

O'REILLY®

OSCON

open source convention



Reviewing PHP Code

Sebastian Bergmann
July 27th 2011



sharing experience



HELLO
my name is
Sebastian



Why so negative?



© 2001 Columbia Pictures Industries, Inc. All Rights Reserved.

“Your code has been weighed, it has been measured,
and it has been found wanting.”





sharing experience



Benefits of Code Review

- Improved code quality
- Fewer defects in code
- Improved communication about code
- Education of junior programmers





2 * 5 Ways of Reviewing Code

- Who
 - Internal
 - External
- How
 - Formal Inspection
 - Over-the-Shoulder
 - E-Mail Pass-Around
 - Pair Programming
 - Tool-Assisted





sharing experience

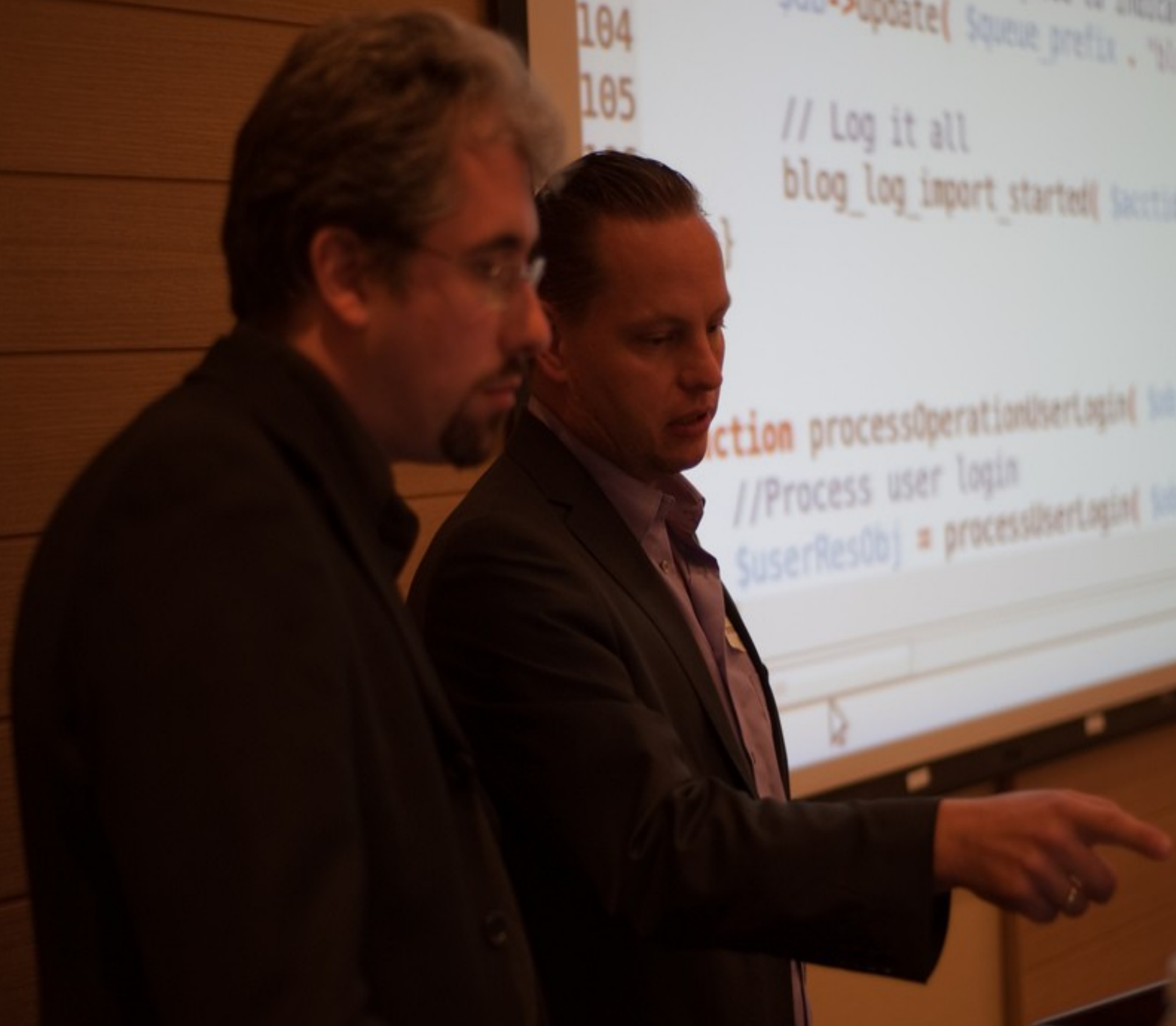


sharing experience



sharing experience

```
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```





sharing experience



sharing experience



sharing experience

Summary

- Formal inspections and pair programming are proven techniques but require large amounts of developer time.
- Over-the-Shoulder reviews are easiest to implement but are not enforceable.
- E-Mail pass-around and tool-assisted reviews strike a balance between time invested and ease of implementation.



READY...
SET...
GO



sharing experience

Where do we start?

- How much code is there?
- How is the code structured?
- Can we find "interesting" code using a tool?
- Which parts of the code are executed during a normal page request?
