A top-down view of a person's hands typing on a laptop keyboard. A large, semi-transparent blue circle is centered over the keyboard. Inside the circle, the text "HURRA FERTIG." is written in white, bold, uppercase letters, and "Ach nee, doch nicht!" is written below it in a similar style. The background is a grayscale image of a desk with a laptop, a pair of headphones, a small potted plant, and a book.

**HURRA FERTIG.**  
**Ach nee, doch nicht!**

# Hallo!

**Vinko Novak**

Lead Architect

Kontakt:

[vinko.novak@scenario.de](mailto: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

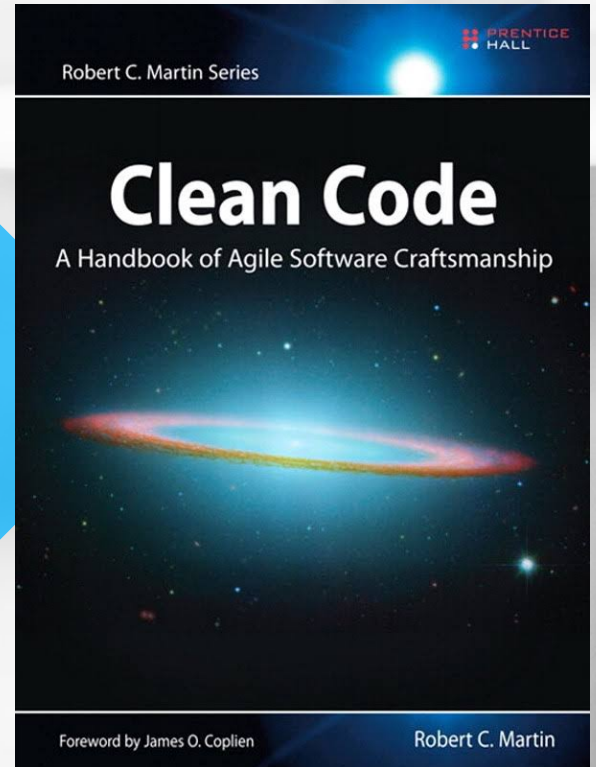


## IoT

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

# CLEAN CODE

Was ist das?





*“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

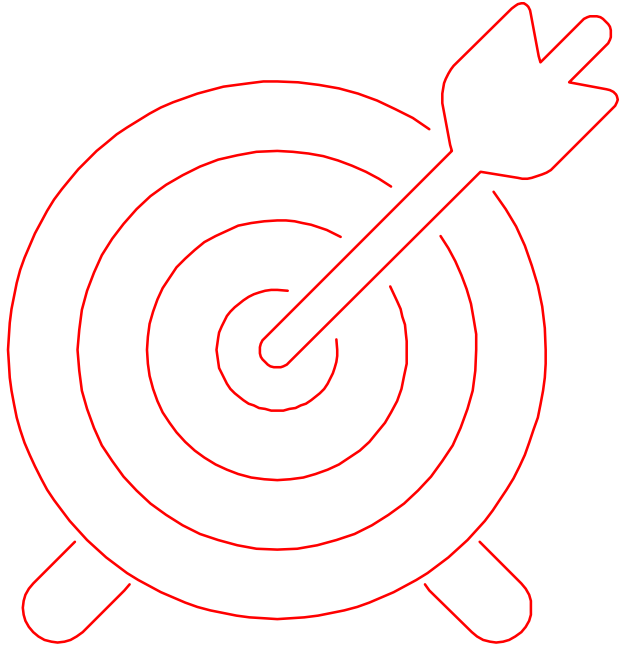
A top-down view of a wooden desk. In the center is a spiral-bound notebook with blank white pages. A silver pen lies on the bottom right page. To the top right is a grey paint palette with several white paint wells. To the top left is a small bowl of fruit. A blue horizontal band is superimposed over the middle of the notebook.

*“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?

