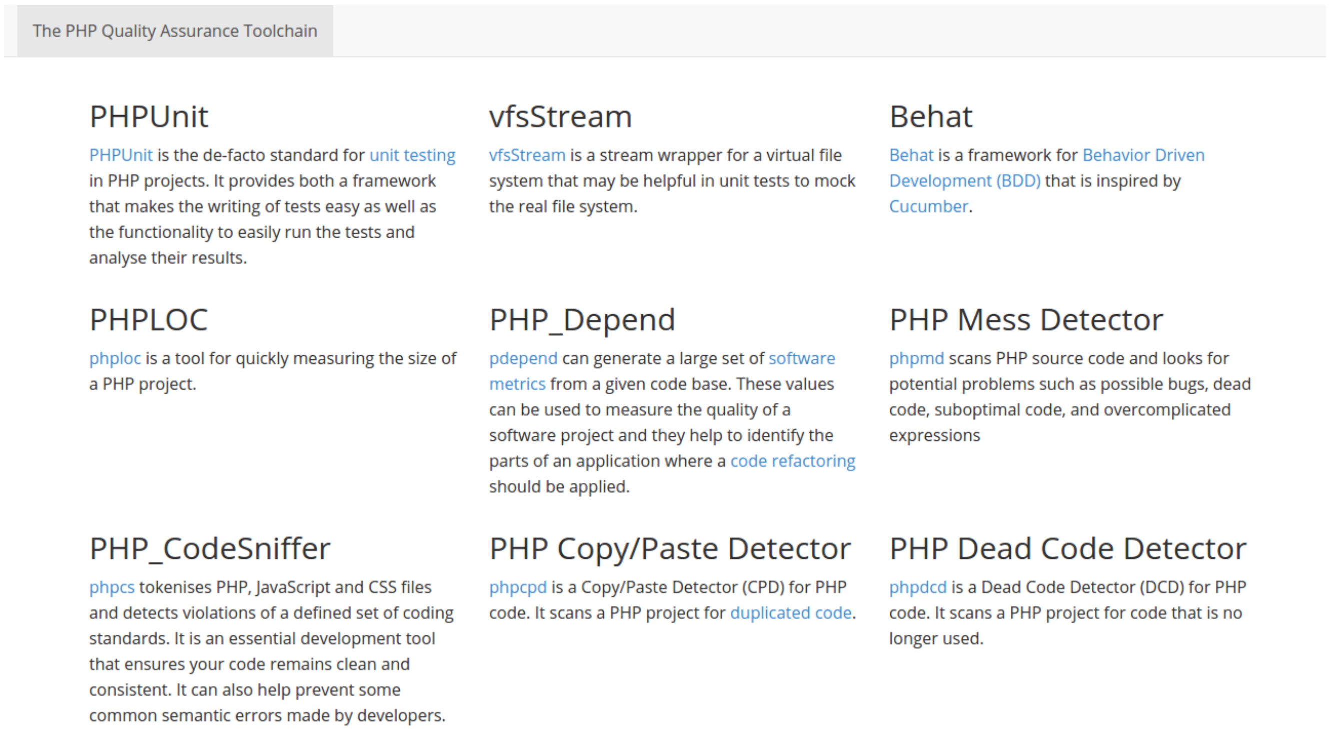


Does Your Code Measure Up?

- **Code "smells"**
  - Cont'd:
    - Lazy Class – class not doing much
    - Speculative Generality – something built for possible future
    - Temporary Field/Variable
    - Message Chains – object asking object asking object
    - Middle Man – directors in place but serve no real purpose
    - Inappropriate Intimacy – classes share private parts
    - Data Class – getters and setters, but nothing else
    - Comments – where comments cover bad code

Does Your Code Measure Up?

- Tools
  - PHPqatools.org


The PHP Quality Assurance Toolchain

PHPUnit
PHPUnit is the de-facto standard for unit testing in PHP projects. It provides both a framework that makes the writing of tests easy as well as the functionality to easily run the tests and analyse their results.

vfsStream
vfsStream is a stream wrapper for a virtual file system that may be helpful in unit tests to mock the real file system.

Behat
Behat is a framework for Behavior Driven Development (BDD) that is inspired by Cucumber.

PHPLOC
phploc is a tool for quickly measuring the size of a PHP project.

PHP_Depend
pdepend can generate a large set of software metrics from a given code base. These values can be used to measure the quality of a software project and they help to identify the parts of an application where a code refactoring should be applied.

PHP Mess Detector
phpmd scans PHP source code and looks for potential problems such as possible bugs, dead code, suboptimal code, and overcomplicated expressions

PHP_CodeSniffer
phpcs tokenises PHP, JavaScript and CSS files and detects violations of a defined set of coding standards. It is an essential development tool that ensures your code remains clean and consistent. It can also help prevent some common semantic errors made by developers.

PHP Copy/Paste Detector
phpcpd is a Copy/Paste Detector (CPD) for PHP code. It scans a PHP project for duplicated code.

PHP Dead Code Detector
phpdcd is a Dead Code Detector (DCD) for PHP code. It scans a PHP project for code that is no longer used.

