Quality assurance in PHP

Thomas Koch

YMC AG

October 2, 2009

Introduction

PHP

History of QA in PHP

PHP QA in Teams

Outline

me

- First programming 1990, Commodore 64
- music, physics, computer science
- PHP since 2006

YMC AG

- 2001: Gründung als Web-Servicedienstleister mit Open-Source-Fokus
- Partnerschaftlich organisiert
- Mitte 2003: Spezialisierung auf eZ Publish und Open Source
- 14 Mitarbeiter
- PHP, content management, eZ Publish Partner

you?

- Entwickler?
- Programmiersprachen? Java / .Net / C, C++ / PHP / Python / Ruby
- Webanwendungen?
- Teamgröße?
- Komplette QA Infrastruktur?

Outline

PHP

Was denken Sie über PHP?

Kategorisierung

- Skriptsprache
- dynamische typisierung
- domain language: web
- “curly braces” Sprache
- seit PHP5: objektorientiert

Charakterisierung

schnell und billig

Charakterisierung

sex, drugs, PHP

Charakterisierung

- schnell erlernbar
- schnell erste Ergebnisse
- weit verbreitet
- viele Programmierer
- 3. populärste Sprache (tiobe)
- viel PHP software (variierender Qualität . . .)

PHP Popularität

Position Sep 2009	Position Sep 2008	Delta in Position	Programming Language	Ratings Sep 2009	Delta Sep 2008	Status
1	1	=	Java	19.383%	-1.33%	A
2	2	=	C	16.861%	+1.48%	A
3	5	↑↑	PHP	10.156%	+0.91%	A
4	3	↓	C++	9.988%	-0.73%	A
5	4	↓	(Visual) Basic	9.196%	-1.29%	A
6	7	↑	Perl	4.528%	-0.31%	A
7	8	↑	C#	4.186%	-0.15%	A
8	6	↓↓	Python	3.930%	-1.08%	A
9	9	=	JavaScript	2.995%	-0.14%	A
10	11	↑	Ruby	2.377%	-0.38%	A

quality assurance

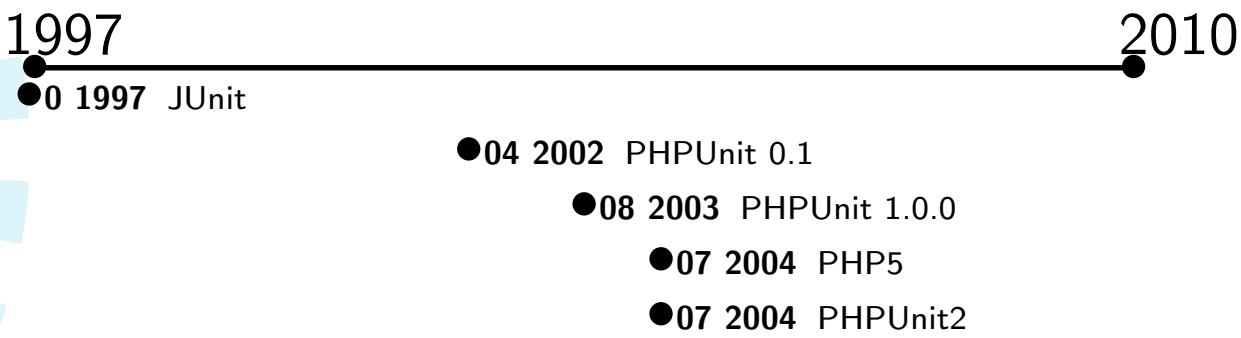
- kostet (anfangs) Zeit
- erfordert geschulte Entwickler
- erfordert Infrastruktur

PHP versus QA

- schnell fertig vs. sorgfältig testen
- billige Entwickler vs. hohe Anforderung
- kleine Firmen vs. komplizierte Infrastruktur

Outline

The dark age of PHP4



PHPUnit

- Eins unter 8 Unit Test Frameworks für PHP
- De-Facto Standard für UnitTest in PHP
- Inspiriert durch JUnit, TestNG
- Hauptentwickler Sebastian Bergmann

