

# Continuous Integration e High Quality Code



Daniele Mondello

2

# Chi Sono



Daniele Mondello

## About me

Nasco nel 1971 a Palermo e dopo un' infanzia fatta di cartoni animati, Super Santos, libri e Coin-Up nell'estate dell'86 ricevo in dono un COMMODORE 64 che mi cambia la vita.



## Skill

### O.S.

- Ubuntu, CentOS
- Mac OSX
- Windows
- IOS
- Android
- Embedded OS: Raspbian.Pidora

### General

- Management
- Team Leader
- Quality Manager
- Web Architect
- Healthcare IT Consultant
- UX Expert

### Developer

- PHP (5 years)
- Javascript (10 years)
- ASP(12 years)
- MySQL (5 years)
- HTML, CSS3 (14 years)
- Ruby, Rails (1 year)

3

1986

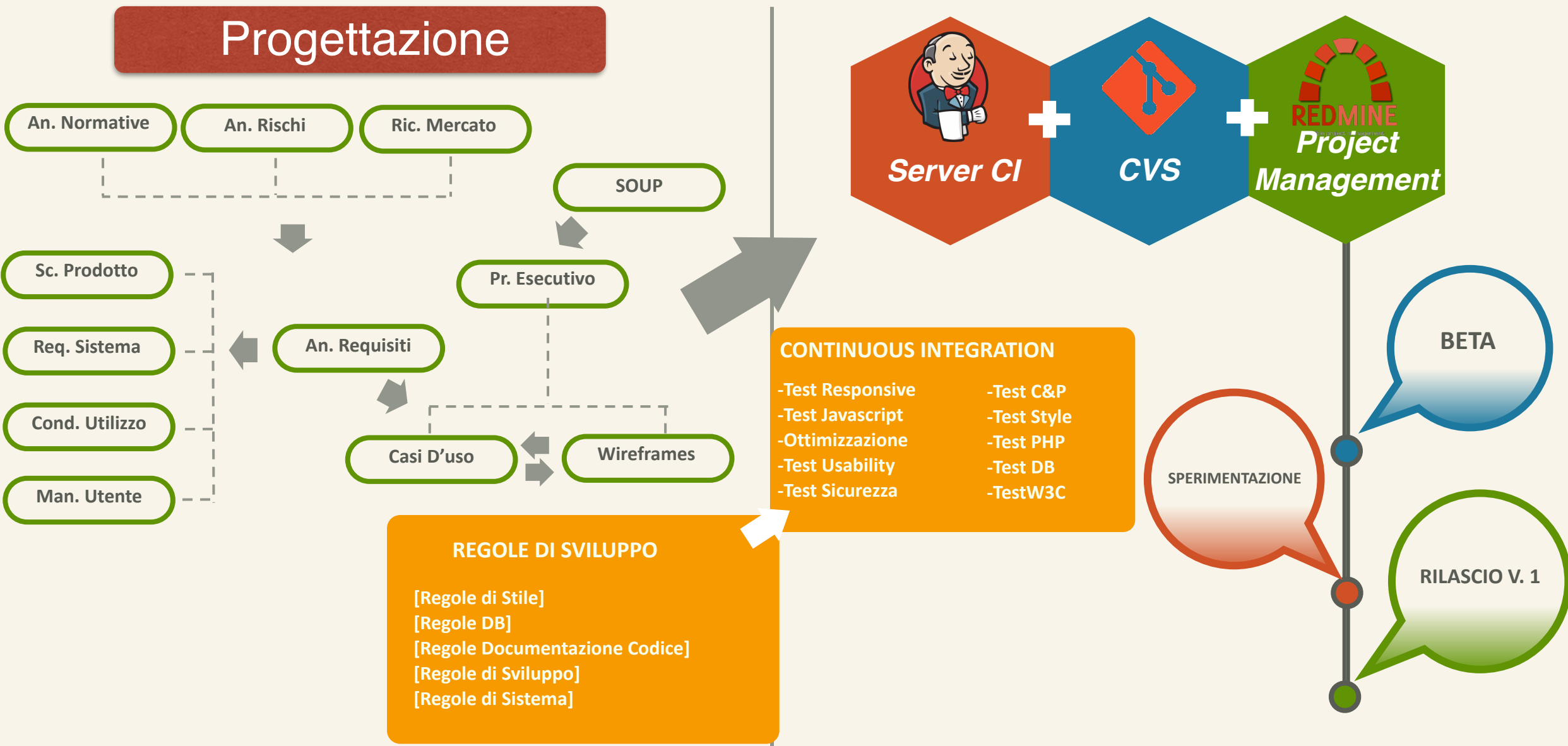


Diego!

