

# La strada per sviluppare più rapidamente: Unit Test & Continuous Integration

by *Enrico Zimuel*

*Senior Consultant & Architect - Zend Technologies*

Email: [enrico.z@zend.com](mailto:enrico.z@zend.com)

Blog: <http://www.zimuel.it/blog>



The PHP Company

Copyright © 2007, Zend Technologies Inc.

# Sommario

- **Metodologie di sviluppo**
- **Introduzione al Testing automatico**
- **Il progetto PHPUnit**
- **Esempi di Unit Test in Zend Framework**
- **Introduzione alla Continuous Integration**
- **Il progetto Bamboo**

# Metodologia di sviluppo

- **Perchè utilizzare una nuova metodologia di sviluppo, Unit Test + Continuous Integration?**
  - **più confidenti nelle modifiche del software;**
  - **ripetibilità/predicibilità;**
  - **modifiche centralizzate, riusabilità del codice;**
  - **limitare lo scope ad una sola istanza del test.**

# Strumenti

zend studio

= IDE per lo sviluppo in PHP



= Testing applicazioni PHP



= Continuous Integration

# Terminologia

- **Unit Test**
  - Il codice da testare
- **Test**
  - Il codice che richiama il codice da testare
- **Assertazione**
  - Condizione di test
- **Code coverage**
  - Le linee di codice che sono state testate
- **Continuous integration**
  - L'applicazione continua del controllo di qualità, es. Unit test

# PHPUnit

- **PHPUnit** è un framework open source per la creazione di test automatici in PHP
- **Gli obiettivi di PHPUnit sono:**
  - facilità d'utilizzo (scrivere, leggere, eseguire test);
  - velocità di esecuzione dei test;
  - creazione di test isolati;
  - test componibili;
- **E' integrato in Zend Studio**
- Per maggiori informazioni: <http://www.phpunit.de/>

