Effektive Softwareentwicklung in der Medizintechnik

Ihr Schlüssel zum Erfolg ...

Agenda

- Begrüßung
- Beispiel – Messgeräteplattform
- Requirementsmanagement
- Software – Architektur
- Implementierungs- und Teststrategien
- Werkzeuge
- Teamzusammensetzung

Vorstellung

- Ingo Beyer, Dipl.-Ing. (FH) Technische Informatik an der Fachhochschule Giessen-Friedberg
- Seit 2005 Mitarbeiter der EXCO GmbH, derzeit Projektleiter im Bereich „Medical Device Verification“
- Kernkompetenz:
  - Projektleitung im regulierten Umfeld
  - Testautomation von Embedded Systemen
  - Blackboxtests, Whiteboxtest, Code- und Designreviews
  - Erstellung von Tools zur Sourcecode Generierung

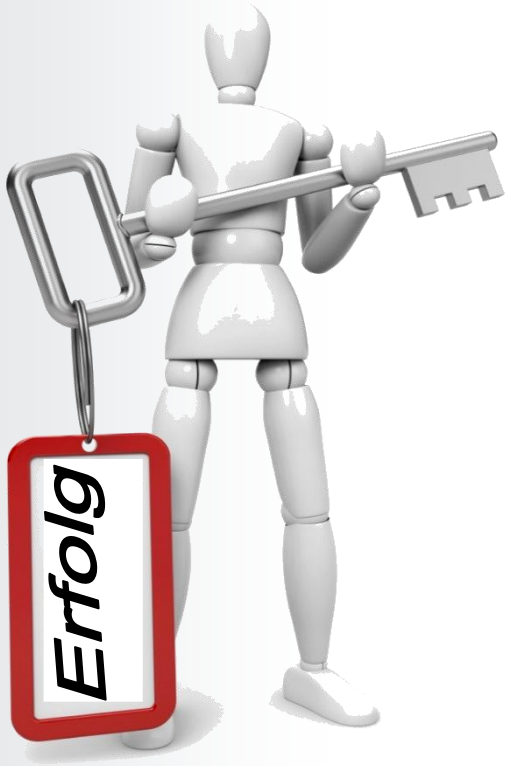
Entwicklung einer Messgeräteplattform

- Beispiel: Ziele bei der Entwicklung einer Messgeräteplattform
  - Weltweiter Verkauf von Geräten
  - Produktion von großen Stückzahlen
  - Identische Basisfunktionalitäten für alle Geräte
  - Gute Wartbarkeit der Software
  - Unterschiedliche HW-Varianten
  - Integration neuer Features mit vertretbarem Aufwand
  - Kurze Entwicklungszyklen für Nachfolgegeräte

Teilaspekte für effektive Softwareentwicklung

Schlüsselelemente für effektive Softwareentwicklung

- Kontinuierliches Requirementsmanagement
- Auswahl der passenden Softwarearchitektur
- Abgestimmte Implementierungs- und Teststrategien
- Effektiver Werkzeugeinsatz
- Interdisziplinäre Teambildung



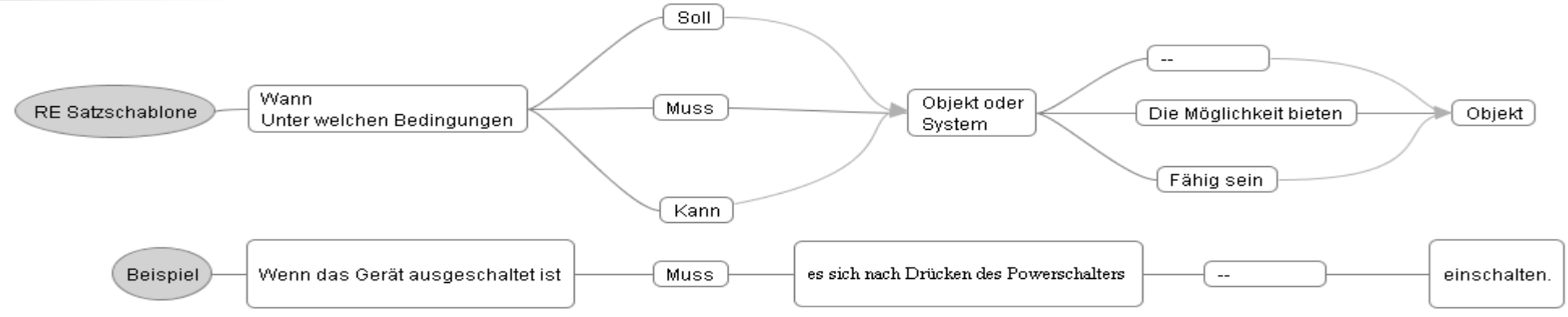
Ihr Schlüssel zum Erfolg ...

Agenda

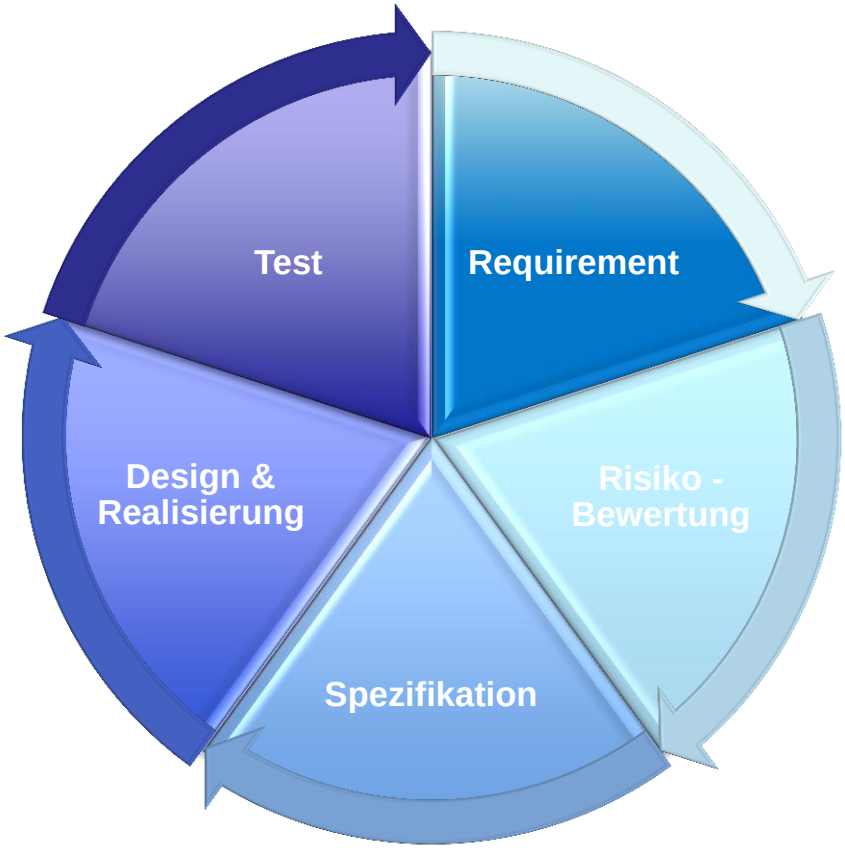
- Begrüßung
- Beispiel – Messgeräteplattform
- Requirementsmanagement
- Software – Architektur
- Implementierungs- und Teststrategien
- Werkzeuge
- Teamzusammensetzung

