



HURRA FERTIG.
Ach nee, doch nicht!

Hallo!

Vinko Novak

Lead Architect

Kontakt:

vinko.novak@scenario.de

SCENARIO

ÜBERBLICK ÜBER SCENARYO



Consulting

Wir bieten Beratungsleistungen in allen Bereichen der Softwareentwicklung und Projektdurchführung



Vorträge

Wir halten Vorträge und Schulungen auf Konferenzen aber immer öfters auch bei unseren Klienten



Review

Wir führen unabhängige Reviews von Code, Architektur und Software-Entwicklungs-Prozessen durch

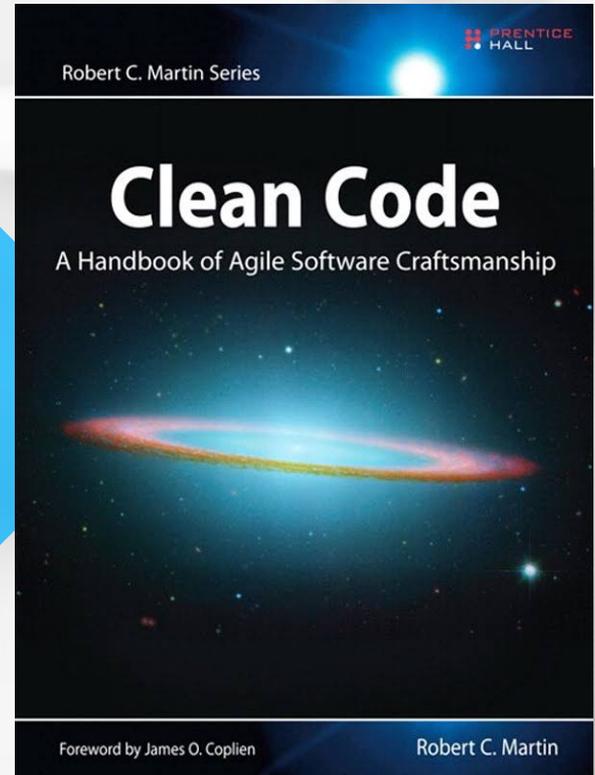


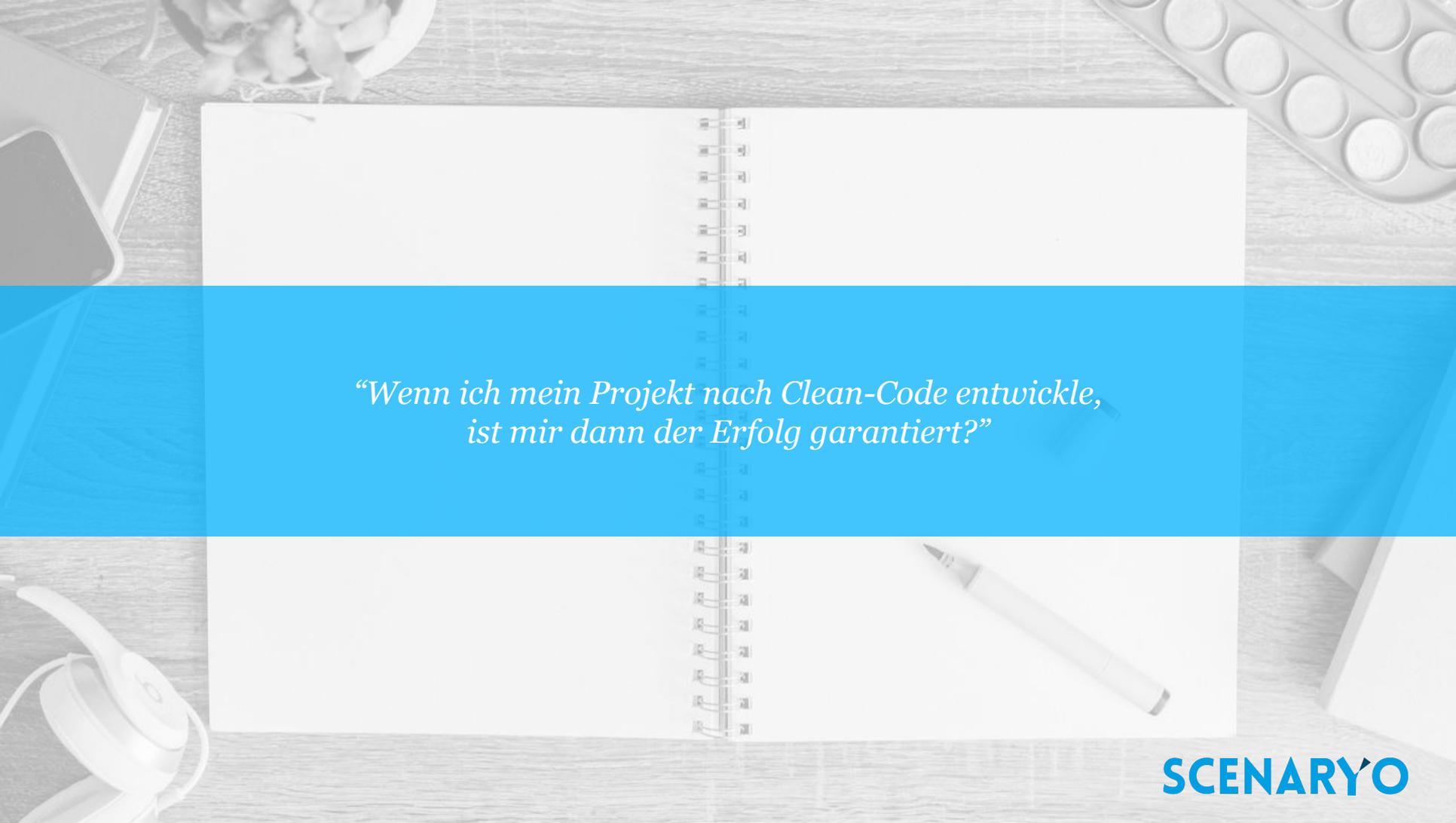
IoT

Unsere Architektur-, Entwicklungs- und Projekterfahrung nutzen wir für die Beratung und Entwicklung neuer vernetzter Lösungen

CLEAN CODE

Was ist das?

