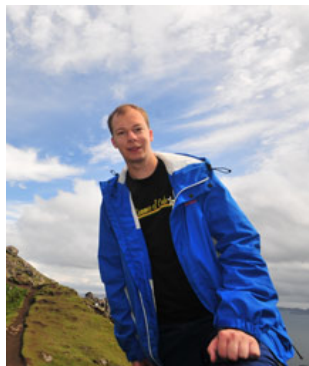


Improving Your Development Process

CodeWorks 2009 - Washington DC, US

Derick Rethans - dr@ez.no - twitter: @derickr

<http://derickrethans.nl/talks.php> — <http://joind.in/810>



Derick Rethans

derick@php.net

- Dutchman living in London
- Project lead for eZ Components at eZ Systems A.S.
- PHP development
- Author of the mcrypt, input_filter and date/time extensions
- Author of Xdebug

- Developer? Team lead? Something else?
- Size per team: 1 or 2 developers, 3-5 developers, 6-10 devs, more?
- New team?
- CVS/SVN? Testing? IDEs? Peer review?

Years ago, this is how I started... and how many new PHP developers still start when they get acquainted with the language:

- PHP files are on the server only
- They are edited with a very simple editor, like vim
- (Alternatively, FTP and Notepad are used)
- Coding standards are optional

In more commercial environments, many things are often missing:

- There are no proper specs
- Source control is not heard off
- Things are released or delivered "when they are ready"
- This is not a sustainable environment

- Overwriting changes happens all the time
- No "accountability" of mistakes
- Who does what from the spec that doesn't exist?
- How well do parts coded by different developers integrate?

- Set-up some sort of process and guidelines: coding standards, local development server, pushing out to staging/live servers, plan time for testing
- Source Control: CVS, SVN, or some of the newer ones
- Write requirements and specifications

- Makes it easier for all participants of the team to read code, as all the code conforms to the same rules.
- Best checked for with a tool: PHP Code Sniffer (http://pear.php.net/package/PHP_CodeSniffer/)
- Standard uses the PEAR Coding Standards, but the rules can easily be modified.

```
FILE: ...home/derick/dev/ezcomponents/trunk/Configuration/src/ini/ini_parser.php
```

```
-----  
FOUND 148 ERROR(S) AND 17 WARNING(S) AFFECTING 89 LINE(S)  
-----
```

```
5 | ERROR | @package tag comment indented incorrectly. Expected 3 spaces  
  |       | | but found 1.  
6 | WARNING | Invalid version "//autogen/" in file comment; consider "CVS:  
  |       | | <cvsn_id>" or "SVN: <svn_id>" instead  
6 | ERROR | The @version tag is in the wrong order; the tag follows  
  |       | | @license  
8 | ERROR | @license tag comment indented incorrectly. Expected 3 spaces
```

Requirements Specification

Define what the software is supposed to do.

Design

Define how the software is supposed to be implemented.

Implementation

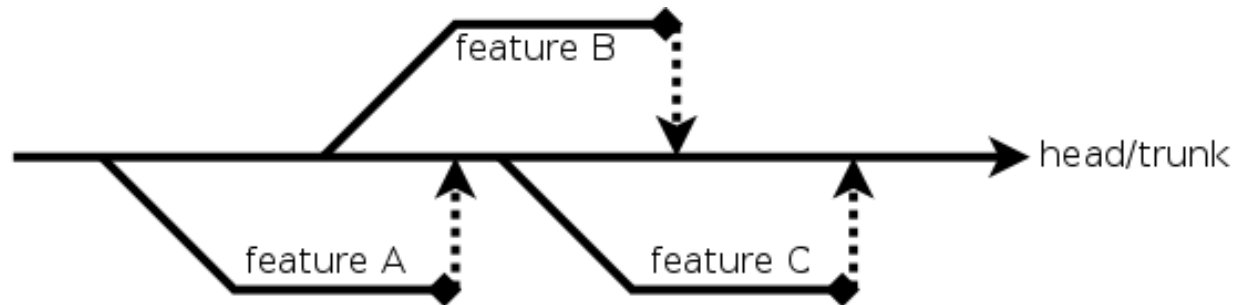
The implementation of the software itself.

