

Clean Code Developer Wertesystem

Praktiken

- 🔴 Tägliche Reflexion
- 🔴 Versionskontrolle
- 🔴 Refactoring
- 🟡 Issue Tracking
- 🟡 Automatisierte Integrationstests
- 🟡 Lesen, lesen, lesen
- 🟡 Code Reviews
- 🟡 Automatisierte Unit Tests
- 🟡 Mockups (Testatruppen)
- 🟡 Codeabdeckungsanalyse
- 🟡 Teilnahme an Fachveranstaltungen
- 🟢 Dependency Injection
- 🟢 Statische Codeanalyse
- 🟢 Erfahrungen weitergeben
- 🟢 Continuous Integration I
- 🟡 Continuous Integration II
- 🟡 Komponentenorientierung
- 🟡 Test first
- 🟡 Iterative Entwicklung

Prinzipien

- 🔴 Don't repeat yourself (DRY)
- 🔴 Keep it simple, stupid (KISS)
- 🟡 Separation of Concerns (SoC)
- 🟡 One level of abstraction
- 🟡 Single Responsibility Principle
- 🟡 Liskov Substitution Principle (OOP)
- 🟡 Interface Segregation Principle
- 🟡 Dependency Inversion Principle / Dependency Location (OOP)
- 🟢 Open-Close Principle (OOP)
- 🟢 Law of Demeter (OOP)
- 🟢 Tell, don't ask
- 🟡 Principle of least astonishment
- 🟡 Entwurf und Implementierung kreuzen nicht
- 🟡 Implementierung spiegelt Entwurf

Regeln

- 🔴 Pfadfinderregel
- 🔴 Root cause analysis
- 🔴 Vorsicht vor Optimierungen!
- 🟡 Source Code Konventionen
- 🟡 You Ain't Gonna Need It (YAGNI)