



UNIVERSITÄT  
MANNHEIM



# Test offener, dynamischer Systeme

**Barbara Paech**

Institut für Informatik

Neuenheimer Feld 326

69120 Heidelberg

<http://www-swe.informatik.uni-heidelberg.de>

[paech@informatik.uni-heidelberg.de](mailto:paech@informatik.uni-heidelberg.de)

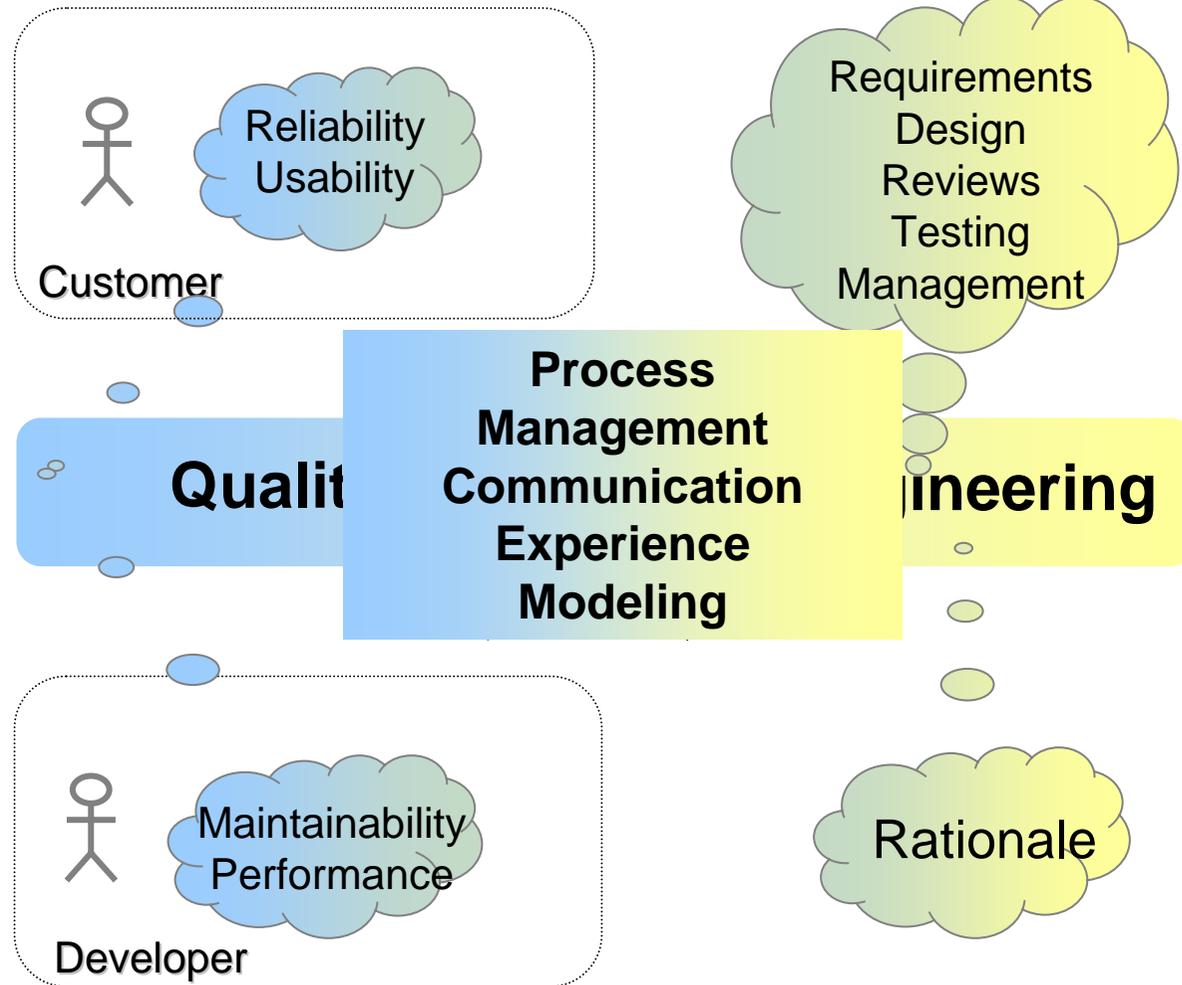


RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG

# AG Software Engineering , Uni HD

Inhalt

- Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung



## Motivation: Offene, dynamische Systeme

- ◆ **Offen:** jede Komponente kann sich mit jeder anderen verbinden, falls die Schnittstelle passt
- ◆ **Dynamisch:** die Partner der Verbindung werden erst zur Laufzeit festgelegt
- ◆ **Beispiel für ODS:** mobile Systeme