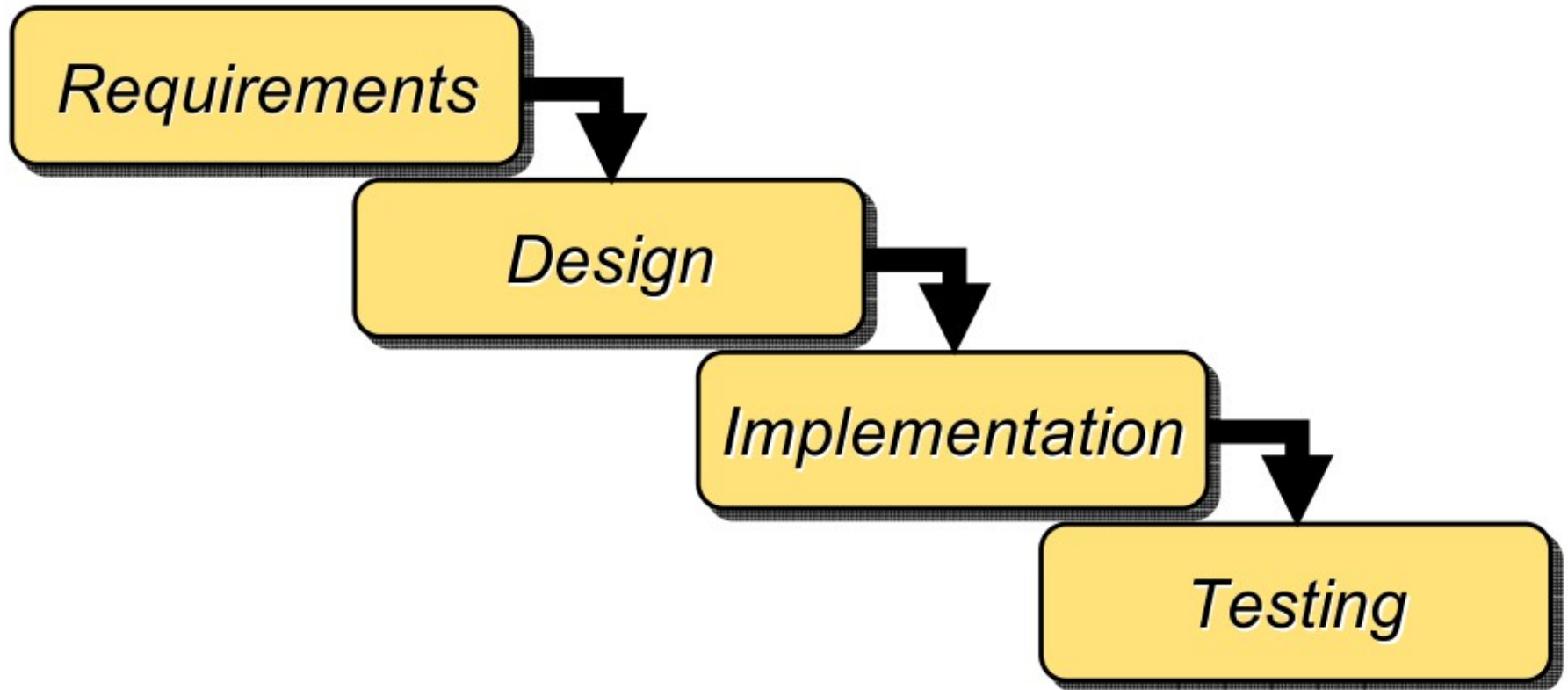


# Demo: PHPUnit & Zend Studio

- Esempio di creazione di un test con PHPUnit per un progetto in Zend Framework
- Integrazione con Zend Studio
- Funzione di Code Coverage

# Che cos'è la Continuous Integration?

# Il vecchio modello: a cascata

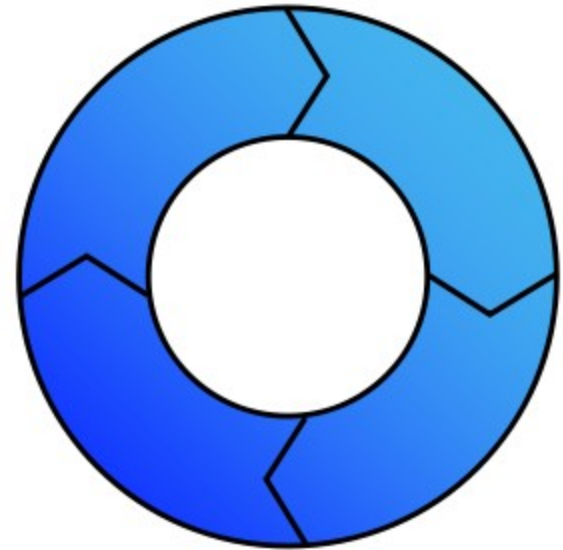


# Problemi del modello a cascata

- E' difficile individuare tutti i requisiti in un singolo step
- I requisiti cambiano continuamente
- Capita spesso di dover tornare indietro, rieseguendo il lavoro più volte
- I tempi ed i costi di sviluppo sono difficilmente valutabili e tendono sempre ad aumentare

# Un modello migliore: Iterazione

- 1) Definire i requisiti
- 2) Specificare i test
- 3) Implementazione, fintanto che i test risultano positivi
- 4) Ripetizione degli step



# Vecchia metodologia: creazione manuale di Test

- **Svantaggi:**
  - Scrivere documentazione dei test
  - Creazione di test manuali per ogni release
  - Processo lento e non privo di errori

# Un modo migliore: creazione di Test automatici

- Test del codice... con altro codice
- Il codice per il testing è importante tanto quanto quello di produzione
- Esecuzione di test su tutto il progetto per scovare problemi di regressione
- Economico, veloce, distribuito