Voraussetzungen für einen kontinuierlichen Requirementsmanagement Prozess

- Anwendung bewährter Requirement Engineering Praktiken
- Hierarchische Strukturierung (vom Groben ins Detail)
- Priorisierung, Versionierung und Zuordnung von Attributen zu Requirements ermöglicht die Erstellung spezifischer Sichten
- Darstellung von Freigabeständen durch Baselines
- Anforderungen jedem jederzeit zur Verfügung stellen
- Traceability von Spezifikationen zu Requirements

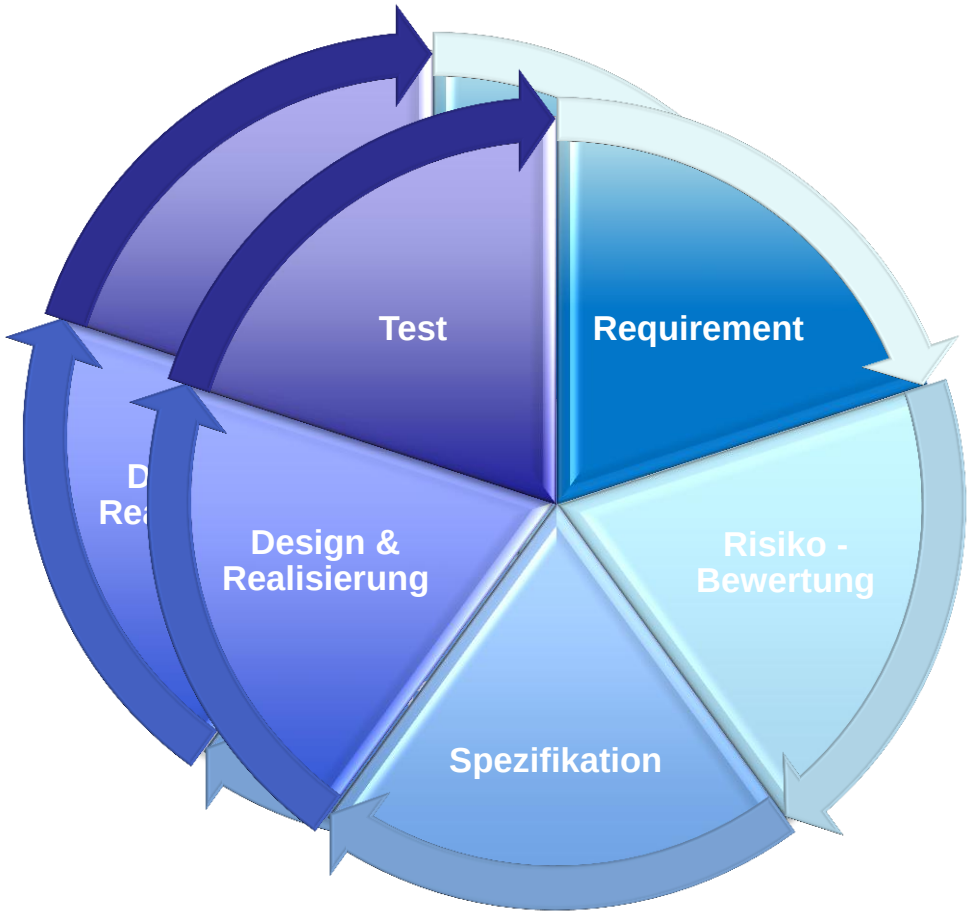


- Kontinuierlicher Entwicklungsprozess



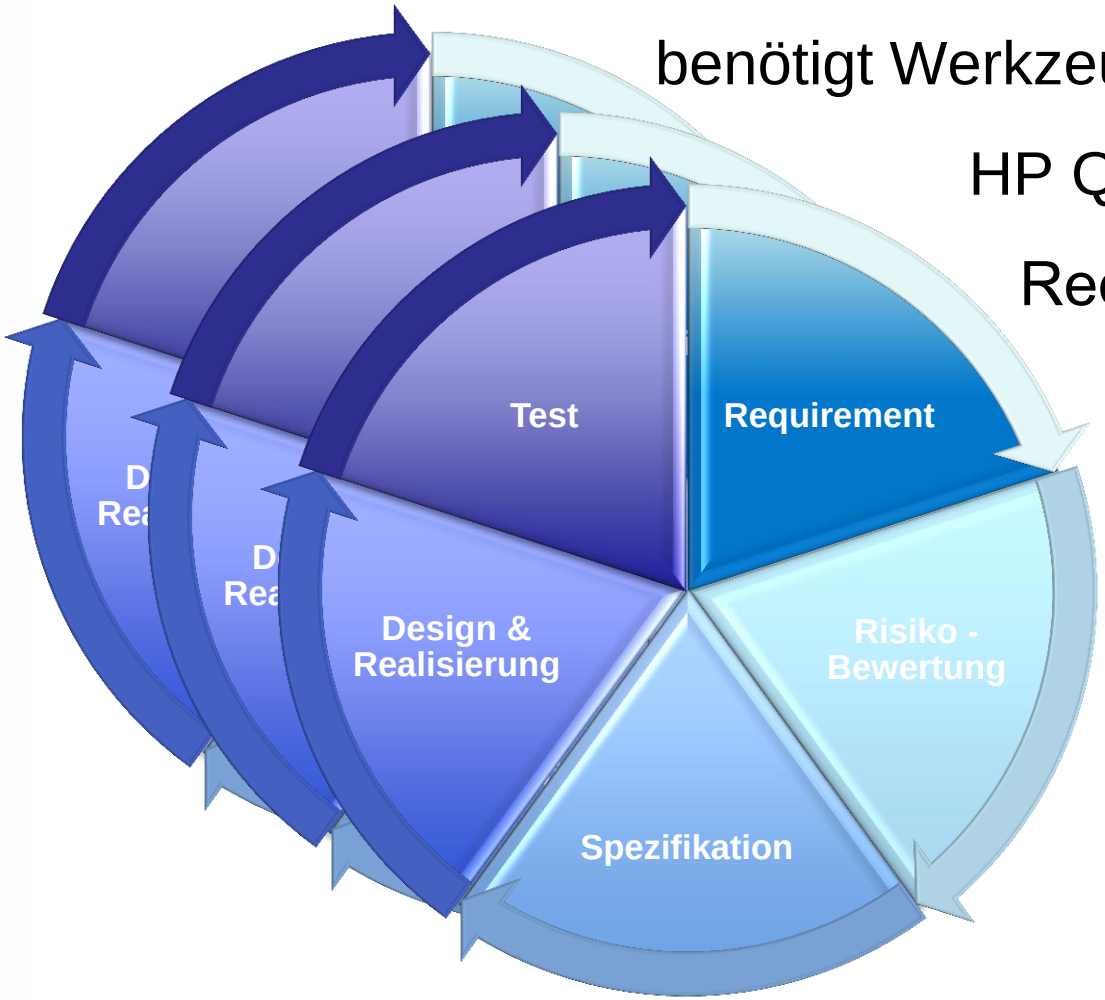
Requirementsmanagement

- kontinuierlicher Entwicklungsprozess



Requirementsmanagement

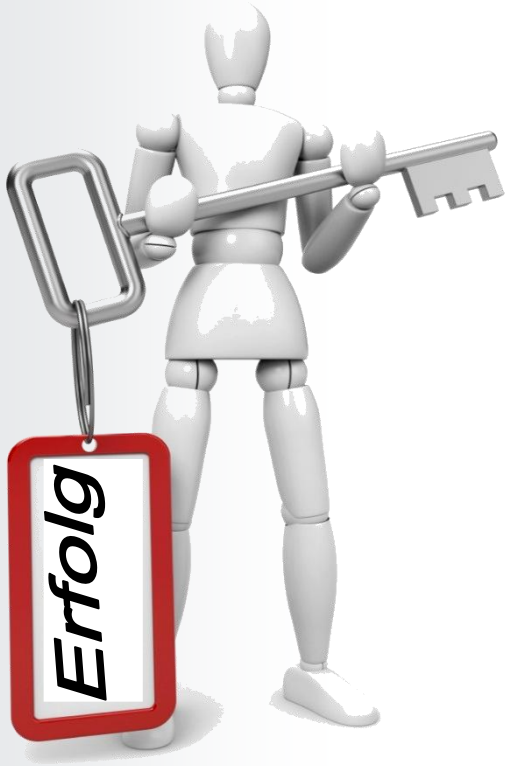
- kontinuierlicher Entwicklungsprozess



benötigt Werkzeuge wie z.B.

HP Quality Center

Requisit Pro ….

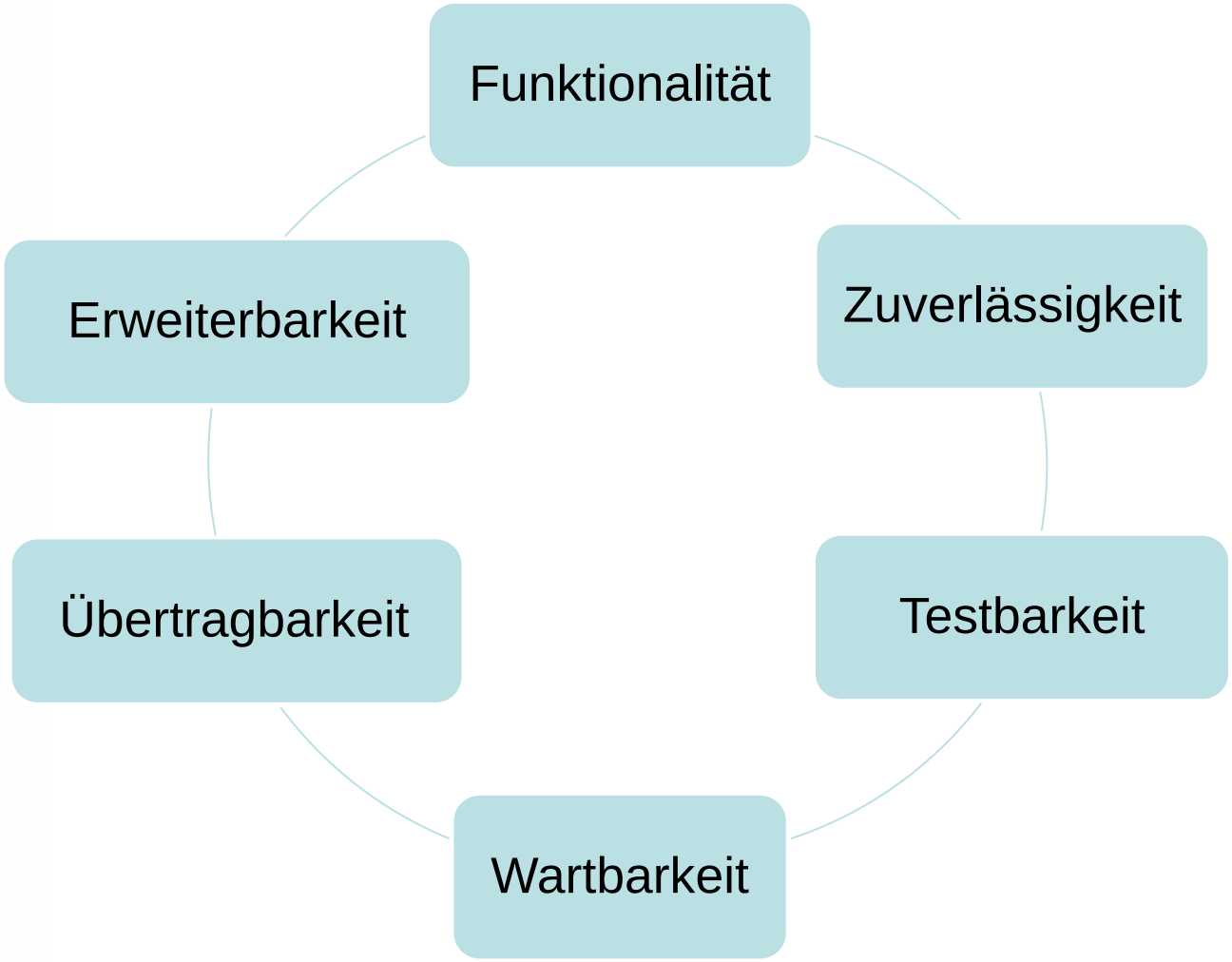


Ihr Schlüssel zum Erfolg ...

