

# Gegenseitige Beeinflussungen von Testautomatisierung, Testmanagement und Entwicklung

Jan Düttmann  
Archimedon Software +  
Consulting GmbH & Co. KG  
Marienstraße 66  
32427 Minden

Stephan Kleuker  
Hochschule Osnabrück  
Fakultät Iul  
Postfach 1940  
49009 Osnabrück

# Inhalt

- Einführung: Begriffsfeld
- Zentrale Fallstudie: Testautomatisierung für admileo (Ein- und Weiterführung)
- (exemplarische) Analyse von Erfolgskriterien der Testautomatisierung abgeleitet aus Fallstudien

# QS-Landkarte

## Testhierarchie

## Testarten

## Testeffizienz

Abnahmetest

Performance-Test

Usability-Test

Testauto-  
matisierung

Systemtest

GUI-Test

Komponententest

Regressionstest

Testwieder-  
verwendbarkeit

Unit-Test

Smoke Test

# Praxisbeispiel: Archimedon

- Unternehmen: Archimedon (Minden + Osnabrück), 15 Mitarbeiter
- zentrales Produkt: admileo
  - Verteilte Anwendung zur Abbildung von Geschäftsprozessen
  - Fokus
    - Multiprojektmanagement im Konzern
    - Product Lifecycle Management für komplexe Produkte
- modular aufgebautes komplexes Client/Server System, Java
- Fat Client mit Java Swing
- systematische klassische Software-Entwicklung mit manueller Qualitätssicherung
  
- Quelle: A. Heidt, Konzeption und Realisierung einer automatisierten Testumgebung in einem Continuous Integration Prozess für admileo, Bachelorarbeit Hochschule Osnabrück, 2012

# Praxisbeispiel: Archimedon

- Ausgangssituation:

Anforderungsanalyse: klassische Dokumentation



Entwicklung: Client-Server; bisher keine drastischen Probleme, kein Stillstand



Architektur: ein- und ausschaltbare Module



Qualitätssicherung: komplett manueller Prozess



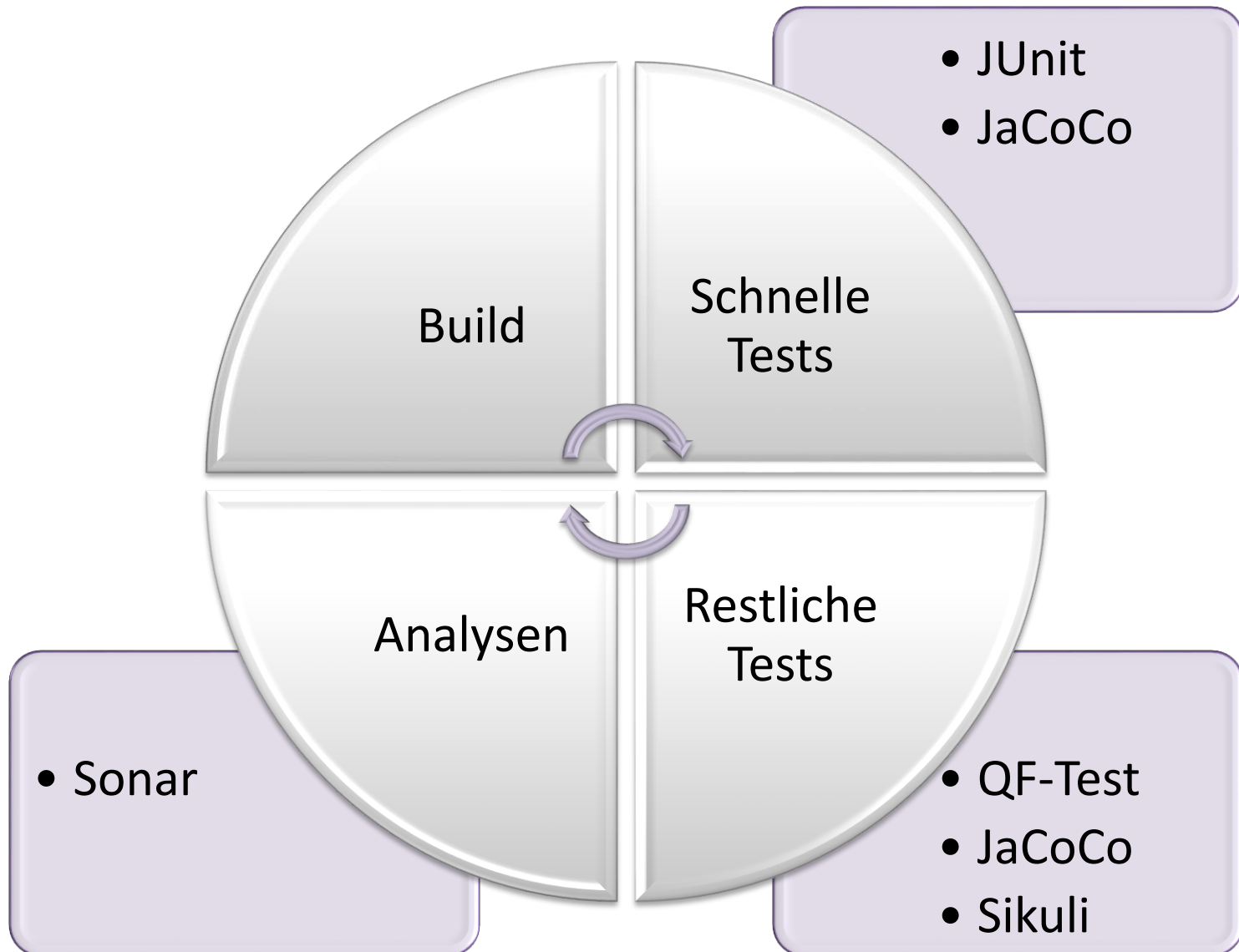
Dokumentation: Studierende schaffen Einarbeitung