# Continuous Integration

=

Sviluppo iterativo

+

Test automatici

# Continuous Integration

- Le modifiche vengono eseguite ad incrementi piccoli e rapidi
- Ogni volta che si esegue una modifica sul codice si eseguono i test automatici ed il processo di build/rilascio
- Se tutti i test **sono positivi** si passa allo step successivo del processo di QA
- Se I test **falliscono** gli sviluppatori devono sistemare l'errore immediatamente

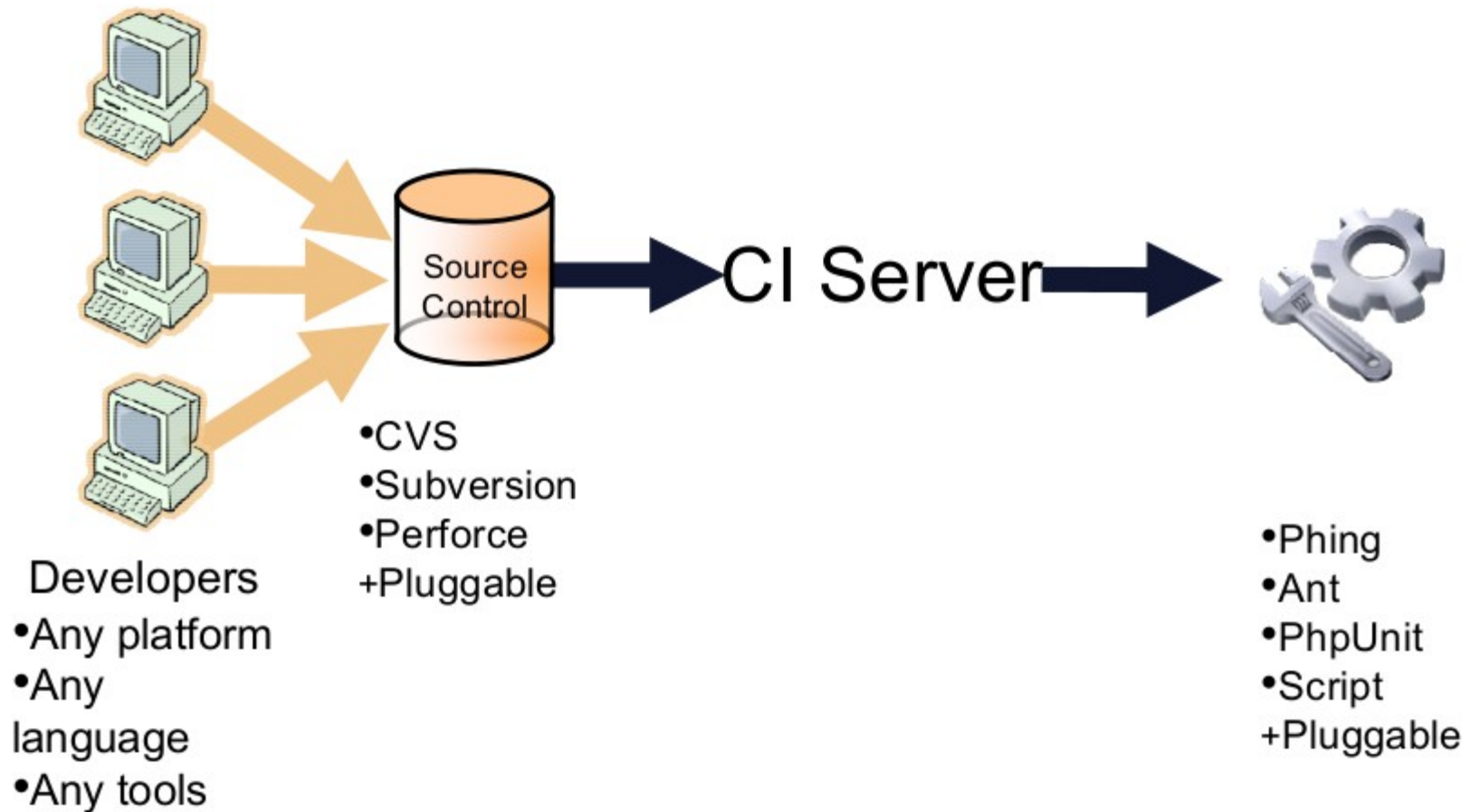
# Continuous Integration: vantaggi

- Migliore rintracciabilità degli errori
- Incremento della visibilità del progetto
- Rapidi feedback dagli utenti finali
- Rilascio incrementale del progetto
- Risparmio di tempo
- Definizione di tempistiche reali e fattibili