Testing

The implemented software is tested.

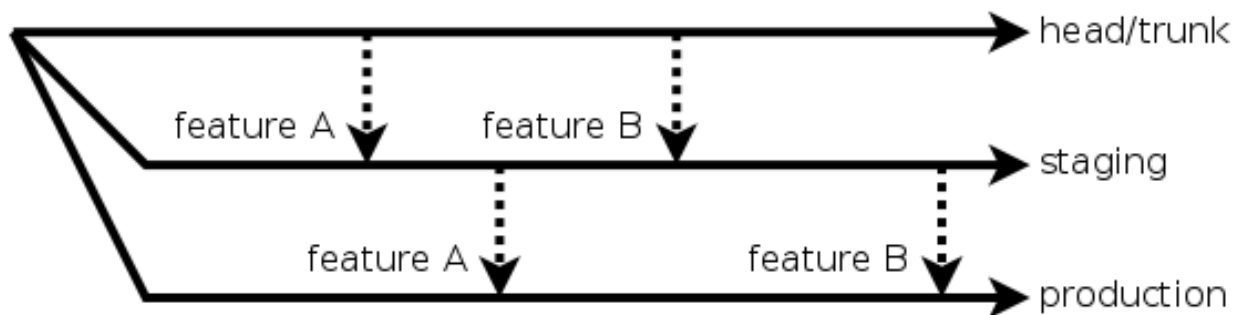
Sometimes.

Different Branching Techniques



Feature branches

All development happens in its own branch



Staging/production branches

All development happens in head/trunk and is merged into the staging/production branches

- Promote peer review of code
- Distributed teams require good communication
- They need to feel "one unit"
- Tools: Mailinglist, IM, IRC, Skype, twitter (TheWire)
- Issue Tracker to follow up on tasks and issues

- VC commit lists
- Pre-commit patch reviews
- Mandatory pre-release reviews
- Extreme Programming?



Warning: fopen(ezcmailtest) [[function.fopen](#)]: failed to open stream: No such file or directory in /home/httpd/html/test/ezc/dr_mail_test.php on line 147

... is kinda useless.

(!) ezcMailTransportException: An error occurred while sending or receiving mail. Failed to connect to the server: dolly.ez.no:ezcomponents. in /dat/dev/ezcomponents/trunk/Mail/src/transport/transport_connection.php on line 57