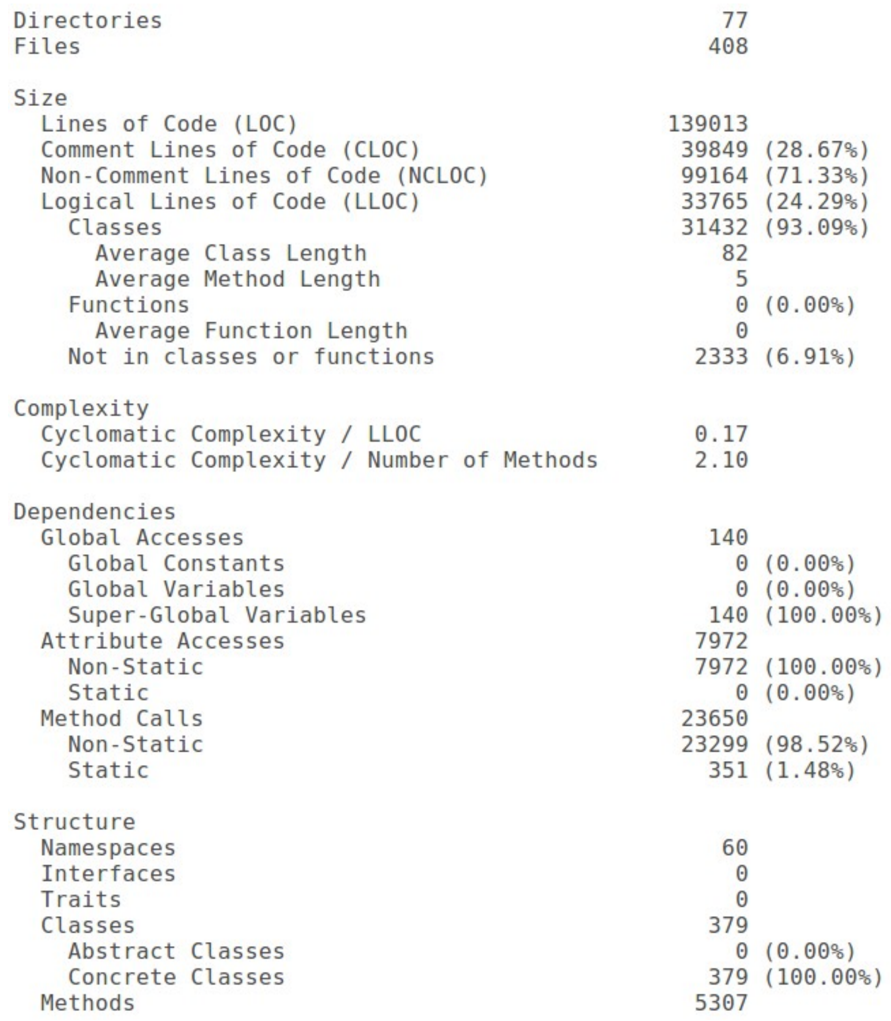
Does Your Code Measure Up?

- Tools
  - PHPqatools.org
    - PHPLoc

```
$ php phploc.phar -v --names "*.php" --exclude 'vendor'
/path/to/project/my-project/module/ >  /path/to/project/phpqatool-
results/phploc.txt
```

Does Your Code Measure Up?

- **PHPLoc Results**



```
Directories                                         77
Files                                              408

Size
  Lines of Code (LOC)                           139013
  Comment Lines of Code (CLOC)                   39849 (28.67%)
  Non-Comment Lines of Code (NCLOC)              99164 (71.33%)
  Logical Lines of Code (LLOC)                   33765 (24.29%)
    Classes                                      31432 (93.09%)
      Average Class Length                          82
      Average Method Length                          5
    Functions                                        0 (0.00%)
      Average Function Length                        0
    Not in classes or functions                   2333 (6.91%)

Complexity
  Cyclomatic Complexity / LLOC                    0.17
  Cyclomatic Complexity / Number of Methods       2.10

Dependencies
  Global Accesses                                  140
    Global Constants                                 0 (0.00%)
    Global Variables                                 0 (0.00%)
    Super-Global Variables                         140 (100.00%)
  Attribute Accesses                              7972
    Non-Static                                    7972 (100.00%)
    Static                                           0 (0.00%)
  Method Calls                                   23650
    Non-Static                                   23299 (98.52%)
    Static                                         351 (1.48%)

Structure
  Namespaces                                        60
  Interfaces                                         0
  Traits                                             0
  Classes                                          379
    Abstract Classes                                 0 (0.00%)
    Concrete Classes                               379 (100.00%)
  Methods                                         5307
```

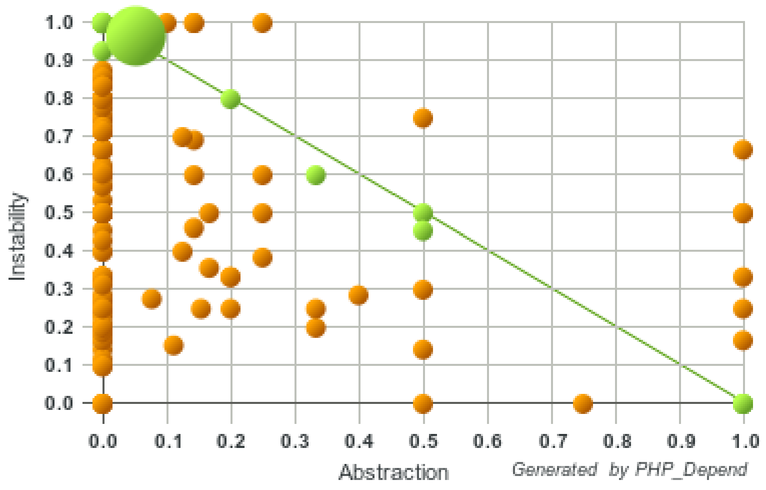
Does Your Code Measure Up?