Agenda

- Begrüßung
- Beispiel – Messgeräteplattform
- Requirementsmanagement
- Software – Architektur
- Implementierungs- und Teststrategien
- Werkzeuge
- Teamzusammensetzung

Auswahl passender Architekturmuster in Bezug auf:



- Architekturmuster sind bewährte Lösungsstrategien
  - OSI 7 Schichten Modell oder Pipes and Filters
  - Modell View Controller
  - Mikrokernel, Monolitischer Kernel
  - Hardware Abstraction Layer (HAL)

- Quellen zur Auswahl von Entwurfsmustern:
  - *Design Patterns. Elements of Reusable Object-Oriented Software* von Erich Gamma et. Al. (Gang of Four)
  - *Pattern-Oriented Software Architecture von* Frank Buschmann et. Al.
  - *Modellgetriebene Softwareentwicklung von Stahl, Voelter, Efftinge, Haase*

Beispiel für eine Anwendung mit Hardware Abstraction Layer

Applikationsschicht
(enthält Subsysteme zur Umsetzung der funktionalen Anforderungen)

Betriebssystem - Kernel

Hardware Abstraction Layer (HAL)

Hardware Treiber (Speicherverwaltung)

Hardware Treiber (Timer)

Subsystem
inkl. Hardware Anbindung

Beispiel für eine Anwendung mit Hardware Abstraction Layer

Applikationsschicht
(enthält Subsysteme zur Umsetzung der funktionalen Anforderungen)

Betriebssystem - Kernel

Hardware Abstraction Layer (HAL)

Neuer Hardware Treiber
(Neuer Speicher)

Hardware Treiber (Timer)

Neues Subsystem
inkl. Hardware Anbindung

Beispiel für eine Anwendung mit Hardware Abstraction Layer

Applikationsschicht
(enthält Subsysteme zur Umsetzung der funktionalen Anforderungen)

Betriebssystem - Kernel

Hardware Abstraction Layer (HAL)

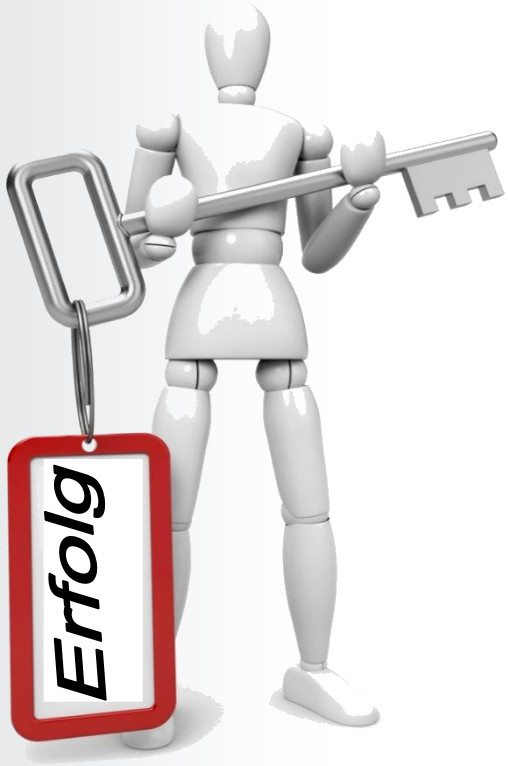
Hardware Simulation (Speicherverwaltung)

Hardware Simulation (Timer)

Simulation Subsystem inkl. Hardware Anbindung

Softwarearchitektur

- Hardware Abstraction Layer (HAL)
  - Definierte Schnittstelle zur Hardware ohne von der Hardware abhängig zu sein
  - Trennung des Software Applikation Layers von der Hardware Treiberschicht
- Vorteile
  - Hardware Komponenten können leichter ersetzt werden
  - Simulation von Hardware durch Software
  - Separate Entwicklung und Test hardwareunabhängiger Funktionen



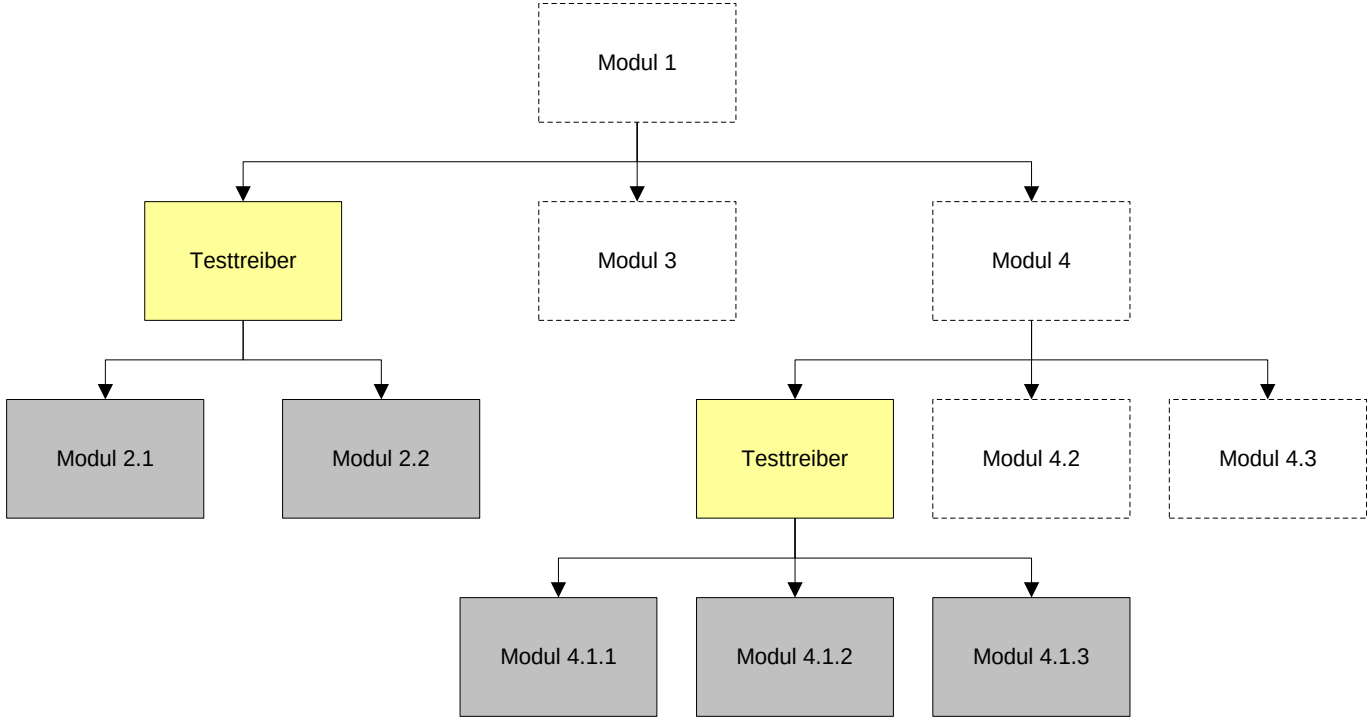
Ihr Schlüssel zum Erfolg ...

