

Don't Repeat Yourself (DRY)



Keep It Simple, Stupid (KISS)



Vorsicht vor Optimierungen



Favour Composition over Inheritance (FCol)



Pfadfinderregel



Root Cause Analysis



Versionskontrolle



Einfache Refaktorisierungen



Täglich reflektieren





Clean Code Developer

Prinzipien und Praktiken für mehr Softwarequalität

www.clean-code-developer.de

Don't Repeat Yourself (DRY)	1	2	3	4	5
Keep It Simple, Stupid (KISS)	1	2	3	4	5
Vorsicht vor Optimierungen	1	2	3	4	5
Favour Composition over Inheritance (FCOI)	1	2	3	4	5
Fladlinenregel	1	2	3	4	5
Root Cause Analysis	1	2	3	4	5
Versionskontrolle	1	2	3	4	5
Einfache Refaktorisierungen	1	2	3	4	5
Täglich reflektieren	1	2	3	4	5

Single Level of Abstraction (SLA)	1	2	3	4	5
Single Responsibility Principle (SRP)	1	2	3	4	5
Separation of Concerns (SoC)	1	2	3	4	5
Sourcecode-Konventionen	1	2	3	4	5
Issue Tracking	1	2	3	4	5
Automatisierte Integrationstests	1	2	3	4	5
Lesen, Lesen, Lesen	1	2	3	4	5
Reviews	1	2	3	4	5

Interface Segregation Principle (ISP)	1	2	3	4	5
Dependency Inversion Principle (DIP)	1	2	3	4	5
Is-Like Substitution Principle (ISLSP)	1	2	3	4	5
Principle of Least Astonishment	1	2	3	4	5
Information Hiding Principle (IHP)	1	2	3	4	5
Automatisierte Unittests	1	2	3	4	5
Mockups	1	2	3	4	5
Code Coverage Analyse	1	2	3	4	5
Teilnahme an Fachveranstaltungen	1	2	3	4	5
Komplexe Refaktorisierungen	1	2	3	4	5

Entwurf und Impl. abkoppeln nicht	1	2	3	4	5
Implementation spiegelt Entwurf	1	2	3	4	5
You ain't gonna need it (YAGNI)	1	2	3	4	5
Continuous Integration (CI) II	1	2	3	4	5
Iterative Entwicklung	1	2	3	4	5
Komponentenorientierung	1	2	3	4	5
Test First	1	2	3	4	5

Open Closed Principle (OCP)	1	2	3	4	5
Tell don't ask	1	2	3	4	5
Law of Demeter (LoD)	1	2	3	4	5
Continuous Integration (CI) I	1	2	3	4	5
Statische Codeanalyse	1	2	3	4	5
Inversion of Control Container	1	2	3	4	5
Erfahrung weitergeben	1	2	3	4	5
Messen von Fehlern	1	2	3	4	5