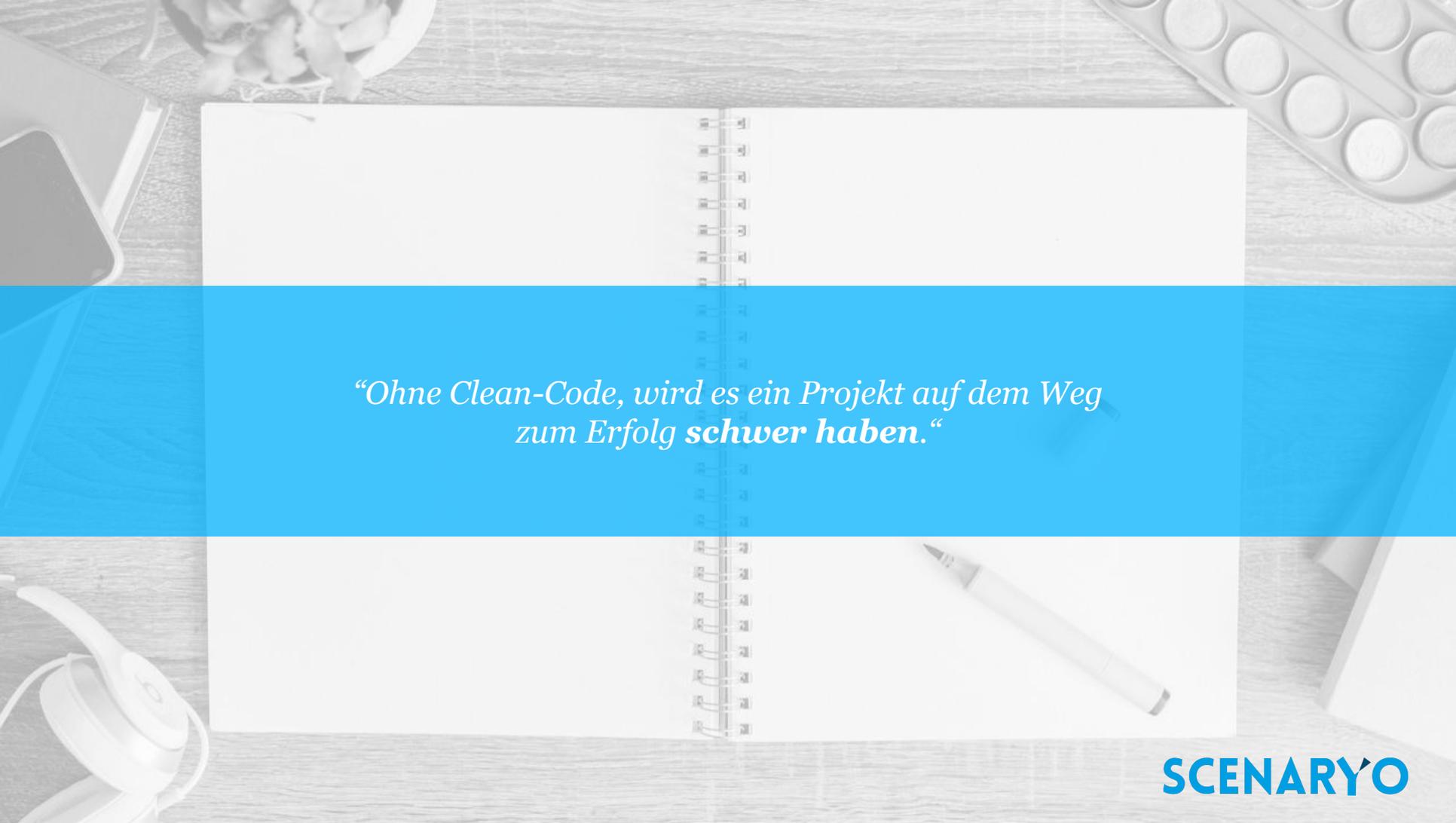


A top-down view of a desk with a spiral notebook, a pen, a water bottle, and a tray of pens. The notebook is open and blank, with a blue horizontal band across the middle containing the text. A white pen lies on the right page. A white water bottle is on the bottom left, and a tray of pens is on the top right.

*“Wenn ich mein Projekt nach Clean-Code entwickle,
ist mir dann der Erfolg garantiert?”*

CLEAN CODE IST MEHR ALS NUR “SCHÖNER WOHNEN”

- Deine Einstellung zur Arbeit
- Praktiken und Prinzipien für professionelle Softwareentwicklung
- Ständig dazulernen und sich verbessern

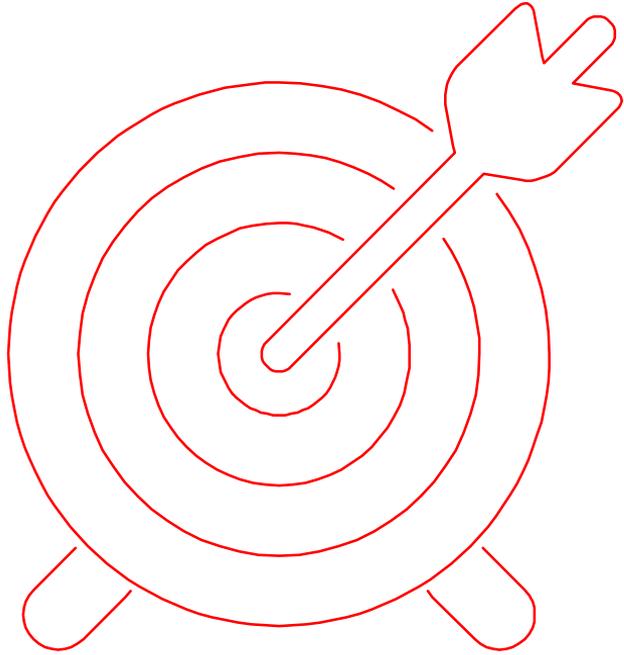
A top-down view of a desk with a spiral notebook, a pen, a water bottle, and a tray of pens. The notebook is open and has a blue horizontal band across the middle. The text is centered on this band.