Agenda

- Begrüßung
- Beispiel – Messgeräteplattform
- Requirementsmanagement
- Software – Architektur
- Implementierungs- und Teststrategien
- Werkzeuge
- Teamzusammensetzung

Implementierungs- und Teststrategien

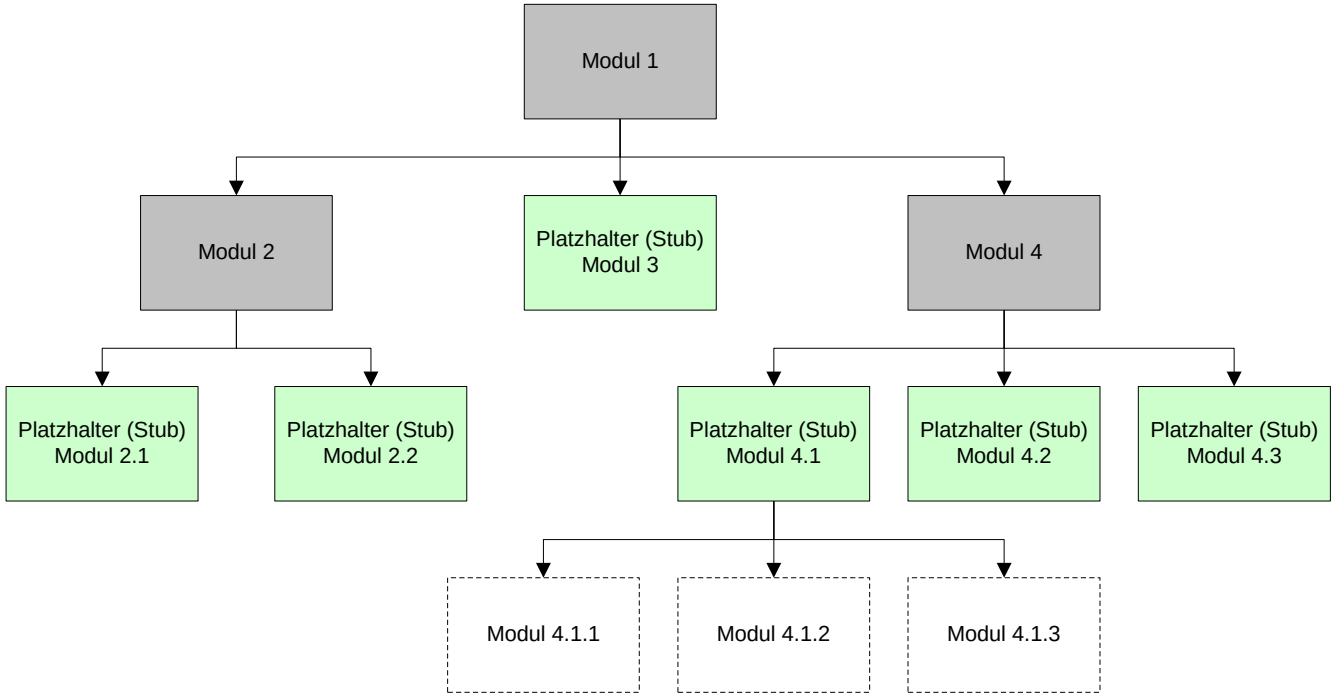
- **Bottom-Up-Integration**



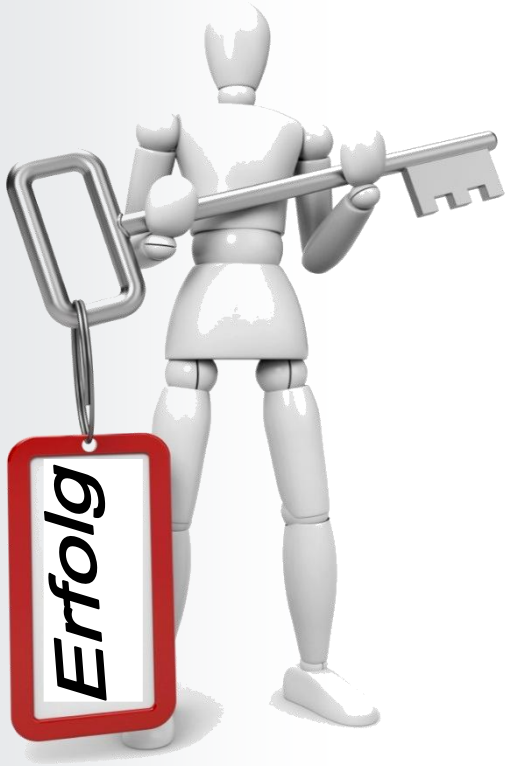
- frühe Umsetzung technisch schwieriger Teile / des Kerns möglich (Critical First)
  ⇨ **Machbarkeitsprüfung**

Implementierungs- und Teststrategien

Top-Down-Integration



- frühzeitige Prüfung der Benutzerschnittstelle (durch Kunden) möglich
  ⇨ **schnelles Kundenfeedback**



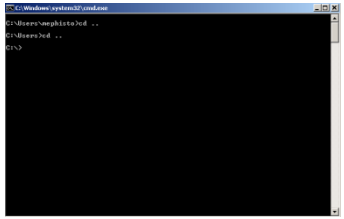
Ihr Schlüssel zum Erfolg ...