Sebastian Bergmann

- Studium in Bonn
- Diplomarbeit bei eZ Systems
- Workflow Engine und Unit Test Runner für eZ Components
- PHP Kernentwickler
- Berater, Ausbilder

PHPUnit Beispiel, Prüfling

```php
class BitSet {
 private $value = 0;

 public function setByPosition($pos, $bool)
 {
  $bit = 1 << $pos;

  // reset the bit
  $this->value = $this->value & ~$bit;

  if( $bool ) {
   $this->value = $this->value | $bit;
  }
 }
}
```

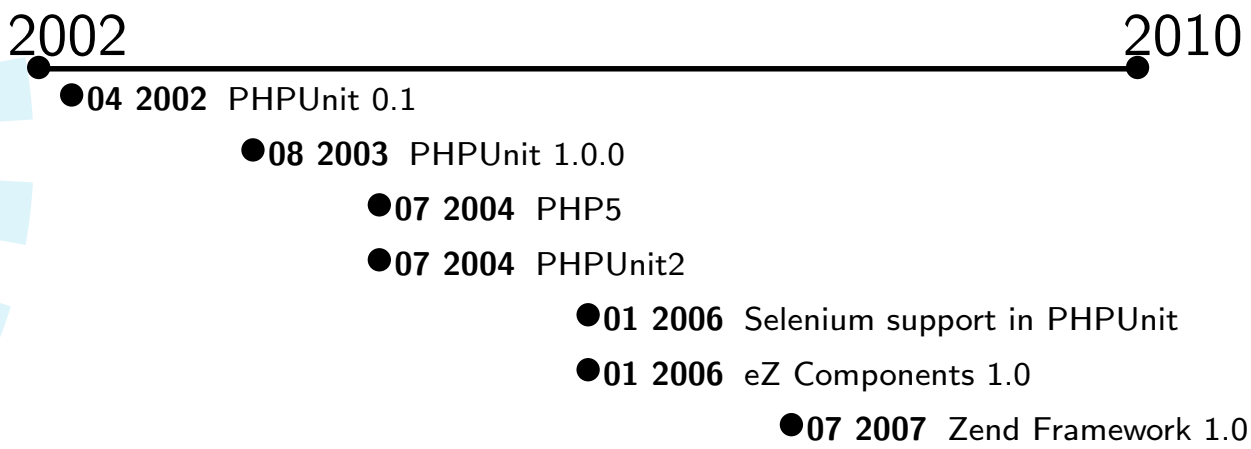
PHPUnit Beispiel, TestKlasse

```
1  class BitSetTest
2        extends PHPUnit_Framework_TestCase
3  {
4   public function testFirstPosMappedToOne()
5   {
6    $bitset = new BitSet;
7
8    $bitset[0] = TRUE;
9    $this->assertEquals(1,
10                       $bitset->getInteger());
11  }
12
13   ...
14 }
```

PHPUnit Beispiel, Asserts

```
1  public function testSetBitByPos()
2  {
3   $bitset = new BitSet;
4
5   $bitset[1] = TRUE;
6
7   $this->assertTrue( $bitset[1] );
8   $this->assertFalse( $bitset[2] );
9  }
```

PHP5


2002
2010
04 2002 PHPUnit 0.1
08 2003 PHPUnit 1.0.0
07 2004 PHP5
07 2004 PHPUnit2
01 2006 Selenium support in PHPUnit
01 2006 eZ Components 1.0
07 2007 Zend Framework 1.0

Selenium und PHPUnit – Key Features

- Runs functional tests in all major browsers
- Compile test scripts in native languages: Java, PHP, Python, Ruby, ...
- Integrate scripts with other test suites and continuous integrations systems.
- Established Project.
- Used and funded by Google.
- Tests can be written as PHPUnit tests

eZ Components

- Komponentenbibliothek für PHP content management
- erstes (?) großes test-driven-developed PHP projekt
- benutzt PHPUnit
- Sebastian Bergmann ist im Entwicklerteam
- Erfahrungen aus der eZ Components Entwicklung fließen zurück in PHPUnit

