

Clean Code Development

Checkliste mit den *wichtigsten* Punkten!

Die nachfolgenden Punkte sollten mit einem „Ja“ beantwortet werden, um sicherzugehen, dass es sich wirklich um Clean Code handelt.

1. Code Basics und Quality – Der Code ist verständlich und selbsterklärend!

- Es sind **nur verwendete Funktionen im Code** und keine offenen Baustellen!
- Die **Kommentare sind hilfreich** aber stellen dennoch **keinen Roman** dar!
- **Variablen und Funktionen** werden so **benannt und verwendet**, dass man sich **fragt**, ob man überhaupt **Kommentare braucht**!
- Der **Code** sollte sich **kaum wiederholen** und in Funktionen **ausgelagert werden**!
- Diese sollten nicht zu groß werden und kaum mehr als einen Zweck erfüllen! (**KISS**)
- Es wird **eine Sprache** verwendet und kein Denglisch!
- Zu erwartende **Fehler werden behandelt** und führen nicht zu einem Absturz!
- Die **Abstraktionsebene** des Codes wird **nicht vermischt** – „Doing“ versus „Calling“!
- **IDEAs bzw. andere Tools** wurden verwendet um den **Code zu überprüfen** und zu formatieren!
- Wenn der **Code häufig geändert** werden soll, sind **automatisierte Test** zu erstellen!
- Die Tests sollten dabei ungefähr **80% des Codes abdecken**!
- Bei **komplexeren Applikationen** ist die Verwendung von **Mockups** zu überprüfen!

2. Architecture, Class Design und Packages – Die Kommunikationen und die Verhältnisse zwischen den Klassen und Paketen sind logisch und nachvollziehbar!

- Eine **Klasse** ist nur für **eine Aufgabe** zuständig!
- Die **Abhängigkeiten** zwischen **Klassen** sollten **geringgehalten** werden!
- Es sind nur die **Informationen sichtbar**, die unbedingt sichtbar **sein müssen**!
- **Module** sollten einfach **erweiterbar** sein!
- Die **Enterprise Patterns** müssen im Blick behalten werden!
- Ein **Paket** ist nur für **ein Thema** zuständig!
- **Keine Kreisabhängigkeiten** zwischen **Paketen**!