CCD

clean-code-developer.de
Principles and Practices
for better Software

Principles:
● Dont Repeat Yourself (DRY)
● Keep it simple, stupid (KISS)
● Beware of Optimizations
● Favour composition over inheritance

Practices:
● Using a Version Control System
● Refactoring Patterns Rename and Extract Method
● Scout Rule (leave a better place)
● Root Cause Analysis
● Daily Reflecton

Principles:
● One level of abstraction
● Single Responsibility Principle
● Separation of Concerns (SoC)
● Source Code Convention

Practices:
● Issue Tracking
● Automatic Integration Tests
● Read, Read, Read
● Reviews

Principles:
● Implementation matches Design
● You Ain't Gonna Need It (YAGNI)
● Separation of Design and Implementation

Practices:
● Continuous Deployment
● Iterative Development
● Component Orientation
● Test first (TDD)

All of it
for 21 days

Principles:
● Information hiding principle
● Principle of least astonishment
● Liskov Substitution Principle (LSP)
● Interface Segregation Principle (ISP)
● Dependency Inversion Principle (DI)

Practices:
● Attend Conferences
● Automatic Unit Tests
● Mockups
● Code Coverage Analysis
● Complex Refactorings

Principles:
● Open Closed Principle
● Tell, don´t ask
● Law of Demeter

Practices:
● Continuous Integration
● Static Code Analysis (Metrics)
● Inversion of Control Container
● Spread your Experience