### Inhalt

- ▶ Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung



## Herausforderungen beim Test von ODS

### Inhalt



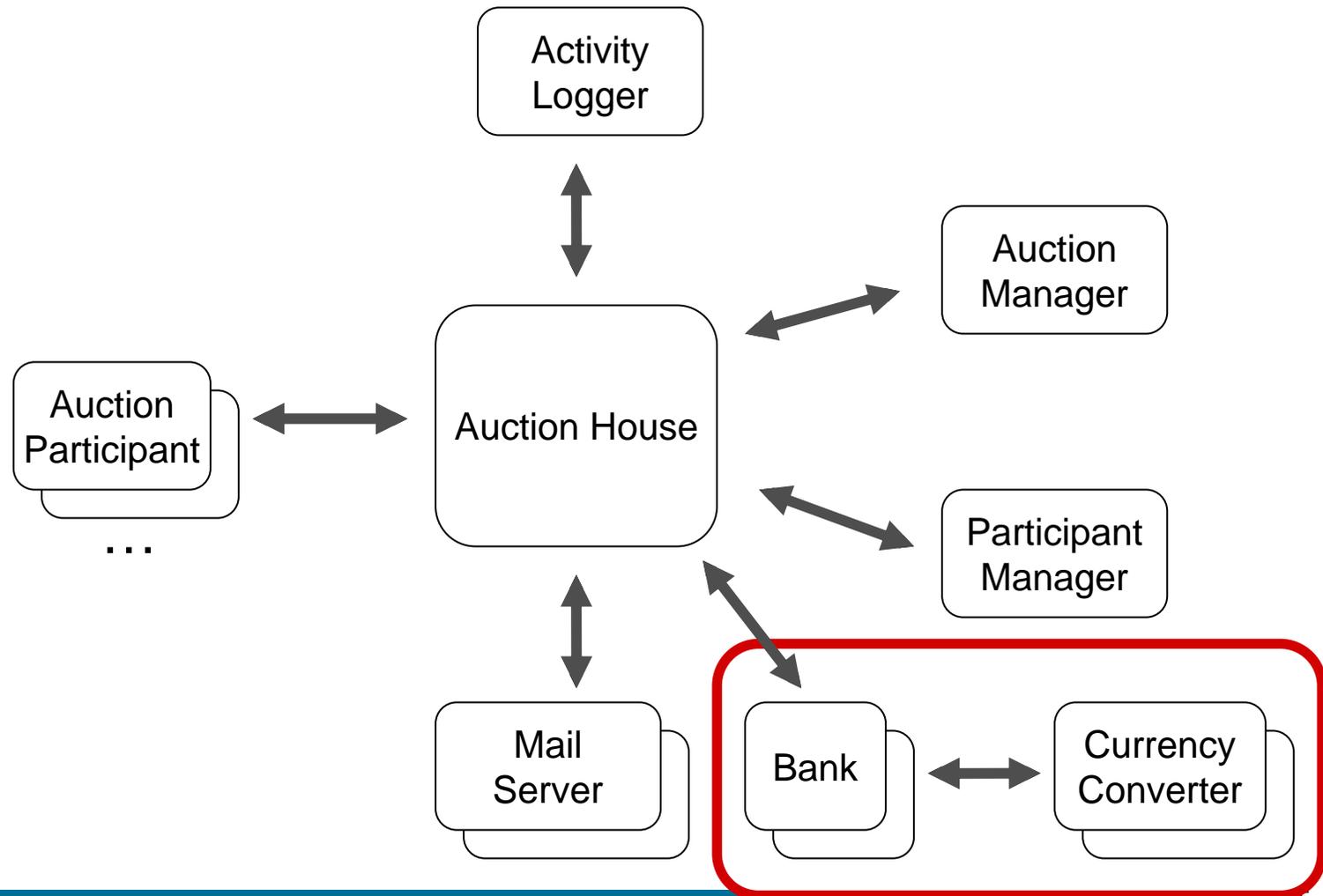
- Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung

- ◆ Bei **Komponenten** Trennung von Hersteller und Nutzer
  - Hersteller kennt Nutzungskontext nicht
  - Nutzer kennt Code nicht
  - => Test zur **Entwicklungszeit nicht ausreichend**
- ◆ Bei **Services** keine Kontrolle über Dienst
  - Z.B. Releasewechsel ohne Wissen des Nutzers
  - => Test zur **Deploymentzeit nicht ausreichend**
- ◆ Falls **häufiger Dienstwechsel**
  - => Test zur **Laufzeit durch Mensch zu umständlich**
- ◆ ODS-Test idealerweise durch **gegenseitigen Test der Komponenten zur Laufzeit**