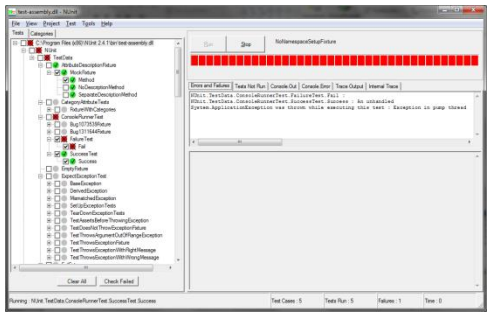
Agenda

- Begrüßung
- Beispiel – Messgeräteplattform
- Requirementsmanagement
- Software – Architektur
- Implementierungs- und Teststrategien
- Werkzeuge
- Teamzusammensetzung

Werkzeuge

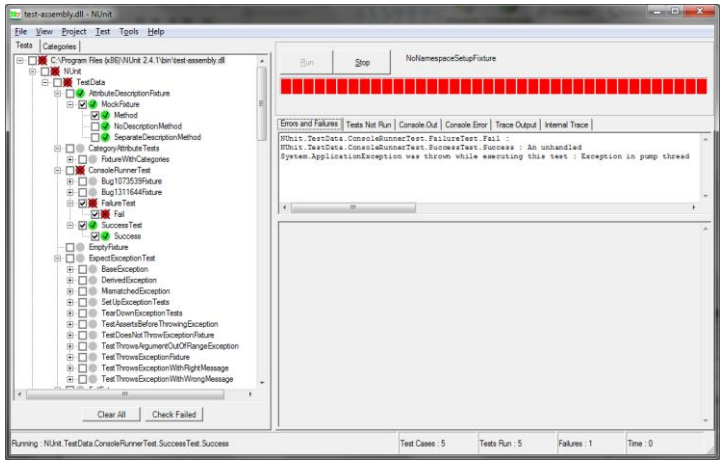
- **Konfigurationsmanagement**
  - **SVN, CSV, Visual Sourcesafe…**
- **Buildautomatisierung**
- **ANT, Team Foundation Server , Hudson …**
- **Testmanagement**
  - **Team Foundation Server (TFS), HP Quality Center …**
- **Testautomatisierung**
  - **xUNIT, Selenium , TFS…**
- **Abweichungsmanagement**
  - **Bugzilla , JIRA, ClearQuest …**



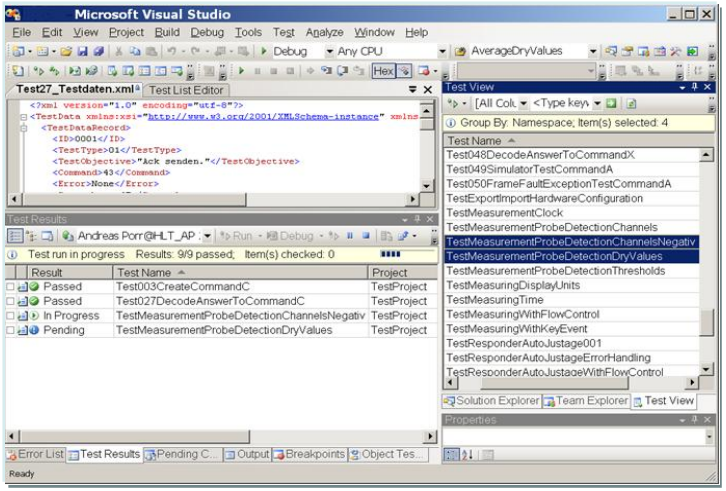


Testautomatisierung

- Externes Programmiersprachen-spezifisches Testframework (xunit)



- Standard Entwicklungsumgebung mit integriertem Testframework



Hardware Simulation

Beispiel:
Embedded Software oder PC-Steuerungs-Software für Geräte ist sehr hardwareabhängig

- **Problem:** Hardware wird parallel zur Software entwickelt

  - Hardware sehr spät und in begrenztem Umfang bzw. begrenzter Funktionalität verfügbar
  - Sonderaufbauten kosten Zeit und Geld
  - Zeitprobleme zum Projektende aufgrund Hardware – Software Integration
  - Qualitätseinbußen

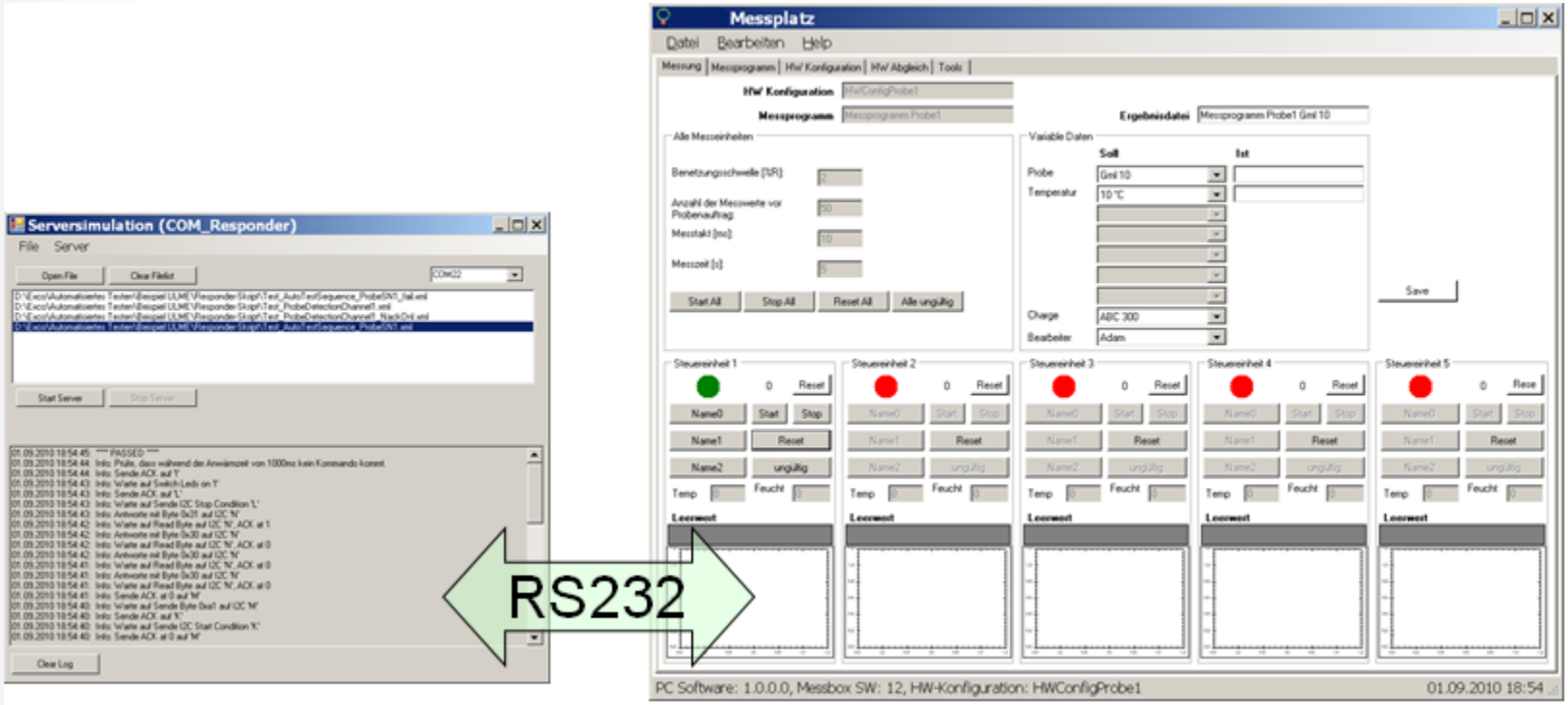
- **Ziel:** hardwareunabhängige Entwicklung und Prüfung der Software
  - Risiken früh erkennen und Gegenmaßnahmen einleiten
  - Kostenreduktion
  - Höhere Qualität des Endproduktes
- **Lösung:** Simulation der Hardware - Komponenten