- Start reading the code.



Tools

- <http://phpqatools.org/>
- <http://gist.github.com/625915>
- <http://gist.github.com/625916>



Software Metrics: How much code is there?

```
> phploc src
phploc 1.6.1 by Sebastian Bergmann.

Directories:          5
Files:               21

Lines of Code (LOC): 482
  Cyclomatic Complexity / Lines of Code: 0.08
Executable Lines of Code (ELOC): 296
Comment Lines of Code (CLOC): 0
Non-Comment Lines of Code (NCLOC): 482

Namespaces:          0
Interfaces:          1
Classes:             19
  Abstract:           1 (5.26%)
  Concrete:           18 (94.74%)
Average Class Length (NCLOC): 21
Methods:             32
  Scope:
    Non-Static:       32 (100.00%)
    Static:           0 (0.00%)
  Visibility:
    Public:           21 (65.62%)
    Non-Public:       11 (34.38%)
Average Method Length (NCLOC): 12
Cyclomatic Complexity / Number of Methods: 1.66

Anonymous Functions: 1
Functions:            0

Constants:           0
  Global constants:  0
  Class constants:   0
```



Software Visualization: Overview Pyramid

```
> pdepend --overview-pyramid=overview_pyramid.svg src
```

```
PHP_Depend 0.10.5 by Manuel Pichler
```

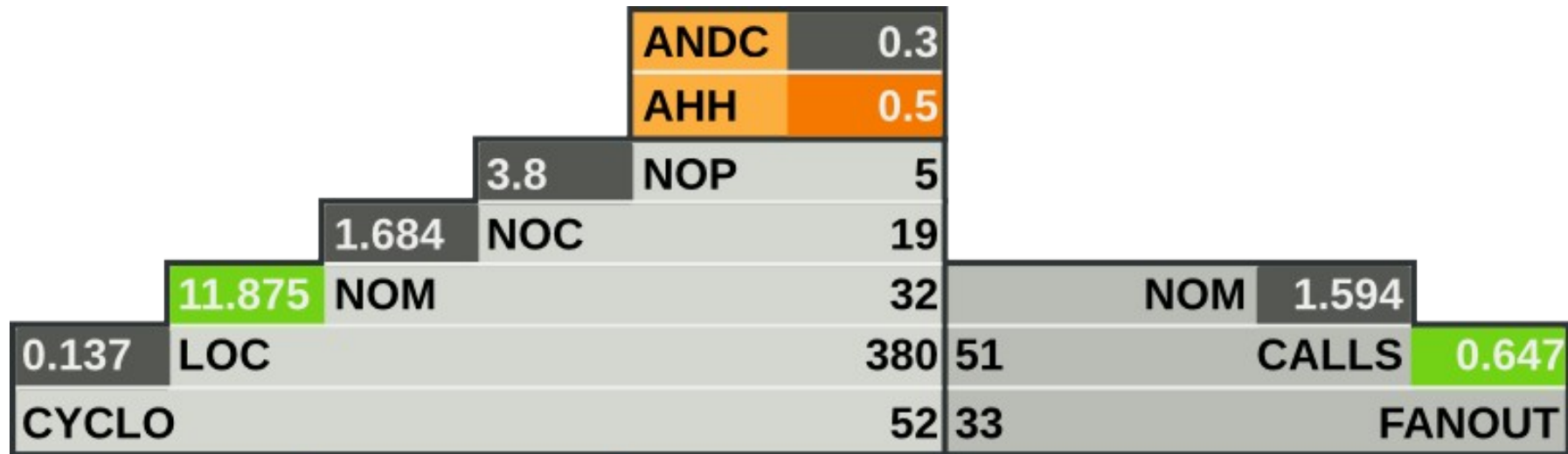
```
Parsing source files: 21
.....
Executing Coupling-Analyzer: 92
....
Executing CyclomaticComplexity-Analyzer: 89
....
Executing Inheritance-Analyzer: 29
.
Executing NodeCount-Analyzer: 57
..
Executing NodeLoc-Analyzer: 77
...
```