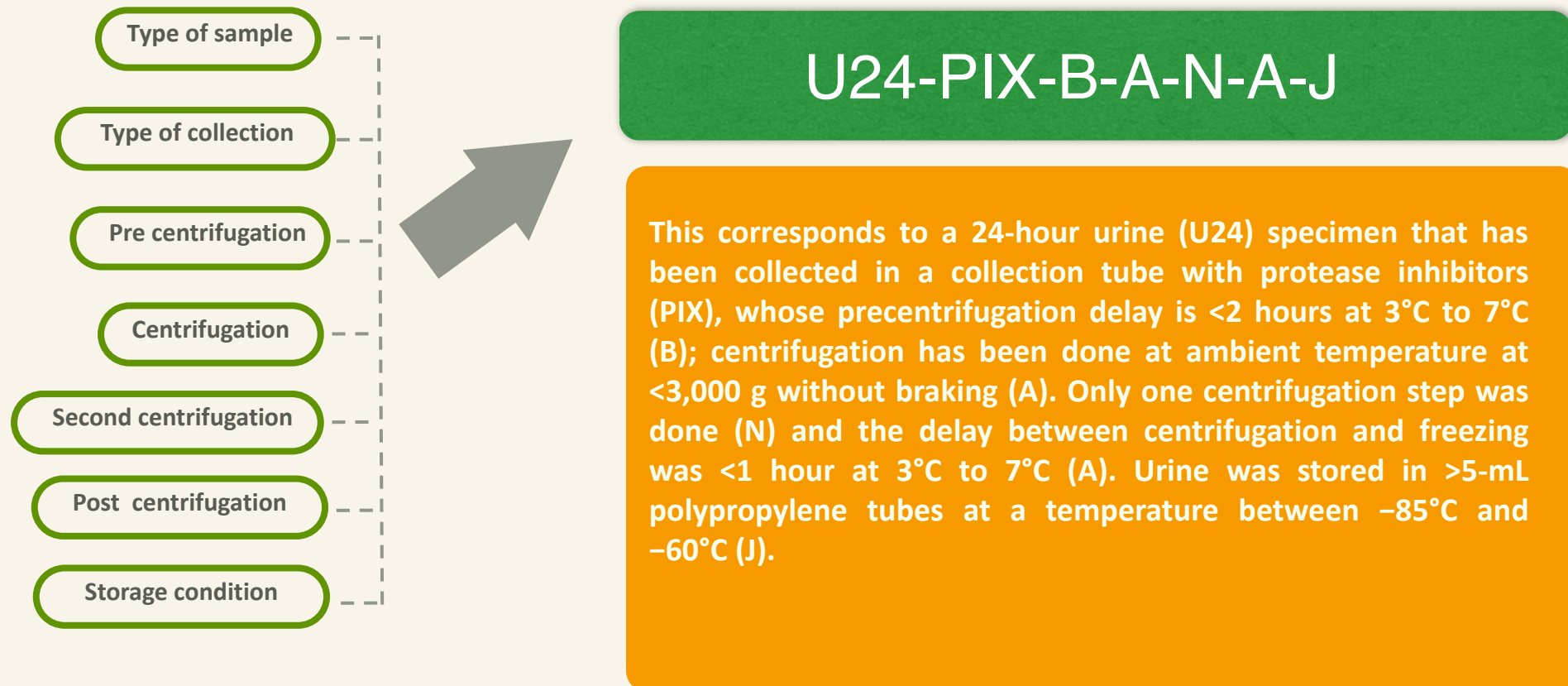
---



# 4 Prodotto Industriale



# SPREC: EXAMPLE



# Cosa è la Continuous Integration?

“E’ una pratica di sviluppo software dove i membri di un team integrano il loro lavoro frequentemente, almeno una volta al giorno. Ogni integrazione è verificata da una build automatica (inclusi i test) per individuare gli errori d’integrazione il più rapidamente possibile.”



# 7 Cosa è un Control Version System?



*E' un tool che permette il controllo di versione (versioning), cioè la gestione di versioni multiple di un insieme di informazioni.*



**Bazaar**

# 8 CVS: GIT



01

*Esaminare lo stato di un progetto nel più breve tempo possibile*



02

*Dividere lo sviluppo del progetto in più linee indipendenti, chiamati "rami", che possono evolvere separatamente*

03

*Periodicamente ricombinare i rami in un processo per riconciliare le modifiche apportate in due o più ram*