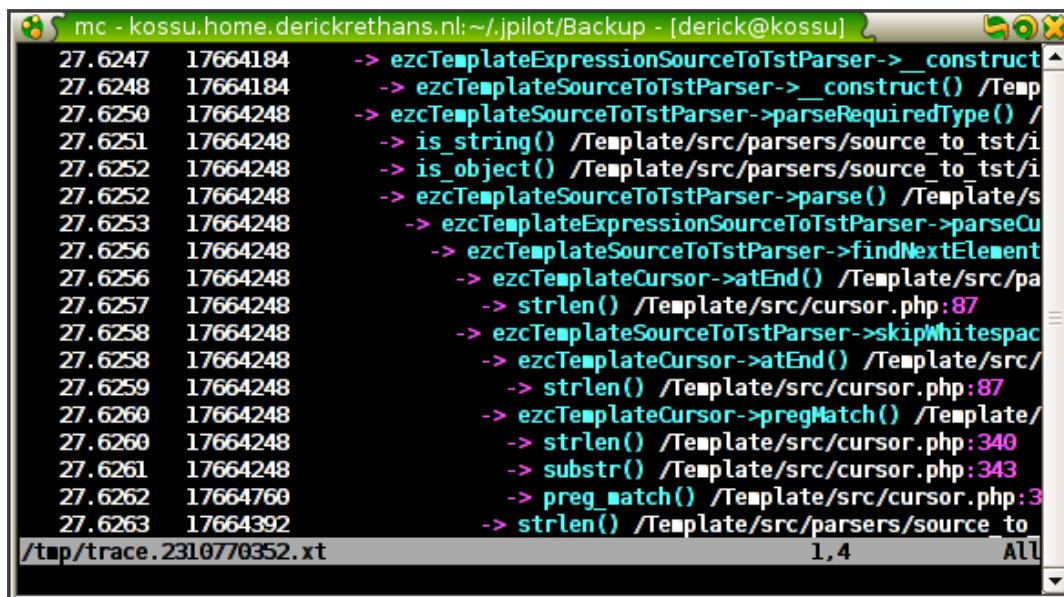
Call Stack

#	Time	Memory	Function	Location
1	0.0005	53512	{main}()	../dr_mail_test.php:0
2	0.0237	554800	ezcMailPop3Transport->__construct(string(11), string(12), string(12))	../dr_mail_test.php:8
3	0.0256	594140	ezcMailTransportConnection->__construct(string(11), string(12))	../pop3_transport.php:101

Variables in local scope (#3)

\$server	= 'dolly.ez.no' (length=11)
\$errno	= 111
\$port	= 'ezcomponents' (length=12)
\$errstr	= 'Connection refused' (length=18)
\$timeout	= 5

xdebug.collect_params=1 and xdebug.show_local_vars=1



```
mc - kossu.home.derickrethans.nl:~/j/pilot/Backup - [derick@kossu]
27.6247 17664184 -> ezcTemplateExpressionSourceToTstParser->__construct
27.6248 17664184 -> ezcTemplateSourceToTstParser->__construct() /Temp
27.6250 17664248 -> ezcTemplateSourceToTstParser->parseRequiredType() /
27.6251 17664248 -> is_string() /Template/src/parsers/source_to_tst/i
27.6252 17664248 -> is_object() /Template/src/parsers/source_to_tst/i
27.6252 17664248 -> ezcTemplateSourceToTstParser->parse() /Template/s
27.6253 17664248 -> ezcTemplateExpressionSourceToTstParser->parseCu
27.6256 17664248 -> ezcTemplateSourceToTstParser->findNextElement
27.6256 17664248 -> ezcTemplateCursor->atEnd() /Template/src/pa
27.6257 17664248 -> strlen() /Template/src/cursor.php:87
27.6258 17664248 -> ezcTemplateSourceToTstParser->skipWhitespac
27.6258 17664248 -> ezcTemplateCursor->atEnd() /Template/src/
27.6259 17664248 -> strlen() /Template/src/cursor.php:87
27.6260 17664248 -> ezcTemplateCursor->pregMatch() /Template/
27.6260 17664248 -> strlen() /Template/src/cursor.php:340
27.6261 17664248 -> substr() /Template/src/cursor.php:343
27.6262 17664760 -> preg_match() /Template/src/cursor.php:3
27.6263 17664392 -> strlen() /Template/src/parsers/source to
/tmp/trace.2310770352.xt 1,4 ALL
```

Some settings:

```
xdebug.auto_trace=1
xdebug.trace_output_dir=/tmp
xdebug.collect_params=1
xdebug.collect_return=1
xdebug.trace_output_name=trace.%c
```

- **Specifiers:**

http://xdebug.org/docs/execution_trace#trace_output_name

- DBGp, common Debugging protocol
- Cross-language: PHP, Python, Perl...
- Supported in Xdebug 2
- Clients: Komodo, Eclipse/PDT, Netbeans (and many others smaller ones)
- Other clients and plugins:

Komodo and Debugging

The screenshot shows the Komodo IDE 5.1 interface with a PHP script named 'show.php' open. The script contains the following code:

```
$c = compact('presentationDir', 'baseDir', 'showScript', 'helpPage', 'baseFontS',  
            'flashFontScale', 'pdfFontScale', 'pdfResourceFile', 'pdf_font',  
            'pdf_font_bold', 'pdf_example_font', 'jsKeyboard', 'css_supports_f'  
require_once 'sniff.php';  
if ($browser_is_IE) {  
    echo "IE is not supported - please use Firefox, Safari, Konqueror or ju  
    exit;  
}  
set_time_limit(0); // PDF generation can take a while  
if(!strlen($ SERVER['PATH INFO'])) {
```

The debugger is paused at line 8, which is the `require_once 'sniff.php';` line. The 'Debug: show.php' window is active, showing the following information:

Break at /home/httpd/pres2/show.php, line 8.