```
Generating pdepend log files, this may take a moment.
```

```
Time: 00:00; Memory: 8.75Mb
```



Software Visualization: Overview Pyramid



● Low ● Average ● High

Generated by PHP_Depend

- ANDC Average Number of Derived Classes
- AHH Average Hierarchy Height
- NOP Number of Packages
- NOC Number of Classes
- NOM Number of Methods
- LOC Lines of Code (non-comment, non-whitespace)
- CYCLO Cyclomatic Complexity
- CALLS Number of Operation Calls
- FANOUT Number of Called Classes



Software Visualization: Abstraction/Instability

```
> pdepend --jdepend-chart=abstraction_instability.svg src
```

```
PHP_Depend 0.10.5 by Manuel Pichler
```

```
Parsing source files:
```

```
.....
```

```
21
```

```
Executing Dependency-Analyzer:
```

```
..
```

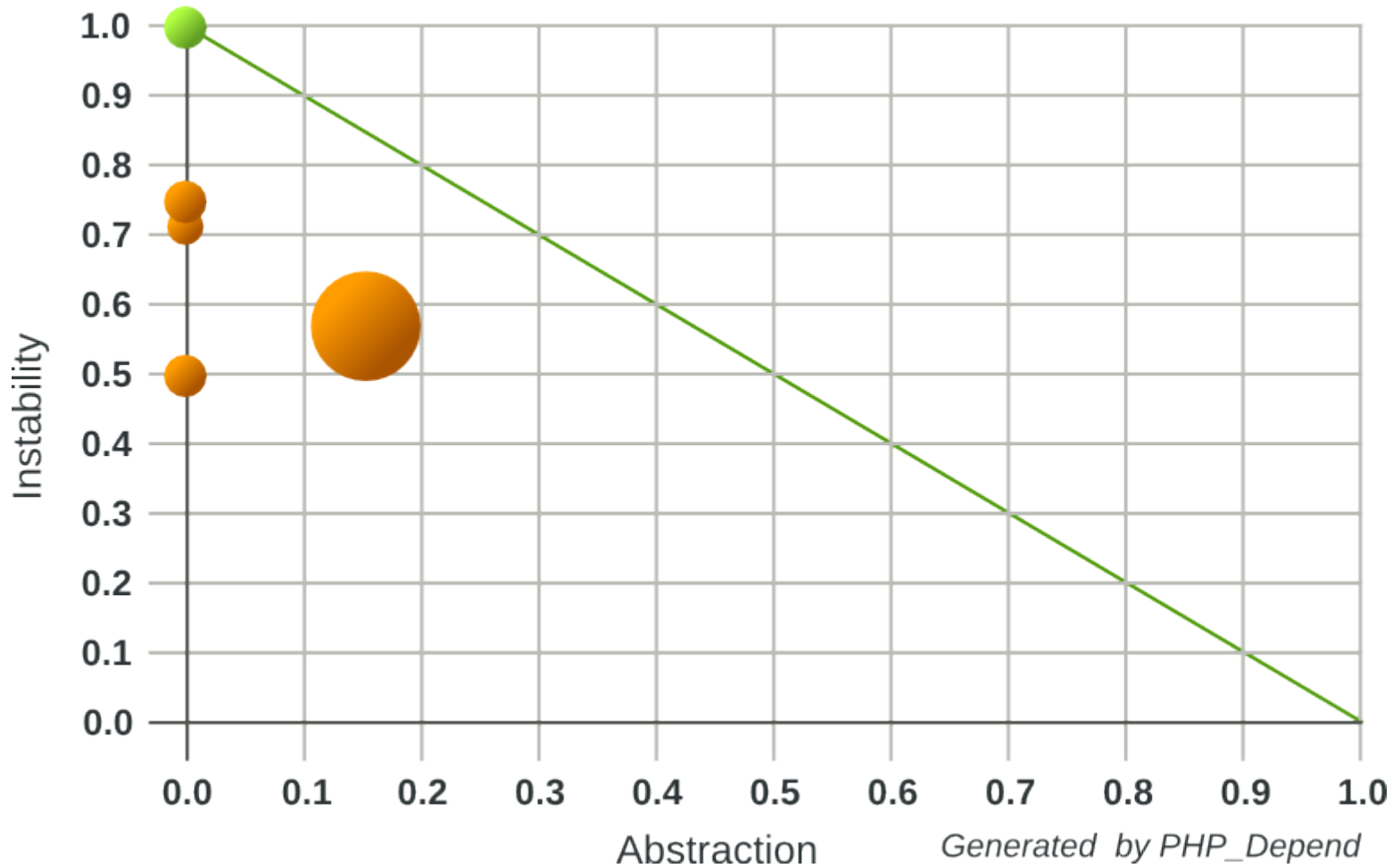
```
57
```

```
Generating pdepend log files, this may take a moment.
```

```
Time: 00:01; Memory: 8.00Mb
```



Software Visualization: Abstraction/Instability



Software Visualization: Dependencies