*“Ohne Clean-Code, wird es ein Projekt auf dem Weg zum Erfolg **schwer haben.**“*



ERFOLG?

Was ist alles notwendig um ein Software-Projekt
erfolgreich durchzuführen?

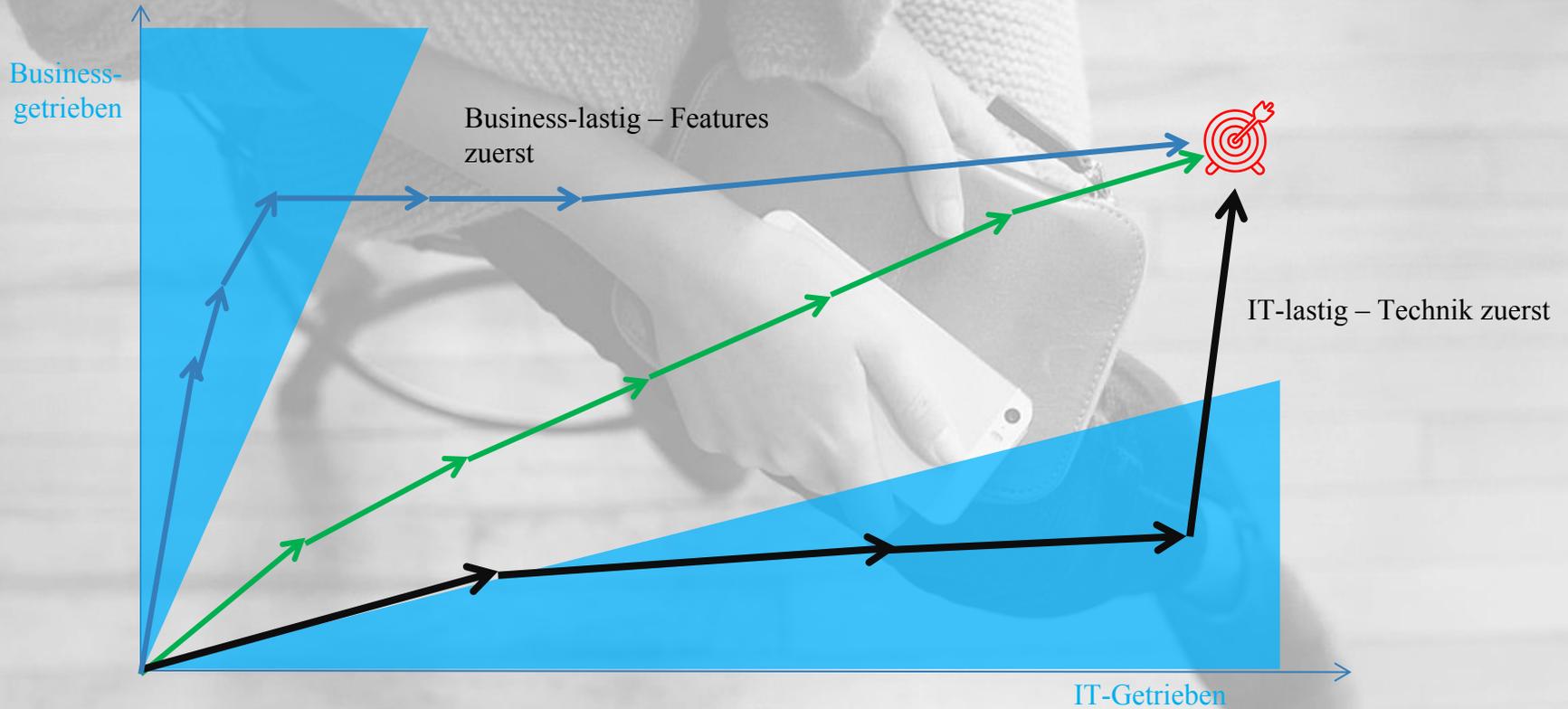


CLEAR GOAL

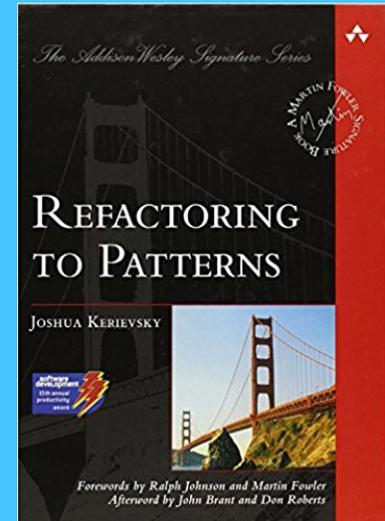
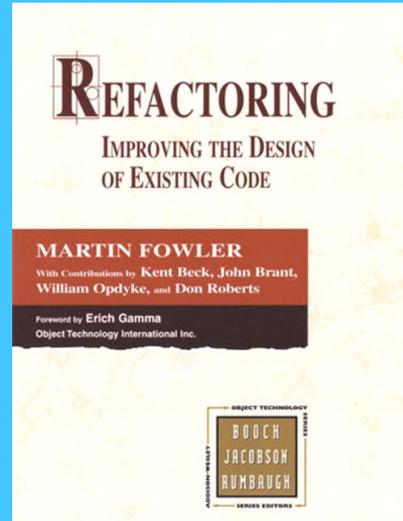
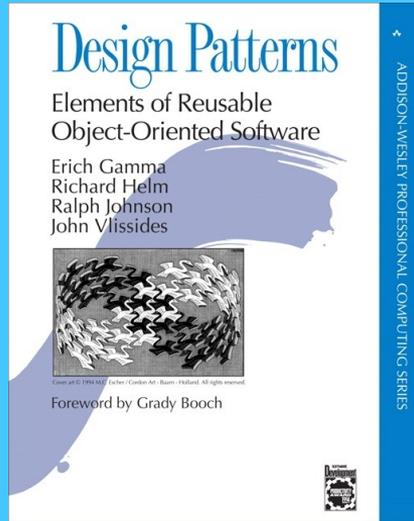