Name	Type	Value
baseDir	uninitialized	
baseFontSize	uninitialized	
browser_is_IE	uninitialized	
c	array	
baseDir	string	/
baseFontSize	string	10pt
flashFontScale	int	18
helpPage	string	help.php

The 'Call Stack Location' window shows the current execution point: `show.php, line 8, in {main}`.

Requirements Specification

Define what the software is supposed to do.

Design

Define how the software is supposed to be implemented.

Implementation \equiv Testing

The implemented software is tested.

The implementation of the software itself.

Tests drive the development

- Tests are written before the code
- There is no code without tests

Test Suites

- Contain tests that check whether the code does what it is supposed to do
- Also cover things that should fail

Present the Idea

Write the Requirements Document

Design the Component

Implementation

- Write API stubs with parameter documentation and descriptions
- Write test cases
- Initial implementation
- Initial implementation review
- Updating implementation according to review
- Implementation review

Pre-release Testing

Write Test Case

Write a test case to test the correct behavior of the method, and verify that the test case fails

Fix the Issue

Fix the implementation

Verify Fix with Test Case

Make sure that the test cases pass with the new implementation

Check Other Test Cases

Verify that the fix for this defect did not break any other test case

PHPUnit

Unit Testing Framework for PHP

```
derick@kossu: ~/dev/ezcomponents/trunk
derick@kossu:~/dev/ezcomponents/trunk$ php UnitTest/src/runtests.php Configuration
on
ezcUnitTest uses the PHPUnit 3.4.0RC2 framework from Sebastian Bergmann.

[Preparing tests]:
eZ Components:
  Configuration:
    ezcConfigurationManagerDelayedInitTest: .
    ezcConfigurationTest: .....
    .....
    ezcConfigurationManagerTest: .....
    ezcConfigurationIniReaderTest: .....S.....
    .....
    ezcConfigurationIniParserTest: ...
    ezcConfigurationIniWriterTest: .....
    ezcConfigurationArrayWriterTest: .....

Time: 1 second

OK, but incomplete or skipped tests!
Tests: 167, Assertions: 314, Skipped: 1.
derick@kossu:~/dev/ezcomponents/trunk$
```


































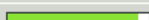
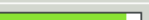
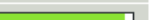















PHPUnit

Code Coverage: No Code Without Test

eZ Components

Current directory: [/home/derick/dev/ezcomponents/trunk/Graph/src](#)

Legend: Low: 0% to 35% Medium: 35% to 70% High: 70% to 100%

	Coverage					
	Classes		Methods		Lines	
Total		94.39% 101 / 107		86.89% 444 / 511		86.32% 8990 / 10415
<u>axis</u>		100.00% 5 / 5		85.25% 52 / 61		91.02% 669 / 735
<u>charts</u>		100.00% 5 / 5		100.00% 36 / 36		99.56% 683 / 686
<u>colors</u>		100.00% 3 / 3		100.00% 17 / 17		100.00% 216 / 216
<u>data_container</u>		100.00% 2 / 2		100.00% 13 / 13		100.00% 39 / 39
<u>datasets</u>		100.00% 9 / 9		89.80% 44 / 49		93.63% 235 / 251
<u>driver</u>		50.00% 3 / 6		49.47% 47 / 95		54.97% 1339 / 2436
<u>element</u>		100.00% 4 / 4		100.00% 21 / 21		99.35% 458 / 461
<u>exceptions</u>		96.15% 25 / 26		96.00% 24 / 25		95.50% 106 / 111
<u>interfaces</u>		100.00% 7 / 7		97.73% 43 / 44		93.74% 943 / 1006
<u>math</u>		100.00% 7 / 7		100.00% 43 / 43		99.72% 350 / 351
<u>options</u>		92.86% 13 / 14		90.32% 28 / 31		94.11% 815 / 866
<u>palette</u>		83.33% 5 / 6		100.00% 0 / 0		83.33% 5 / 6
<u>renderer</u>		100.00% 8 / 8		100.00% 67 / 67		96.19% 3005 / 3124
<u>structs</u>		100.00% 3 / 3		100.00% 7 / 7		100.00% 35 / 35
<u>graph.php</u>		100.00% 1 / 1		100.00% 0 / 0		100.00% 1 / 1
<u>tools.php</u>		100.00% 1 / 1		100.00% 2 / 2		100.00% 91 / 91