```
> pdepend --jdepend-xml=jdepend.xml src
```

```
PHP_Depend 0.10.5 by Manuel Pichler
```

```
Parsing source files:
```

```
.....
```

21

```
Executing Dependency-Analyzer:
```

```
..
```

57

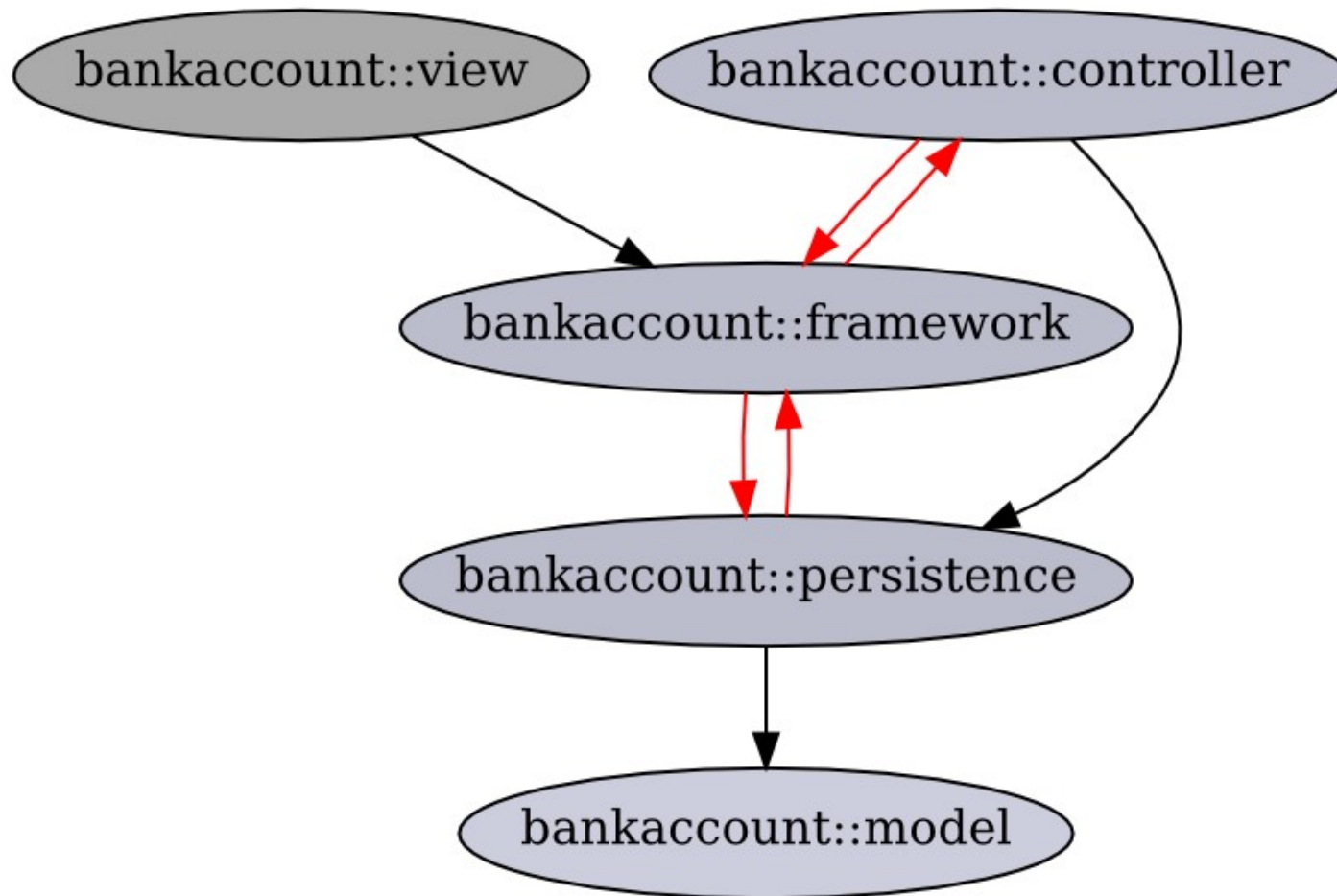
```
Generating pdepend log files, this may take a moment.
```

