

# Welcome!

## **phpUnderControl** **A Quick Start to Continuous Integration**

Sebastian Bergmann  
<http://sebastian-bergmann.de/>

September 17<sup>th</sup> 2008

# Who I am



- Sebastian Bergmann
- Involved in the PHP Project since 2000
- Developer of PHPUnit
- Author, Consultant, Coach, Trainer

# Continuous Integration

- Software development practice where members of a team integrate their work frequently

# Continuous Integration

- Software development practice where members of a team integrate their work frequently
  - Usually each person integrates at least daily, leading to multiple integrations per day

# Continuous Integration

- Software development practice where members of a team integrate their work frequently
  - Usually each person integrates at least daily, leading to multiple integrations per day
- Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible

# Continuous Integration

- Software development practice where members of a team integrate their work frequently
  - Usually each person integrates at least daily, leading to multiple integrations per day
- Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible
- This approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly

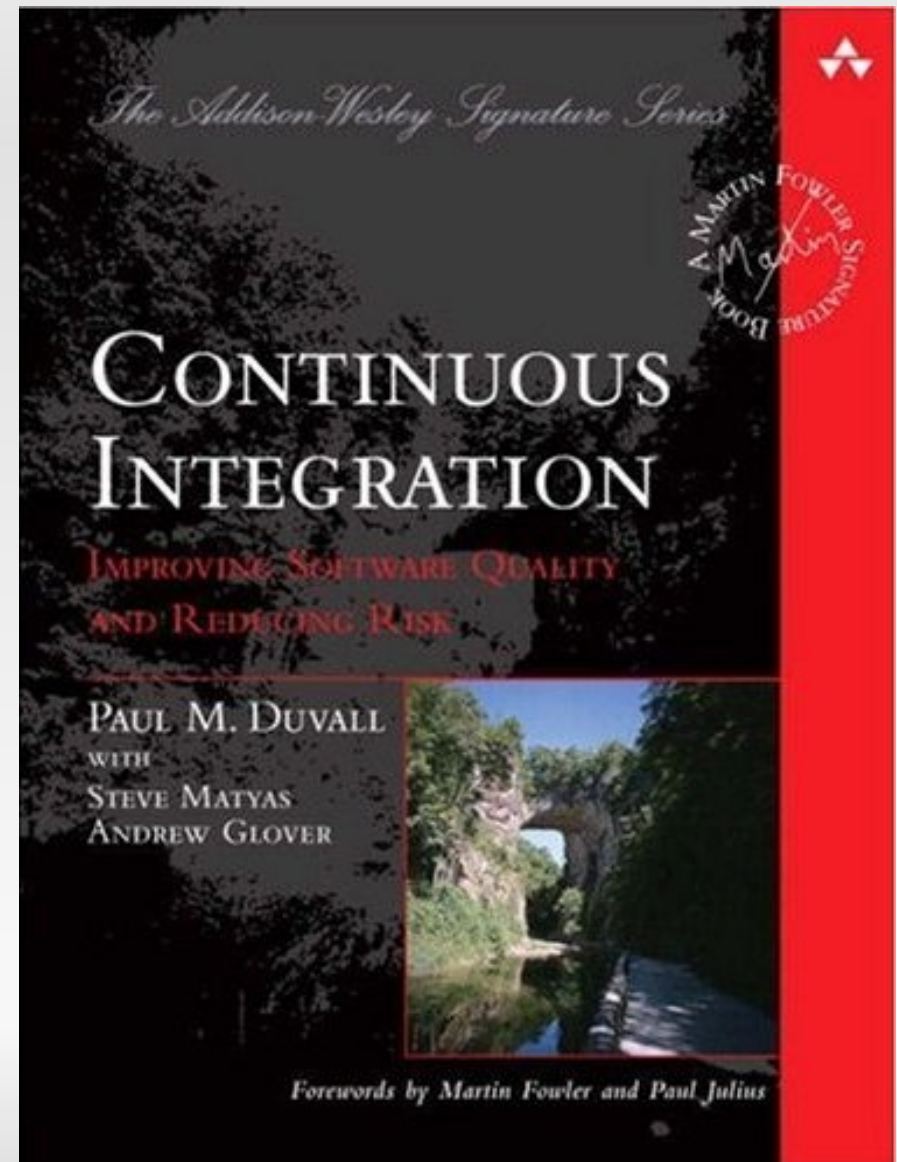
# Continuous Integration

Continuous Integration

Paul M. Duvall

Addison-Wesley, 2007

ISBN 978-0321336385



# Continuous Integration

## Software

- CruiseControl
  - phpUnderControl
- Bamboo
- BuildBot
- Hudson
- Xinc
- ...