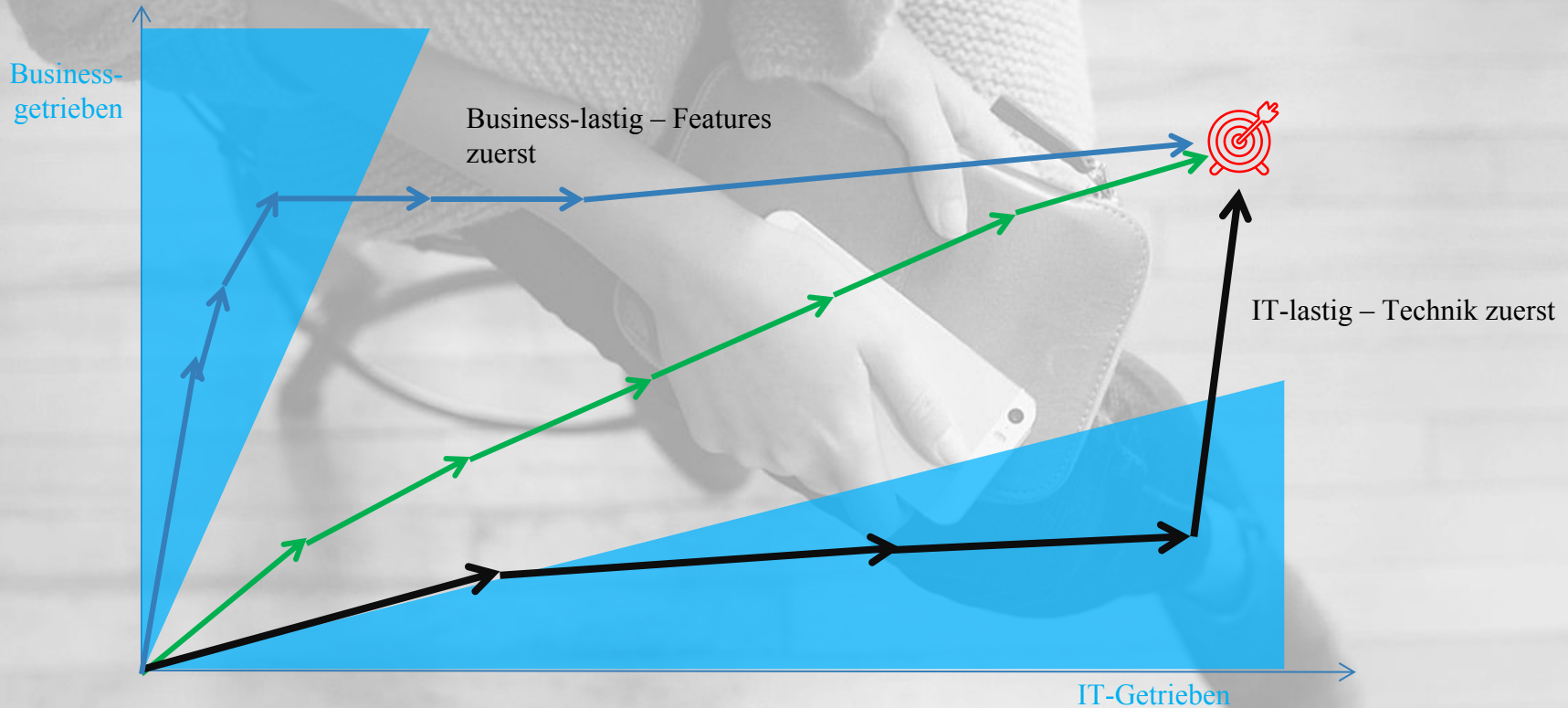


## CLEAR GOAL

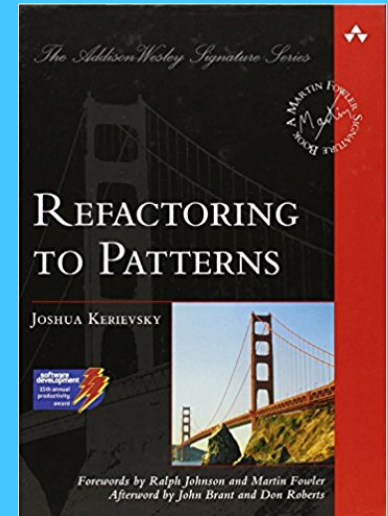
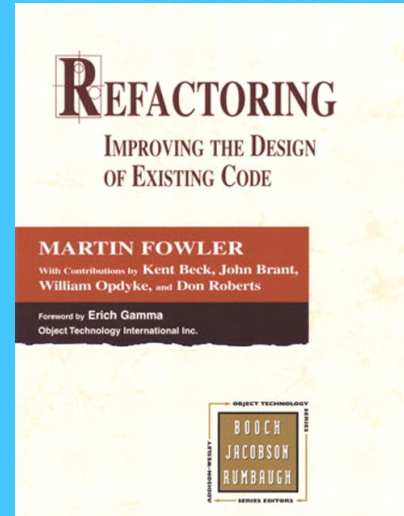
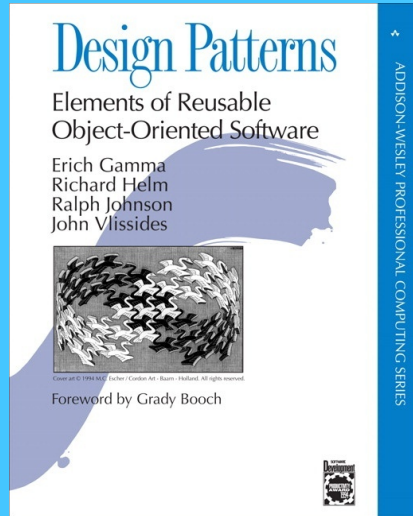
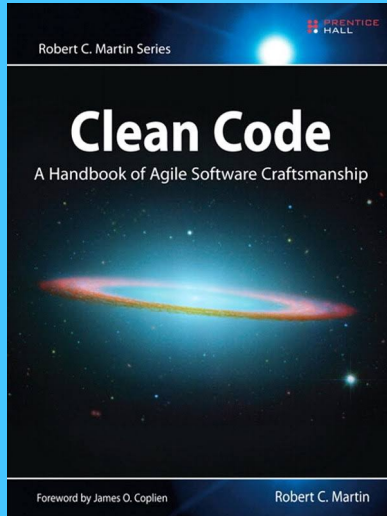
- Es ist klar, was der Kunde will
- Ich bin erfolgreich, wenn...

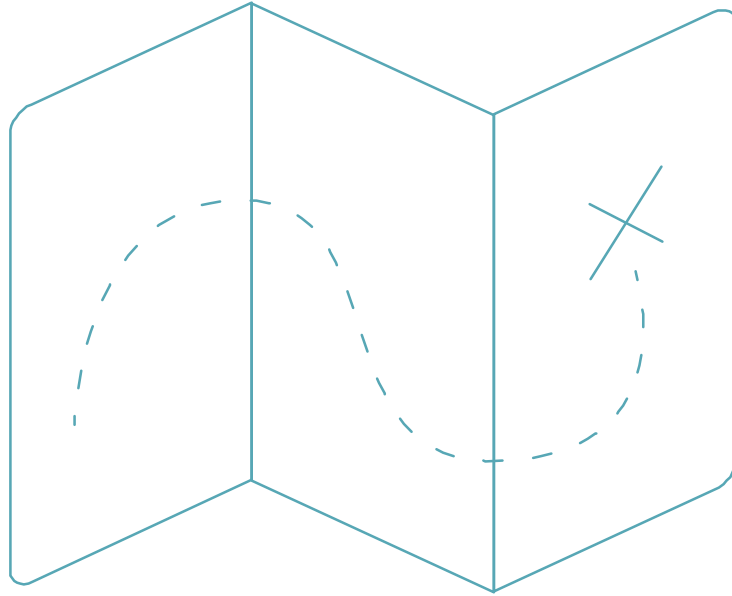


# UNTERSCHIEDLICHE STRATEGIEN FÜHREN ZUM GLEICHEN ZIEL



# 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)

Fehlermeldungen

GUI Framework (JSF, SWT)

i18n l10n

Validierung

Kommunikation mit Backend

→ Das Modell berücksichtigt alle wesentlichen Anforderungen

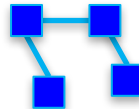
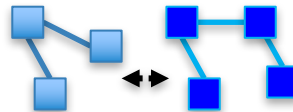
Backend