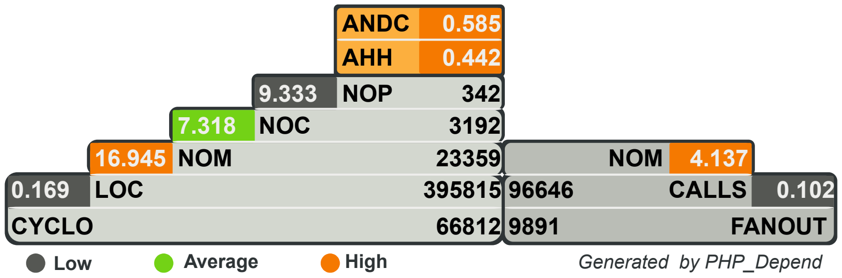
- Tools
  - PHPqatools.org
    - PHPLoc
    - PHP_Depend

```
$ php pdepend.phar --ignore='vendor' --summary-xml=' /path/to/project/phpqatool-
results/pdepend_output.xml' --jdepend-chart=' /path/to/project/phpqatool-
results/pdepend_chart.png' --overview-pyramid=' /path/to/project/phpqatool-
results/pdepend_pyramid.svg'  /path/to/project/my-project/module/
```

Does Your Code Measure Up?

- **PHP_Depend Result**
  - Graphs and XML output





		ANDC	0.585		
		AHH	0.442		
	9.333	NOP	342		
	7.318	NOC	3192		
	16.945	NOM	23359	NOM	4.137
0.169	LOC		395815	96646 CALLS	0.102
CYCLO			66812	9891 FANOUT	

● Low ● Average ● High

Generated by PHP_Depend

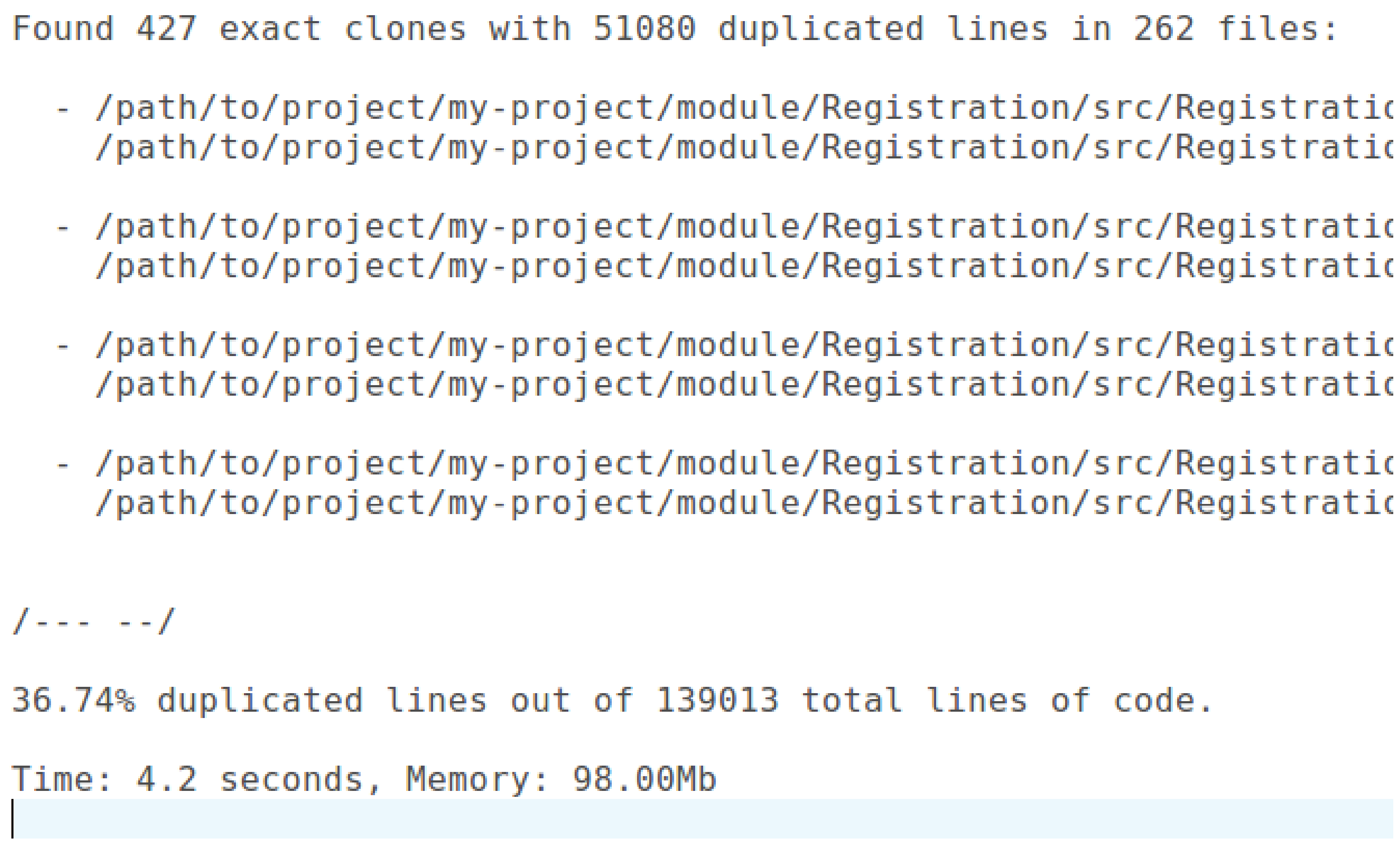
Does Your Code Measure Up?

- Tools
  - PHPqatools.org
    - PHPLoc
    - PHP_Depend
    - PHP Copy/Paste Detector

```
$ php phpcpd.phar  /path/to/project/my-project/module/ --exclude 'vendor' >
/path/to/project/phpqatool-results/phpcpd_output.txt
```

Does Your Code Measure Up?