Generated by [PHPUnit 3.3.0RC1](#) and [Xdebug 2.1.0-dev](#) at Mon Sep 8 10:30:06 CEST 2008.

```
550 :  
551 19 :      switch ( $this->interval )  
552 :      {  
553 19 :      case self::MONTH:  
554 4 :          $time = strtotime( '+1 month', $time );  
555 4 :          break;  
556 15 :      case self::YEAR:  
557 4 :          $time = strtotime( '+1 year', $time );  
558 4 :          break;  
559 11 :      case self::DECADE:  
560 0 :          $time = strtotime( '+10 years', $time );  
561 0 :          break;  
562 11 :      default:  
563 11 :          $time += $this->interval;  
564 11 :          break;  
565 19 :      }  
566 19 :      }  
567 :  
568 19 :      return $steps;  
569 :      }  
570 :  
571 :      /**
```

Green: Covered code.

Red: Not covered code.

Grey: Unreachable code.

White: No code.

How much coverage should you aim for?

<http://www.developertesting.com/archives/month200705/20070504-000425.html>

< 100% Coverage \equiv Not fully tested code

100% Coverage \neq Fully tested code


Code Coverage: 100% is not good enough

```
1      : <?php
2      : class PathTest
3      : {
4      :     static public function testMethod( $a, $b )
5      :     {
6      2 :         $nr = 1;
7      2 :         if ( $a == true )
8      2 :         {
9      1 :             $nr *= 2;
10     1 :         }
11     2 :         if ( $b == true )
12     2 :         {
13     1 :             $nr *= 3;
14     1 :         }
15     2 :         return $nr;
16     :     }
17     : }
```

```
<?php require "PathTest.php";
class CcTest extends PHPUnit_Framework_TestCase
{
    public function testPath1()
    {
        self::assertEquals( 2, PathTest::testMethod( true, false ) );
    }
    public function testPath2()
    {
        self::assertEquals( 3, PathTest::testMethod( false, true ) );
    }
}
?>
```

phpUnderControl

Continuous integration with CruiseControl


Project: **ezcDocument**
Build: **More builds**

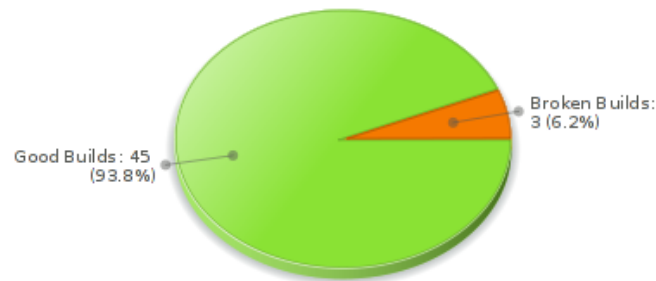
waiting for next time to build since
09/01/2009 16:44:16
progress: 16:44:16 next build in 5 minutes