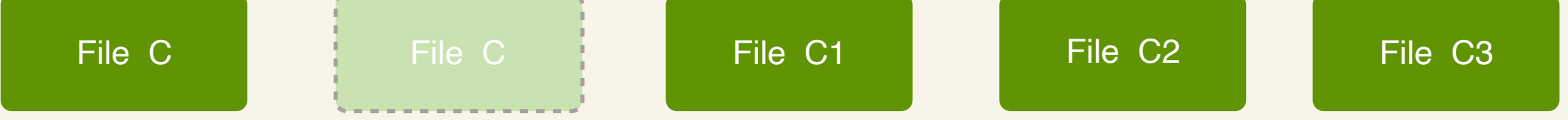
04

*Mostrare le differenze tra le varie versioni del progetto*

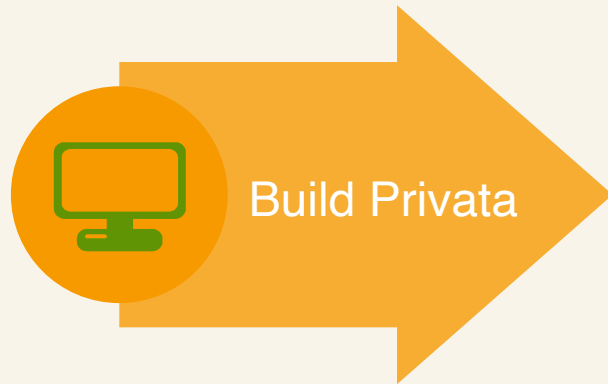




# 9 Funzionamento GIT

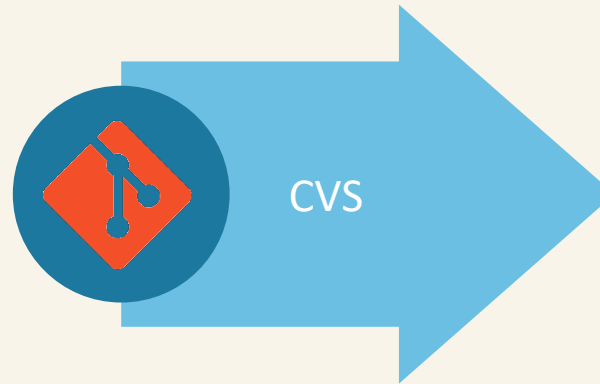


# FLUSSO DI LAVORO



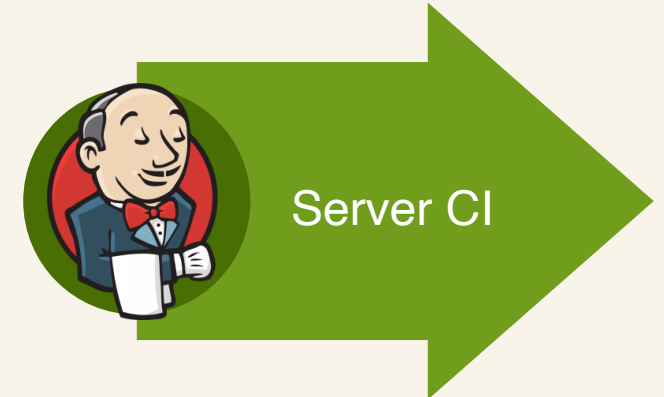
## *Lancio di una build privata*

Lo sviluppatore effettua una Build Privata sulla propria postazione così da integrare eventuali cambiamenti effettuati dagli altri membri del team e verificato che l'integrazione dia un sistema funzionante.



## *Commit sul CVS del codice*

Lo sviluppatore conclude l'attività effettuando un commit sul repository del sistema di controllo di versione. 5



## *Lancio SCRIPT su server CI*

Il server di CI accortosi del cambiamento scarica il codice aggiornato e lancia uno script di build così da integrare questi cambiamenti e da ricostruire il sistema e ritestarlo ed a seguito del risultato della build, genera un feedback (es: email) visibile al team

# MOTORE CI e BUILD



Il sistema di CI viene utilizzato per:

- collezionare il codice dai repository remoti
- lanciare i test automatici
- lanciare test statici sul codice
- generare report più o meno complessi
- effettuare delle misurazioni rispetto a delle metriche
- notificare a chi di dovere

## Differenti tipi di BUILD



### Privata

Test sul codice  
Test PHP  
Test Javascript



### Pubblica

Test Integrazione  
Prod. Documentazione  
Test Sicurezza



### Deploy

Minifizzazione  
Cambio URL  
Disattivazione Warning

**Jenkins**  [admin](#) | [esci](#)

Jenkins [ENABLE AUTO REFRESH](#) [aggiungi descrizione](#)

[Nuovo Job](#)  
[Utenti](#)  
[Cronologia build](#)  
[Configura Jenkins](#)  
[Mie Viste](#)

**Coda di build**  
 Nessun build in coda.

**Stato esecutore build**

#	Stato
1	Inattivo
2	Inattivo

Tutto +

S	W	Nome ↓	Ultimo successo	Ultimo fallimento	Durata ultimo
		<a href="#">bellomo-develop</a>	22 days (#123)	1 mo 15 days (#113)	9 min 27 sec
		<a href="#">garofalo-develop</a>	11 mo (#145)	11 mo (#141)	5 min 52 sec
		<a href="#">querrera-develop</a>	4 mo 29 days (#22)	N.D.	8 min 11 sec
		<a href="#">job-template</a>	N.D.	N.D.	N.D.
		<a href="#">lobuono-develop</a>	14 days (#23)	3 mo 17 days (#19)	10 min
		<a href="#">ologram-develop</a>	16 days (#90)	1 yr 3 mo (#26)	10 min
		<a href="#">ologram-master</a>	3 mo 26 days (#7)	N.D.	3 min 25 sec
		<a href="#">oloturni</a>	N.D.	1 mo 9 days (#1)	10 sec
		<a href="#">vetri-develop</a>	1 yr 3 mo (#1)	N.D.	1 min 44 sec
		<a href="#">vetro-develop</a>	16 days (#54)	4 mo 10 days (#50)	10 min