- PHP Copy/Paste Detector Result



```
Found 427 exact clones with 51080 duplicated lines in 262 files:

  - /path/to/project/my-project/module/Registration/src/Registratio
    /path/to/project/my-project/module/Registration/src/Registratio

  - /path/to/project/my-project/module/Registration/src/Registratio
    /path/to/project/my-project/module/Registration/src/Registratio

  - /path/to/project/my-project/module/Registration/src/Registratio
    /path/to/project/my-project/module/Registration/src/Registratio

  - /path/to/project/my-project/module/Registration/src/Registratio
    /path/to/project/my-project/module/Registration/src/Registratio


/--- --/

36.74% duplicated lines out of 139013 total lines of code.

Time: 4.2 seconds, Memory: 98.00Mb
```

SoFloPHP Users Group
WWW.SOFLOPHP.COM

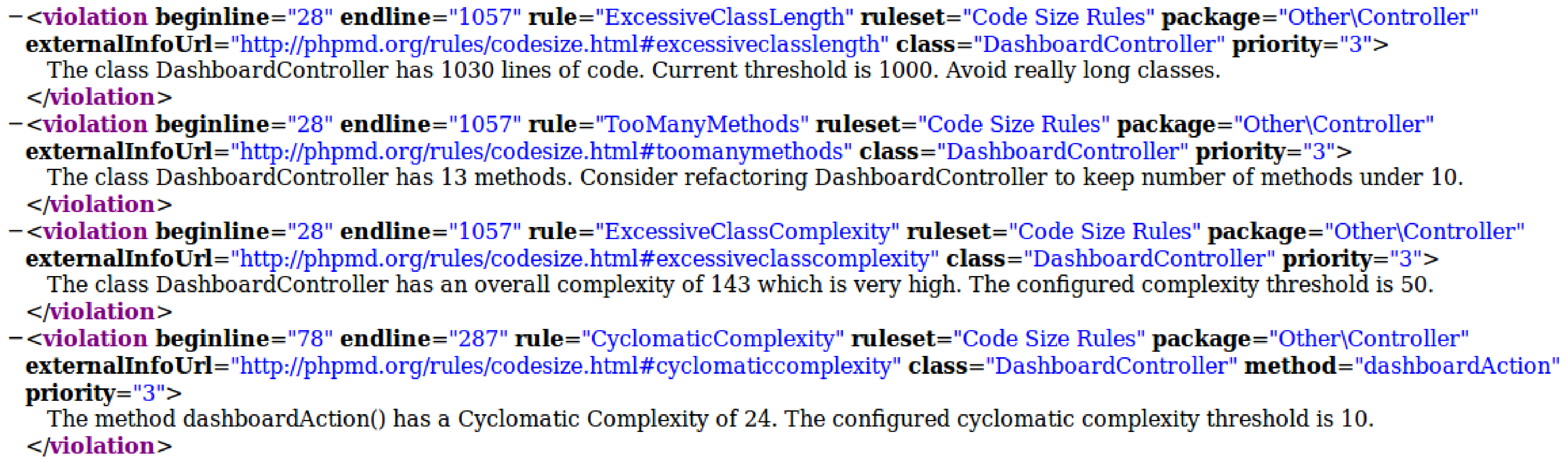
Does Your Code Measure Up?

- Tools
  - PHPqatools.org
    - PHPLoc
    - PHP_Depend
    - PHP Copy/Paste Detector
    - PHP Mess Detector
      - Codesize

```
$ php phpmd.phar  /path/to/project/my-project/module/ xml codesize --exclude
'vendor' --reportfile ' /path/to/project/phpqatool-
results/phpmd_codesize_output.xml'
```

Does Your Code Measure Up?

- PHP Mess Detector Result (codesize)



```
-<violation beginline="28" endline="1057" rule="ExcessiveClassLength" ruleset="Code Size Rules" package="Other\Controller"
 externalInfoUrl="http://phpmd.org/rules/codesize.html#excessiveclasslength" class="DashboardController" priority="3">
   The class DashboardController has 1030 lines of code. Current threshold is 1000. Avoid really long classes.
 </violation>
-<violation beginline="28" endline="1057" rule="TooManyMethods" ruleset="Code Size Rules" package="Other\Controller"
 externalInfoUrl="http://phpmd.org/rules/codesize.html#toomanymethods" class="DashboardController" priority="3">
   The class DashboardController has 13 methods. Consider refactoring DashboardController to keep number of methods under 10.
 </violation>
-<violation beginline="28" endline="1057" rule="ExcessiveClassComplexity" ruleset="Code Size Rules" package="Other\Controller"
 externalInfoUrl="http://phpmd.org/rules/codesize.html#excessiveclasscomplexity" class="DashboardController" priority="3">
   The class DashboardController has an overall complexity of 143 which is very high. The configured complexity threshold is 50.
 </violation>
-<violation beginline="78" endline="287" rule="CyclomaticComplexity" ruleset="Code Size Rules" package="Other\Controller"
 externalInfoUrl="http://phpmd.org/rules/codesize.html#cyclomaticcomplexity" class="DashboardController" method="dashboardAction"
 priority="3">
   The method dashboardAction() has a Cyclomatic Complexity of 24. The configured cyclomatic complexity threshold is 10.
 </violation>
```

Does Your Code Measure Up?

- Tools
  - PHPqatools.org
    - PHPLoc
    - PHP_Depend
    - PHP Copy/Paste Detector
    - PHP Mess Detector
      - Codesize
      - Unused

```
$ php phpmd.phar  /path/to/project/my-project/module/ xml unusedcode --exclude
'vendor' --reportfile ' /path/to/project/phpqatool-results/phpmd_unused_output.xml'
```

Does Your Code Measure Up?

- Tools
  - PHPqatools.org
    - PHPLoc
    - PHP_Depend
    - PHP Copy/Paste Detector
    - PHP Mess Detector
      - Codesize
      - Unused
    - PHP Dead Code Detector
      - Searches for code no longer used

Does Your Code Measure Up?