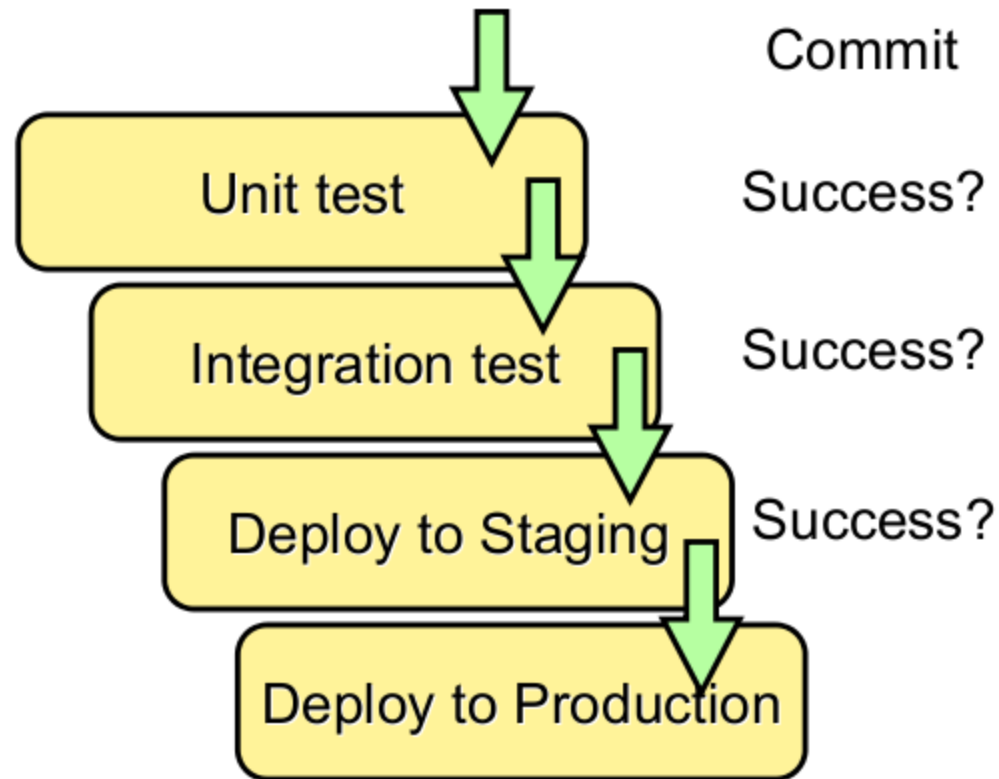
# Continuous Integration: per lo sviluppatore

- Gestione dei conflitti istantanea
- Ultima versione del progetto sempre disponibile
- L'aggiornamento frequente facilita ed incoraggia la creazione di moduli semplici
- Le metriche sul processo di CI migliorano la qualità del codice

# Un'implementazione tipica del processo di CI



# Ciclo di build e release



# Bamboo

- **Bamboo** è un software per facilitare l'implementazione della metodologia di Continuous Integration
- Funzionalità:
  - istant continuous integration;
  - facile integrazione in scenari diversi;
  - build distribuiti;
  - report e analisi;
  - integrazione con PHPUnit
- Per maggiori info:  
<http://www.atlassian.com/software/bamboo/>