Icona: [S](#) [M](#) [L](#)

[Legenda](#) [RSS tutti](#) [RSS solo fallimenti](#) [RSS solo le ultime build](#)

**Jenkins** cerca admin | esci

Jenkins guerrera-develop ENABLE AUTO REFRESH

[Ritorna alla Dashboard](#)

**Stato**

[Modifiche](#)

[Workspace](#)

[Effettua build](#)

[Elimina Progetto](#)

[Configura](#)

[Embeddable Build Status](#)

[Checkstyle Warnings](#)

[Duplicate Code](#)

[Plots](#)

[Clover HTML Report](#)

[Documentazione](#)

[Code Browser](#)

[Violations](#)

[Git Polling Log](#)

**Build History (andamento)**

- #22 21-mag-2014 15.19.39
- #21 30-set-2013 9.31.54
- #20 30-set-2013 5.28.46
- #19 30-set-2013 5.09.38
- #18 27-set-2013 12.33.16
- #17 27-set-2013 12.02.52
- #16 27-set-2013 7.47.07
- #15 27-set-2013 7.41.34

**Progetto guerrera-develop**

0.174	13.106	6.8	3	ANDC	0.578
LOC	NOM	NOC	NOP	AHH	1.808
					90
					612
				NOM	2.497
				CALLS	0.203
CYCLO					FANOUT
					1398
					310

● Low ● Average ● High *Generated by PHP\_Depend*

[modifica descrizione](#)

Disabilita Progetto

**Code Coverage - 74.1% (2692/3635 elements)**

**Methods** 62.3% **Statements** 75.4%

— method — statement — total

#15	<a href="#">27-set-2013 7.41.34</a>
#14	<a href="#">26-set-2013 8.26.23</a>
#13	<a href="#">26-set-2013 8.21.06</a>
#12	<a href="#">25-set-2013 12.48.28</a>
#11	<a href="#">31-ago-2013 2.58.42</a>
#10	<a href="#">26-ago-2013 19.44.22</a>
#9	<a href="#">26-ago-2013 17.42.35</a>
#8	<a href="#">11-lug-2013 9.42.02</a>
#7	<a href="#">8-lug-2013 12.41.55</a>
#6	<a href="#">5-lug-2013 17.22.25</a>
#5	<a href="#">5-lug-2013 17.20.55</a>
#4	<a href="#">5-lug-2013 17.19.00</a>
#3	<a href="#">30-giu-2013 18.05.49</a>
#2	<a href="#">28-giu-2013 10.37.33</a>
#1	<a href="#">18-giu-2013 15.53.16</a>

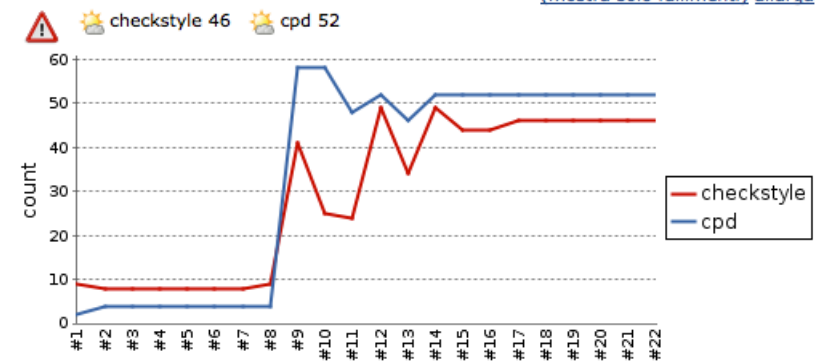
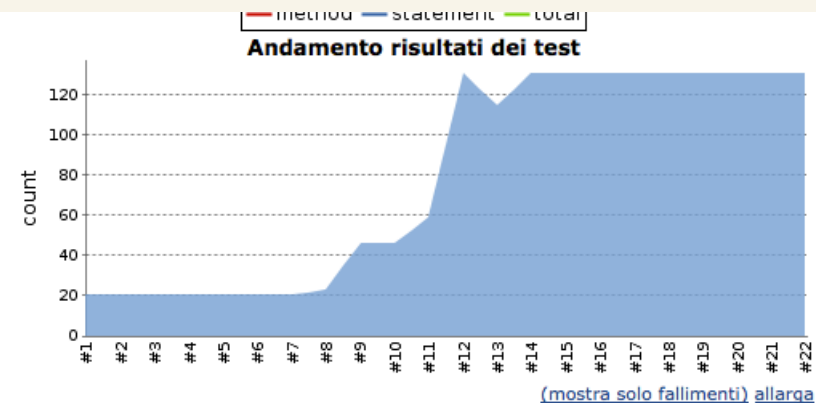
[RSS per tutto](#)
[RSS per i fallimenti](#)