- Tools
  - PHPqatools.org
  - PHPMetrics
    - http://phpmetrics.org

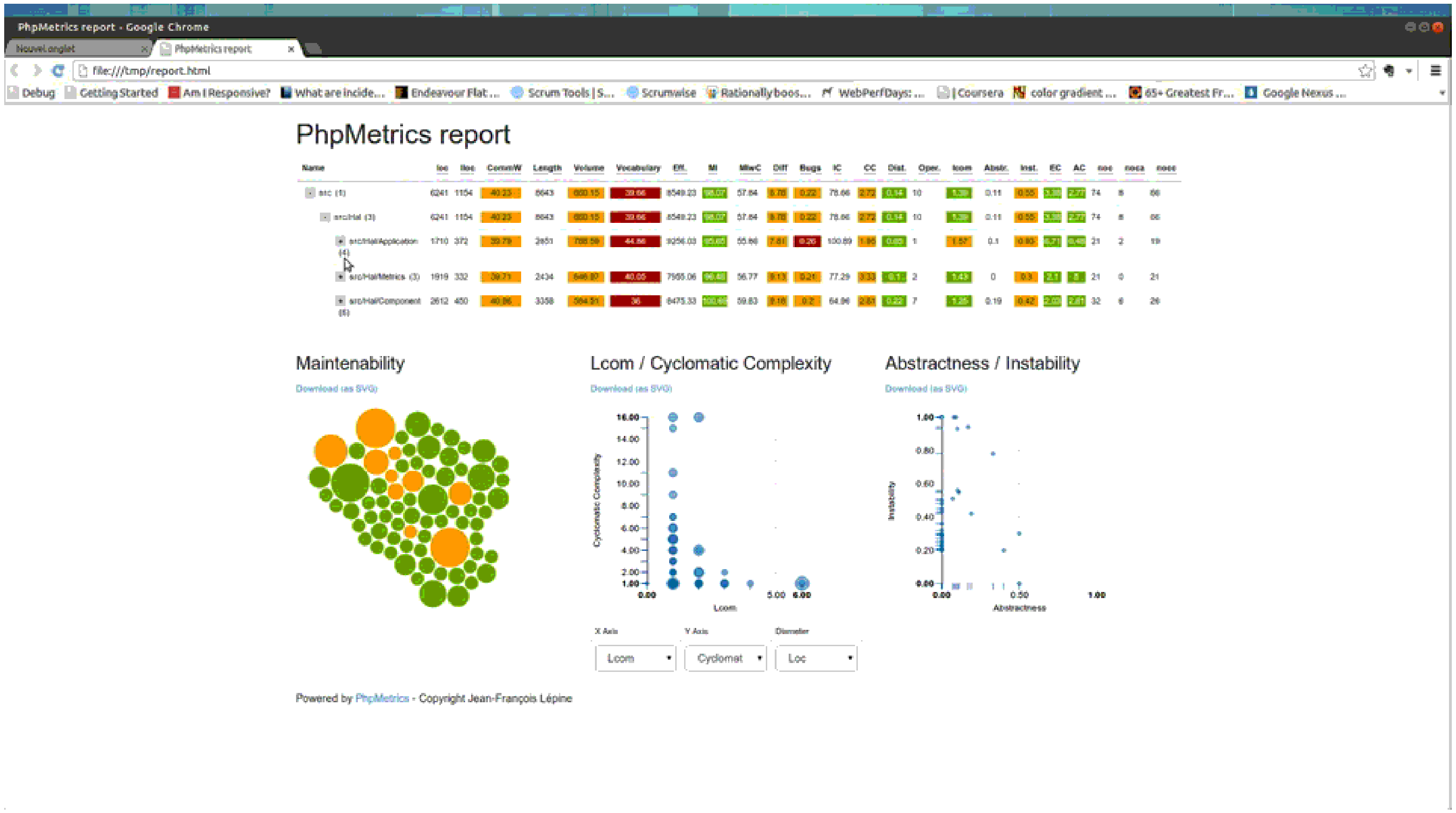
```
wget https://github.com/Halleck45/PhpMetrics/raw/master/build/phpmetrics.p
har
chmod +x phpmetrics.phar
mv phpmetrics.phar /usr/local/bin/phpmetrics
```

Usage:

```
phpmetrics --report-html=/path/of/report.html <folder or filename>
```

Does Your Code Measure Up?

- **PHPMetrics Results**



PhpMetrics report - Google Chrome

Nouvel onglet | PhpMetrics report

file:///tmp/report.html

Debug | Getting Started | Am I Responsive? | What are incide... | Endeavour Flat ... | Scrum Tools | S... | Scrumwise | Rationallyboos... | WebPerfDays: ... | | Coursera | color gradient ... | 65+ Greatest Fr... | Google Nexus ...

PhpMetrics report

Name	loc	lloc	CommW	Length	Volume	Vocabulary	Eff.	MI	MIwC	Diff	Bugs	IC	CC	Dist.	Oper.	lcom	Abstr.	Inst.	EC	AC	noc	noca	nocc
src (1)	6241	1154	40.23	6643	600.15	39.66	8549.23	98.07	57.84	8.78	0.22	78.66	2.72	0.14	10	1.39	0.11	0.55	3.38	2.77	74	8	66
src/Hal (3)	6241	1154	40.23	6643	600.15	39.66	8549.23	98.07	57.84	8.78	0.22	78.66	2.72	0.14	10	1.39	0.11	0.55	3.38	2.77	74	8	66
src/Hal/Application (4)	1710	372	39.79	2851	788.59	44.86	9256.03	95.65	55.66	7.81	0.26	100.89	1.95	0.03	1	1.57	0.1	0.93	6.71	0.48	21	2	19
src/Hal/Metrics (3)	1919	332	39.71	2434	646.97	40.05	7955.06	96.48	56.77	8.13	0.21	77.29	3.33	0.1	2	1.43	0	0.9	2.1	1	21	0	21
src/Hal/Component (5)	2612	450	40.86	3358	564.51	36	8475.33	100.60	59.83	9.18	0.2	64.96	2.81	0.22	7	1.25	0.19	0.42	2.03	2.81	32	6	26