Inhalt

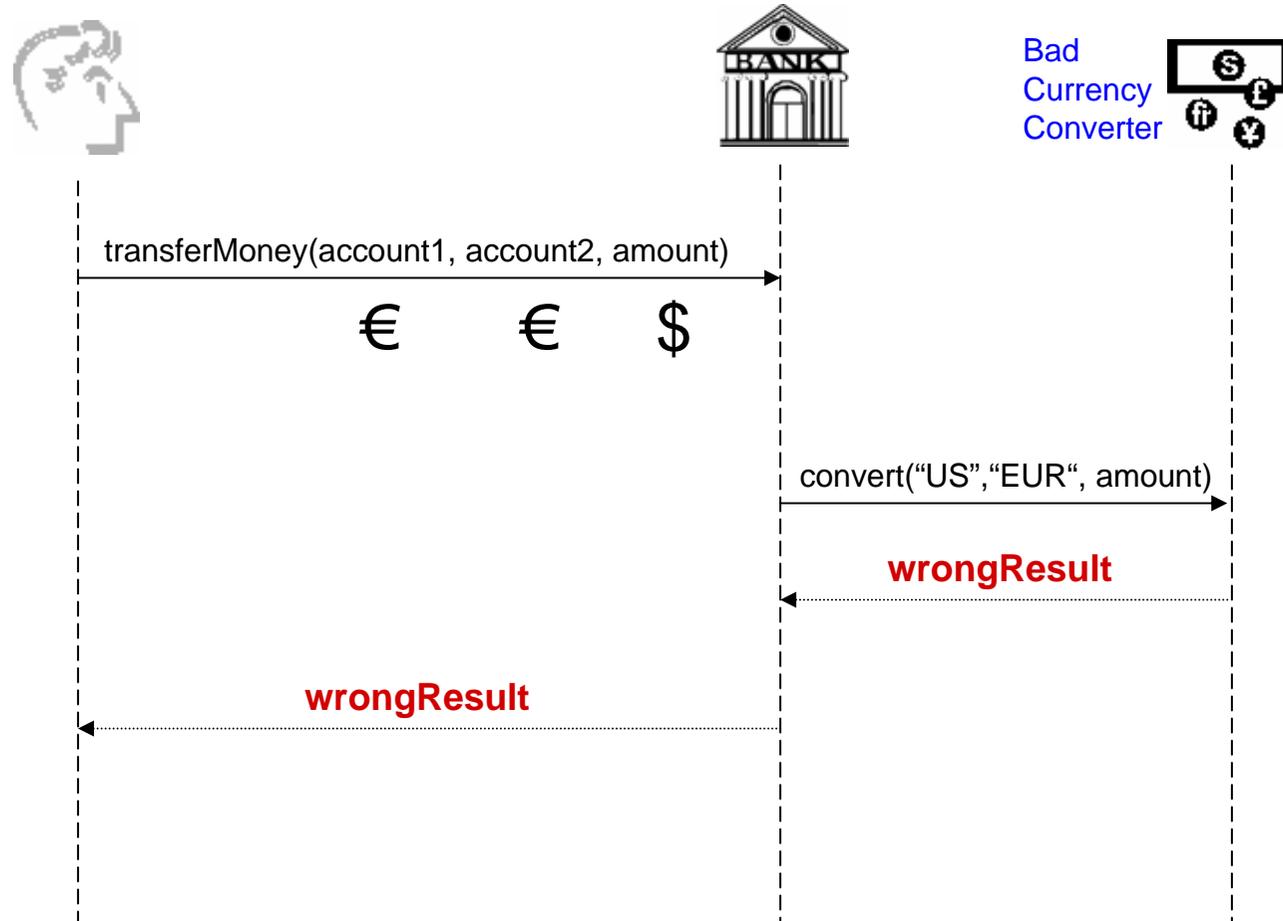
- Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung



# Fehlerhafter Ablauf ohne Test

Inhalt

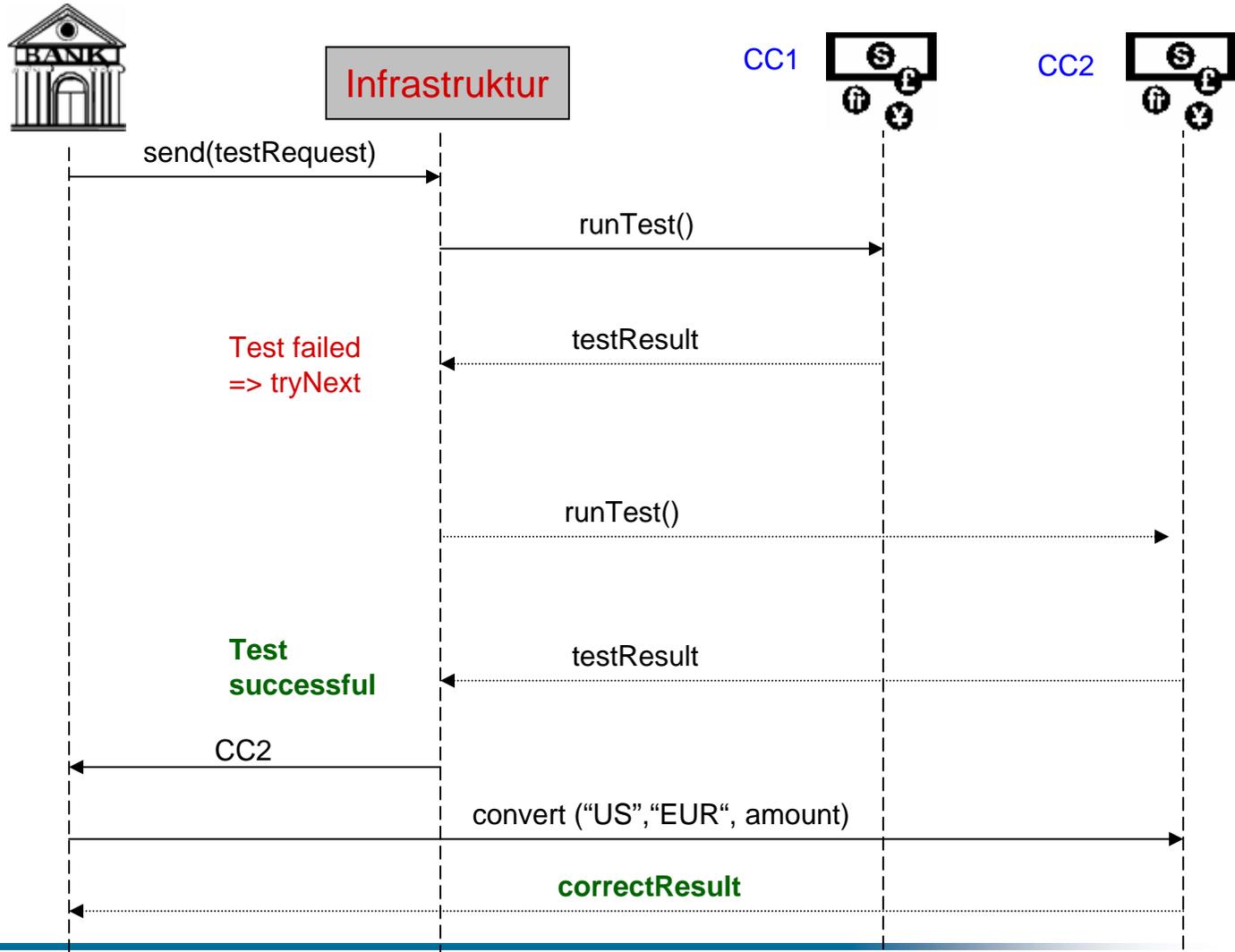
- ▶ Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung



# Ziel: Korrekter Ablauf mit Test

Inhalt

- ▶ Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung



# Kontinuierliche Tests zur Laufzeit

Inhalt



- Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung

## ◆ Testbarkeit

- Bereitstellung von Informationen für Tests (ohne Angabe des Codes), z.B. Funktionen um Zustand zu setzen oder Testfälle des Herstellers
- => Built-In-Test

## ◆ Teststrategie

- Automatisierung der Entwicklungszeittätigkeiten beim Test

## ◆ Test in verschiedenen Umgebungen

- Regressionstests
- Konfiguration von Testfällen und Testumgebung



## Inhalt

Motivation  
Built-In-Test  
Testprozess  
Testisolation  
Ressourcen  
Infrastruktur  
Methode  
Zusammenfassung



- ◆ Forschungsprojekte Component+ und MORABIT ([www.morabit.org](http://www.morabit.org))
- ◆ Bereitstellung einer **Testschnittstelle**
  - Zusätzliche Funktionen und Attribute für Test
- ◆ Bereitstellung von **Testfällen**
  - Für **Selbsttest**: d.h. um die Komponente im aktuellen Umfeld zu testen
  - Für **Vertragstest**: d.h. um zu testen, ob die von der Komponente verwendeten Komponenten die **Erwartungen** der Komponente erfüllen

# MORABIT Testprozess (1)

Inhalt

- Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung



## Testanfrage

Test-zeit

Test-fälle

Test-reaktion

Komponente



## Inhalt

Motivation  
Built-In-Test  
Testprozess  
Testisolation  
Ressourcen  
Infrastruktur  
Methode  
Zusammenfassung



## ◆ Testausführungszeitpunkte

- Lebenszyklus (z.B. creation, lookup, call)
- Systemzustand (z.B. idle, topology change)
- Zeitpunkte (z.B. periodic, random)

## ◆ Testreaktionen

- Infrastruktur: shutdown, tryNext, chooseBest
- Komponente: Anpassung der Algorithmen