[Latest Test Result](#) (nessun errore)

### Permalink

- [Last build \(#22\), 4 mo 29 days fa](#)
- [Last stable build \(#22\), 4 mo 29 days fa](#)
- [Last successful build \(#22\), 4 mo 29 days fa](#)



# Metriche del software

Le metriche software sono degli standard per misurare alcune proprietà del software. Le metriche del software permettono di valutare funzionalità e il rapporto costi/efficacia, così da realizzare software di qualità.

---

Affidabilità

Efficienza

Sicurezza

Manutenibilità

High Quality Code

# Tools: PHPLOC

Permette di misurare le dimensioni di un progetto PHP, ottenendo informazioni utili riguardanti il numero di elementi presenti.

---



***PHPLOC***  
PHP Lines of Code

## Informazioni sulle dimensioni di un progetto

linee di codice, linee di codice commentate e non commentate, numero di classi, dimensione media delle classi, media di funzioni nelle classi.



***PHP DEPEND***

## Informazioni sulle complessità di un progetto

Complessità ciclomatica rispetto alle linee di codice.



***PHPMD***  
PHP Mass Detector

## Informazioni sulle struttura

Numero di Namespace, Interfacce, Classi( Astratte e Concrete), Numero di metodi (pubblici e Non), Numero Funzioni e Numero Costanti.



***PHPCS***  
PHP Code Sniffer

**Autore:** S. Bergmann



***PHPCPD***  
Copy/Paste Detector



# 17 Tools: PHP DEPEND

Software che esegue l'analisi statica del codice sorgente.



**PHPLOC**  
PHP Lines of Code



**PHP DEPEND**



**PHPMD**  
PHP Mass Detector



**PHPCS**  
PHP Code Sniffer

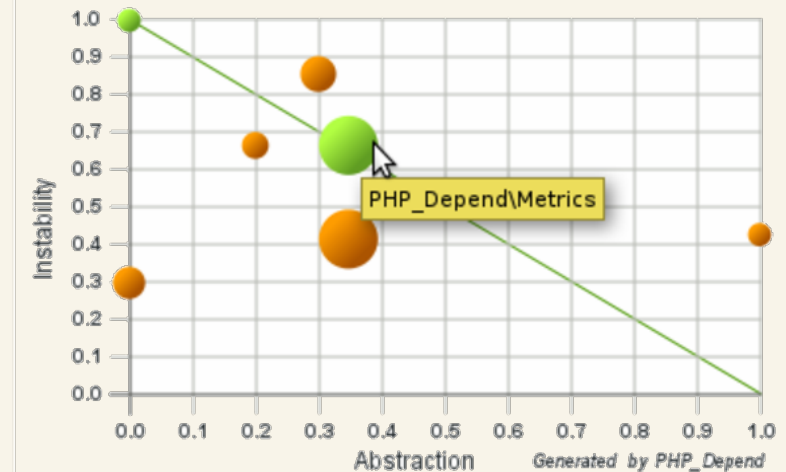
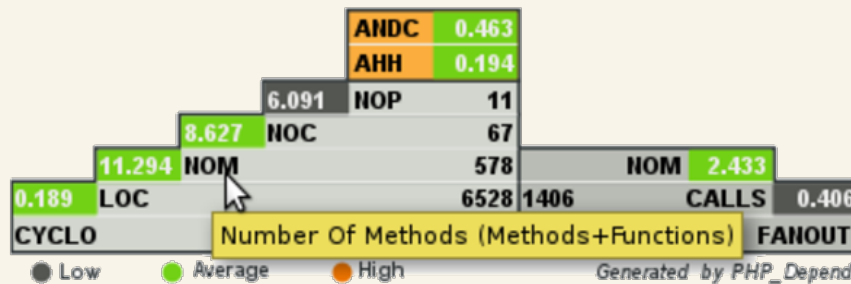


**PHPCPD**  
Copy/Paste Detector

In output genera:  
due grafici



Analizza il codice sorgente di un applicativo PHP e ne produce un albero sintattico astratto (AST)



# Tools: PHP DEPEND

Software che esegue l'analisi statica del codice sorgente.