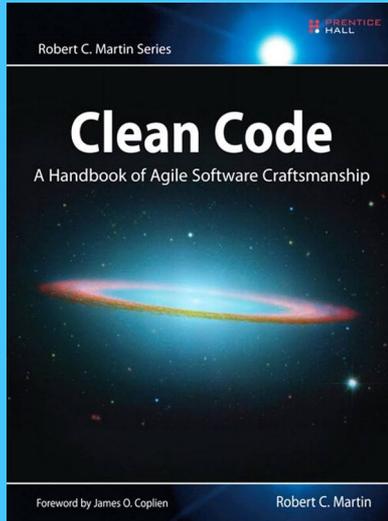
- Es ist klar, was der Kunde will
- Ich bin erfolgreich, wenn...

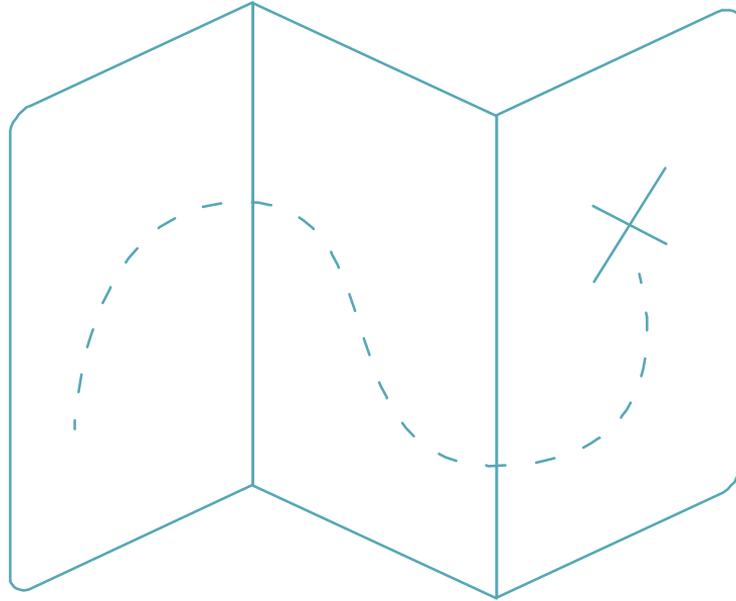


UNTERSCHIEDLICHE STRATEGIEN FÜHREN ZUM GLEICHEN ZIEL



CLEAN CODE DEVELOPMENT





CLEAR STRATEGY

- Es ist klar, wie ich zum Ziel komme
- Es ist klar kommuniziert, was ich vorhabe



WAS IST DAS?