Qualifikation / Schulung: Entwicklung durch Informatiker



# Neu eingeführter Testprozess



# Praxisbeispiel: Archimedon

- Ausgangssituation:

Anforderungsanalyse: klassische Dokumentation

Entwicklung: Client-Server; bisher keine drastischen Probleme, kein Stillstand

Architektur: ein- und ausschaltbare Module

**Qualitätssicherung: voll automatisierter Prozess**

Dokumentation: Studierende schaffen Einarbeitung

Qualifikation / Schulung: Entwicklung durch Informatiker



# Benefit

- Ständige Verfügbarkeit von ausführbaren Testsystemen
- Vollständig automatisierter Aufruf und Reporting der Testfälle
- Risikominimierung bei Release-Wechseln

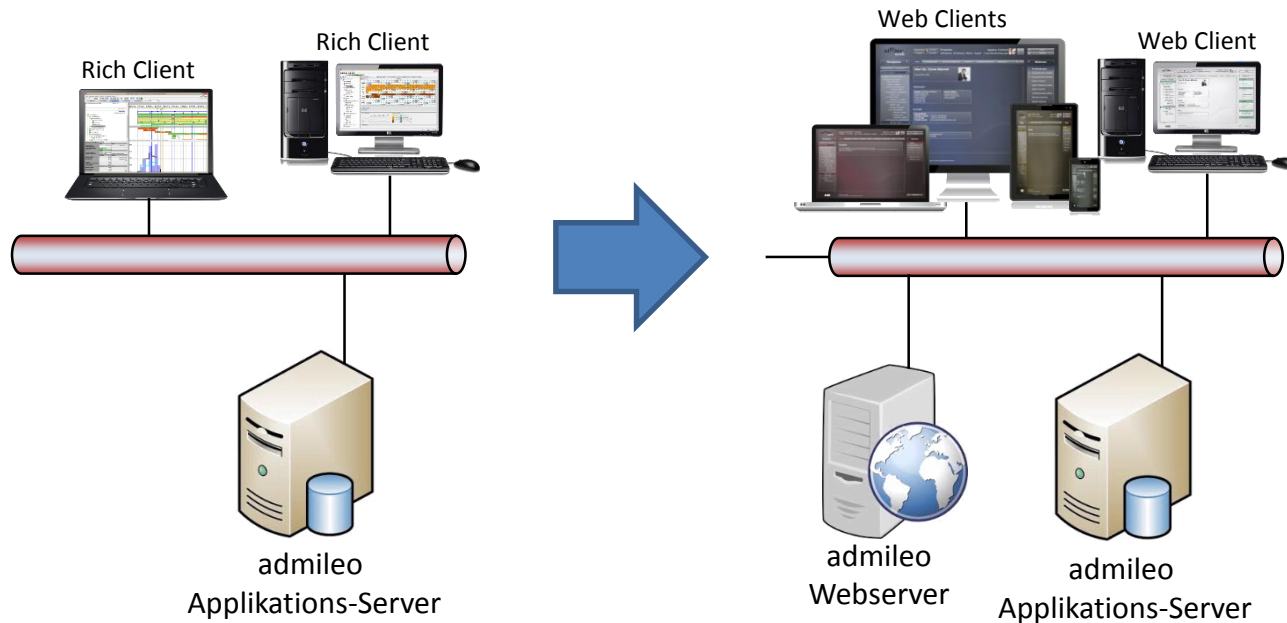




# Erfahrungen

- Technisch:
  - Verwendete Tools sind ausgereift und stabil
  - Implementierung einer Continuous Integration und Testing Umgebung gestaltet sich als reibungslos
  - Betrieb erfordert geringen Aufwand (wenige Mannstunden pro Monat)
- Organisatorisch:
  - Einführung einer zentralen QS-Stelle sinnvoll
    - => Einarbeitungstiefe minimieren
  - Definition von Schnittstellen zwischen QS und Entwicklung
  - Wiederverwendung, Aufbau von Libraries, Testarchitektur