# Bamboo: esempio di report

## Bamboo - Acceptance tests

[BAM-BAMFUNC-2782](#), Updated by [Belinda Teh](#), 6 hours ago 109 passed

## Bamboo - HEAD

[BAM-MAIN-5714](#), Updated by [Belinda Teh](#), 6 hours ago 1646 passed

## Bamboo - Remote Agent Functional Tests

[BAM-REMOTE-1837](#), Manual build by [Mark Chaimungkalanont](#), 2 hours ago 2 out of 12 failed

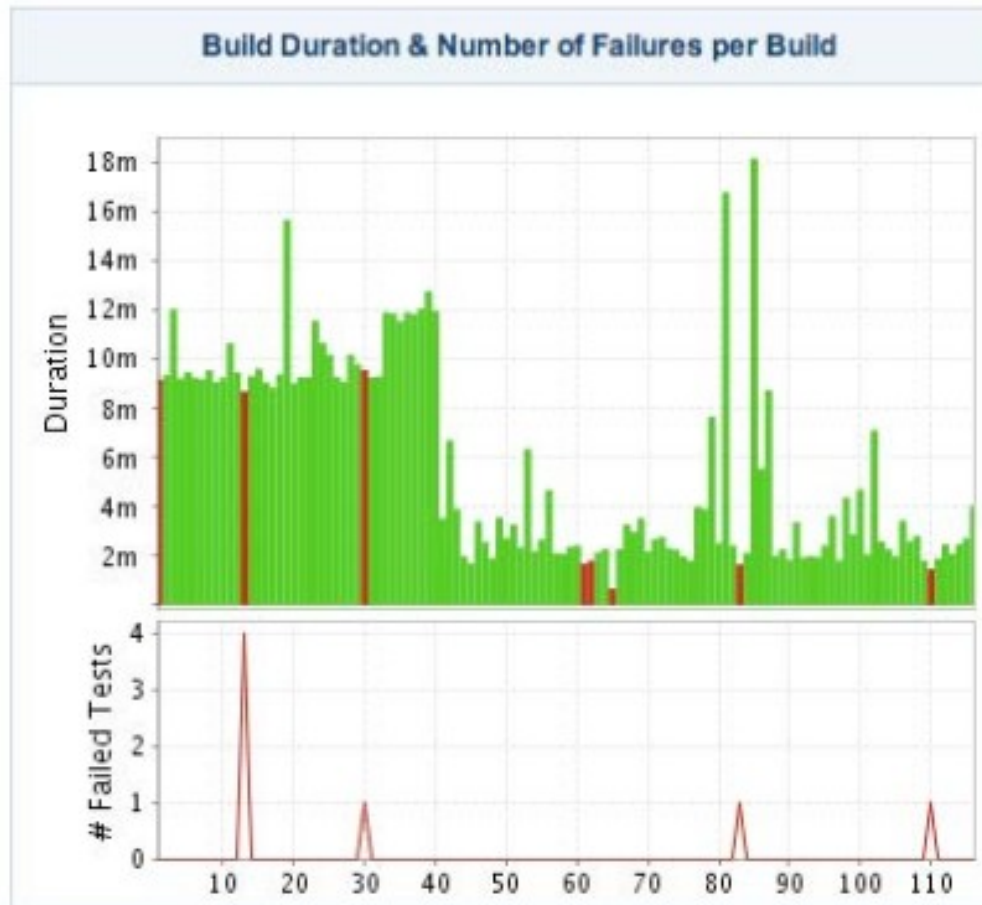