Facade-Layer

Business-Layer

Integration-Layer

Resource-Layer



Entitäten

Authentifizierung /  
Autorisierung

M2M Transformation

Dienste

Monitoring / Logging

Transaktionsmanagement

Datenmapping / Datenzugriff  
(DAO)

Datenspeicherung

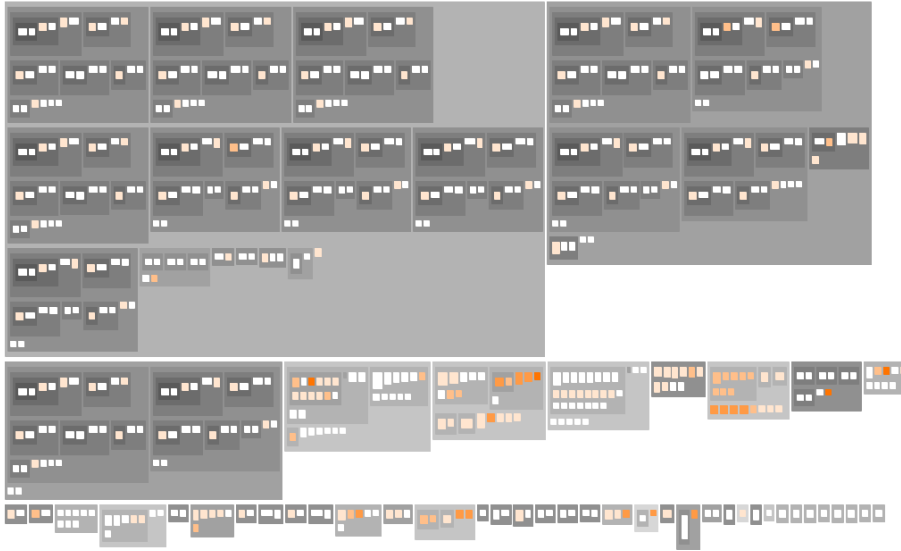
→ Kommuniziert die Konzepte



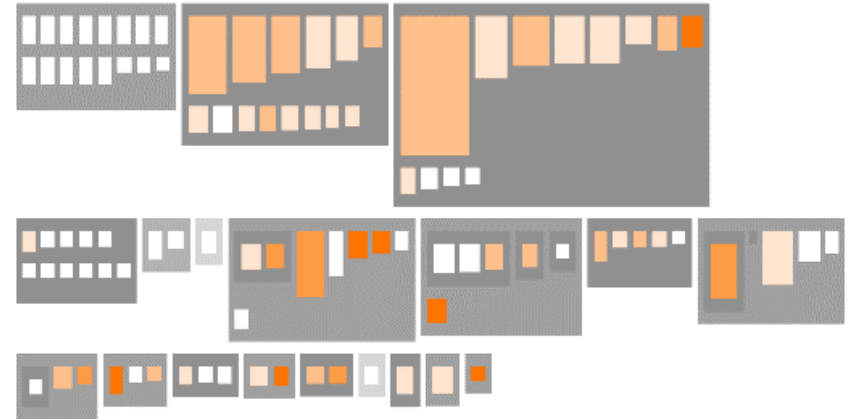


**DER CODE IST DIE  
ARCHITEKTUR**

# 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 {

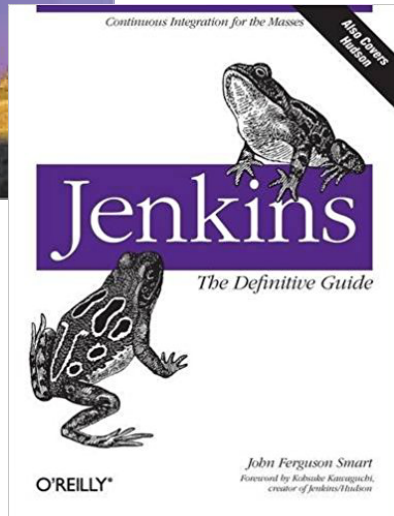
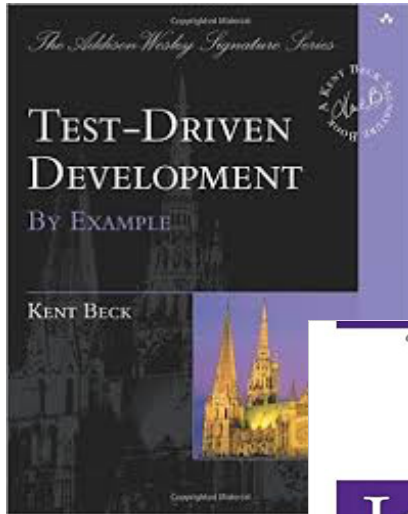
    // GIVEN