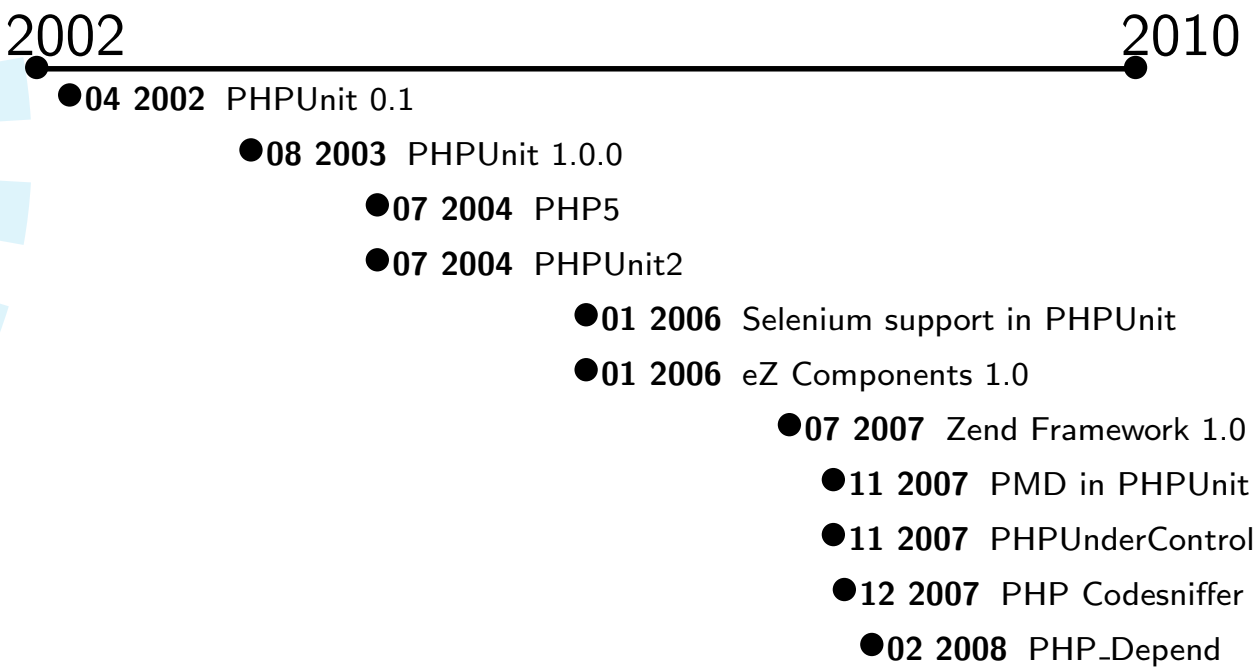
Zend Framework

- Die Idee von Qualitätsstandards in PHP verbreitet sich
- . . . leider wird Zend Framework den eigenen Ansprüchen nicht gerecht

"Durchgängig hohe Qualität" in Bezug auf das Zend Framework kann man meiner Ansicht nach keinesfalls behaupten. Ich sehe das Zend Framework eher als Einäugigen unter Blinden, von wenigen löblichen Ausnahmen wie den ezComponents abgesehen. Die Code Coverage der Zend Framework-Testsuite ist teilweise Makulatur, was bedeutet, dass das Framework längst nicht so gut getestet ist, wie das Marketing glauben macht.[1]

[1] Stefan Priebsch, `https://www.xing.com/net/php/php-q-a-6531/struktur-entwickler-workflow-24357744/p0`

Feature complete (?)