## Bamboo - Stable Acceptance Tests

[BAM-STABFUNC-887](#), Updated by [David Cheney](#), 3 days ago 96 passed

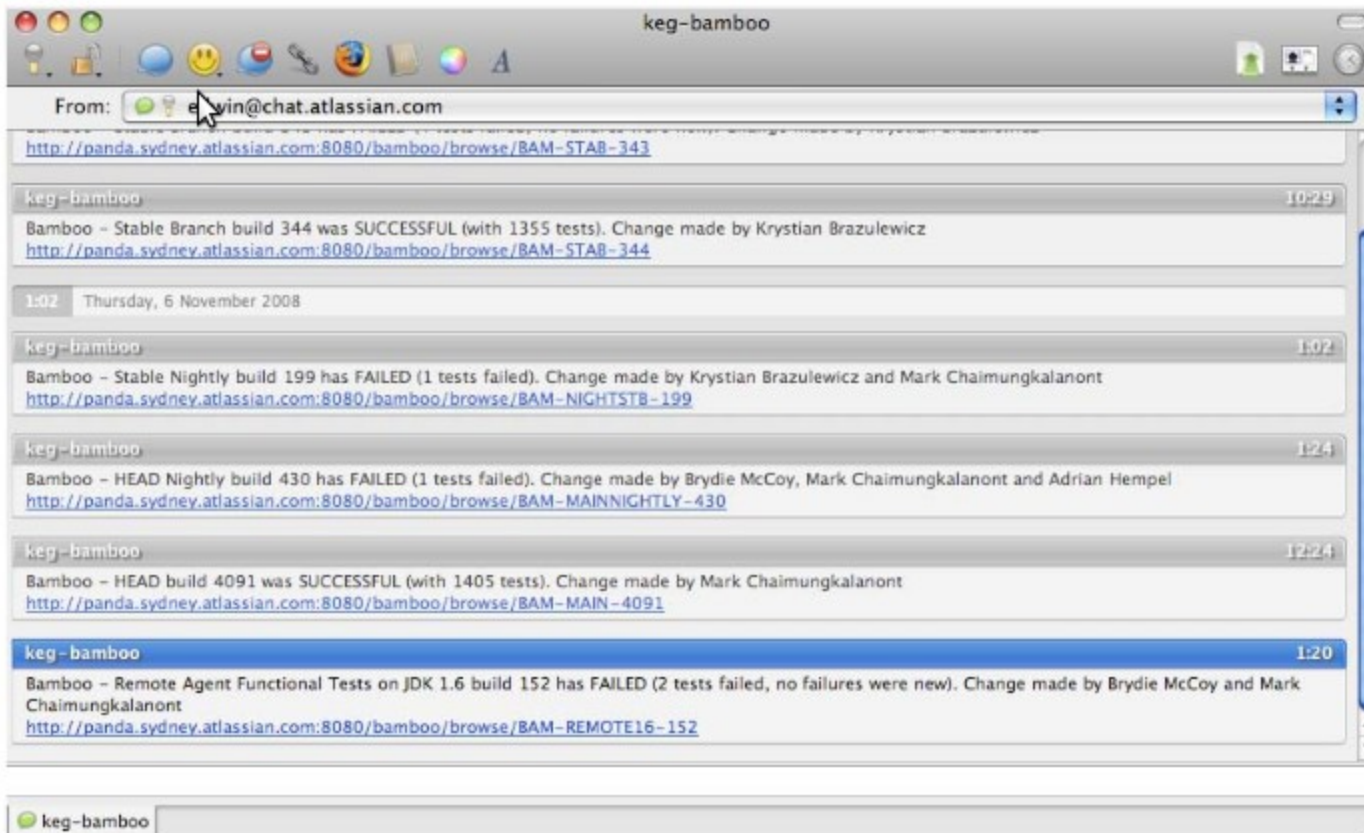
## Bamboo - Stable Branch

[BAM-STAB-671](#), Updated by [David Cheney](#), 3 days ago 1680 passed

# Bamboo: metrice



# Bamboo: collaborazione istantanea



Grazie!

Per ulteriori informazioni:

<http://www.zend.com>

<http://www.atlassian.com>



The PHP Company