# MORABIT Komponenten mit Testfällen

## Inhalt

Motivation

Built-In-Test

Testprozess

Testisolation

Ressourcen

Infrastruktur

Methode

Zusammenfassung



MORABIT RuntimeEngine Inspector

S4.J

Morabit Overview | Test Execution History | Resources | Main Components

- Configuration
  - Infrastructure configuration: Default Morabit Infrastructure (configuration with the standard Infrastructure with test mode and gui turned on)
    - Gui = true
    - InfrastructureType = class de.emlr.morabit.infrastructure.DefaultInfrastructure
    - TestMode = resource-aware
    - OptionalComponentDescriptionPostProcessor = class de.emlr.morabit.examples.ExampleComponentDescriptionPostProcessor
  - Installed Components
    - ComponentDescription: AccountManagementComponent (AccountManagementComponent)
    - ComponentDescription: BankComponent (BankComponent)
      - ImplementingClass = class org.morabit.ma.bank.BankComponent
      - Singleton = true
      - TestRequests
        - TestRequestDescription: TR for CurrencyConverter (TR for CurrencyConverter)
          - TestReaction = TryNextComponent
          - Confidence = 0.5
          - Reliability = 0.9
        - TestSuiteDescription: TS for CurrencyConverter (TS for CurrencyConverter)
          - TestCases
            - TestCaseDescription: Argument order TC (tests whether client and currencyconverter share the same understanding of invocati
              - ImplementingClass = class org.morabit.hd.testing.currencyconverter.OrderTest
            - TestCaseDescription: Currency Test (tests whether client and currencyconverter share the same understanding of currency)
              - ImplementingClass = class org.morabit.hd.testing.currencyconverter.CurrencyTest
            - TestCaseDescription: Accuracy Test (tests whether the currencyconverter result is with 2 Accuracy)
              - ImplementingClass = class org.morabit.hd.testing.currencyconverter.AccuracyTest
            - TypeUnderTest
              - TheInterface = interface org.morabit.app.interfaces.currencyconverter.CurrencyConverter
              - SimpleName = CurrencyConverter
          - TestTime = LookupTime
        - TestRequestDescription: TR for AccountManagement (TR for AccountManagement)
      - ComponentDescription: CurrencyConverterComponent (CurrencyConverterComponent)
      - ComponentDescription: DemoScenario2 (Bank-CurrencyConverter Demo Scenario)

## Inhalt

Motivation  
Built-In-Test  
Testprozess  
Testisolation  
Ressourcen  
Infrastruktur  
Methode  
Zusammenfassung



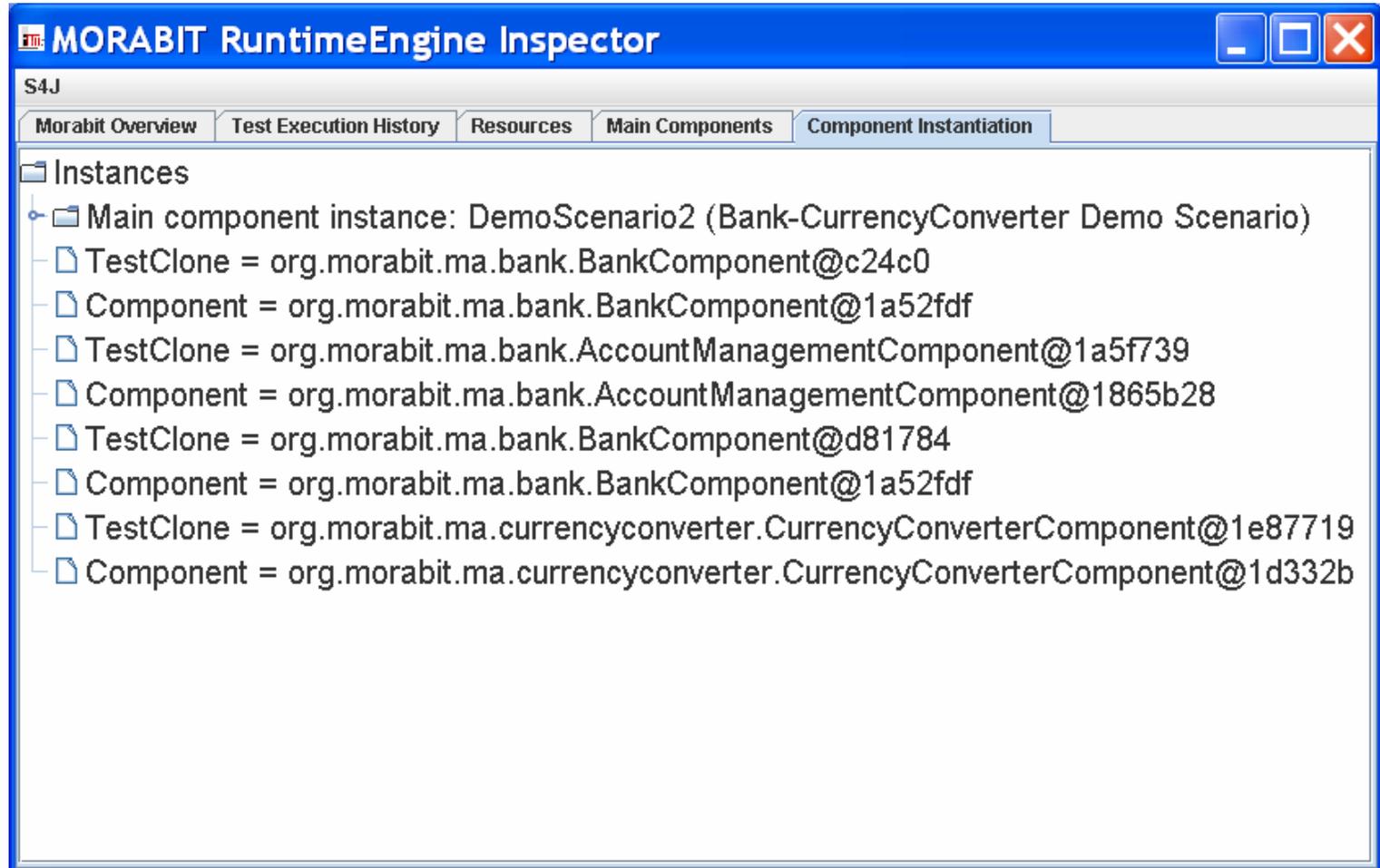
- ◆ **Trennung** der Ausführung der Testanfragen und normaler Anfragen
- ◆ **Zustandslose** Komponenten: Keine Testisolation nötig
- ◆ **Zustandsbehaftete** Komponenten: Komponente wird geklont (ggf. Komponentenspezifisches Klonen)