Project Mess Detection

- statische Codeanalyse zur Qualitätsmessung
- zuerst Bestandteil von PHPUnit
- später selbstständig als PHP_Depend und PHPMD
- Hauptentwickler Manuel Pichler, Dortmund
- PHPLOC von Sebastian Bergmann

Code Metriken

- Umfang
- Komplexität
- Kopplung & Abstraktion
- Lesbarkeit
- Laufzeiten
- Fehleranfälligkeit

Umfangsmetriken

- LOC - Lines Of Code
- ELOC - Executable Lines Of Code
- CLOC - Comment Lines Of Code
- NCLOC - None Comment Lines Of Code
- NOC - Number Of Classes
- NOM - Number Of Methods
- NOP - Number Of Packages

Komplexitätsmetriken

- Maß für Komplexität sind Kontrollstrukturen:
  if, elseif, for, while, foreach, catch, case, xor, and, or, ...
- Cyclomatic Complexity (CCN)
  = Anzahl der Verzweigungen
- NPath Complexity
  = Anzahl der Ausführungspfade
  Berücksichtigt die Struktur von Blöcken

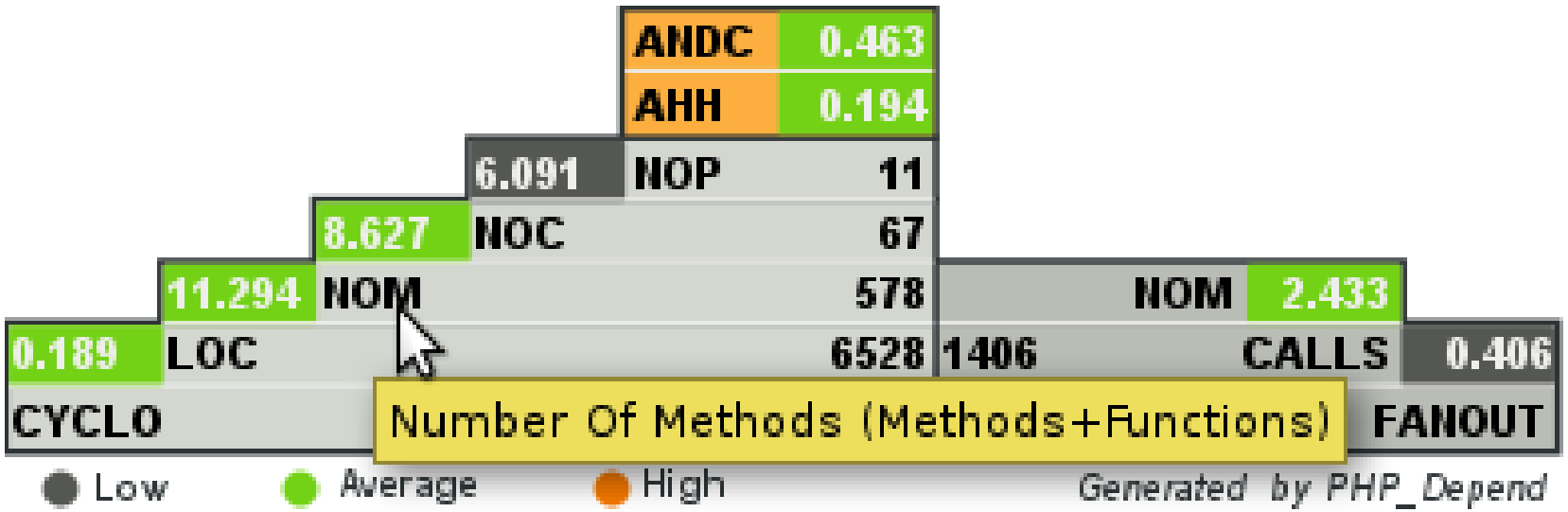
Objektorientierte Metriken (Auswahl)

- Depth of Inheritance Tree (DIT)
- Afferent Coupling / Efferent Coupling
- Code Rank (wie Page Rank)

Nutzen von Metriken?

- Messwerte müssen interpretiert werden
- Entwickler können jedes System betrügen
- zeigt Stellen, an denen refactoring notwendig ist
- kann mir helfen besser zu werden
- fremden Code schnell beurteilen (große PHP codebase)
- . . . sollte übersichtlich aufbereitet werden . . .