# CruiseControl

CruiseControl is a framework for a continuous build process

- It includes, but is not limited to, plugins for email notification, Ant, Phing, and various source control tools
- A web interface is provided to view the details of the current and previous builds

# CruiseControl

## phpUnderControl

phpUnderControl is customization of CruiseControl that caters to the needs of PHP projects

- PHPUnit
- PHPDocumentor
- PHP\_CodeSniffer
- (PHP\_Depend)
- (PHP\_CompatInfo)
- ...

# Software Metrics

- A software metric is a measure of some property of a piece of software or its specifications.
- „You cannot control what you cannot measure.“ (Tom DeMarco)

# Software Metrics

## Lines of Code

- Text-based metric for code size
- Various definitions
  - Lines of Code (LOC)
  - Comment Lines of Code (CLOC)
  - Non-Comment Lines of Code (NCLOC)
  - Executable Lines of Code (ELOC)
- Ratios can be interesting
  - $CLOC / (E)LOC$

# Software Metrics

## Code Coverage

- Which statements, branches, and paths are executed when the tests run?
  - $C^0$ -Coverage: Statement Coverage
  - $C^1$ -Coverage: Branch Coverage
  - $C^\infty$ -Coverage: Path Coverage
- 100% Code Coverage is a required, but not a sufficient criteria for test completeness

# Software Metrics

## Cyclomatic Complexity

- Metric for the complexity of a code unit
  - Originally defined in graph theory
    - $ccn(G) = \text{Edges} - \text{Vertices} + 2$
- Equivalent: Counting the branching points
  - `if`, `for`, `foreach`, `while`, `case`, `catch`, `&&`, `||`, ternary operator (`?:`)
- Interpretation
  - Higher complexity leads to more errors
  - Higher complexity makes testing harder

# Software Metrics

## Cyclomatic Complexity

```
<?php
1 function isLeapYear($year)
  {
    $result = false;

2    if ($year % 4 != 0) {
      $result = false;
    }

3    if ($year % 100 != 0) {
      $result = true;
    }

4    if ($year % 400 == 0) {
      $result = true;
    }

    return $result;
  }
```

# Software Metrics

## Change Risk Analysis and Predictions (CRAP)

- Estimates effort, pain, time, ... to change a unit of code based on code coverage and complexity
  - Low Code Coverage  $\Rightarrow$  High CRAP Index
  - High Code Coverage  $\Rightarrow$  CRAP Index  $\approx$  Complexity

$$CRAP(m) = CCN(m)^2 * \frac{1 - COV(m)}{100}^3 + CCN(m)$$

# Software Metrics

## Tools for PHP

- PHP\_CodeSniffer
- PHP\_Depend
- Support for software metrics in PHPUnit
  - Added in PHPUnit 3.2
  - Deprecated in PHPUnit 3.4
  - Removed in PHPUnit 4.0
    - Seamless transition from PHPUnit to PHP\_CodeSniffer and PHP\_Depend when using phpUnderControl

# The End

- Thank you for your interest!
- These slides will be available shortly on <http://sebastian-bergmann.de/talks/>.

# License

This presentation material is published under the Attribution-Share Alike 3.0 Unported license.

You are free:

- ✓ **to Share** – to copy, distribute and transmit the work.
- ✓ **to Remix** – to adapt the work.

Under the following conditions:

- **Attribution.** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Share Alike.** If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

For any reuse or distribution, you must make clear to others the license terms of this work.

Any of the above conditions can be waived if you get permission from the copyright holder.

Nothing in this license impairs or restricts the author's moral rights.