    Work work = new WorkBuilder().withTitle("GIRL ON FIRE").build();
    Work notMatchingWork = new WorkBuilder().withTitle("GIR NON MATCHING")
        .build();

    // 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));
}
```





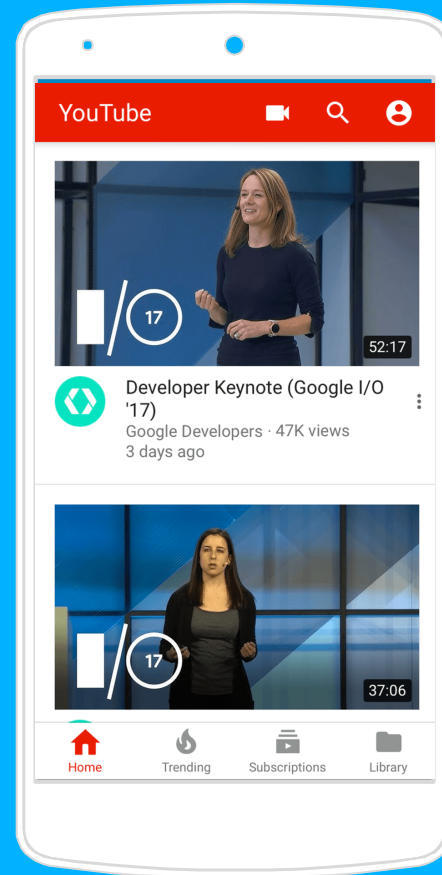
# CLEAN TESTS

- 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.



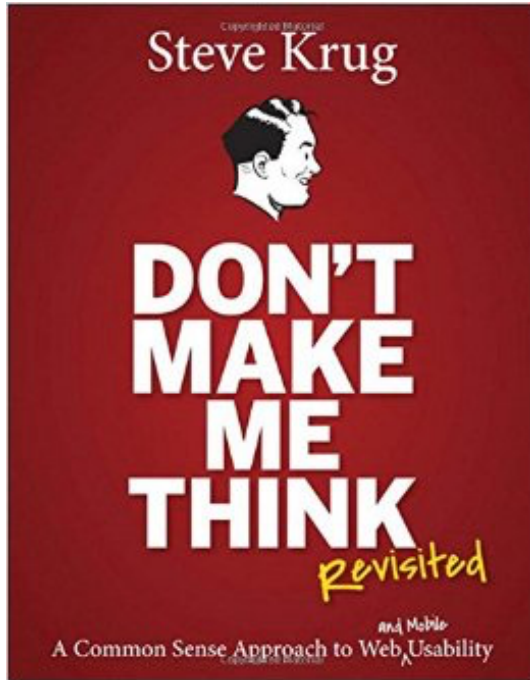
## → BENUTZEROBERFLÄCHE

→ 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](mailto: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](#))