PHP Depend Pyramide



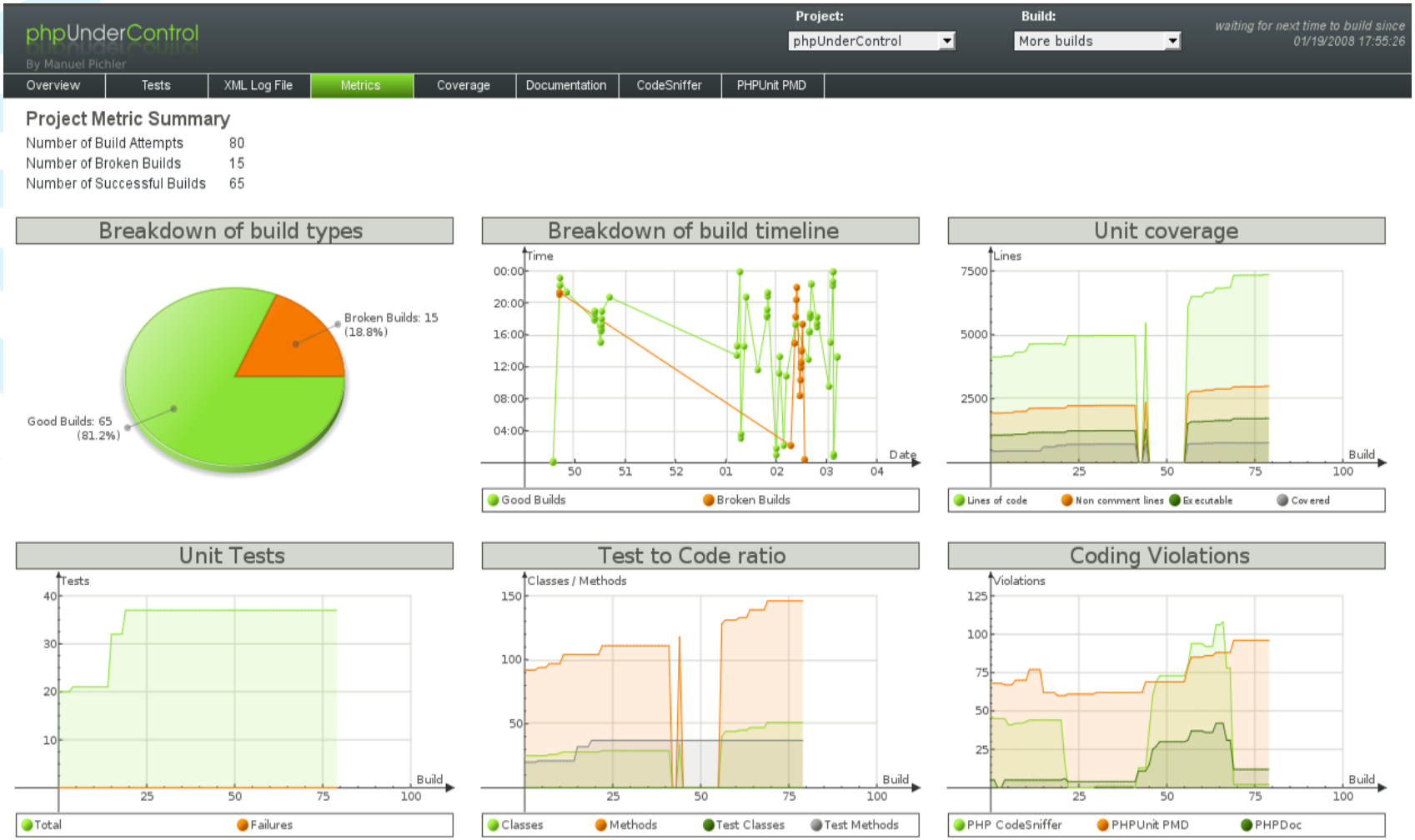
PHP CodeSniffer

- erlaubt die automatische Überprüfung von coding conventions
- gibt checkstyle XML aus
- teilweise überschneidung mit PMD