Maintenability

Download (as SVG)

Lcom / Cyclomatic Complexity

Download (as SVG)

Abstractness / Instability

Download (as SVG)

X Axis: Lcom | Y Axis: Cyclomat | Diameter: Loc

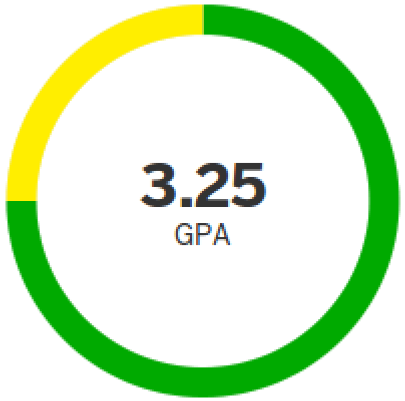
Powered by PhpMetrics - Copyright Jean-François Lépine

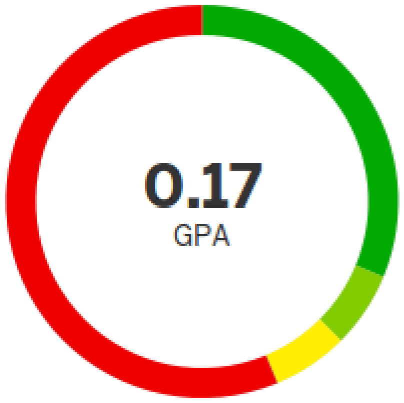
Does Your Code Measure Up?

- Tools
  - PHPqatools.org
  - PHPMetrics
  - PHP_Codesniffer
    - Create rules/sniffs to ensure standards are followed
    - From CLI

Does Your Code Measure Up?

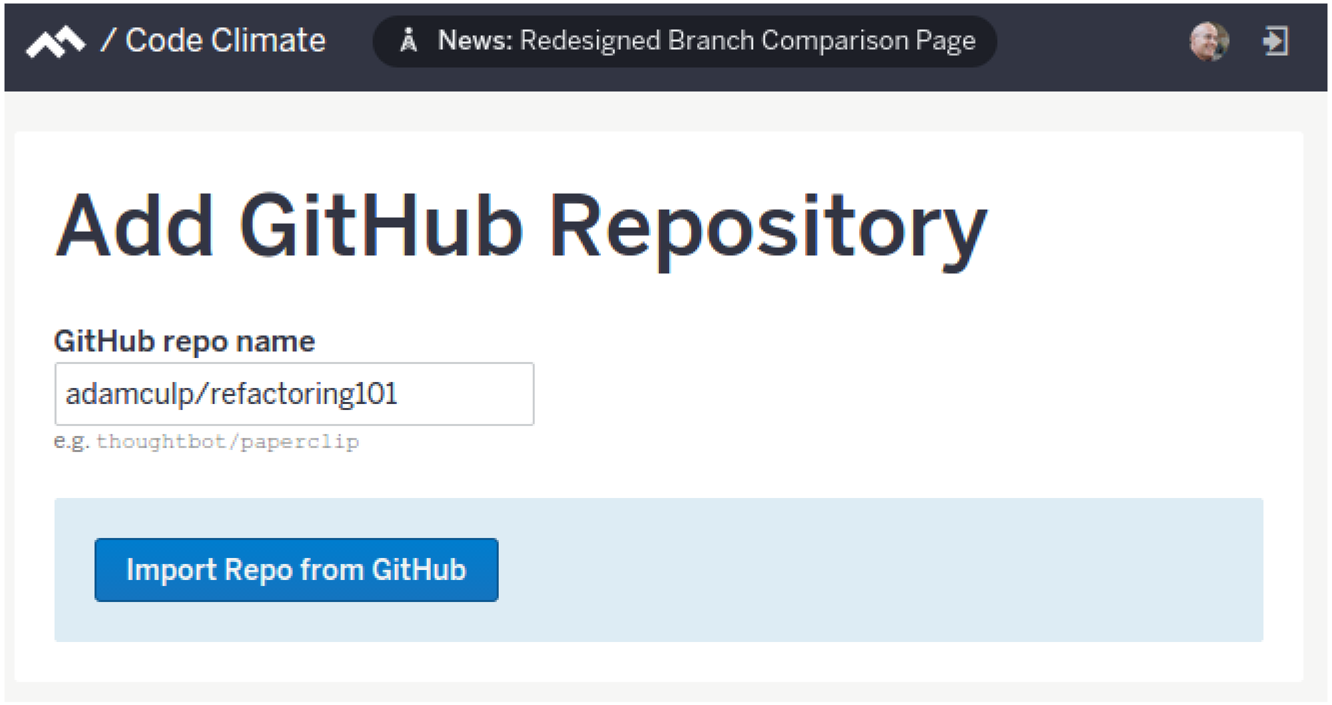
- Tools
  - PHPqatools.org
  - PHPMetrics
  - PHP_Codesniffer
  - Code Climate
    - Build from Git/Github repo
      - Open = free
      - Private = $
    - GPA – like in high school, simple





Does Your Code Measure Up?