# Zukünftige Herausforderungen

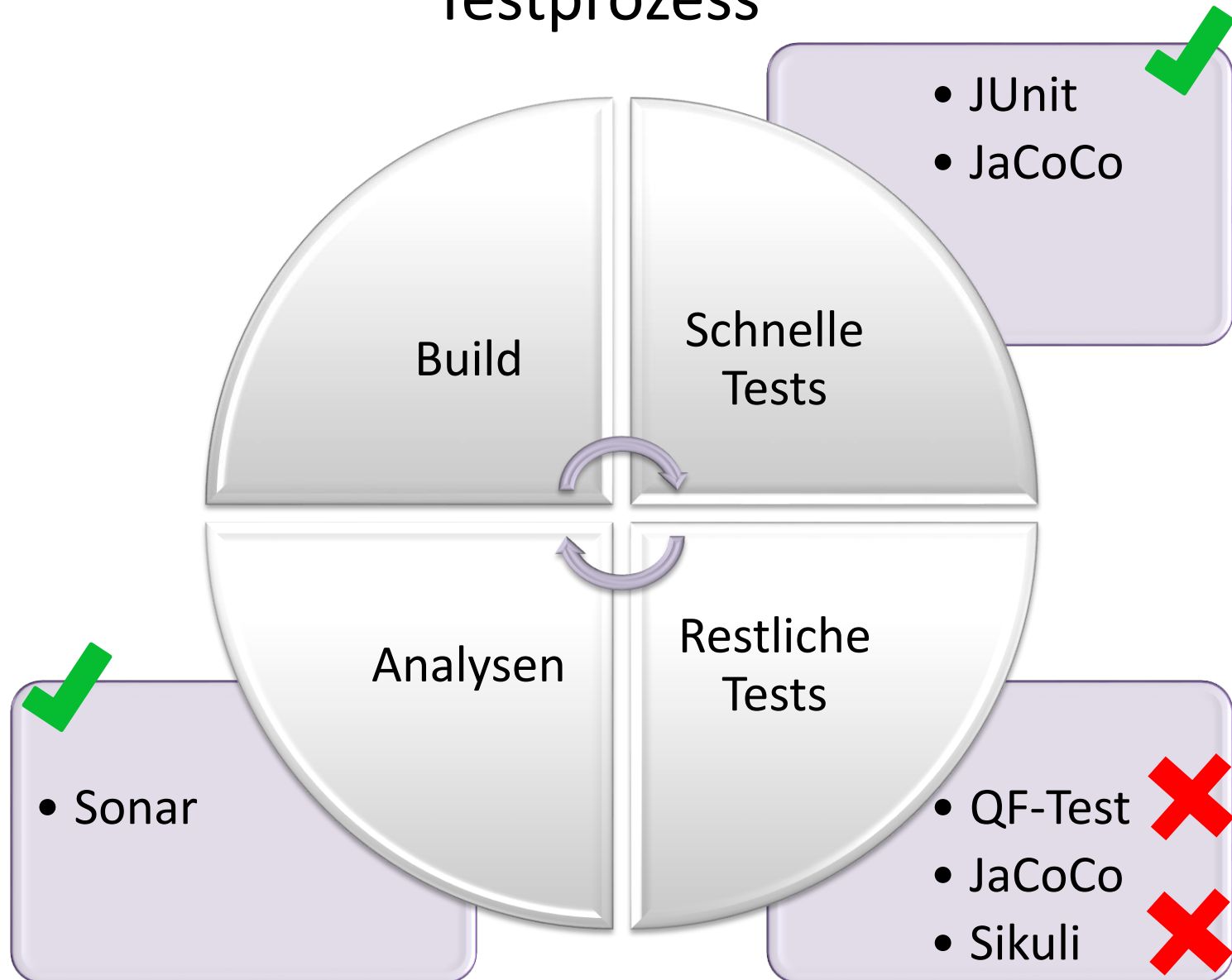


Umstieg Rich Client => Web Client (RIA)