CLEAN ARCHITECTURE

Frontend

Presentation-Layer
(MVC)



Data Transfer
Objects (DTO)

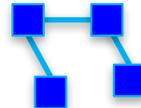
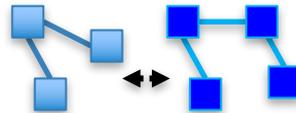
Backend

Facade-Layer

Business-Layer

Integration-Layer

Resource-Layer



Entitäten

Fehlermeldungen

GUI Framework (JSF, SWT)

i18n l10n

Validierung

Kommunikation mit Backend

Authentifizierung /
Autorisierung

M2M Transformation

Dienste

Monitoring / Logging

Transaktionsmanagement

Datenmapping / Datenzugriff
(DAO)

Datenspeicherung

→ Das Modell berücksichtigt alle wesentlichen Anforderungen

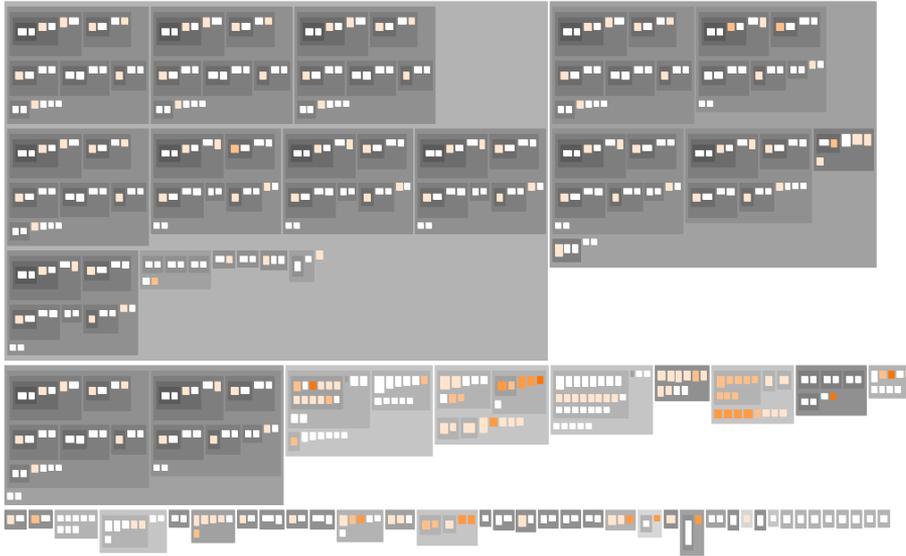
→ Kommuniziert die Konzepte



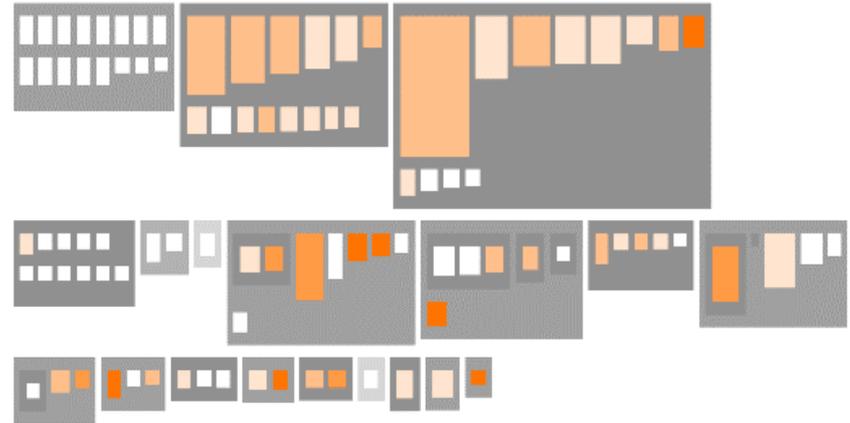


**DER CODE IST DIE
ARCHITEKTUR**

DER CODE IST DIE ARCHITEKTUR

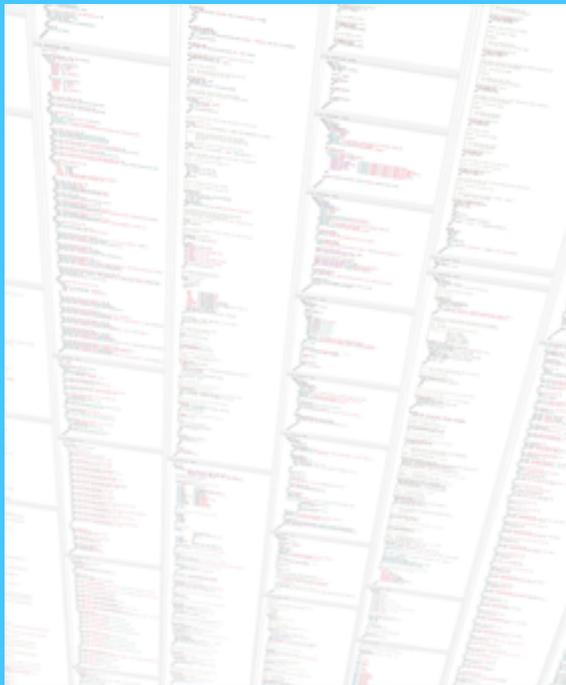


Beispiel für ein gut strukturiertes System. Die einzelnen Sub-Module sind als Maven-Artefakte definiert und durch Schnittstellen voneinander getrennt. Änderungen in der Implementierung haben keine Auswirkungen auf das restliche System.



Komplexitätssicht auf XYZ zeigt viele große und komplexe Klassen. Die Verteilung der Klassen auf Packages und die Größe der Klassen ist unregelmäßig, was auf Probleme im Design hindeutet.

SIND DIE TESTS AUCH ARCHITEKTUR?