PHP CodeSniffer - Anwendungen

- oft wechselndes Team
- deprecating von Funktionen
- Reduzierung der Möglichkeiten von PHP, schlechten Code zu schreiben
- coding style: Klammersetzung, Einrückung, Kapitalisierung

PHPUnderControl



PHPUnderControl pmd summary

phpUnderControl
By Manuel Pichler

Project: php-under-control | Build: More builds

waiting for next time to build since 11/23/2007 22:46:58

Overview | Tests | XML Log File | Metrics | Coverage | Documentation | CodeSniffer | PHPUnit PMD

PHPUnit PMD Summary

Files: 1
Violations: 7

PHPUnit PMD rule	Files	Error/Warnings
/ NPathComplexity	1	2
/ CodeCoverage	1	2
/ CopyPasteDetection	2	1

Math.php (6)

90	The CRAP index is 132. The Change Risk Analysis and Predictions (CRAP) index of a function or method uses cyclomatic complexity and code coverage from automated tests to help estimate the effort and risk associated with maintaining legacy code. A CRAP index over 30 is a good indicator of crappy code.
90	The NPath complexity is 83520. The NPath complexity of a function or method is the number of acyclic execution paths through that method. A threshold of 200 is generally considered the point where measures should be taken to reduce complexity.
90	The code coverage is 0.00 which is considered low.
154	The CRAP index is 132. The Change Risk Analysis and Predictions (CRAP) index of a function or method uses cyclomatic complexity and code coverage from automated tests to help estimate the effort and risk associated with maintaining legacy code. A CRAP index over 30 is a good indicator of crappy code.
154	The NPath complexity is 83520. The NPath complexity of a function or method is the number of acyclic execution paths through that method. A threshold of 200 is generally considered the point where measures should be taken to reduce complexity.
154	The code coverage is 0.00 which is considered low.

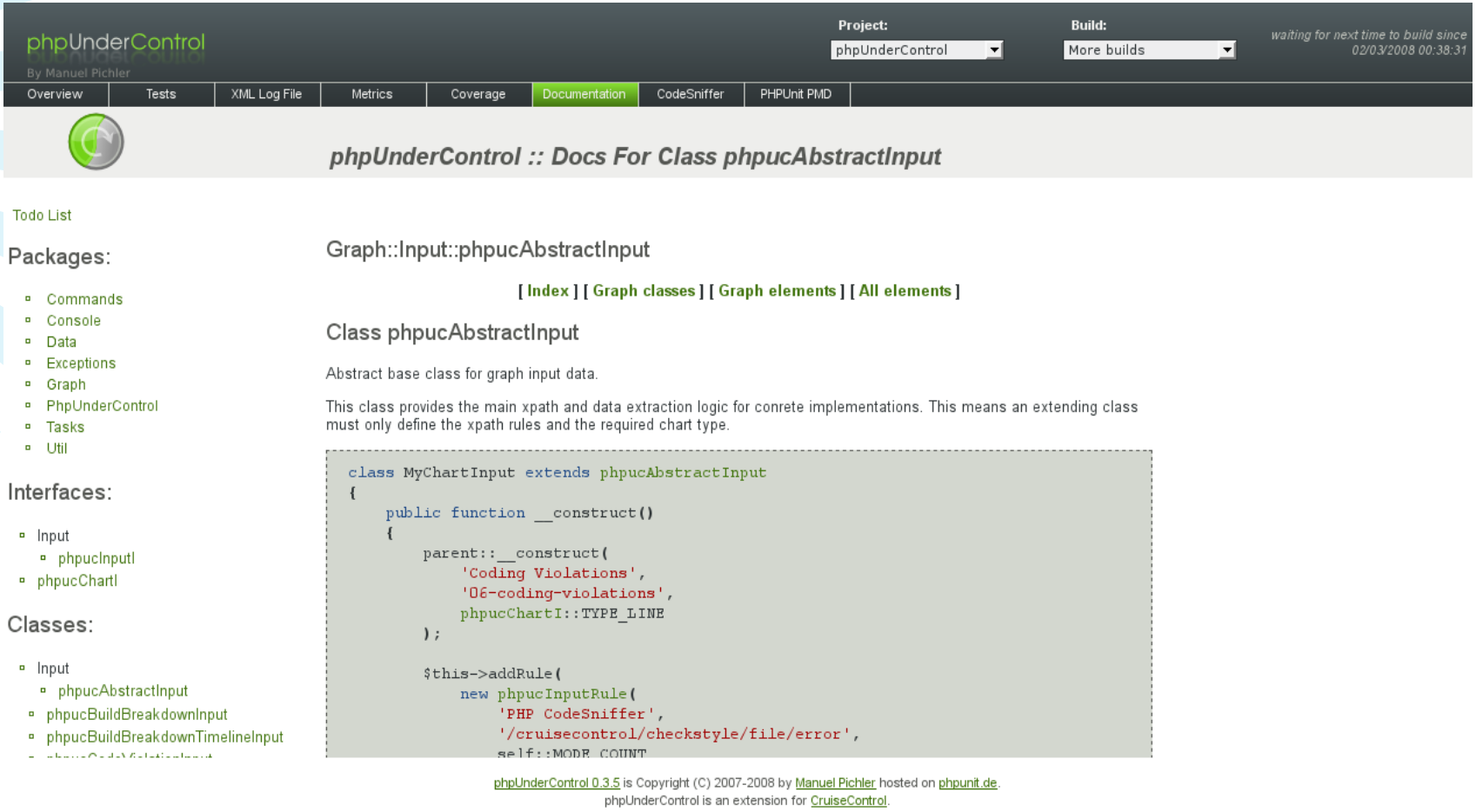
Duplication (Files: 2, Lines: 28, Tokens: 68)