# Zukünftige Herausforderungen



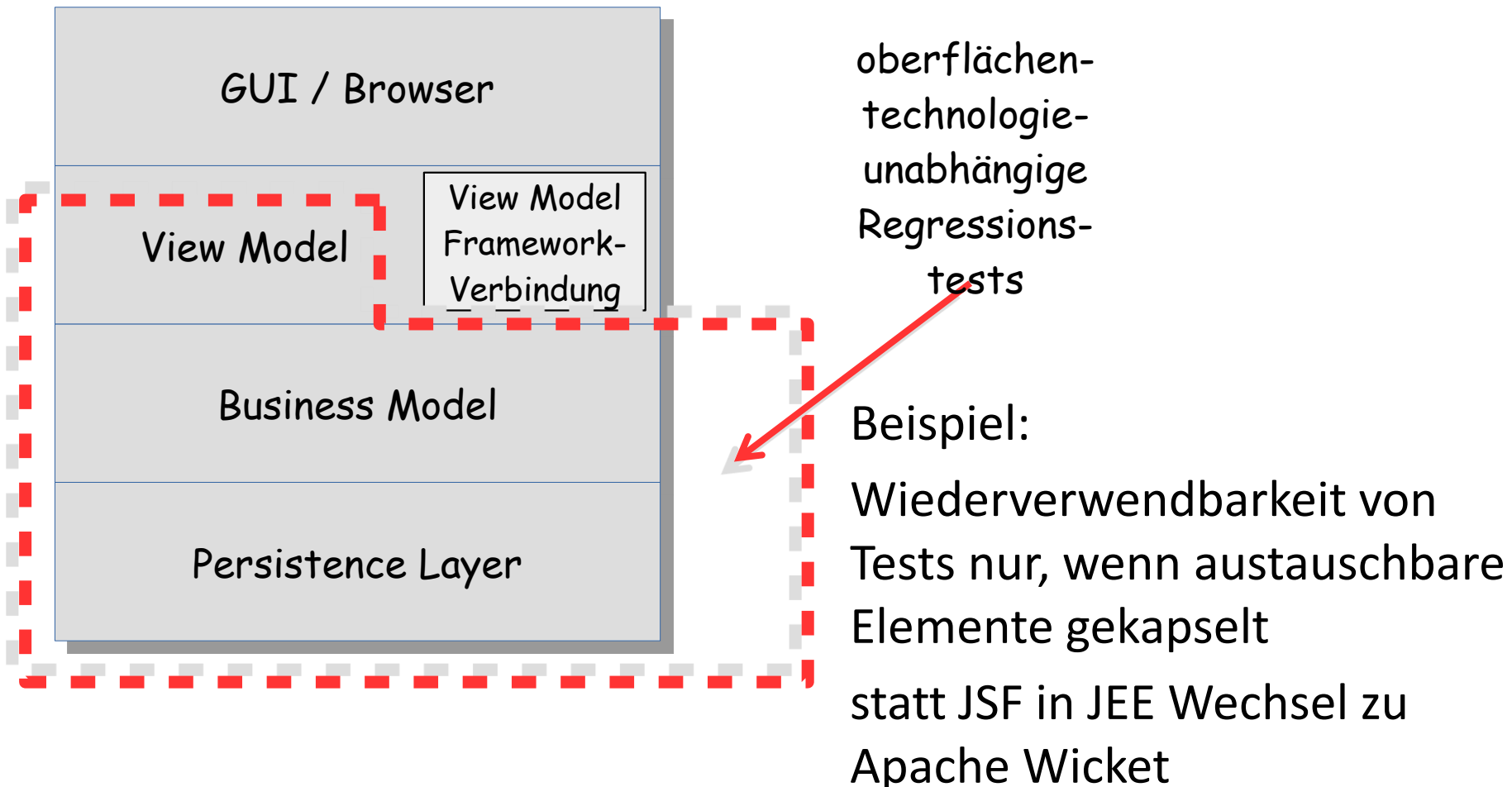
# Testprozess



# Erfolgskriterium: Testarchitektur (1/2)

## Architektur der Software (testbare Architektur)

- modular / komponentenbasiert / gegen Schnittstellen entwickelt



# Erfolgskriterium: Testarchitektur (2/2)

Architektur der Tests (wart- und erweiterbar)

- modular / hierarchisch / strukturiert

## **strukturiert**

```