**PHPLOC**  
PHP Lines of Code



**PHP DEPEND**



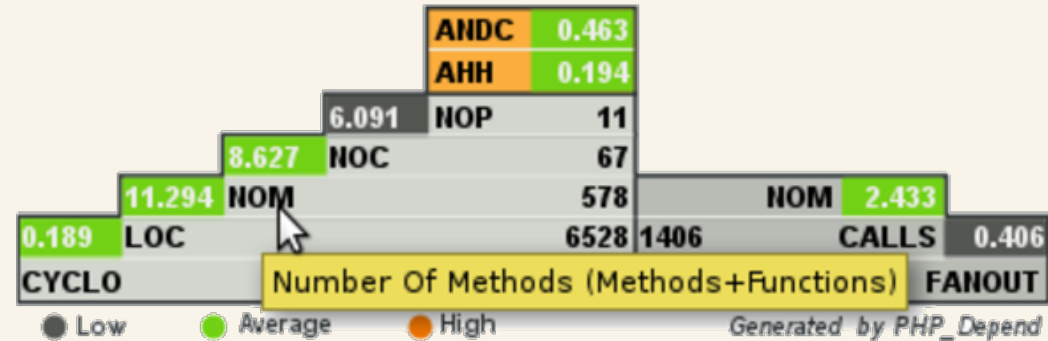
**PHPMD**  
PHP Mass Detector



**PHPCS**  
PHP Code Sniffer



**PHPCPD**  
Copy/Paste Detector



Size &  
Complexity

Inheritance

Coupling

Size & Complexity

Number Of Packages

Number of Classes

Number Of Methods

Lines Of Code

Cyclomatic Complexity

Inheritance

Average Num of Derived Classes

Average Hierarchy Height

Coupling

CALLS

FANOUT

# Tools: PHP DEPEND

Software che esegue l'analisi statica del codice sorgente.



**PHPLOC**  
PHP Lines of Code



**PHP DEPEND**



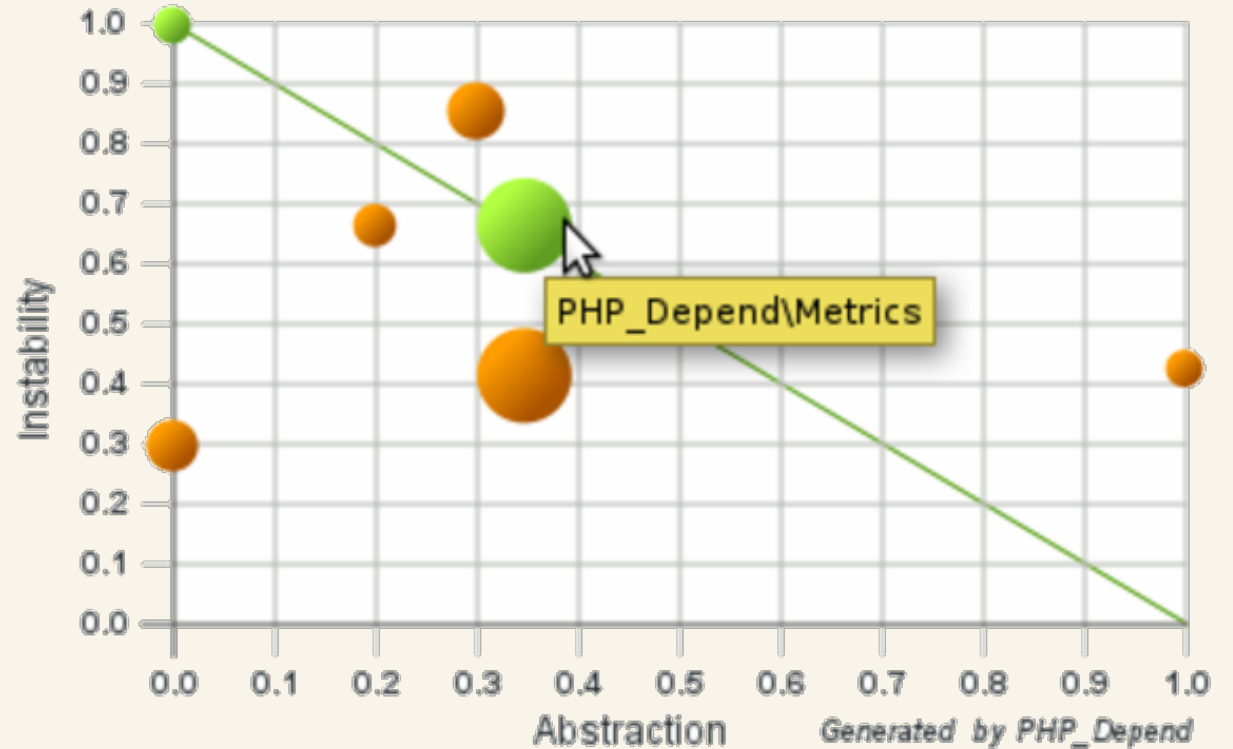
**PHPMD**  
PHP Mass Detector



**PHPCS**  
PHP Code Sniffer

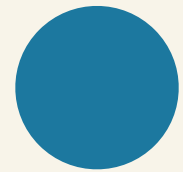


**PHPCPD**  
Copy/Paste Detector



# Tools: PHP Mess Detector

Analizza il codice sorgente di un progetto PHP e cerca di estrarne i potenziali problemi, applicando un insieme predefinito di regole per l'analisi del codice.