- Code Climate
  - Build from Git/Github repo
    - Open = free
    - Private = $



Add GitHub Repository

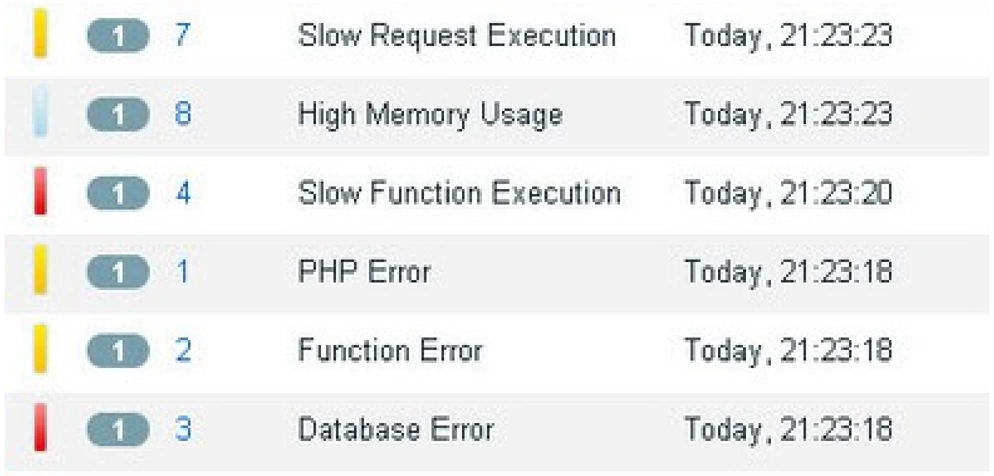
GitHub repo name

adamculp/refactoring101

e.g. thoughtbot/paperclip

Does Your Code Measure Up?

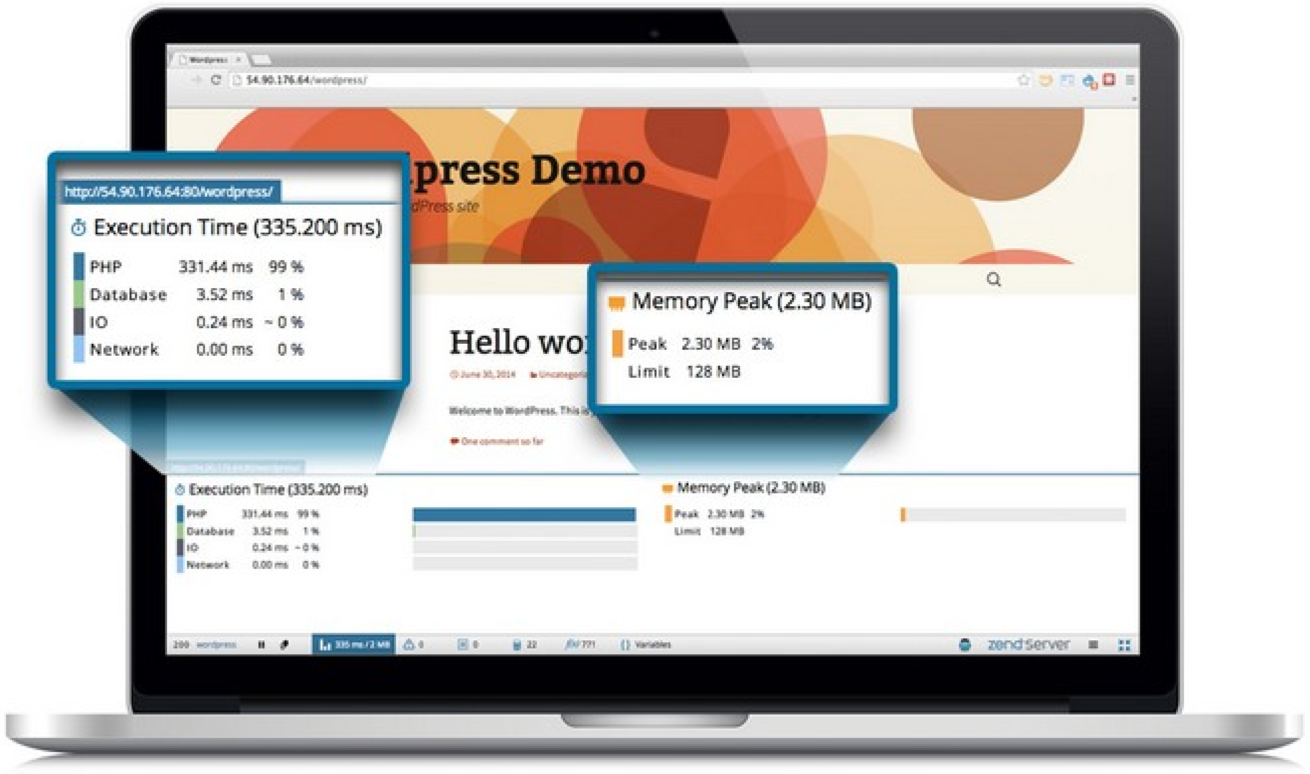
- Tools
  - PHPqatools.org
  - PHPMetrics
  - PHP_Codesniffer
  - Code Climate
  - **Zend Server and Z-Ray**
    - Debugging, Code tracing, Error reporting



🟨	1	7	Slow Request Execution	Today, 21:23:23
🟦	1	8	High Memory Usage	Today, 21:23:23
🟥	1	4	Slow Function Execution	Today, 21:23:20
🟨	1	1	PHP Error	Today, 21:23:18
🟨	1	2	Function Error	Today, 21:23:18
🟥	1	3	Database Error	Today, 21:23:18

Does Your Code Measure Up?

- **Realtime profiling**
  - Z-Ray



http://54.90.176.64:80/wordpress/

Execution Time (335.200 ms)

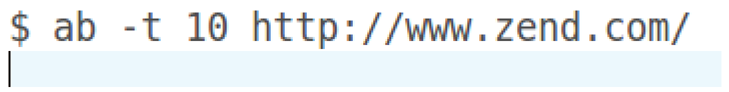
```
PHP        331.44 ms   99 %
Database     3.52 ms    1 %
IO           0.24 ms  ~ 0 %
Network      0.00 ms    0 %
```

Memory Peak (2.30 MB)

```
Peak   2.30 MB  2%
Limit  128 MB
```

Does Your Code Measure Up?