```
Time: 00:00; Memory: 8.00Mb
```

```
> dependencies.php jdepend.xml -o dependencies.svg
```



Software Visualization: Dependencies



Software Metrics

```
> pdepend --summary-xml=summary.xml src
PHP_Depend 0.10.5 by Manuel Pichler

Parsing source files:
..... 21

Executing CyclomaticComplexity-Analyzer:
..... 89

Executing ClassLevel-Analyzer:
... 70

Executing CodeRank-Analyzer:
. 25

Executing Coupling-Analyzer:
..... 92

Executing Hierarchy-Analyzer:
... 72

Executing Inheritance-Analyzer:
. 29

Executing NPathComplexity-Analyzer:
..... 89

Executing NodeCount-Analyzer:
.. 57

Executing NodeLoc-Analyzer:
... 77

Generating pdepend log files, this may take a moment.
Time: 00:00; Memory: 9.00Mb
```



Software Metrics: Cyclomatic Complexity

Number of possible decision paths in a program or program unit

Thomas J. McCabe, "A Complexity Measure," IEEE Transactions on Software Engineering 2, No. 4 (IEEE Computer Society Press, Los Alamitos, CA, USA, 1976).



Software Metrics: Cyclomatic Complexity

```
> inspect.php summary.xml --metric@ ccn
```

Name	Value
BankAccountMapper::findById()	4.0000
Router::route()	4.0000
BankAccountController::execute()	3.0000
Request::__call()	3.0000
ControllerFactory::getController()	3.0000
BankAccount::setBalance()	2.0000
MapperFactory::getMapper()	2.0000
BankAccountMapper::getAllIds()	2.0000
BankAccountMapper::insert()	2.0000
BankAccountMapper::delete()	2.0000
BankAccountMapper::update()	2.0000
BankAccountListView::render()	2.0000
HashMap::get()	2.0000
BankAccount::depositMoney()	1.0000
.	
.	
.	



Software Metrics: Cyclomatic Complexity

```
class Router extends HashMap {  
1  public function route(Request $request) {  
    $parts = explode('/', $request->getServer('REQUEST_URI'));  
    unset($parts[0]);  
    $controller = array_shift($parts);  
  
2    if ( !isset($this->values[$controller])  
3        || count($parts) % 2 != 0) {  
        throw new RouterException;  
    }  
  
    $keys = array_keys($parts);  
    $count = count($keys);  
  
4    for ($i = 0; $i < $count; $i += 2) {  
        $request->set($parts[$keys[$i]], $parts[$keys[$i+1]]);  
    }  
  
    return $this->values[$controller];  
    }  
}
```



Software Metrics: NPath Complexity

Number of acyclic execution paths in a program or program unit

Brian A. Nejmeh, "NPATH: A Measure of Execution Path Complexity and its Applications," Communications of the ACM 31, Issue 2 (February 1988): 188–200. ISSN 0001-0782.



Software Metrics: NPath Complexity

```
> inspect.php summary.xml --metric0 npath
```

Name	Value
Router::route()	8.0000
Request::__call()	6.0000
BankAccountMapper::findById()	6.0000
BankAccountController::execute()	4.0000
ControllerFactory::getController()	3.0000
BankAccountMapper::getAllIds()	2.0000
BankAccountListView::render()	2.0000
BankAccount::setBalance()	2.0000
MapperFactory::getMapper()	2.0000
BankAccountMapper::update()	2.0000
BankAccountMapper::delete()	2.0000
BankAccountMapper::insert()	2.0000
HashMap::get()	2.0000
BankAccount::withdrawMoney()	1.0000
.	
.	
.	

Software Metrics: NPath Complexity

```
class Router extends HashMap {
1  public function route(Request $request) {
    $parts = explode('/', $request->getServer('REQUEST_URI'));
    unset($parts[0]);
    $controller = array_shift($parts);

2    if ( !isset($this->values[$controller])
4      || count($parts) % 2 != 0) {
        throw new RouterException;
    }

    $keys = array_keys($parts);
    $count = count($keys);

8    for ($i = 0; $i < $count; $i += 2) {
        $request->set($parts[$keys[$i]], $parts[$keys[$i+1]]);
    }

    return $this->values[$controller];
  }
}
```



Software Metrics: Code Rank

Application of Google's PageRank algorithm to software

Blair Neate and Warwick Irwin and Neville Churcher, "CodeRank: A New Family of Software Metrics", Proceedings of the Australian Software Engineering Conference 2006: 369–378. ISBN 0-7695-2551-2.



Software Metrics: Code Rank