Unendlich lang

```
15 public class ArabicNumbersTest {  
16  
17     @Test  
18     public void shouldReturnEnglishValueForArabicNumber() {  
19         String expected = "ZERO";  
20         String actual = ArabicNumber.ZERO.getEnglishUppercaseNumber();  
21         assertEquals("String should be equal", expected, actual);  
22     }  
23 }
```

Sinnlos

```
15 public class ArabicNumbersTest {  
16  
17     @Test  
18     public void shouldReturnEnglishValueForArabicNumber() {  
19  
20  
21     }  
22  
23 }
```

Leer

CLEAN TESTS

```
@Test
public void testWorkWithoutAlternativeTitlesAndBeginsWithFilter()
    throws Exception {

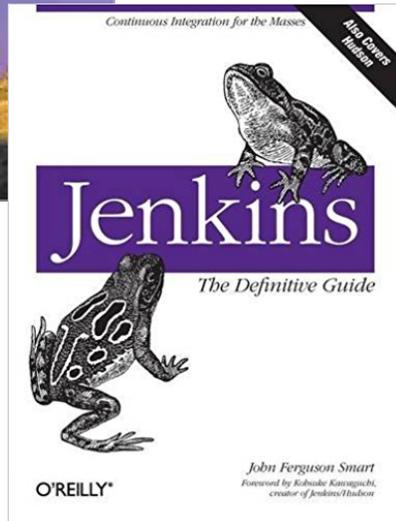
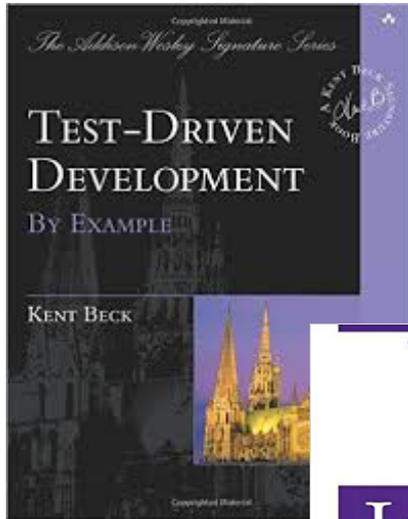
    // GIVEN
    Work work = new WorkBuilder().withTitle("GIRL ON FIRE").build();
    Work notMatchingWork = new WorkBuilder().withTitle("GIR NON MATCHING")
        .build();

    // WHEN
    WorkFilter workFilter = new FilterBuilder().withWildCards("^GIRL*")
        .buildTitleWildcard();

    // THEN
    assertTrue("Title begins with GIRL. Filter should accept it.",
        workFilter.accept(work));

    assertFalse("Title begins with GIR. Filter should not taccept it.",
        workFilter.accept(notMatchingWork));
}
```