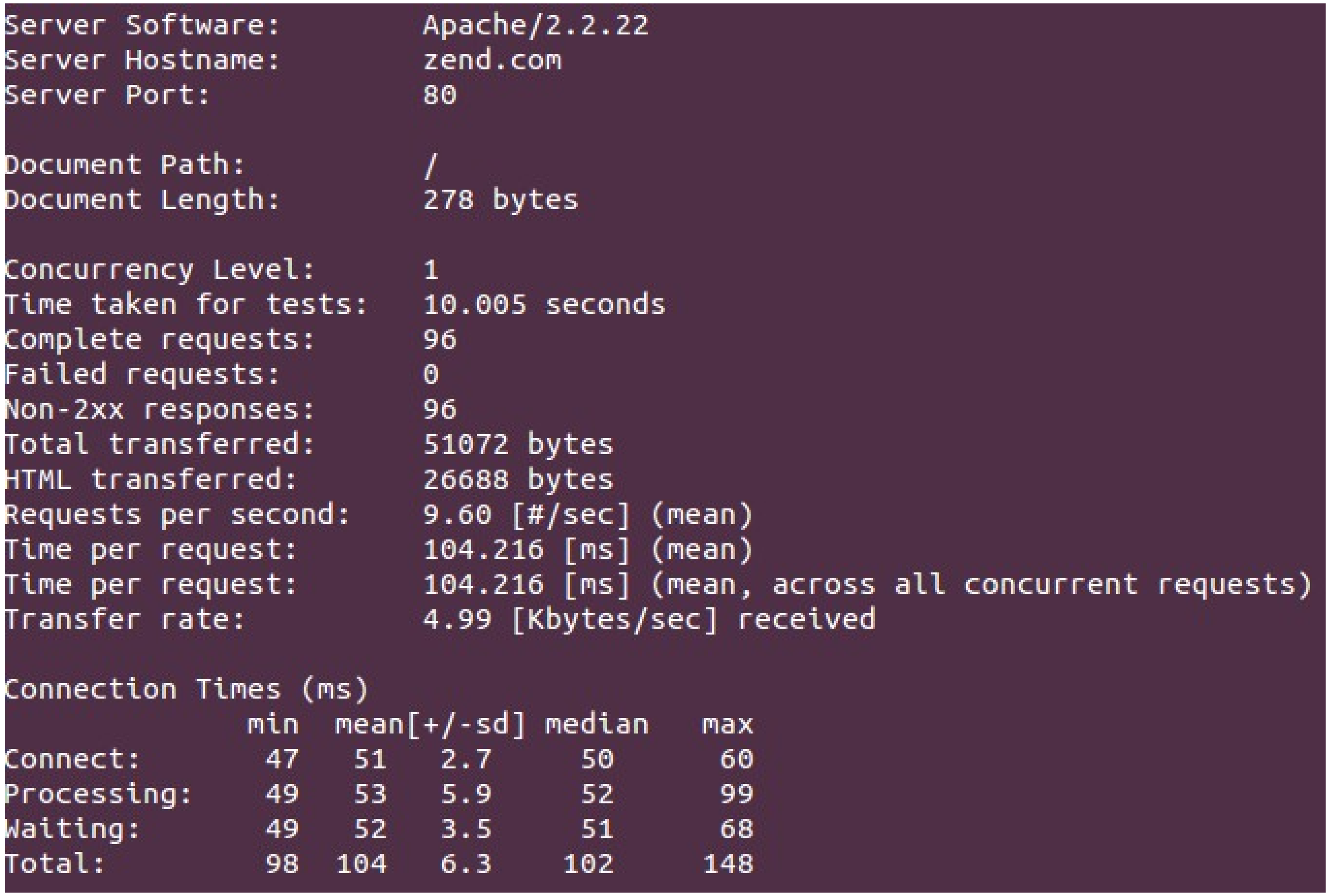
- **Tools**
  - PHPqatools.org
  - PHPMetrics
  - PHP_Codesniffer
  - Code Climate
  - Zend Server and Z-Ray
  - Apache Bench (apache2-utils in apt)
    - How much traffic can you handle?



```
$ ab -t 10 http://www.zend.com/
```

Does Your Code Measure Up?

- Apache Bench Result



```
Server Software:        Apache/2.2.22
Server Hostname:        zend.com
Server Port:            80

Document Path:          /
Document Length:        278 bytes

Concurrency Level:      1
Time taken for tests:   10.005 seconds
Complete requests:      96
Failed requests:        0
Non-2xx responses:      96
Total transferred:      51072 bytes
HTML transferred:       26688 bytes
Requests per second:    9.60 [#/sec] (mean)
Time per request:       104.216 [ms] (mean)
Time per request:       104.216 [ms] (mean, across all concurrent requests)
Transfer rate:          4.99 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       47   51   2.7     50      60
Processing:    49   53   5.9     52      99
Waiting:       49   52   3.5     51      68
Total:         98  104   6.3    102     148
```

Does Your Code Measure Up?

- **Conclusion**
  - Measure all the things!
  - Don't fear results, share them
  - Reduce complexity
  - Leave code cleaner than you got it
  - Learn to “smell” problems
  - Use refactoring to fix shortcomings
  - Love iteration!

- Thank you!
  - Code: https://github.com/adamculp/refactoring101
  - http://phpqatools.org
  - http://phpmetrics.org
  - http://zend.com
  - http://codeclimate.com
  - Please rate at: https://joind.in/13300

Adam Culp

http://www.geekyboy.com

Twitter @adamculp