```
> inspect.php summary.xml --metric0 cr --type class
```

Name	Value
HashMap	0.5325
View	0.4050
BankAccountController	0.2775
BankAccount	0.1500
ViewFactory	0.1500
BankAccountException	0.1500
BankAccountView	0.1500
BankAccountListView	0.1500
BankAccountMapper	0.1500
RouterException	0.1500
Response	0.1500
ControllerFactory	0.1500
ControllerException	0.1500
BankAccountListController	0.1500
FrontController	0.1500
MapperException	0.1500
Request	0.1500
MapperFactory	0.1500
Router	0.1500



Software Metrics: Reverse Code Rank

```
> inspect.php summary.xml --metric@ rcr --type class
```

Name	Value
BankAccountListController	0.3859
BankAccountController	0.2775
BankAccountListView	0.2137
BankAccountView	0.2137
Request	0.1925
Router	0.1925
Response	0.1925
RouterException	0.1819
BankAccountException	0.1819
ControllerException	0.1819
MapperException	0.1819
BankAccountMapper	0.1500
ControllerFactory	0.1500
BankAccount	0.1500
View	0.1500
HashMap	0.1500
FrontController	0.1500
MapperFactory	0.1500
ViewFactory	0.1500



Detecting Rule Violations (with PHPMD)

```
> phpmd wordpress/wp-includes text phpmd.xml
```

```
wp-includes/admin-bar.php:120 Avoid unused local variables such as '$wpdb'.
wp-includes/admin-bar.php:176 The function wp_admin_bar_edit_menu() has a Cyclomatic Complexity of 22.
wp-includes/admin-bar.php:239 The function wp_admin_bar_new_content_menu() has a Cyclomatic Complexity of 13.
wp-includes/admin-bar.php:239 The function wp_admin_bar_new_content_menu() has an NPath complexity of 1296.
wp-includes/admin-bar.php:294 The function wp_admin_bar_appearance_menu() has a Cyclomatic Complexity of 10.
wp-includes/admin-bar.php:294 The function wp_admin_bar_appearance_menu() has an NPath complexity of 288.
wp-includes/admin-bar.php:325 The function wp_admin_bar_updates_menu() has an NPath complexity of 640.
wp-includes/atomlib.php:65 The class AtomParser has an overall complexity of 61 which is very high.
wp-includes/atomlib.php:90 Classes should not have a constructor method with the same name as the class
wp-includes/atomlib.php:104 Avoid unused parameters such as '$log_level'.
wp-includes/atomlib.php:104 Avoid unused parameters such as '$error_file'.
wp-includes/atomlib.php:104 Avoid unused parameters such as '$error_line'.
wp-includes/atomlib.php:149 The method start_element() has a Cyclomatic Complexity of 21.
wp-includes/atomlib.php:149 The method start_element() has an NPath complexity of 204.
wp-includes/atomlib.php:149 Avoid unused parameters such as '$parser'.
wp-includes/atomlib.php:228 The method end_element() has a Cyclomatic Complexity of 12.
wp-includes/atomlib.php:228 Avoid unused parameters such as '$parser'.
wp-includes/atomlib.php:282 Avoid unused parameters such as '$parser'.
wp-includes/atomlib.php:287 Avoid unused parameters such as '$parser'.
wp-includes/atomlib.php:291 Avoid unused parameters such as '$parser'.
wp-includes/atomlib.php:298 Avoid unused parameters such as '$parser'.
wp-includes/atomlib.php:298 Avoid unused parameters such as '$data'.
wp-includes/atomlib.php:303 The method ns_to_prefix() has a Cyclomatic Complexity of 10.
.
.
.
```



PHPMD Rule Set Configuration

```
<?xml version="1.0"?>  
  
<ruleset name="my-rule-set">  
  <description>My Rule Set for PHPMD</description>  
  
  <rule ref="rulesets/codesize.xml/CyclomaticComplexity" />  
  <!-- ... -->  
</ruleset>
```



Detecting Rule Violations (with PHP_CodeSniffer)