CLEAN TESTS

→ Es ist klar, wie die Architektur funktioniert

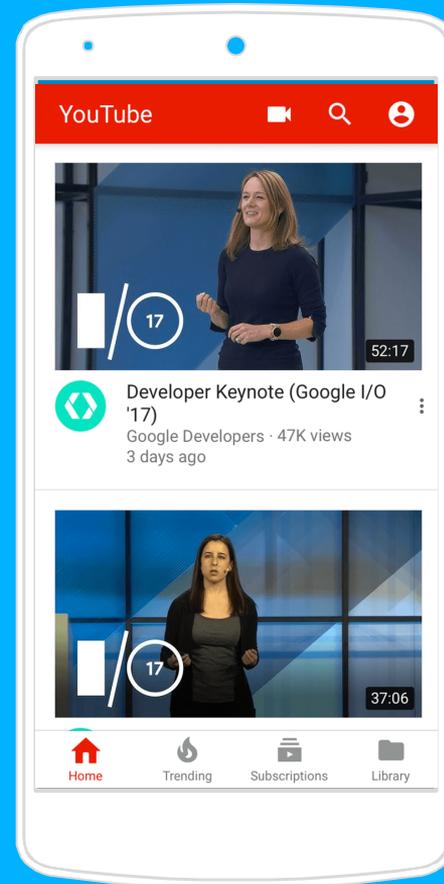
→ Es ist klar, was getestet wird

→ Es ist sichergestellt, dass die Software richtig funktioniert



BENUTZEROBERFLÄCHE

Für den Benutzer ist die
Benutzeroberfläche die Software.



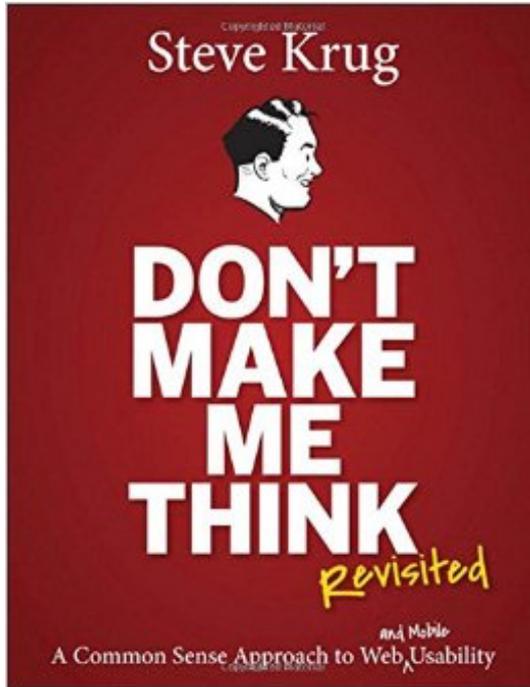
→ BENUTZEROBERFLÄCHE

→ Für den Benutzer ist die Benutzeroberfläche die Software.

→ Schlechte Benutzeroberfläche, bedeutet schlechte Software.

→ Im Corporate-Umfeld ist UX komplett vernachlässigt.





CLEAN UI/UX

- Es ist klar, wie die Software funktioniert
- Benutzer fühlt sich gut dabei



Danke!

**Starten wir eine gemeinsame
Diskussion...**

Sie können mich später unter folgender Adresse
erreichen:

vinko.novak@scenario.de

SCENARIO

CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation by [SlidesCarnival](#)
- Photographs by [Death to the Stock Photo](#) ([license](#))