90	/opt/cruisecontrol-bin-2.7/projects/php-under-control/source/src/Math.php
154	/opt/cruisecontrol-bin-2.7/projects/php-under-control/source/src/Math.php

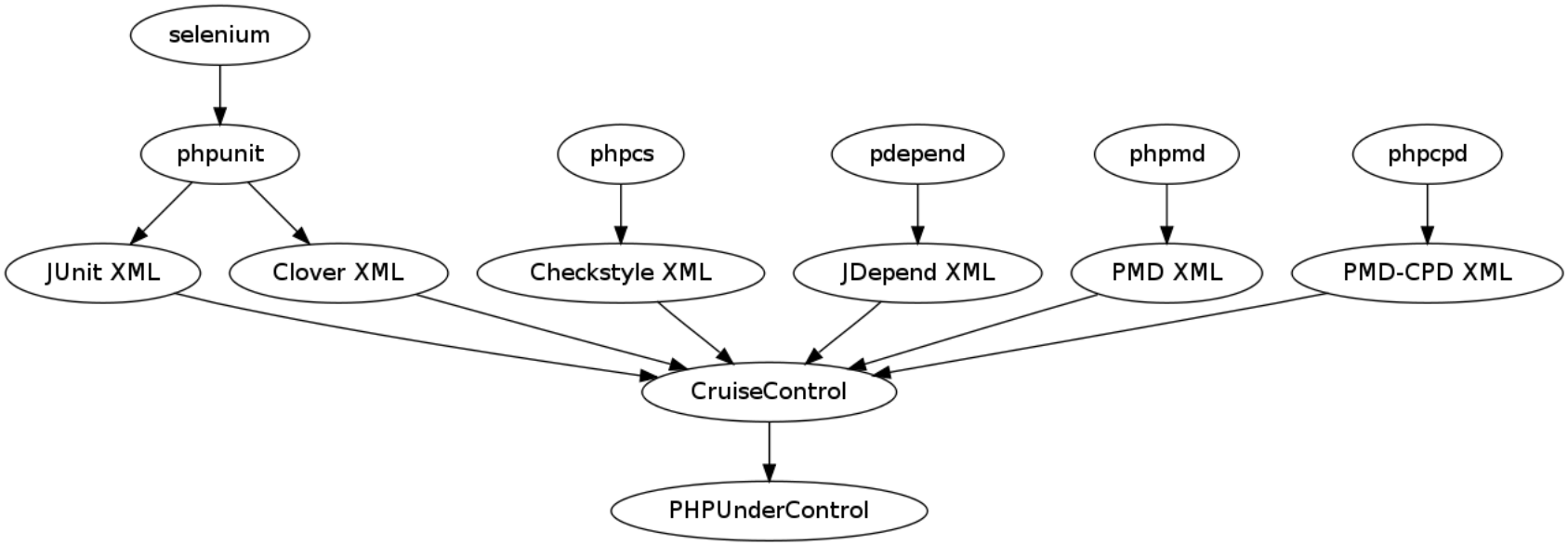
view plain print ?

```
01.  public function div($v1, $v2)
02.  {
03.      $v3 = $v1 / ($v2 + $v1);
04.      if ($v3 > 14)
05.      {
06.          $v4 = 0;
07.          for ($i = 0; $i < $v3; $i++)
08.          {
09.              $v4 += ($v2 * $i);
```