class NutzeranmeldungTest  
class NutzeraktualisierungTest
```

...

„eine Test-Suite pro Web-Seite“

## **hierarchisch**

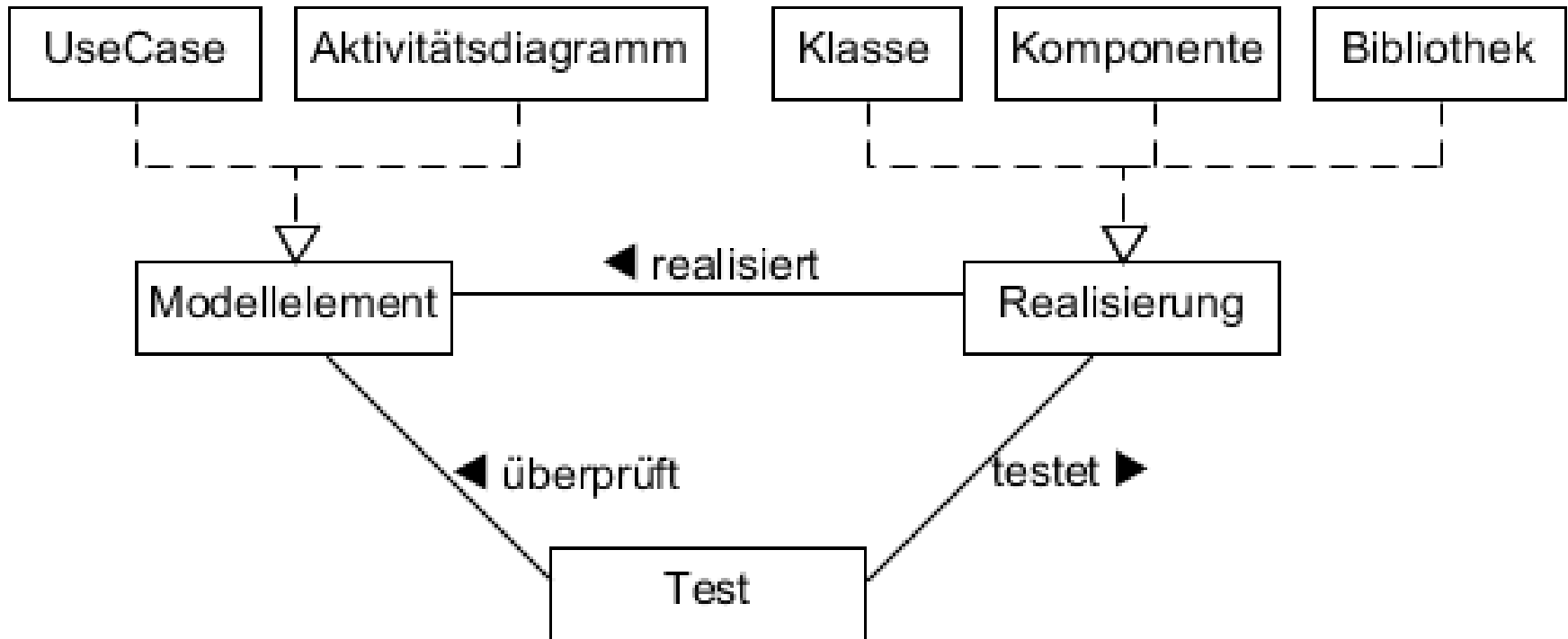
```
class NutzeranmeldungTest {  
    gui.textEingeben("Login", "Ute")
```

```
class MeineGUIToolSteuerung{  
    public void textEingeben(String feld, ...  
        Input in = X.find(feld);  
        mouse.click(in); ...
```