**PHPLOC**  
PHP Lines of Code



**PHP DEPEND**



**PHPMD**  
PHP Mass Detector



**PHPCS**  
PHP Code Sniffer



**PHPCPD**  
Copy/Paste Detector

Regole sulle  
Dimensioni

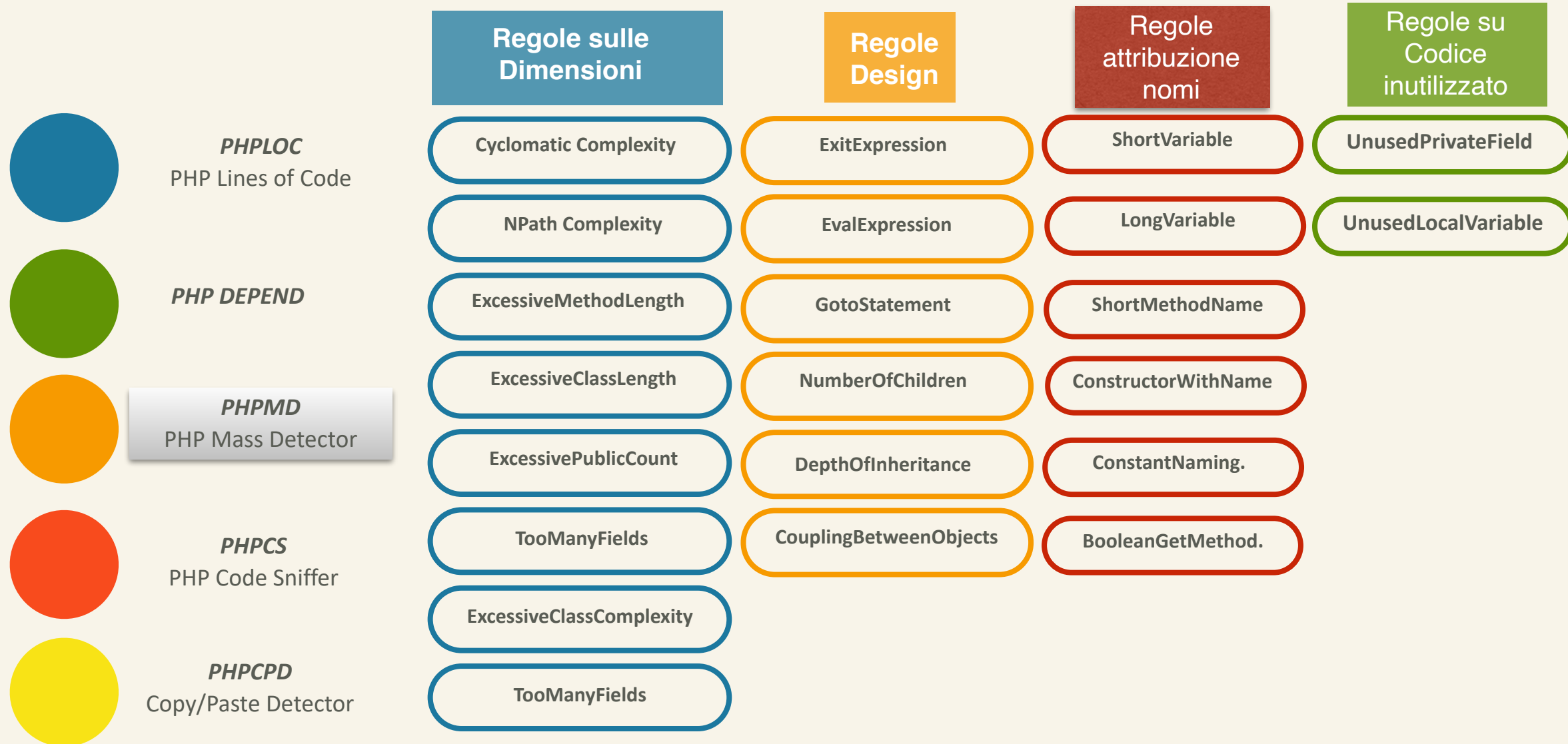
Regole  
Design

Regole  
attribuzione  
nomi

Regole su  
Codice  
inutilizzato



# Tools: PHP Mess Detector



# Tools: PHP Code Sniffer

Tool per l'analisi del codice che rileva violazioni delle convenzioni di scrittura (coding standard), così da aiutare a mantenere il codice pulito, consistente e manutenibile.