Hardware Simulation

- Hardware Simulation
  - Einsatz des Hardware Abstraction Layers
  - Nutzbar für Entwicklung, Verifizierung oder sonstige Parteien
- Vorteile:
  - Kein Mengen- und Verfügbarkeitsproblem
  - Schnell aktualisierbar
  - Frühe Verfügbarkeit
- Risiken:
  - Erhöhter Entwicklungsaufwand für die Hardware Simulation
  - Verhalten kann von realer Hardware abweichen (Race conditions, Timing)

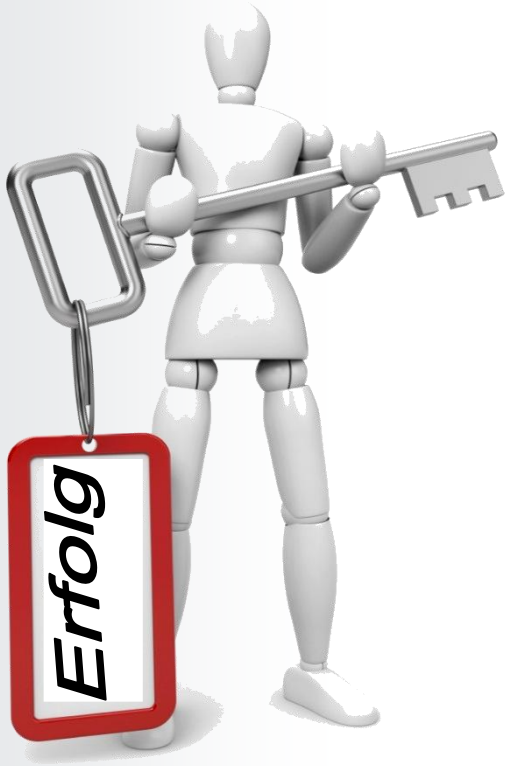
Beispiel:

Simulation eines Laborgerätes zum Test der seriellen Schnittstellenkommunikation der PC-Steuersoftware



Lösungsansatz für den Test der Schnittstellenkommunikation

- Datengetriebene Tests an internen Modulschnittstellen
  - Testdaten von Testskripten trennen
  - Erzeugung eines generischen Skriptes, für beliebig viele Testdaten, die mit einem beliebigen Editor oder Generator erzeugt werden
  - Kommandos und Kommandoantworten automatisiert prüfen
- Hardware Simulation des externen Gerätes
  - Simuliert die Gegenstelle für externe Schnittstellen
  - Wiederverwendbarkeit der Daten aus Modultests
- Integrationstest mit fertigem Gerät:
  - Nur noch wenige Fehler zu erwarten (z.B. Timingprobleme)