# MORABIT Instantiierungssicht: Komponenten und Klone

## Inhalt

- Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung



**MORABIT RuntimeEngine Inspector**

S4J

Morabit Overview | Test Execution History | Resources | Main Components | **Component Instantiation**

Instances

- Main component instance: DemoScenario2 (Bank-CurrencyConverter Demo Scenario)
  - TestClone = org.morabit.ma.bank.BankComponent@c24c0
  - Component = org.morabit.ma.bank.BankComponent@1a52fdf
  - TestClone = org.morabit.ma.bank.AccountManagementComponent@1a5f739
  - Component = org.morabit.ma.bank.AccountManagementComponent@1865b28
  - TestClone = org.morabit.ma.bank.BankComponent@d81784
  - Component = org.morabit.ma.bank.BankComponent@1a52fdf
  - TestClone = org.morabit.ma.currencyconverter.CurrencyConverterComponent@1e87719
  - Component = org.morabit.ma.currencyconverter.CurrencyConverterComponent@1d332b

## Inhalt

- Motivation
- Built-In-Test
- Testprozess
- Testisolation
- ▶ Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung

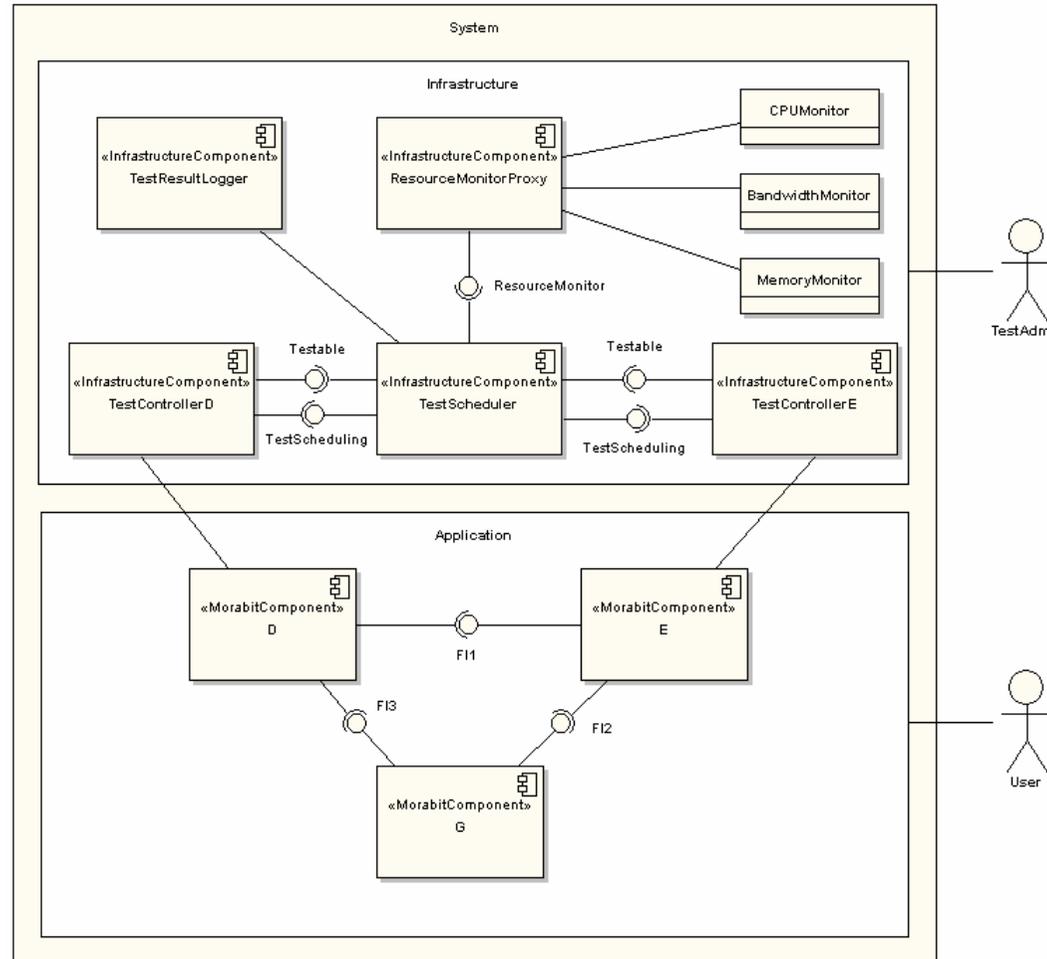


- ◆ sind beschränkt auf kleinen Endgeräten
- ◆ wichtige Beispiele:
  - CPU
  - Speicher
  - Bandbreite/Übertragungsrate
  - Batterie
- ◆ Messung als Grundlage für **Ressourcen-adaptive Teststrategien**
  - Zurückweisung oder Verzögerung von Tests
  - Priorisierte Ressourcenzuweisung
  - Partielle Testfallmengen

# MORABIT Konzeptionelle Architektur

Inhalt

- Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung



# MORABIT Testausführungshistorie

## Inhalt

Motivation

Built-In-Test

Testprozess

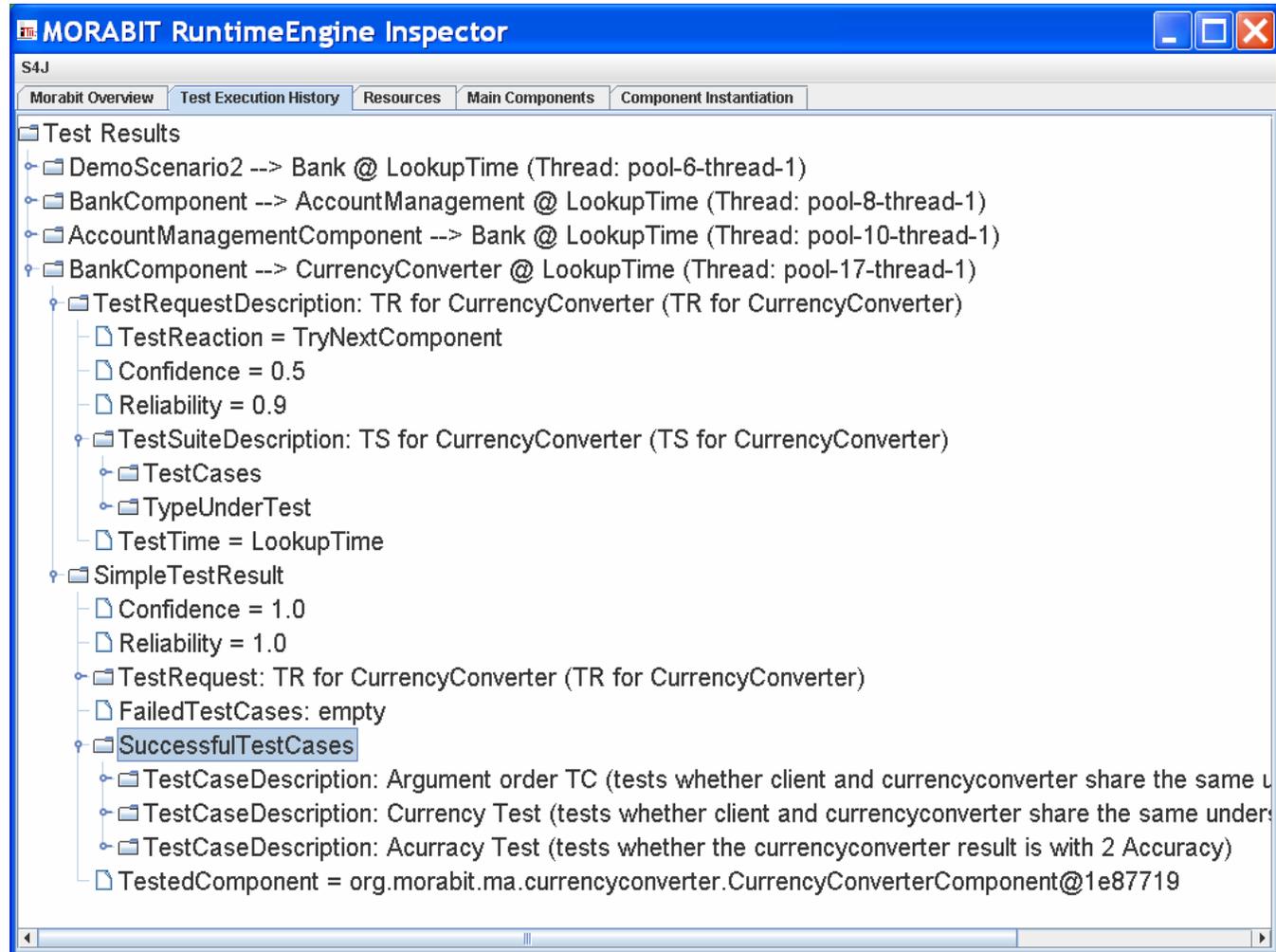
Testisolation

Ressourcen

Infrastruktur

Methode

Zusammenfassung

**MORABIT RuntimeEngine Inspector**

S4J

Morabit Overview | Test Execution History | Resources | Main Components | Component Instantiation

