

Open Closed Principle (OCP)



Tell don't ask



Law of Demeter (LoD)



Continuous Integration (CI) I



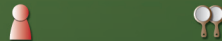
Statische Codeanalyse



Inversion of Control Container



Erfahrung weitergeben



Messen von Fehlern





Clean Code Developer

Prinzipien und Praktiken für mehr Softwarequalität

www.clean-code-developer.de

Don't Repeat Yourself (DRY)

Keep It Simple, Stupid (KISS)

Vorsicht vor Optimierungen

Favour Composition over Inheritance (FCoI)

Pladinderregel

Root Cause Analysis

Versionskontrolle

Einfache Refaktorisierungen

Täglich reflektieren

Single Level of Abstraction (SLA)

Single Responsibility Principle (SRP)

Separation of Concerns (SoC)

Sourcecode-Konventionen

Issue Tracking

Automatisierte Integrationstests

Lesen, Lesen, Lesen

Reviews

Interface Segregation Principle (ISP)

Dependency Inversion Principle (DIP)

Is-Like Substitution Principle (ISLSP)

Principle of Least Astonishment

Information Hiding Principle (IHP)

Automatisierte Unittests

Mockups

Code Coverage Analyse

Teilnahme an Fachveranstaltungen

Komplexe Refaktorisierungen

Entwurf und Impl. abkoppeln nicht

Implementation spiegelt Entwurf

You ain't gonna need it (YAGNI)

Continuous Integration (CI) II

Iterative Entwicklung

Komponentenorientierung

Test First

Open Closed Principle (OCP)

Tell don't ask

Law of Demeter (LoD)

Continuous Integration (CI) I

Statische Codeanalyse

Inversion of Control / Container

Erfahrung weitergeben

Messen von Fehlern