```
> phpcs --standard=CodeReview --report-source wordpress
```

```
PHP CODE SNIFFER VIOLATION SOURCE SUMMARY
```

STANDARD	CATEGORY	SNIFF	COUNT
CodeRevi	Functions	Global function found	2297
CodeRevi	PHP	Global keyword not allowed	938
Generic	PHP	No silenced errors discouraged	523
CodeRevi	PHP	Forbidden functions found	77
Generic	Code analysis	For loop with test function call not allowe	53
Generic	Code analysis	Empty statement not allowed warning	34
Generic	PHP	Deprecated functions found	28
Generic	Code analysis	Useless overriding method found	4
Generic	Classes	Duplicate class name found	2
Generic	Code analysis	Unconditional if statement found	1

```
A TOTAL OF 3957 SNIFF VIOLATION(S) WERE FOUND IN 10 SOURCE(S)
```

```
Time: 08:02, Memory: 1750.25Mb
```



Detecting Rule Violations (with PHP_CodeSniffer)

```
> phpcs --standard=CodeReview wordpress
```

```
FILE: /tmp/wordpress/wp-includes/admin-bar.php
```

```
-----  
FOUND 7 ERROR(S) AND 16 WARNING(S) AFFECTING 23 LINE(S)  
-----
```

```
18 | WARNING | Consider refactoring "_wp_admin_bar_init" to avoid global  
   |         | functions.  
19 | ERROR   | Use of the "global" keyword is forbidden  
52 | WARNING | Consider refactoring "wp_admin_bar_render" to avoid global  
   |         | functions.  
53 | ERROR   | Use of the "global" keyword is forbidden  
78 | WARNING | Consider refactoring "wp_admin_bar_my_account_menu" to avoid  
   |         | global functions.  
79 | ERROR   | Use of the "global" keyword is forbidden  
101| WARNING | Consider refactoring "wp_admin_bar_dashboard_view_site_menu"  
   |         | to avoid global functions.  
119| WARNING | Consider refactoring "wp_admin_bar_my_sites_menu" to avoid  
   |         | global functions.  
120| ERROR   | Use of the "global" keyword is forbidden  
154| WARNING | Consider refactoring "wp_admin_bar_shortlink_menu" to avoid  
   |         | global functions.  
176| WARNING | Consider refactoring "wp_admin_bar_edit_menu" to avoid global  
   |         | functions.
```

```
.  
.  
.
```



PHP_CodeSniffer Rule Set Configuration

```
<?xml version="1.0"?>  
  
<ruleset name="my-rule-set">  
  <description>My Rule Set for PHP_CodeSniffer</description>  
  
  <rule ref="Generic.PHP.DeprecatedFunctions" />  
  <!-- ... -->  
</ruleset>
```



Detecting Code Duplication (with phpcpd)

```
> phpcpd wordpress-3.2.1  
phpcpd 1.3.2 by Sebastian Bergmann.
```

```
Found 26 exact clones with 459 duplicated lines in 4 files:
```

- wp-content/plugins/akismet/admin.php:488-500
wp-content/plugins/akismet/admin.php:537-549
- wp-content/plugins/akismet/legacy.php:234-248
wp-content/plugins/akismet/legacy.php:301-315
- .
- .
- .
- wp-includes/class-snoopy.php:165-172
wp-includes/class-snoopy.php:225-232
- wp-includes/class-snoopy.php:181-187
wp-includes/class-snoopy.php:339-345
- wp-includes/class-snoopy.php:317-331
wp-includes/class-snoopy.php:384-398