- Test Results
  - DemoScenario2 --> Bank @ LookupTime (Thread: pool-6-thread-1)
  - BankComponent --> AccountManagement @ LookupTime (Thread: pool-8-thread-1)
  - AccountManagementComponent --> Bank @ LookupTime (Thread: pool-10-thread-1)
  - BankComponent --> CurrencyConverter @ LookupTime (Thread: pool-17-thread-1)
    - TestRequestDescription: TR for CurrencyConverter (TR for CurrencyConverter)
      - TestReaction = TryNextComponent
      - Confidence = 0.5
      - Reliability = 0.9
      - TestSuiteDescription: TS for CurrencyConverter (TS for CurrencyConverter)
        - TestCases
        - TypeUnderTest
        - TestTime = LookupTime
      - SimpleTestResult
        - Confidence = 1.0
        - Reliability = 1.0
        - TestRequest: TR for CurrencyConverter (TR for CurrencyConverter)
        - FailedTestCases: empty
        - SuccessfulTestCases
          - TestCaseDescription: Argument order TC (tests whether client and currencyconverter share the same u
          - TestCaseDescription: Currency Test (tests whether client and currencyconverter share the same under
          - TestCaseDescription: Accuracy Test (tests whether the currencyconverter result is with 2 Accuracy)
        - TestedComponent = org.morabit.ma.currencyconverter.CurrencyConverterComponent@1e87719

## Inhalt

- Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- ▶ Methode
- Zusammenfassung

- ◆ **Komponentenentwicklung**
  1. Komponentenspezifikation
  2. Architekturentwurf (Testschnittstelle!)
  3. Entwurf der Testanfragen (insbes. Reaktionen)
  4. Feinentwurf
  5. Implementierung
  6. Entwicklungszeittest
  7. “Verpackung”
  
- ◆ **Komponentennutzung**
  1. Entpackung, Einsatz und Überprüfung
  2. Anpassung und Weiterentwicklung der Testanfragen
  3. Einsatz und Ausführung



# MORABIT Entwurf der Testanfragen

## Inhalt

- Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung

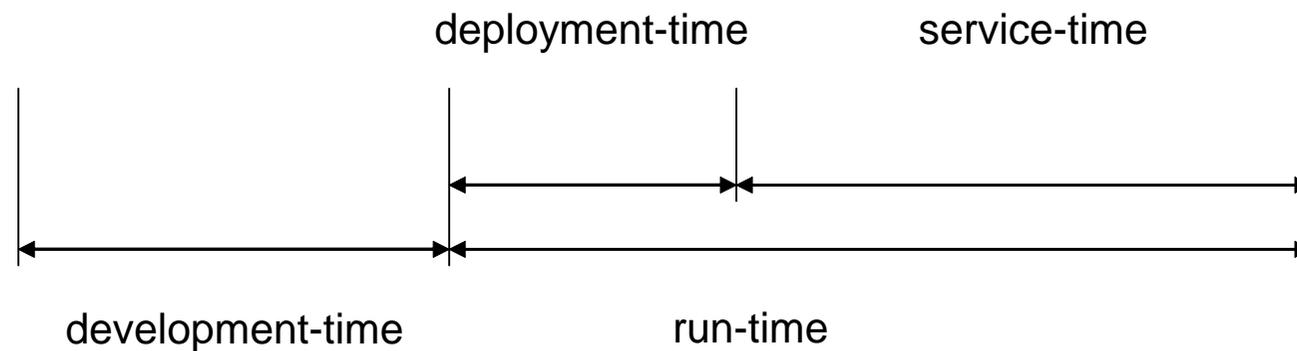


- ◆ Risikoanalyse, um kritische Interaktionen zu erkennen
- ◆ Qualitative Tests (falls wenige Testfälle Fehler aufdecken können) und quantitative Tests (basierend auf Nutzungsstatistik)
- ◆ Typische Qualitative Tests: Semantische Probleme, wie
  - Falsche Reihenfolge bei Eingangs- oder Ausgangsparametern (z.B. Empfänger/ Absenderkonto)
  - Falsche Interpretation von Parametern (z.B. höchster oder niedrigster Bietspreis)
  - Falscher Zustand (z.B. nicht registriert)
  - Unterschiedliche Interpretation von Ausnahmen
  - Qualitätsprobleme (z.B. Antwortzeiten)

- ◆ Built-In-Test auch zur Entwicklungszeit und Deploymentzeit nützlich!
- ◆ Vereinfachen Integrationstest

## Inhalt

- Motivation
- Built-In-Test
- Testprozess
- Testisolation
- Ressourcen
- Infrastruktur
- Methode
- Zusammenfassung



# Quality Engineering in der Zukunft