Overview | Tests | Metrics | Coverage | Documentation | CodeSniffer | PHPUnitPMD

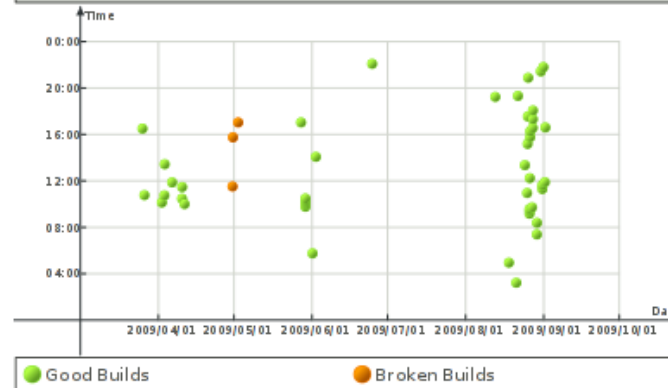
Project Metric Summary

Number of Build Attempts	48
Number of Broken Builds	3
Number of Successful Builds	45

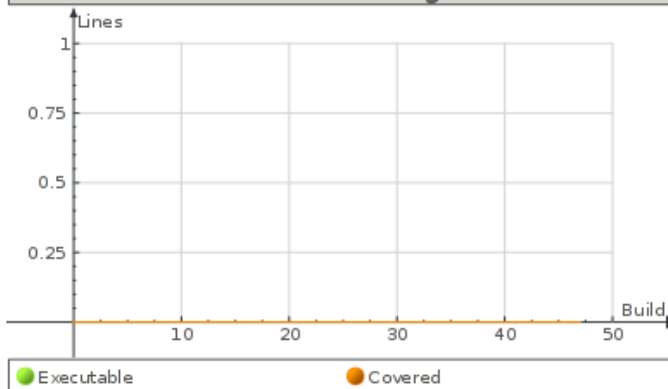
Breakdown of Build Types



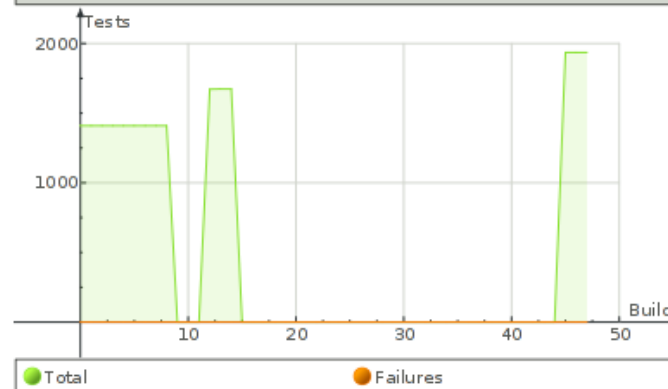
Breakdown of Build Timeline



Unit Coverage



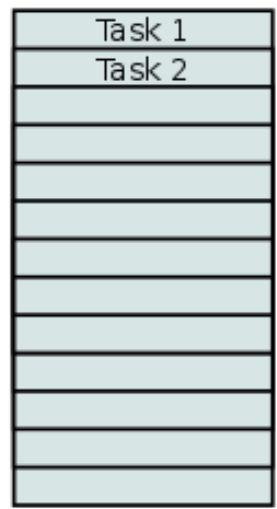
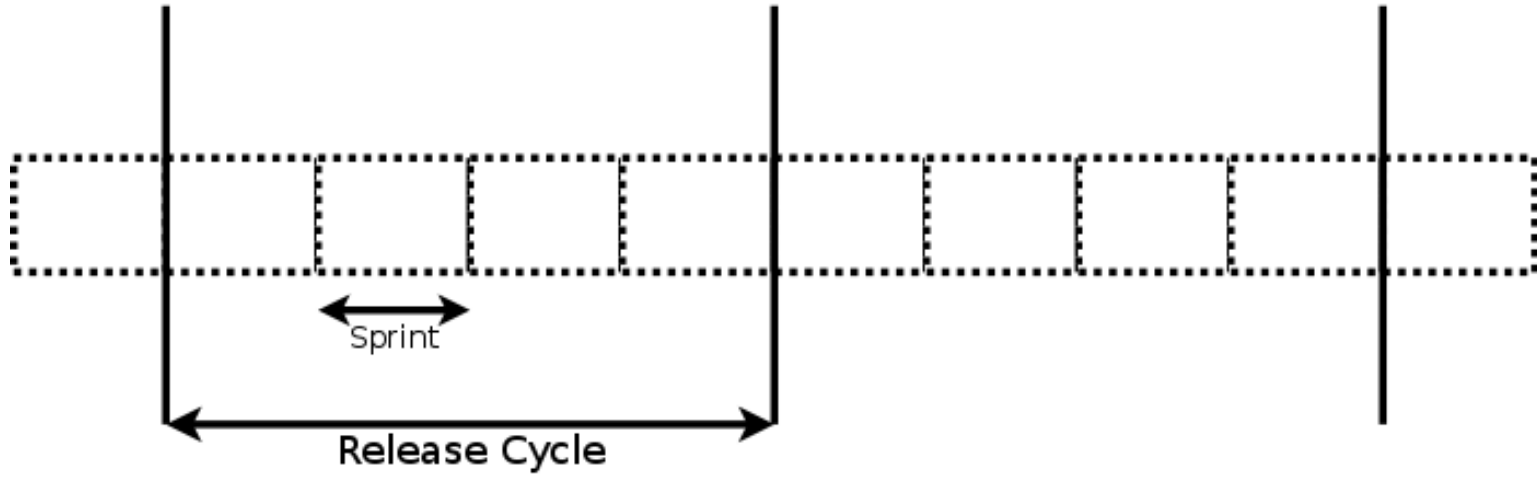
Unit Tests



Test to Code Ratio

Coding Violations

Timeboxing

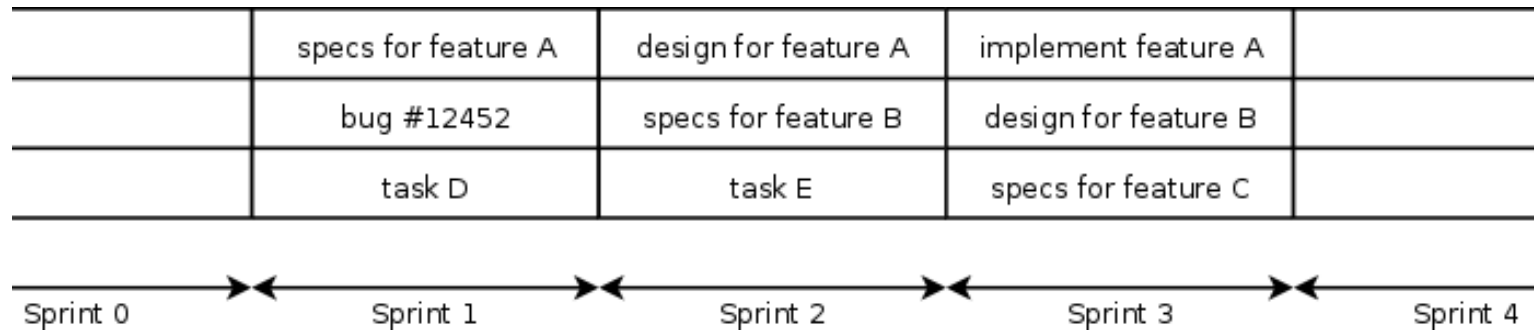


Backlog of tasks



Sprint

Timeboxing Sprints



- Sprints are generally 3 to 4 weeks
- Every aspect needs to have a user task: specs, design, testing, peer review, releasing
- A feature can be split over multiple sprints if needed
- After every sprint you should have something that you can immediately release. (Doesn't mean every functionality should be exposed yet of course).



- <http://homepage.mac.com/hey.you/lessons.html>
- Testuvius: <http://www.artima.com/weblogs/viewpost.jsp?thread=203994>

- PHP Unit: <http://phpunit.de>
- phpUnderControl: <http://phpundercontrol.org>
- Xdebug: <http://xdebug.org>
- These Slides: <http://derickrethans.nl/talks.php>

- PHP: <http://www.php.net>