

PRINZIPIEN

Don't Repeat Yourself (DRY)



Keep it simple, stupid (KISS)



Vorsicht vor Optimierungen



Favour Composition over Inheritance (FCoI)



PRAKTIKEN

Pfadfinderregel beachten



Root Cause Analysis



Versionskontrollsystem einsetzen



Einfache Refaktorisierungen



Täglich reflektieren



PRINZIPIEN

Single Level of Abstraction (SLA)



Single Responsibility Principle (SRP)



Separation of Concerns (SoC)



Source Code Konventionen



PRAKTIKEN

Issue Tracking



Automatisierte Integrationstests



Lesen, Lesen, Lesen



Reviews



PRINZIPIEN

Interface Segregation Principle (ISP)



Dependency Inversion Principle



Liskov Substitution Principle



Principle of Least Astonishment



Information Hiding Principle



PRAKTIKEN

Automatisierte Unit Tests



Mockups (Testattractanten)



Code Coverage Analyse



Teilnahme an Fachveranstaltungen



Komplexe Refaktorisierungen



<http://clean-code-developer.de>

PRINZIPIEN

Open Closed Principle (OCP)



Tell, don't ask



Law of Demeter (LoD)



PRAKTIKEN

Continuous Integration (CI)



Statische Codeanalyse (Metriken)



Inversion of Control Container



Erfahrung weitergeben



Messen von Fehlern



PRINZIPIEN

Entwurf & Implementierung überlappen nicht



Implementierung spiegelt Entwurf



You Ain't Gonna Need It (YAGNI)



PRAKTIKEN

Continuous Delivery



Iterative Entwicklung



Komponentenorientierung



Test First



<http://clean-code-developer.de>

PRINZIPIEN

PRAKTIKEN

- Don't Repeat Yourself (DRY)
- Keep it simple, stupid (KISS)
- Vorsicht vor Optimierungen
- Favour Composition over Inheritance (FCoI)

- Pfadfinderregel beachten
- Root Cause Analysis
- Versionskontrollsystem einsetzen
- Einfache Refaktorisierungen
- Täglich reflektieren

- Single Level of Abstraction (SLA)
- Single Responsibility Principle (SRP)
- Separation of Concerns (SoC)
- Source Code Konventionen

- Issue Tracking
- Automatisierte Integrationstests
- Lesen, Lesen, Lesen
- Reviews

- Interface Segregation Principle (ISP)
- Dependency Inversion Principle
- Liskov Substitution Principle
- Principle of Least Astonishment
- Information Hiding Principle

- Automatisierte Unit Tests
- Mockups (Testattrappen)
- Code Coverage Analyse
- Teilnahme an Fachveranstaltungen
- Komplexe Refaktorisierungen

- Open Closed Principle (OCP)
- Tell, don't ask
- Law of Demeter (LoD)

- Continuous Integration (CI)
- Statische Codeanalyse (Metriken)
- Inversion of Control Container
- Erfahrung weitergeben
- Messen von Fehlern

- Entwurf und Implementierung überlappen nicht
- Implementierung spiegelt Entwurf
- You Ain't Gonna Need It (YAGNI)

- Continuous Delivery
- Iterative Entwicklung
- Komponentenorientierung
- Test First