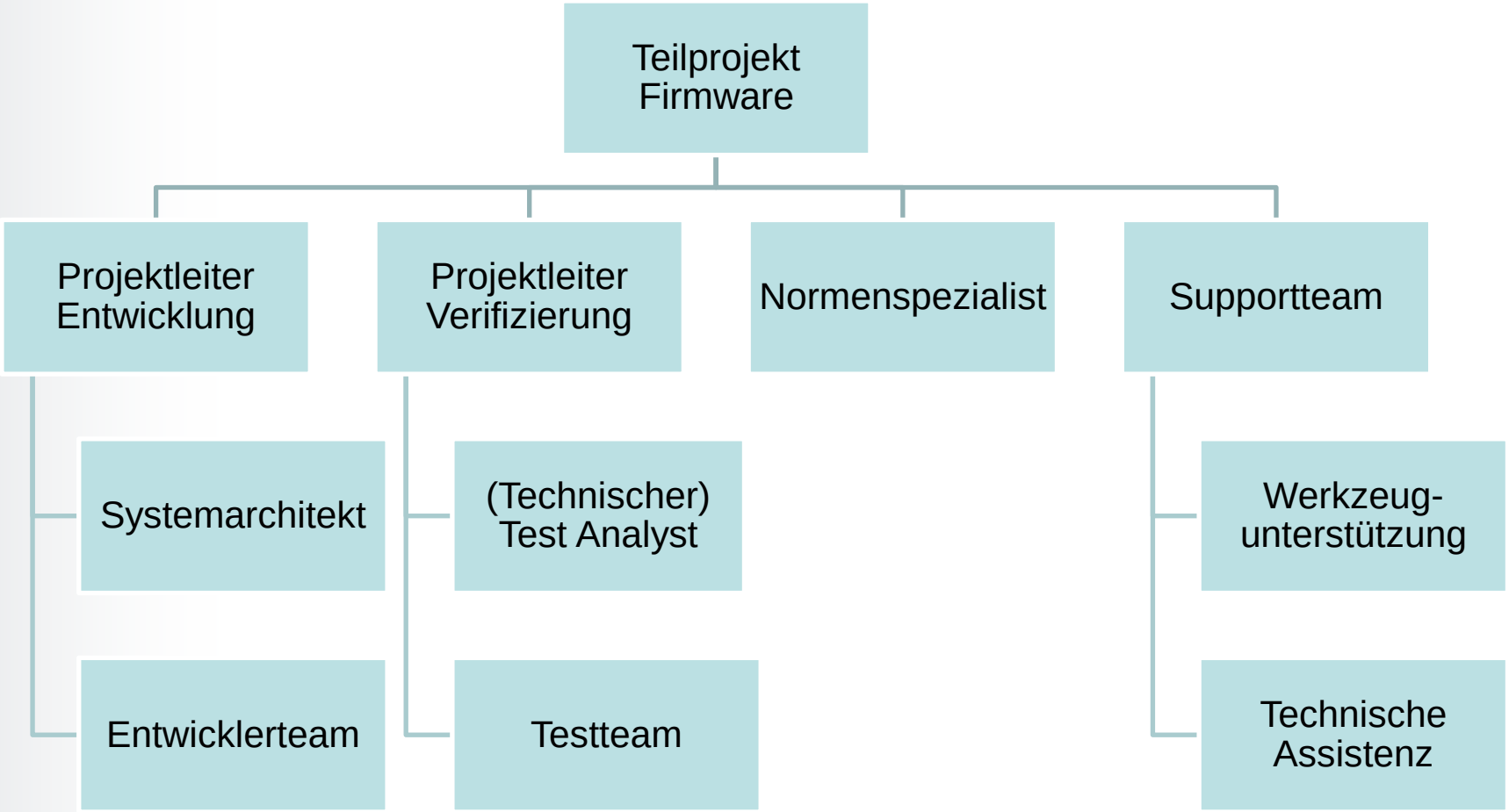


Ihr Schlüssel zum Erfolg ...

Agenda

- Begrüßung
- Beispiel – Messgeräteplattform
- Requirementsmanagement
- Software – Architektur
- Implementierungs- und Teststrategien
- Werkzeuge
- Teamzusammensetzung

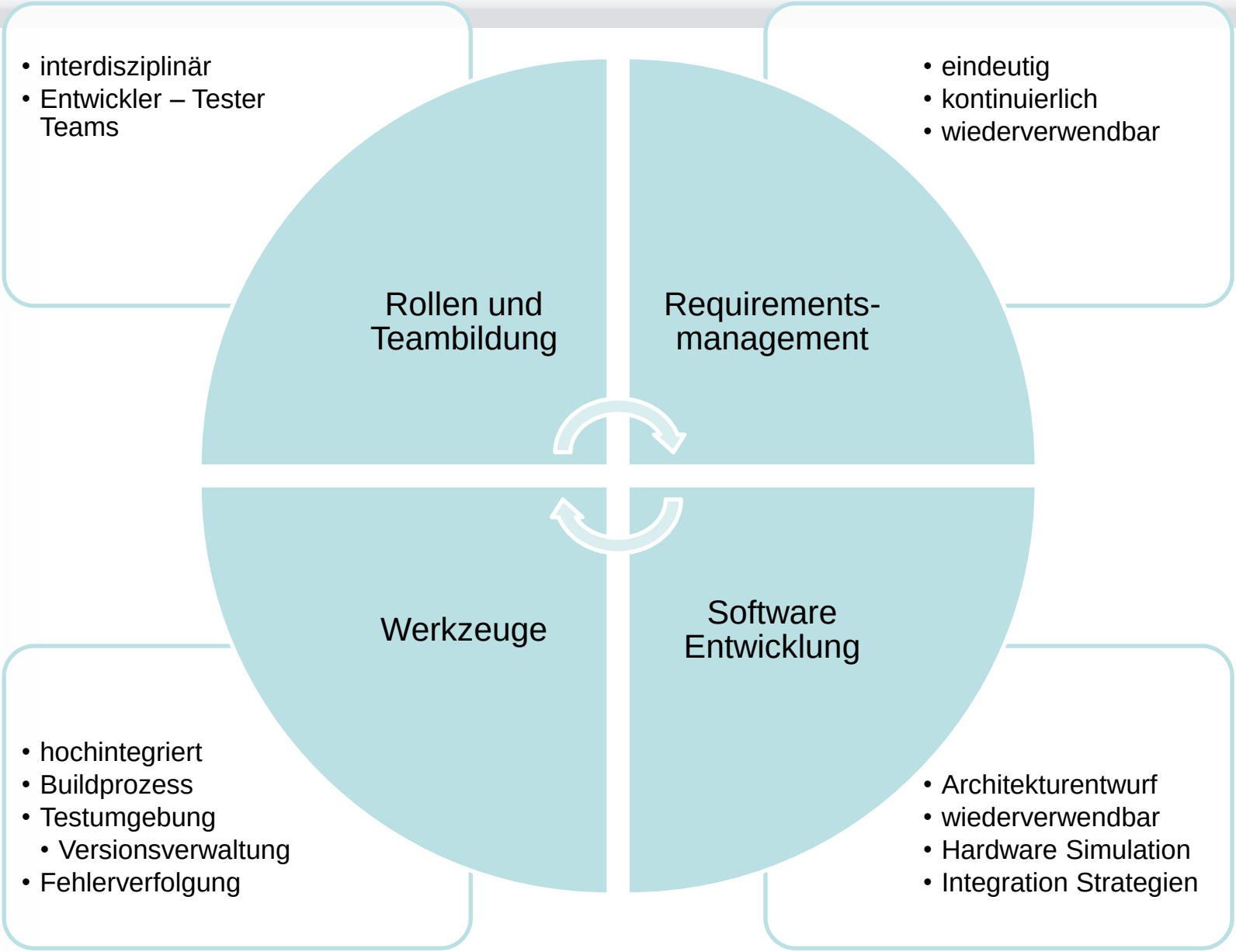
Rollen in der Entwicklung



Teambildung

- Teambildung Entwickler - Tester
  - Permanente Interaktion zwischen Entwicklern und Testern
  - Gleichzeitige Entwicklung und Testerstellung
- Vorteile:
  - Kurze Wege, effektive Kommunikation
  - Frühzeitige Fehlerfindung und –behebung
  - Hohe Qualität durch 4-Augen-Prinzip
  - Unterschiedliche Interpretationen (Entwickler, Tester) ermöglichen das frühzeitige Entdecken von Unstimmigkeiten in Requirements

Zusammenfassung



Danke für Ihre Aufmerksamkeit

Haben Sie noch Fragen?

Deutschland:

EXCO GmbH
Adam-Opel-Strasse 9-11
D-67227 Frankenthal
Deutschland

fon +49 (0) 6233 73778-0
fax +49 (0) 6233 73778-190

www.exco.de
info@exco.de

Schweiz:

EXCO Consulting GmbH
D4 Business Center Luzern, Platz 5
CH-6039 Root Längenbold
Schweiz

fon +41 (0) 41 455 24-70
fax +41 (0) 41 455 24-79

www.exco-consulting.ch
info@exco-consulting.ch

Frankreich:

Verbindungsbüro Frankreich
50 Avenue d'Alsace
F-68025 COLMAR Cedex
Frankreich

fon +33 (0) 389 2928-19
fax +33 (0) 389 2043-79

www.exco.de
france@exco.eu