---



**PHPLOC**  
PHP Lines of Code



**PHP DEPEND**



**PHPMD**  
PHP Mass Detector



**PHPCS**  
PHP Code Sniffer



**PHPCPD**  
Copy/Paste Detector

# Tools: PHP Code Sniffer

## PHP Framework Interoperability Group



**PHPLOC**  
PHP Lines of Code



**PHP DEPEND**



**PHPMD**  
PHP Mass Detector



**PHPCS**  
PHP Code Sniffer



**PHPCPD**  
Copy/Paste Detector

[www.php-fig.org](http://www.php-fig.org)  
PHP Framework Interop Group

Agavi

AWS SDK for PHP (Amazon Web Services)

Apache log4php

Assetic and Buzz

Aura Project and Solar Framework

CakePHP

Composer and Packagist

Contao Open Source CMS

Doctrine

Drupal

eZ Publish

TYPO3 Flow

Jackalope

Joomla

Laravel

Phing

phpBB

phpDocumentor

PPI Framework

PrestaShop

Propel

PyroCMS

SabreDAV

Sculpin

Stash

SugarCRM

Symfony2

The community at large

Wikibase and Semantic MediaWiki

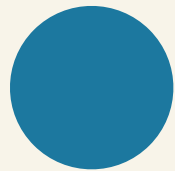
Yii framework



# Tools: PHP Code Sniffer

Tool per l'analisi del codice che rileva violazioni delle convenzioni di scrittura (coding standard), così da aiutare a mantenere il codice pulito, consistente e manutenibile.

[www.php-fig.org](http://www.php-fig.org)  
PHP Framework Interop Group



**PHPLOC**  
PHP Lines of Code



**PHP DEPEND**



**PHPMD**  
PHP Mass Detector



**PHPCS**  
PHP Code Sniffer



**PHPCPD**  
Copy/Paste Detector

**PSR-1** Garantisce un elevato livello di interoperabilità tecnica tra codice PHP condiviso.

**PSR-2** Fornisce una guida dello stile di codifica da adottare nella scrittura codice estendendo il PSR-1

**PSR-0** descrive i requisiti necessari ai quali ci si deve uniformare per garantire l'interoperabilità tra gli autoloader

**PSR-3** Descrive una interfaccia comune per le librerie di log.

**PSR-4** Descrive un nuovo standard per l'autoloading





# Tools: PHPCPD

Strumento che rivela la presenza di codice duplicato all'interno del codice sorgente



**PHPLOC**  
PHP Lines of Code



**PHP DEPEND**



**PHPMD**  
PHP Mass Detector



**PHPCS**  
PHP Code Sniffer



**PHPCPD**  
Copy/Paste Detector

```
testing@testing-VirtualBox:~/Code/ConwaysGameOfLife$ phpcpd .
phpcpd 1.3.2 by Sebastian Bergmann.

Found 1 exact clones with 203 duplicated lines in 2 files:

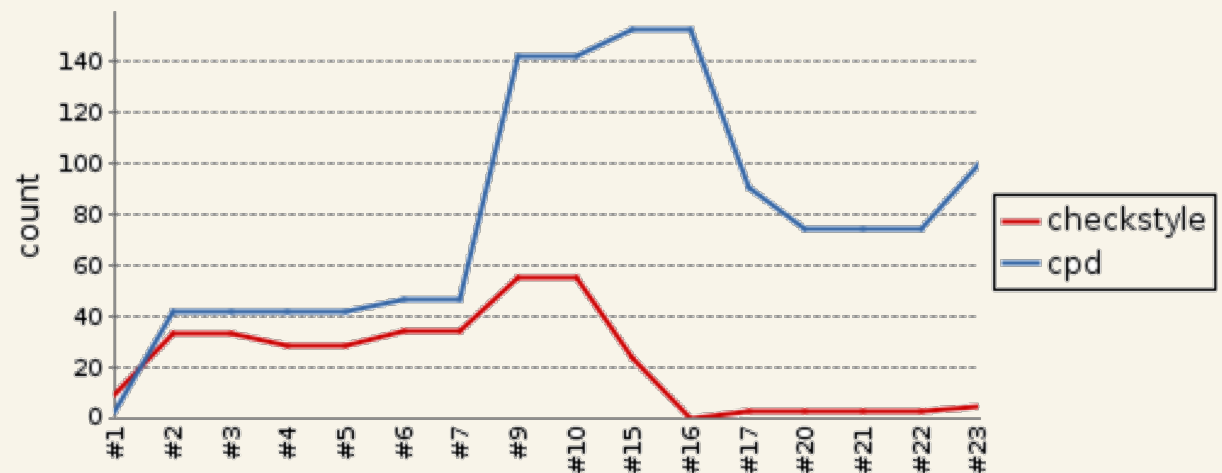
- GridCopy.php:25-228
  Grid.php:25-228

15.88% duplicated lines out of 1278 total lines of code.

Time: 0 seconds, Memory: 2.75Mb
testing@testing-VirtualBox:~/Code/ConwaysGameOfLife$
```



☀️ checkstyle 4 (+2) ☁️ cpd 100 (+26)



# Continuous Integration e High Quality Code

## GRAZIE!



Tiziano Lo Giudice  
<http://olomedia.it>



@dmondello



OlomediaFanPage



[info@danielemondello.it](mailto:info@danielemondello.it)