PHPUnderControl phpDocumentor



PHPUnderControl summary



Overview

2002 — 2010

- **04 2002** PHPUnit 0.1
- **08 2003** PHPUnit 1.0.0
- **07 2004** PHP5
- **07 2004** PHPUnit2
- **01 2006** Selenium support in PHPUnit
- **01 2006** eZ Components 1.0
- **07 2007** Zend Framework 1.0
- **11 2007** PMD in PHPUnit
- **11 2007** PHPUnderControl
- **02 2008** PHP_Depend
- **03 2009** thePHP.cc
- **03 2009** quality assurancein phpprojects.com

thePHP.cc
the php consulting company

What | Who | Where | Contact

Who

Sebastian Bergmann

Sebastian Bergmann holds a degree in computer science and is a long-time contributor to various PHP projects, including PHP itself. He is the creator of PHPUnit and other Open Source tools that help developers develop better software. As a Co-Founder and Principal Consultant with thePHP.cc, Sebastian Bergmann offers consulting, training, and coaching services to help enterprises improve the quality assurance process for their PHP-based software projects.

Arne Blankerts

Arne Blankerts is a Co-Founder and Principal Consultant with thePHP.cc and head of development at NonFood advertising agency in Hamburg, Germany. Furthermore he offers security consulting and training with thePHP.cc, is a regular author writing articles for the German PHP Magazine, php|architekt and others. Arne is the inventor and lead developer of an open source site system called "fCMS" (http://fcms.de) and in the unusual case of spare time he helps maintaining the German translation of the PHP manual.

Stefan Priebsch

Stefan Priebsch is a Co-Founder and Principal Consultant with thePHP.cc. He holds a degree in computer science and is author of various books and technical articles. As a consultant, he helps customers make better use of PHP, with a focus on software architecture, design patterns, and tools and methods. He served on the advisory board of International PHP Conference in 2008.

Outline

wie führe ich PHP QA ein?

z.B. Webtreff

- 14-tägiges Treffen von Webentwicklern
- ursprünglich in den Räumen von YMC
- jetzt in einer Kneipe (wie die Dortmunder)
- jeder stellt vor, was ihm wichtig ist
- Ziel: Die Entwickler sollen es cool finden

virale Einführung

- Manche Entwickler benutzen PHPQA tools in ihren Projekten
- Wer in diesen Projekten mitarbeitet, benutzt auch diese Tools
- auch für Versionskontrollsysteme! SVN $\rightarrow$ GIT

Bibliothekscode vs. Präsentationscode

- Präsentationscode geht durch viele Iterationen
- UnitTesting wäre dann overhead
- Umso stabiler der Code wird, umso höher die Abdeckung durch PHPQA
- Bibliothekscode von Anfang an testgetrieben und “sauber” entwickeln