„Kapselung von Logik und  
Werkzeugdetails“

# Erfolgskriterium: Testorganisation

- Nachvollziehbarkeit, was wie getestet wird, Zusammenhänge



# Erfolgskriterium: Nutzung neuer Technologie (1/2)

- neue Technologie in der Entwicklung und bei Testwerkzeugen
- Beispiel: konsequente Nutzung von Dependency Injection

`@Inject`

```
private PersistenzService db;
```

...

```
public void persist(Object object) {  
    db.persist(object);  
}
```

- Vorteil Entwicklung: Entkopplung von benötigten Ressourcen vom eigentlichen Nutzer
- Vorteil Test: Einfache Integration von Mocks
- Beispiel: Contexts and Dependency Injection (JSR 299, 346) in JEE (z. B. `@Alternative`)



# Erfolgskriterium: Nutzung neuer Technologie (2/2)

- Beispiel 2 CDI: Interceptor (Aspektorientierung 2001)

```
class NutzeranmeldungTest{                                @InterceptorQualifier
    @InterceptorQualifier                                  @Interceptor
    public void beobachtbareMethode(                       class MeinInterceptor{
        ...                                                @AroundInvoke
                                                            public void beobachten(InvocationContext ctx)
                                                            ...
    }
```

- Vorteil Entwicklung: zentrale Aufgaben (Rechte, Logging, Validierung, Transaktion) an zentraler Stelle
- Nachteil Entwicklung: unübersichtlich, welcher wann aktiv
- Vorteil Qualitätssicherung: einfacher Eingriff in Ausführung
- Nachteil Qualitätssicherung: wann wie Interceptor testen

# Erfolgskriterium: Richtiger Metrikeinsatz

- Richtige Nutzung von Überdeckungsmetriken (Zweigüberdeckung)

korrekt:

- Testfälle klassisch aus typischer Nutzung und möglichen Extremfällen konstruieren
- dann Überdeckung messen
- dann analysieren, warum bestimmte Bereiche nicht überdeckt
- dann ggfls. weitere Testfälle für Überdeckung ergänzen

falsch:

- festes Maß (95% Überdeckung) als Ziel ausgeben und hieraus Korrektheit annehmen

# Erfolgskriterium: Qualifikation

- nur gute Programmierer können gute Tests programmieren
- nicht jeder gute Programmierer schreibt gute Tests
- nur gute SW-Architekten können gute Testarchitektur erstellen
  
- zentrales Hilfsmittel: ISTQB-Zertifizierung

# Zusammenfassung

- Testautomatisierung kann Zeit sparen und Qualität erhöhen
- viele dynamische Erfolgskriterien: Personal, Prozess, Technologie
- gute QS nur bei guter Entwicklung möglich