```
0.27% duplicated lines out of 171170 total lines of code.
```

```
Time: 5 seconds, Memory: 73.25Mb
```



Disassembling Bytecode (with bytekit-cli)

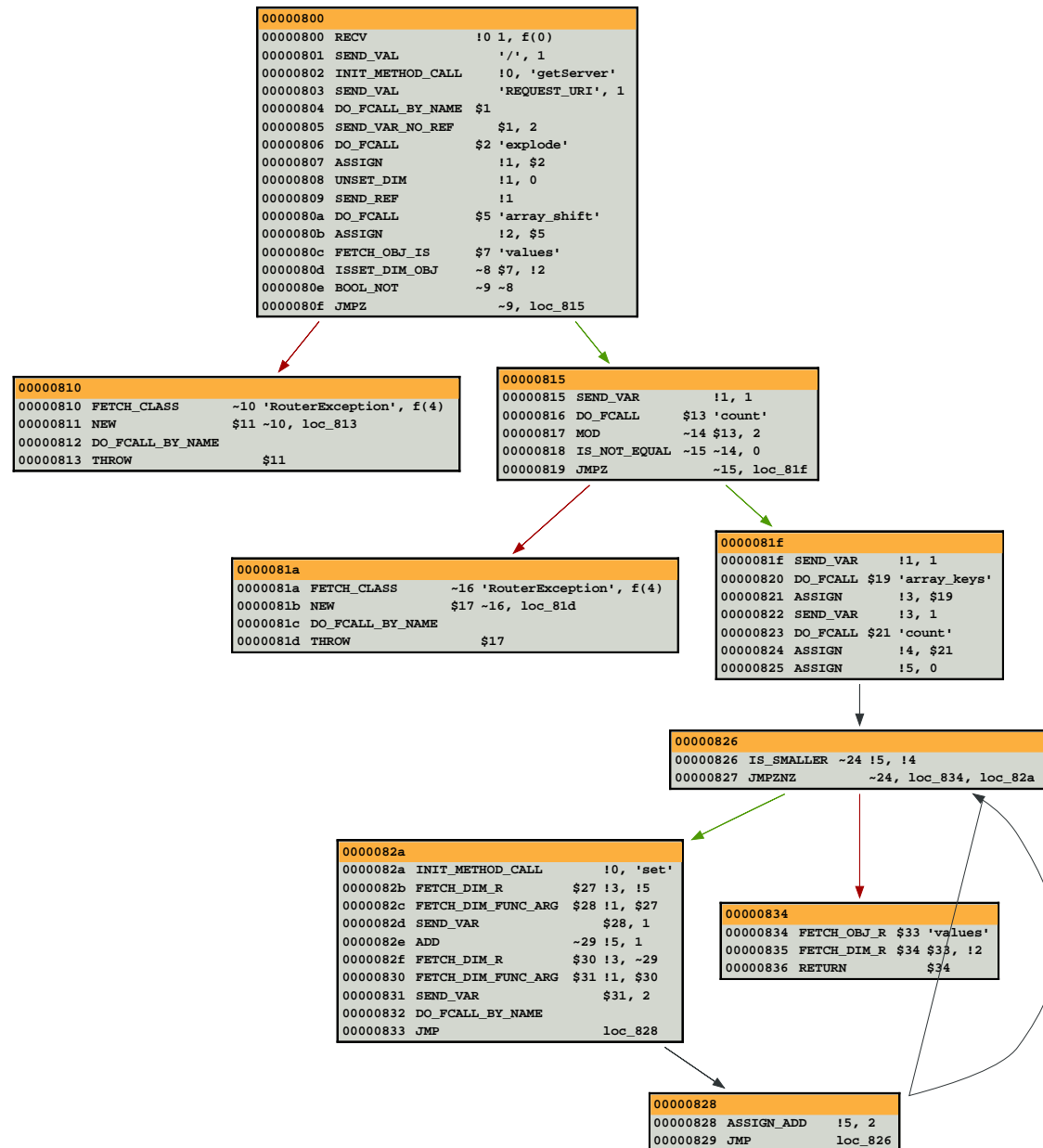
```
> bytekit src/framework/Router.php
Bytekit-cli 1.1.1 by Sebastian Bergmann.
```

```
Filename:      /usr/local/src/bankaccount/src/framework/Router.php
Function:      Router::route
Number of opcodes: 56
Compiled variables: !0 = $request, !1 = $parts, !2 = $controller, !3 = $keys, !4 = $count, !5 = $i
```

line #	opcode	result	operands
8	0	RECV	!0 1, f(0)
10	1	SEND_VAL	'/', 1
	2	INIT_METHOD_CALL	!0, 'getServer'
	3	SEND_VAL	'REQUEST_URI', 1
	4	DO_FCALL_BY_NAME	\$1
	5	SEND_VAR_NO_REF	\$1, 2
	6	DO_FCALL	\$2 'explode'
	7	ASSIGN	!1, \$2
11	8	UNSET_DIM	!1, 0
13	9	SEND_REF	!1
	10	DO_FCALL	\$5 'array_shift'
	11	ASSIGN	!2, \$5
15	12	FETCH_OBJ_IS	\$7 'values'
	13	ISSET_DIM_OBJ	~8 \$7, !2
	14	BOOL_NOT	~9 ~8
	15	JMPZ	~9, ->21
16	16	FETCH_CLASS	~10 'RouterException', f(4)
	17	NEW	\$11 ~10, ->19
	18	DO_FCALL_BY_NAME	
	19	THROW	\$11
17	20	JMP	->21
	:		
	:		



Software Visualization: Code Flow



The End

- Web: <http://thePHP.cc/>
<http://sebastian-bergmann.de/>
- Mail: sebastian@thePHP.cc
- Twitter: [@s_bergmann](https://twitter.com/s_bergmann)
- Slides: <http://talks.thePHP.cc/>

